SCAPE final evaluation and methodology report

Authors

Rune Bruun Ferneke-Nielsen, Bolette Ammitzbøll Jurik, Bjarne Andersen (State & University Library Denmark), William Palmer (British Library), Daniel Pop (West University of Timisoara), Sven Schlarb (Austrian National Library), Alastair Duncan (Science and Technology Facilities Council), Ivan Vujic (Microsoft Research), Ondřej Klíma (Brno University of Technology), Opher Kutner (ExLibris), Tomasz Parkola (Poznań Supercomputing and Networking Center), Frank Asseg (Fachinformationszentrum Karlsruhe), Stanislav Barton, Leïla Medjkoune (Internet Memory)

September 2014

This work was partially supported by the SCAPE Project. The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).

Executive Summary

The SCAPE project's focus has been to develop scalable tools, services and infrastructure for the efficient planning and execution of preservation strategies and workflows for large-scale, heterogeneous collections of complex digital objects. Through this, digital preservation state-of-the-art is enhanced threefold:

- by developing infrastructure and tools for scalable preservation actions;
- by developing a framework for automated, quality assured preservation workflows;
- by integrating these components with a policy-based preservation planning and watch system.

These concrete project results have been driven by requirements from and in turn validated within four large-scale testbeds from diverse application areas: Web Content from the web archiving community, Digital Repositories from the library community, Research Data Sets from the scientific community and Data Center Infrastructure from data and supercomputing centers.

Within each of the testbeds, we have been validating tools and techniques developed within SCAPE against very large heterogeneous digital object collections. The focus has been on the reliability and scalability of the services. We have also used the testbed environments to demonstrate the robustness of our platform, the feasibility of our approach, and the advances being made by the project to the various communities of interest.

This report presents the results of the Evaluation of Results work package (TB.WP.4), as a part of the testbeds in the SCAPE project. In this work package, we have developed a structured and systematic evaluation methodology based on goals, objectives, metrics and evaluation through experiments coupled to SCAPE-defined user stories.

The evaluations carried out in the SCAPE project show that the tools, components, workflows, applications and platform, developed within the project scope, have added tremendous value to the digital preservation environment and community. Now, it is possible to create scalable solutions, which are able to handle very large amount of data, and the processing time can therefore be reduced to days or weeks instead of months or years.

Developments of SCAPE have been tested on various new areas through the extension of the SCAPE project in the final project phase with new partners representing data centers and several new content types – e.g. data from hospitals. The results show great impact on the possibility to process large amounts of data within a reasonable amount of time.

Components have been tested by commercial partners Ex Libris and Microsoft Research. Participation in the SCAPE project has contributed to their understanding of the evolving needs and emerging solutions in this community. On top of that SCAPE technology has proved to work in other technical setups than the standard Hadoop setup implemented as the reference system of SCAPE.

Finally, in addition to the implemented software solutions and greatly promising results, the project has facilitated a vast increase of experience and knowledge within the preservation community, and very many new bonds have been created among institutions and people across Europe.

Table of Contents

1 Introduction

This report is the second and final evaluation report from the SCAPE[1] project and describes the outcome of the testbeds work package *Evaluation of Results*, TB.WP.4. It presents the evaluation methodology developed in the project as well as the results from the testing performed within the project period.

User stories, experiments and evaluations span across four different testbeds[2], and thereby four business domains, namely *Web Content*, *Large Scale Digital Repository*, *Research Dataset* and *Data Center*. The four testbeds, and the work carried out, are described fully in the following SCAPE deliverables: D15.2[3], D16.2[4], D17.2[5], and the upcoming D23.2[6].

Every user story has one or more of the following topics as focal points:

- Accessing content
- Analysis of medical data
- Validation of content against specification and/or institutional policy
- Ingest of content
- Feature extraction from content
- Identification of content
- Characterisation of content
- Migration of content
- Validation of action
- Quality assurance of content

The work of TB.WP.4 is primarily concerned about reporting of the solutions that have been implemented as part of the work in the four testbeds. Secondary, the work is also about facilitating the reporting process of the evaluators.

1.1 Follow up of first evaluation, D18.1

The first evaluation report, the SCAPE D18.1[7] deliverable, was written in month 22-24 of the project period, and it describes the state of the testbeds work at that time.

As there has been a lot of progress and development in the SCAPE project since the first report was written, we have also embraced the opportunity to make some changes for the second report. This means that some parts of this report are continued, while others have been removed. Some of the

[1] http://www.scape-project.eu
[2] An outline of the four testbeds can be found in section 6
[3] http://www.scape-project.eu/deliverable/d15-2-web-content-executable-workflows-for-large-scale-execution
[4] http://www.scape-project.eu/deliverable/d16-2-lsdr-executable-workflows-for-large-scale-execution
[5] http://www.scape-project.eu/deliverable/d17-2-research-data-sets-executable-workflows-for-large-scale-execution
[6] The report is not created yet, but will be available at http://www.scape-project.eu/category/deliverable
[7] http://www.scape-project.eu/deliverable/d18-1-first-evaluation-report-draft

topics have been described in other publicly available reports see below for those not discussed further in this report.

- SCAPE functional review and development guidelines, found in D2.3[8]
- Evaluation of planning case study[9]

1.2 Moving from scenario to user story

In the second half of the SCAPE project, we decided to structure and describe the use cases differently. It was sometimes difficult to extract the relevant information from a scenario (consisting of a data set, one or more issues, and one or more solutions) that would enable a developer to construct an appropriate solution or to ensure that the solution would be successfully evaluated.

Therefore, scenarios were refactored into a simpler format that captures requirements succinctly and provides a starting point for development and a defined set of criteria for evaluation. The new format consists of three levels: user story, experiment and evaluation.

The retained list of scenarios on the SCAPE wiki[10] is a valuable resource and present lots of useful problems and ideas for solutions that SCAPE might address. And we have ensured that the refactoring does not impact work already completed that either references or makes use of the existing scenarios.

User Story

A user story represents a top level summary of the problem and user requirements. The story should not reference any given organisation, but rather try to generalise the problem such that someone outside of SCAPE could read through the user stories and see which, if any, could be useful to them. For example, rather than saying the "BL's Web Archive Dataset", it is better to say "A large collection of WARC files". This allows us to use the user stories as a showcase for the kinds of problems SCAPE may solve.
The user requirements should identify what needs to be done rather than the tools to do it. Development of functional requirements and identification of tools - existing or required - should be done outside of the user stories and is not part of the testbeds development work.

Experiment

Each user story will then have one or more experiments associated with it. An experiment is a real-life application of a use case, outlining an existing dataset, the business needs of the dataset owner, a workflow (for instance a Taverna workflow) and the set of evaluation criteria. We view the evaluation criteria as dataset-specific requirements - for example, throughput may not be a great concern, when migrating an existing collection, but may become paramount when verifying content as it arrives from a digitisation agency, where problems need to be identified, before the contract ends.

[8] http://www.scape-project.eu/deliverable/d2-3-technical-architecture-report-v2
[9] http://ifs.tuwien.ac.at/~becker/pubs/archiving2013.pdf
[10] http://wiki.opf-labs.org

Evaluation
The evaluation will contain information about the experiment and the evaluation criteria, and is likely to summarize a series of experiment executions. The series of experiments will show progress, aiming at reaching the proper solution, and for each new experiment some parameter is changed to alter the outcome.
The evaluation criteria may be measurable as well as non-measurable, and the evaluator has to be aware of what will be the best way to provide the results. An overall conclusion will sum up the results and findings, and thereby ensure that other people will comprehend the message.

1.3 Outline

In chapter 2, the methodology for evaluating experiments is described, which includes 10 goals for the testbeds work, and a reference to the central metrics catalogue. Also, we have described the connection between the evaluation work and the SCAPE project objectives.

Chapter 3 is a review of the SCAPE testbeds user stories, where individual goals are stated and potential solutions, results and findings presented. Moreover, we look at the progress of continued user stories from the first evaluation report.

In chapter 4, testbed partners have given their view of the SCAPE project achievements in relation to their own organisation. Their reflections tell a story about gained experiences and challenges.

In chapter 5, Ex Libris has described the SCAPE project achievements seen from a commercial point of view.

Appendix A contains the testbed descriptions.

Appendices B1 - B4 describes the user stories.

Appendices C1 - C4 describes the experiments.

Appendix D describes the platforms.

Appendix E describes the datasets.

Appendices F1 - F4 describes the evaluations.

Appendix G contains the templates used for the testbed work.

2 Evaluation Methodology

The evaluation methodology defines the measurable objectives of the testbeds and accordingly identifies suitable evaluation methods, on which an evaluation plan can be built. It explains how to perform a testbed experiment in the SCAPE project, as well as describing reporting templates and other relevant information in use.
We have made all the testbed work understandable, comparable and meaningful to all interested parties by using a common methodology across the four testbeds.

The structure chosen to organise the testbed elements and the many relations between testbeds, user stories, experiments and evaluations can be seen below.

testbed
  user story
    experiment
      evaluation

2.1 Experiment and Evaluation

A few things should be considered and be in place to get the best outcome of an experiment and an evaluation. As these are closely linked, it can make sense to work on both elements in small iterations, switching back and forth between the two as new knowledge is gained and reflected upon. The steps below describe the content of this process in a listed form, which is not iterative but provides a clearer picture.

1. Define top-10 goals and objectives that will be evaluated
2. For these goal-objective pairs choose what and how to evaluate
    a. Set up an experiment, and execute.
    b. Write an evaluation of findings and results, both measurable and non-measurable.

Each evaluation follows a basic scheme

1. Set up evaluation page
    a. Define metrics (Metrics Catalogue)
    b. Define metric baseline (ground truth) - e.g. this could be current state for a tool running on a single machine
    c. Define metric goal - what result do we want to achieve
    d. Define non-measurable evaluation points
2. Write relevant technical details, use WebDAV where suited (see evaluation template)
3. Write up results, and other important information in the evaluation template
    a. Especially results for the defined metrics (1a) must be recorded

The results were gathered manually and entered into the evaluation pages on SCAPE wiki[11] pages. We expected that many of the experiments were evaluated several times; ultimately until the

[11] http://wiki.opf-labs.org/display/SP/Stories+and+Experiments

defined metric goal (1c - see above) was reached. Thus resulting in multiple findings and results (3a - see above) showing progress of SCAPE developments. For some objectives (e.g. organisational fit), it was not be applicable to define a precise measure, and for such objectives a more qualitative human understandable evaluation (in form of a textual section) was written.

2.2 Top-10 goals and objectives

The top-10 goals and objectives (also mentioned as testbed goals) were defined by testbed work packages leads and reviewed by subproject leads. The testbed goals have been selected to cover a broad range of activities within the project, as well as covering main aspects across the work packages. The following resources were used in the process of defining the overall goals and objectives.

- An overview of scenarios (now named user stories) and how they relate to work packages[12]
- An overview of goals, objectives and suggested metrics defined by this work package (TB.WP.4) and reviewed by subproject leads[13]
- The SCAPE D14.1[14] deliverable, which builds upon the SQUARE[15] quality model

The list of goals was created before any of the user stories, experiments or evaluations were defined and written, and it was therefore not possible to use those as guide lines for creating the list. As a result, some of the goals were not used in the evaluations in the first report, as a match between the goals and the experiments was not found at this early stage in the SCAPE project.
For this second evaluation report, many more experiments are being evaluated, and during the refinement of the evaluation methodology it was decided to keep the list of goals for the second round. The reasoning being that it would still be possible to include more goals in the upcoming evaluations. And that the goals in use would be the same between the first and the second round of evaluations, making it easier to track progress for continued experiments. However, reality has turned out to be somewhat different, and not all goals are used; and there are a number of reasons for this.

The top-10 goals list can be found in Table 1, followed by a brief explanatory discussion about the use or omission of the goals.

No	
1	**Goal:** Performance efficiency **Sub-goal:** Capacity; Resource utilization; Time behaviour **Objective:** Improve DP technology to handle large preservation actions within a reasonable amount of time on a multi node cluster **Comments:** Evaluates different kinds of performance - e.g. throughput, time per MB, time per

[12] http://wiki.opf-labs.org/download/attachments/14352645/Scenarios_WPs_matrix.pdf?version=1&modificationDate=1340274253000
[13] http://wiki.opf-labs.org/download/attachments/14352645/TB4-objectives-metrics-evaluation-20120530.docx?version=1&modificationDate=1340270446000
[14] http://www.scape-project.eu/deliverable/d14-1-report-on-decision-factors-and-their-influence-on-planning
[15] http://www.iso.org/iso/catalogue_detail.htm?csnumber=35733

	sample, memory per sample, maximum files
2	**Goal:** Reliability **Sub-goal:** Stability indicators **Objective:** Package tools with known methods and run development with good open source practices **Comments:** Support available, release cycle, active community. Not directly relevant for testbeds but components developed in SCAPE in connection with scenarios in all testbeds could be used to evaluate this
3	**Goal:** Reliability **Sub-goal:** Runtime stability **Objective:** Improve DP technology (platform and tools) to run automated with proper error handling and fault tolerance **Comments:** E.g. ability to handle invalid input, error codes
4	**Goal:** Functional suitability **Sub-goal:** Completeness **Objective:** Improve number of file formats correctly identified within a heterogeneous corpus **Comments:** Identification, Automated Watch
5	**Goal:** Functional suitability **Sub-goal:** Correctness **Objective:** Develop and improve components to do preservation actions more correctly **Comments:** Valid and well-formed objects from action tools QA accuracy (e.g. correct similarity between two files) Automated Watch: Correct information
6	**Goal:** Organisational maturity **Sub-goal:** Dimensions of maturity: Awareness and Communication; Policies, Plans and Procedures; Tools and Automation; Skills and Expertise; Responsibility and Accountability; Goal Setting and Measurement **Objective:** Improve the capabilities of organisations to monitor and control preservation operations to a point where SCAPE methods, models and tools enable a best-practice organisation to be on level 4 **Comments:** This is the compound effect of policy-based planning and watch, cf. the vision described in the paper at ASIST-AM 2011[16]
7	**Goal:** Maintainability **Sub-goal:** Reusability **Objective:** Increase number of tools registered in components catalogue making them discoverable **Comments:** This is more like a platform/watch evaluation - not directly linked to any specific scenarios or components
8	**Goal:** Maintainability **Sub-goal:** Organisational fit **Objective:** Ensure SCAPE technology fits organisational needs and competences **Comments:** How does it fit in an organisation? How easy is it to integrate with existing infrastructure and processes? Should be implicit in all we're doing rather than an explicit testbed requirement. We should be able to evaluate this in any solutions actually implemented in real organisations within the project
9	**Goal:** Planning and monitoring efficiency

[16] http://asis.org/asist2011/proceedings/submissions/124_FINAL_SUBMISSION.pdf

	Sub-goal: Information gathering and decision making effort **Objective:** Drastically reduce the effort required to create and maintain a preservation plan **Comments:** cf. the metrics described in the paper at ASIST-AM 2011[17]
10	**Goal:** Commercial readiness **Sub-goal:** - **Objective:** Evaluate to what extent SCAPE technology is going in a direction that makes it ready for commercial exploitation **Comments:** -

Table 1 Top-10 goals and objectives

Almost every experiment has one or more performance aspects included, and as a result most evaluations have successfully included goal 1, *Performance efficiency*. This is very much in line with one of the overall goals of the SCAPE project. This goal will be processed in further details in chapter 3, and for those user stories where applicable.

The goals 2 and 3, *Reliability – Stability indicators* and *Reliability – Runtime stability*, as well as goals 4 and 5, *Functional suitability – Completeness* and *Functional suitability – Correctness*, have in some degree been directly evaluated through experiments. Perhaps more important, they have spawned valuable feedback to the other SCAPE work packages, being used to drive and improve their work and development. These goals will be processed in further details in chapter 3, and for those user stories where applicable.

The planning aspects were part of the first report, as part of the Plato case study. Since then a second round of the Plato case study has taken place, and the evaluation is documented in its own report[18]. Moreover, none of the experiments in the testbed work package have focused on planning aspects. Therefore, the goals 6, *Organisational maturity,* and 9, *Planning and monitoring efficiency*, are thus not discussed further in this report.

Goal 7, *Maintainability – Reusability*, has also been handled outside of the testbed work; more specifically in the SCAPE D7.3[19] report.

Maintainability – Organisational fit, which is goal 8, is part of many evaluations. It is worth noting that a great part of the experiments are exploratory work which is responding to an organisational need but will not necessarily go into production as the outputs of the evaluation will influence future implementation decisions.
For this report, participating institutions reflected on what SCAPE has brought their organisation, benefits and challenges, and where they are heading. These thoughts are described in chapter 4 Retrospective by SCAPE partners.

Goal 10, *Commercial readiness*, has been evaluated by commercial partners in the SCAPE project, described in chapter 5. Also, the SCAPE information days held by SCAPE partners have given some insight on this topic, and can be studied in the SCAPE demonstration report[20].

[17] http://asis.org/asist2011/proceedings/submissions/124_FINAL_SUBMISSION.pdf
[18] http://ifs.tuwien.ac.at/~becker/pubs/archiving2013.pdf
[19] http://www.scape-project.eu/deliverable/d7-3-design-and-implementation-of-the-preservation-component-catalogue
[20] http://www.scape-project.eu/news/d19-2-final-demonstration-report

2.3 Mapping testbed goals to SCAPE objectives

A number of project wide objectives are defined in the projects Description of Work[21] (DoW) Part-B page 9-11; three of these objectives are explicitly mentioned as being relevant to the work carried out in the testbed work package. Furthermore, a number of the other project objectives are also taken into consideration, when the evaluations are performed through the testbed work.
In this section, we will state relevant project objectives and the connection between these and the testbed goals.

The three project objectives, 1, 3 and 6 are stated below, and are according to the DoW directly related to testbed work. All three objectives are addressed through the experiments, and can be linked to the testbed goals 1, 3, 4, 5.

DoW-1
Objectives: Addressing the problem of scalability in four dimensions: number of objects, size of objects, complexity of objects, and heterogeneity of collections
Actions: Improved preservation characterisation tools; improved preservation action tools; parallel processing service; complex workflow support
Related to testbed goals: 1, 3, 4, 5

DoW-3
Objectives: Answering the question, what tools and technologies are optimal for scalable preservation actions, given a defined set of institutional policies
Actions: Requirements and development of preservation components; preservation plans generated by the planning component; testbed summative assessment reports
Related to testbed goals: 1, 3, 5

DoW-6
Objectives: Validating and demonstrating the scalability and reliability of this system against large collections from three different Testbeds
Actions: Producing a set of metrics, evaluation methods and timetable for monitoring progress towards those objectives; periodically assessing the state of progress in each area
Related to testbed goals: 1, 3, 4, 5

Moreover, the project objectives 2, 5, 8, 10, 11, 12 and 13 play a secondary part in the evaluation work, meaning that the testbed work is passing on results and findings to other work packages in the project. In this way, there is an interactive correspondence between several work areas; as there should be, to make the best out of the project.

DoW-2
Objectives: Introducing automation and scalability in the areas of (2a) Preservation actions, (2b) Quality assurance, (2c) Technical watch, and (2d) Preservation planning
Actions: Development of new automated quality assurance approaches; development of a scalable preservation planning tool; development of an automated technical watch component
Related to testbed goals: 1, 3, 4, 5

[21] This is an internal document, which have not been published.

DoW-5
Objectives: Producing a reliable, robust integrated preservation system prototype within the timeframe of the project
Actions: Platform reference implementation releases; testbed demonstrators
Related to testbed goals: 1, 3, 5

DoW-8
Objectives: Ensuring a viable future for the results of this and other successful digital preservation projects and engaging with users, vendors, and stakeholders from outside the digital preservation community
Actions: Communication plan and related dissemination activities; training activities; sustainability efforts, in particular engagement through the OPF
Related to testbed goals: 2, 8

DoW-10
Objectives: Increase the variety of SCAPE deployments to include data center environments and new hardware facilities
Actions: Deploy and integrate the SCAPE preservation platform with data center environments and resolve domain-specific preservation requirements
Related to testbed goals: 10

DoW-11
Objectives: Extend the functionality of SCAPE services to ensure the integrity and privacy of data that is preserved by remote and third party institutions
Actions: Develop a method for anonymous and/or encrypted ingest and access of medical data into/form external data facilities
Related to testbed goals: 10

DoW-12
Objectives: Extend the SCAPE user-base by large-scale preservation scenarios from domain scientists and data-center customers
Actions: Development of a methodology for preserving voluminous 3D-object models and raw video materials
Related to testbed goals: 10

DoW-13
Objectives: Extend the SCAPE user-base by large-scale preservation scenarios from domain scientists and data-center customers
Actions: Demonstrate scalability through preservation scenarios from two different domains utilizing large external data center facilities
Related to testbed goals: 10

2.4 Metrics Catalogue

We have agreed a common set of metrics, which makes it possible to recognise and relate to the properties across the many SCAPE work packages. Among some of the places, where the metrics are used, are the applications Plato and Scout that have been developed in the work packages concerning planning and watch. Overall, we have adopted a standardised approach.

When choosing metrics for an evaluation, all existing metrics can be found in the central catalogue[22] and the appropriate one(s) can be selected. If no such metric exists, a request about defining the new metric in the central catalogue can be made.

The first round of evaluations used metrics that were not defined in the central catalogue; at that time the catalogue was in the process of being built. During the refinement of the evaluation methodology, these metrics were added to the catalogue – some metrics were created as no such metric existed, while others were transformed into an already existing metric. The process and discussion can be found as GitHub issues 2 - 13[23]

As a result, some evaluations use the original metric, and others the metrics found in the central catalogue. We have decided to allow the use of both, and the table below shows the link between the new and the original metric.

Metric	Previously known as	URL
number of objects per second	~~NumberOfObjectsPerHour~~	http://purl.org/DP/quality/measures#418
IdentificationCorrectnessInPercent	~~IdentificationCorrectnessInPercent~~	http://purl.org/DP/quality/measures#417
max object size handled in bytes	~~MaxObjectSizeHandledInGbytes~~	http://purl.org/DP/quality/measures#404
min object size handled in bytes	~~MinObjectSizeHandledInMbytes~~	http://purl.org/DP/quality/measures#405
N/A	~~PlanEfficiencyInHours~~	see https://github.com/openplanets/policies/issues/6
throughput in bytes per second	~~ThroughputGbytesPerMinute~~	http://purl.org/DP/quality/measures#406
throughput in bytes per second	~~ThroughputGbytesPerHour~~	http://purl.org/DP/quality/measures#406
stability judgement	~~ReliableAndStableAssessment~~	http://purl.org/DP/quality/measures#108
failed objects in percent	~~NumberOfFailedFiles~~	http://purl.org/DP/quality/measures#407
N/A	~~NumberOfFailedFilesAcceptable~~	see https://github.com/openplanets/policies/issues/11
QAFalseDifferentPercent	~~QAFalseDifferentPercent~~	http://purl.org/DP/quality/measures#416
N/A	~~AverageRuntimePerItemInHours~~	see https://github.com/openplanets/policies/issues/13

[22] http://ifs.tuwien.ac.at/dp/vocabulary/quality/measures
[23] https://github.com/openplanets/policies/issues

3 Evaluating the user stories

In the second half of the SCAPE project, the experiments focussed on large scale execution in the sense of large amounts of data and improved performance through scaling. This followed the planned design of the work package tasks.

> Task 10: Large scale workflow execution and benchmarking (M34 - M36)
> Task 11: Improve large scale preservation workflows (M37 - M40)
> Task 12: Improved large scale workflow execution and benchmarking (M41-M42)

For the fourth testbed, there is a similar task stating:

> Task 3: Execution and benchmarking in data centers (M35-44)
> This task corresponds to tasks 10 and task 12 defined in the SCAPE work plan for its three testbeds.

This chapter will focus on the scalability aspects, for user stories, experiments and evaluations, and aims at reviewing all user stories with this in mind.
To better understand the structure, note that experiments are tied together with a platform and a dataset. As an outcome, it is often not possible to compare experiments and their evaluations directly, as the used hardware, software and dataset need to be considered as well.

3.1 User Story Review

The following sections reviews the user stories that are part of the SCAPE testbeds and discusses the results that have been obtained through the associated experiments. The review is built of three elements: (1) user story goals, (2) solutions, (3) results and findings.
It was decided to present as many of the important results as possible, without writing all the technical and non-technical details for all user stories, experiments and evaluations. Therefore, in those cases where you as a reader want more information, we refer to the appendices in this report.
In this section, we have decided to present the

Some user stories are new, compared to the list in the SCAPE deliverable D18.1, *First evaluation report*, and some have been transferred from one or more scenarios into a user story. For these continued user stories, there will be a note in the description about the names of the previous scenarios.

3.1.1 Web Content Testbed

In the Web Content testbed, three user stories have been worked upon by SCAPE partners, where focus, as the title also indicates, is on handling web data.

ARC to WARC Migration

This user story has two goals, one being about migrating ARCs to WARCs in a timely fashion, and a second about ensuring the completeness of the migration.

As part of a solution, a tool named Hawarp[24] has been developed, which performs the migration from ARC format to the WARC format. Also, the ToMaR[25] tool, which is used to wrap or package other tools into a Hadoop component, has been assessed.
The purpose of the experiment is to test the performance of two different approaches of implementing a large-scale ARC to WARC migration workflow. The findings[26] are visualised in the charts below.

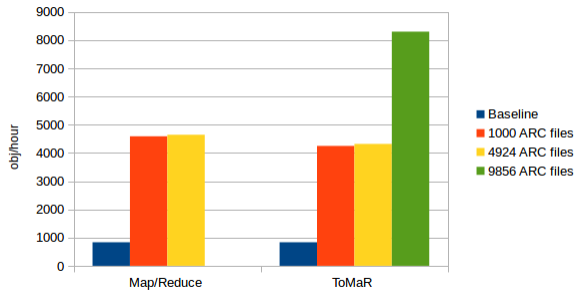


Figure 1 Performance without Apache Tika

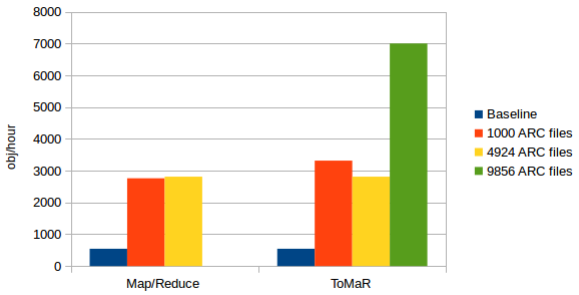


Figure 2 Performance with Apache Tika

These findings show an increase in performance for a native Map/Reduce implementation compared to integrating the ToMaR tool; and that running Apache Tika[27] on top[28] of the migration will slow the overall performance, as expected.

Comparison of Web Snapshots

This goal of this user story is to provide an automated and scalable visual comparison tool to web archives. To reach this goal, the IM team worked closely with the UPMC team, developer of the comparison tool Pagelyzer, through several iterations of development, annotation and evaluation.
The workflow defined for this story consists in generating images of a web page found in a WARC/ARC file and/or on the live web and then to compare these two images to provide a similarity score.
In the first evaluation of results report D18.1, this scenario was named "*WCT1: Comparison of web pages for quality assurance (Internet Memory Foundation)*". The visual comparison tool named Marcalizer, the first version of Pagelyzer, was developed during the work carried out and the evaluation is found in Section 17.6.

In the second experiment of this user story, the workflow evolved and the tool was wrapped.
The input is a list of URLs and a list of web browsers. In the list of web browsers the first one is taken as a reference. Hence, for each URL on the input - the URL pointing to the web archive - a screen shot is taken for each browser from the input list. Subsequently, each of these screenshots is compared to the one described as the reference one. The scores are stored for reporting purpose. There were 3 goals to watch in this experiment:

- parallelization – the scenario got parallelized using MapReduce

[24] https://github.com/openplanets/hawarp/tree/master/arc2warc-migration-cli
[25] https://github.com/openplanets/tomar
[26] Further information can be found in sections 11.2, 17.2, 17.3, 17.4, 17.5
[27] http://tika.apache.org/
[28] Running Apache Tika is will add value, as data is characterised; though not a functional requirement.

- integration – the results are properly communicated to SCOUT[29]
- performance – should be similar to the one from the previous experiment

The full description of the second experiment can be found in sections 11.4 of this document. Its evaluations can be found in sections 17.6.

Figure 3 depicts the comparison between the throughputs of the first and second experiment. The throughput in is comparisons per hour. There are several reasons for such a growth:

- parallelization – usage of 4 cores instead of one
- usage of better CPUs (roughly 2x faster)
- code refactoring – in order to parallelize the task the wrapper code has been rewritten from Python to Java and is executed within one VM

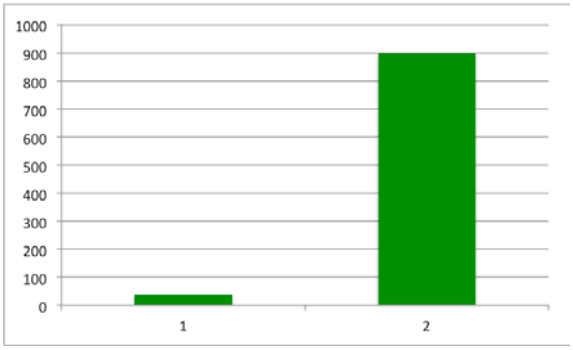


Figure 3 Throughput of comparisons per hour (experiment 1 and 2)

Within the third experiment, the IMF Platform 2 (extended central instance, see the evaluation's platform description for full details) was used. It consists of 43 nodes running the SCAPE platform (16 MapReduce nodes, 43 HDFS nodes, 10 HBase nodes), where MapReduce and HBase are not hosted on the same nodes and all nodes take part in the distributed HDFS file system. The dataset (WARC files representing the web crawl + HBase table with metadata needed to address the web archive) is hosted on the same set on nodes. Therefore, when rendering a page from the archive, the same set of machines participates in the process.

The experiment lasted for 160 hours. It took as an input 2.6 millions of URLs from an IMF large scale crawl, rendered the archived page and the live page and compared the result using Pagelyzer. In the evaluation, the times recorded to get the page and to render the page were consistent with the previous experiments as shown within their evaluations. We therefore present below the comparisons scores rather than the performance metrics.

[29] http://scout.scape.keep.pt/web/

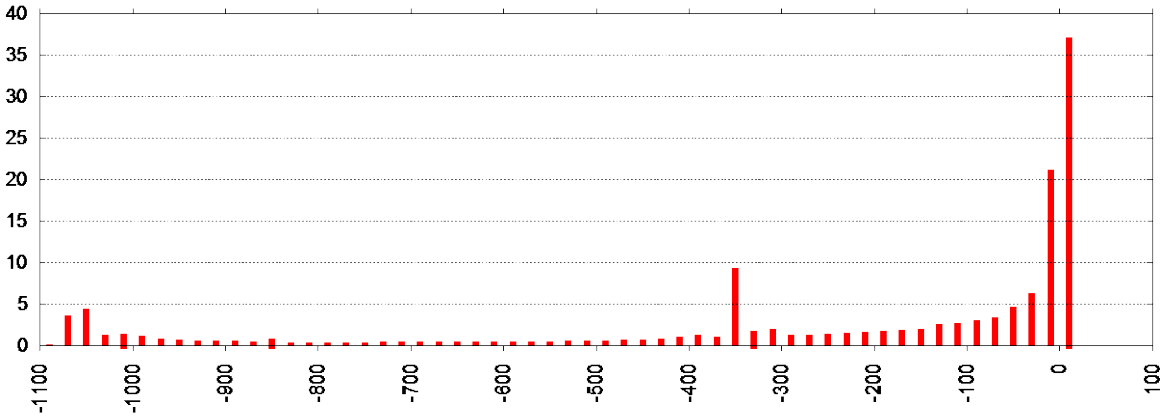


Figure 4: Pagelyzer score frequency (Experiment 3)

In Figure 4 the frequencies of the acquired scores are depicted. The x-axis represents the scores aggregated into bins – the size of bin is 20. On the y-axis, the percentage of the overall scores in the particular bin is presented. Recall that the Pagelyzer results range from [-10 000, 1], where the higher the score the more similar the pages are. It can be noted that about 37% of the pages have score between 0 and 1 thus identified as very similar. The next bin [-20, 0] records 22%. This together gives about 59% of the comparisons as very similar.
The low scores might have three causes:

1. the live page changed significantly since the time of crawl (as the experiment was launched after rather than during the crawl).
2. the web archive page is not complete or not correctly rendered.
3. Pagelyzer failed to correctly assess similarity of pages (see correctness benchmarking of Pagelyzer in MS54 report[30]).

This third experiment will be followed by a correctness evaluation on a sample of similar and dissimilar pages. Further experiments are also planned after the project end on samples of running crawls to finalise a QA workflow that could be used in-house.
The full experiment is described within section 11.5 and its evaluation is available within 17.7.

File Format Identification and Characterisation of Web Archives

The goal for this user story is to process both ARC and WARC files, and identify file formats and characterise items contained in these archives. As part of these goals, the tools FITS[31], Nanite[32] and ToMaR have been assessed, details can be found in the experiments and evaluations sections. Nanite and FITS both utilizes a range of other characterisation and identification components, such as Tika, DROID and file, and they can be seen as alternatives to each other.

[30] https://portal.ait.ac.at/sites/Scape/Management/Lists/Milestones List/DispForm.aspx?ID=54
[31] https://code.google.com/p/fits/
[32] https://github.com/openplanets/nanite

This is another user story, the origins of which can be found in scenarios "*WCT3 Characterise web content in ARC and WARC containers at State and University Library Denmark (SB)*", "*WCT4 Web Archive Mime-Type detection at Austrian National Library (ONB)*" from the first evaluation report.

Findings[33] about throughput are illustrated below. Please note that the numbers from the three organisations are not directly comparable, it is necessary to take other aspects into consideration. For instance, the hardware platforms are widely different; as are the tools in use.
What can be seen from the charts is the progress that has happened for the individual experiments.

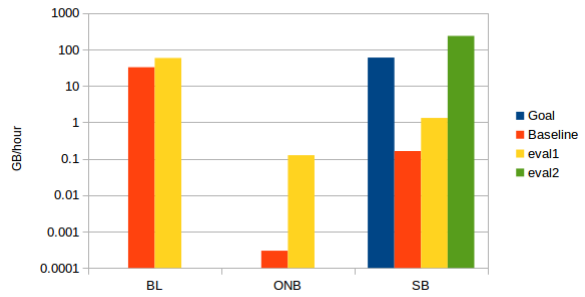


Figure 5 Throughput in GB/hour (logarithmic scale)

All three organisations have successfully obtained better performance over time. The internal goal at SB was reached in the second iteration; no goals were set for experiments performed by BL and ONB.

The decision to use and develop Nanite further for this experiment has proved to have been a sound one. Nanite benefits greatly due to being tightly coupled with Hadoop, and making use of pure-Java libraries so no external applications are called. After initially reducing the runtime by almost 50%, further work was undertaken to add in full characterisation of the input files, which proved to perform better and compared favourably to other methods of characterisation at scale. Nanite is a good base for future work on gleaning more information from web archives and can be easily extended further. One of the BL web archive collections totals 30TB of compressed (W)ARC files, and using Nanite to characterise that data on the same test cluster would be expected to take 68 days, which is acceptable.

3.1.2 Large Scale Digital Repository Testbed

The Large Scale Digital Repository testbed contains seven user stories, spanning a wide variety of different workflows and datasets.

Characterisation of Large Audio and Video Files

This user story is about characterising very large audio and video files, and thereby having the option to evaluate the collection for preservation risks and ongoing risk management.
A single iteration[34] of the experiment has taken place, but the large scale execution was not initiated[35].

[33] Further information can be found in sections 11.6, 11.7, 11.8, 11.9, 17.7, 17.9, 17.10, 17.11
[34] Experiment and evaluation are found in sections 12.1 and 18.1
[35] Resources were re-prioritised and the experiment had to be paused

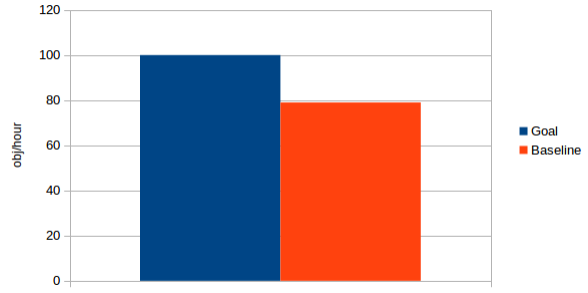


Figure 6 Processed objects per hour

The chart above shows that the internal goal was set to 100 objects to be processed in one hour, and the experiment reached close to 80 objects as the initial non scaled approach. If a large scale approach was introduced to the experiment, it would almost certainly reach the goal.

Large Scale Audio Migration

As the title states, this user story concerns migration of a large numbers of audio files from one format to another; but also ensuring that the migration is a good and complete copy of the original. This story can also be found in the first report, as the scenario "*LSDR6 Large scale migration from mp3 to wav (SB)*".
The development of the xcorrSound[36] tool was part of this solution, and a range of other tools were integrated into the migration workflow with the purpose of migration and quality assurance.
Even though the goal of 1000 objects handled per hour was not accomplished, a great performance gain has been achieved[37] and distributing work seems like a promising solution.

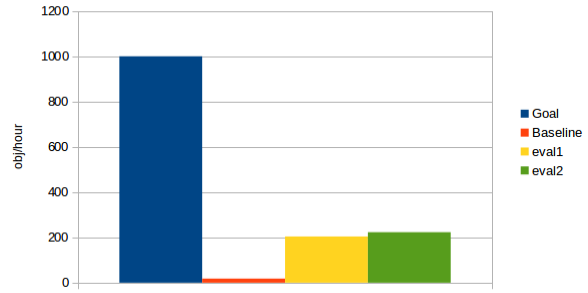


Figure 7 Processed objects per hour

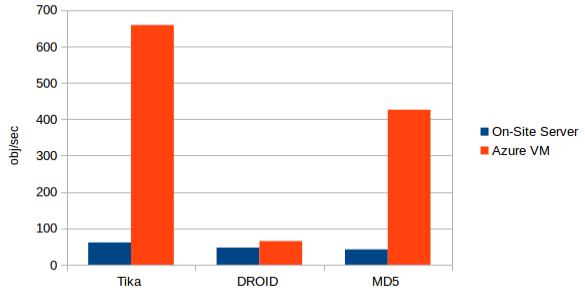
As for correctness, we believe[38] that the automatic QA correctly identifies the "questionable" migrations, such that these can be checked in a manual QA process. However, we must ensure that the number of migrations to check manually is minimal, as this is a very resource demanding process. The goal for QAFalseDifferentPercent has been changed to 2%. This means that we would have to check 3500 migrated 2 hour wav files manually. This is already too resource demanding. However the poor quality of the original files is a great challenge for the content comparison tool.

[36] http://openplanets.github.io/scape-xcorrsound/
[37] Evaluations are found in sections 18.2 and 18.3, experiments are found in section 12.2 and 12.3
[38] This was observed in evaluation 18.2

The workflow does scale, and even though the migration of the collection cannot be done in 1 week, it will be possible to do in one month on the SB Hadoop cluster; which is considerably better than the one year needed without Hadoop. Another solution would be to scale the cluster with a factor of 5, which should give the necessary performance increase. This is backed by the fact that there are no dependencies between processes or data; it requires a lot of reading and writing of data, so the storage devices need to keep up.

Large scale document characterization and identification with Tika and DROID on SCAPE Azure platform

The main goal is to evaluate which platform users should be using to run characterization and identification tools. The Azure platform[39] is compared with a traditional on-site server.
More details can be found in section 12.4 and 18.4, being experiment and evaluation.



The chart shows significant differences between the two platforms, where the Azure platform reveals better performance. An interesting point is the DROID results, which do not show the same differences between the two platforms; further investigation into this matter is needed to reveal more details.

Large Scale Image Migration

For this user story, the goal is to be able to do a migration of a large number of images from one format to another, ensuring that the migrated images conform to an institutional profile and that no image data is lost. It has its origin in the scenarios "*LSDR2 Validating files migrated from TIFF to JPEG2000 (BL)*" and *"LSDRT3 Validating Migrated Images 'Visually'"*.
Two organisations have been working on this topic, providing two different approaches to solutions; and the outcomes are different, as expected. There are also similarities, and the charts below give some insight into the performance. More details can be found in experiment sections 12.5, 12.6, 12.7 and evaluation sections 18.5, 18.6 and 18.7
One of the tools being used for quality assurance, Jpylyzer, was developed during the SCAPE project.

[39] See section 15.4 - MSR Azure Platform

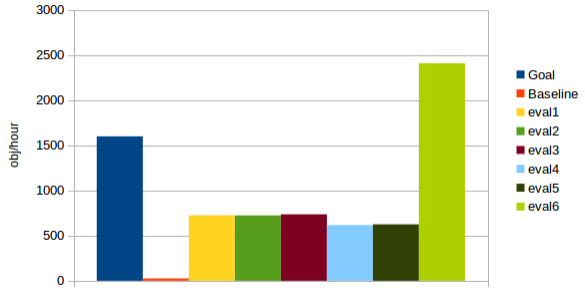


Figure 8 Experiment conducted by BL

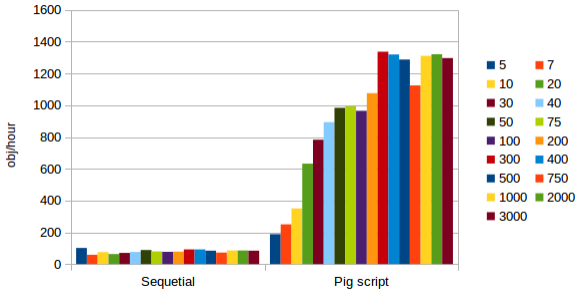


Figure 9 Experiment conducted by ONB

The experiment by ONB has input size as legend, spanning from 5 to 3000, showing that processing large datasets gives better performance compared to small datasets.

An internal metric goal (set by BL), being able to process 1600 objects every hour, was reached by the BL experiment, while the ONB experiment was close. The numbers from BL and ONB experiments are not directly comparable, as it is necessary to take into account the differences in software and hardware. Software and hardware have impact on the numbers, and BL and ONB are using different hardware and software.

Large Scale Ingest

The user story examine ingestion of a large number of digital objects and associated metadata into a digital repository - securely, correctly and with acceptable performance. The experiments carried out have been focusing on Fedora 4[40] as the digital repository, and investigate the possibilities of a clustered solution.

The conclusion[41] arrived at, after a number of performance tests, was that Fedora 4 performance drops significantly when using more than a single node, and the average throughput in a distributed environment is not sufficient for real life large scale ingests.
Fedora 4 is built on top of Modeshape[42] and the plain performance of this JCR implementation was assessed in a distributed environment, and it was shown that Modeshape itself was not capable of producing satisfactory results. After gathering these results, the Fedora 4 steering committee[43] decided to drop scalability as a key feature for the first release and plans to better integrate horizontal scalability in the second release of Fedora 4.

The integration of Fedora 4 with SCAPE is still being developed further to keep up to date with the development of Fedora 4.

We first measured the performance of a stand-alone node on a single desktop machine. This result is called “local” in the plot. Next we measured the ingest performance on the local desktop box and the

[40] https://wiki.duraspace.org/display/FF/Fedora+Repository+Home
[41] For more information, see section 12.8
[42] http://modeshape.jboss.org
[43] https://fedoraproject.org/wiki/Fedora_Engineering_Steering_Committee

Fedora as a single node on the OpenNebula Cluster (behind a load balancer and without a load balancer). The result can be seen in the following box plot

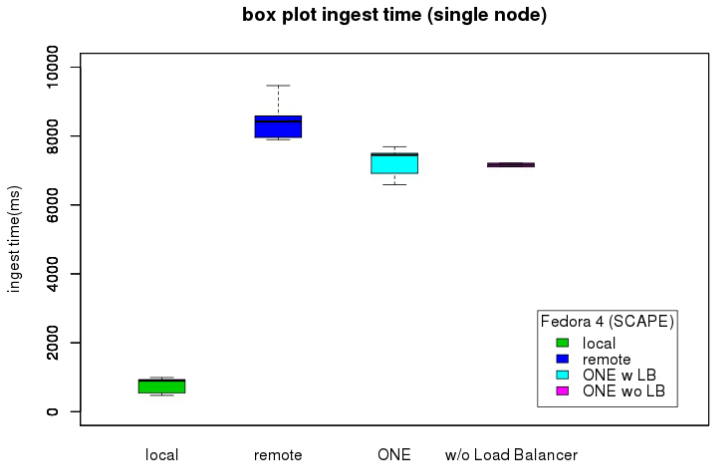


Figure 10 Single node ingest duration

As can been seen the Load Balancer has no influence on the average value of the ingest time, only the spread of the values is broader.

Second we set up a cluster with 3 nodes and with 6 nodes and ran the test with the Load Balancer in front again.

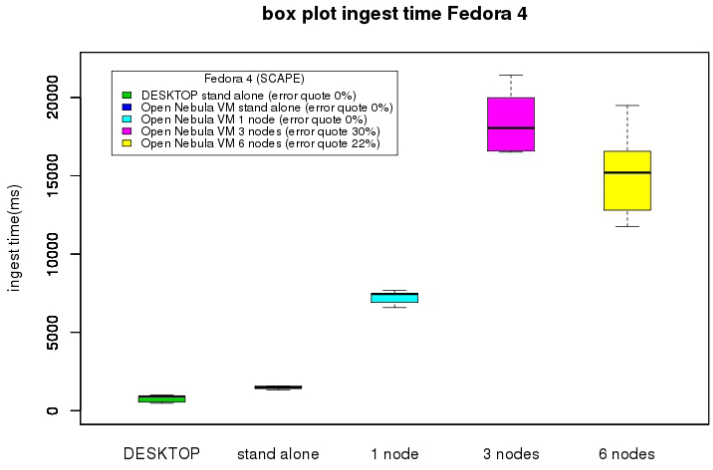


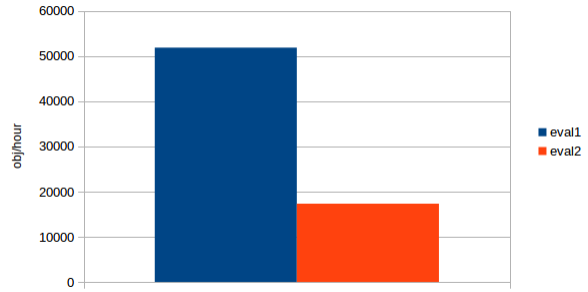
Figure 11 Ingest duration using load balancing

As can be seen from the plot the ingest time increases dramatically if one puts more nodes in. The difference between 3 and 6 nodes is statistically not relevant. Compared to a non-clustered environment (called stand-alone in the plot) the ingest time rises from 900ms on a Desktop Suse box and ca. 1200ms on a VM at the Open Nebula cluster in stand-alone configuration to around 17000ms, which is a factor of 15.

It was concluded that Fedora 4 in its current state is not an option for using as a well performing distributed large scale repository. As Fedora 4 is still in development performance might improve over the course of time. Therefore the SCAPE implementations on top of Fedora 4 will be kept up to date with Fedora 4 development.

Policy-Driven Identification of Preservation Risks in Electronic Document Formats

The goal is to sustainably manage collections by identifying specific preservation risks, either at ingest or at some later stage. Holding large numbers of electronic documents from various sources are potential risks for long-term accessibility and preservation.



For this experiment, two evaluations can be found in the chart above. The rather big difference is as expected, as the second run is doing more content checking, which takes longer. Testing with the policy checks takes approximately three times as long as the basic checks.
As the two runs are quite different, it is not possible to compare them to each other. It is the end results of approximately 51000 and 18000 objects processed per hour that are of interest.
Extrapolating from the test dataset for this evaluation[44], it would be possible to process 1TB of PDF files, with policy checks, in less than 4.5 days on the same Hadoop cluster. This is acceptable for using on a routine basis, should that be necessary.

Validation of Archival Content against an Institutional Policy

The goal here is to ensure that content conforms both to its file format specification and (where appropriate) the profile of that format as specified by the institutional policies.
The tool, Jpylyzer, has been used as one example, and prototypes of the SCAPE loader, SCAPE stager and SCAPE connector API were implemented[45].

[44] More information can be found in section 18.8, experiment is found in section 12.9
[45] https://github.com/openplanets/scape-stager-loader-SB

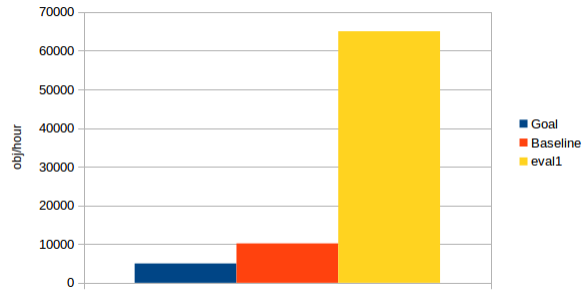


Figure 12 First iteration

Through the first iteration[46], the internal goal was reached, and with the current setup there is a significant positive difference between the goal and eval1 results.
A second iteration was planned, and a prototype was run showing promising functionality and results; no large scale tests have been performed. The second iteration will test integrating with repositories.

3.1.3 Research Datasets Testbed

The user stories in this testbed are different from the two first testbeds in the sense that they mostly concentrate their use cases around scientific data.

Migration from local format to domain standard format

In this user story, the aim is to migrate scientific data held in a local format into a domain standard format to reduce the risks of losing the ability to read/use and reuse the data contained within the file format. Its origin can be tracked back to the scenario "*RDST2: Format migration from RAW to NeXus: moving from a local format to the domain support standard*"
Many different experiments have been carried out with many different configurations and input sample collections, and they have shown some interesting results. Some performance gains have been seen, but some major issues have arisen, see experiment and evaluations in sections 13.1, 19.1, 19.2 and 19.3

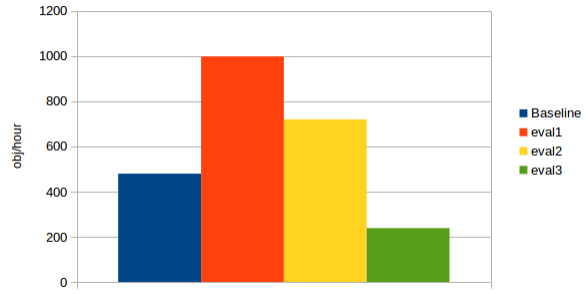


Figure 13 Evaluation: raw2nexus small dataset evaluation

As can be seen from the chart above, some experiments have shown far better performance than the baseline. However, from the large scale experiments that have been run and the monitoring data

[46] Evaluation is found in section 18.9, experiment is found in section 12.10

from the Ganglia system it can be seen that the migration process is Input Output (IO) bound. Processing of files which are small < 129Mb is possible but when larger files are processed the system struggles if the tasks are run in parallel and the migration process fails. The file size of the test datasets are quite modest with the largest at < 500Mb and these are relatively small files that can be produced by the systems at ISIS and Diamond, where the size of files being produced are often in the 10s of Gb. Moving the files onto the specialised HDFS and then off again after processing is very time consuming.

Identification, validation and checksumming of a complex corpus

Being a content holder of geospatial data, here the goal is to ensure files can be identified, to create/check fixity and validate formats, where appropriate, thereby being confident that the data formats are valid and that the data does not change over time.

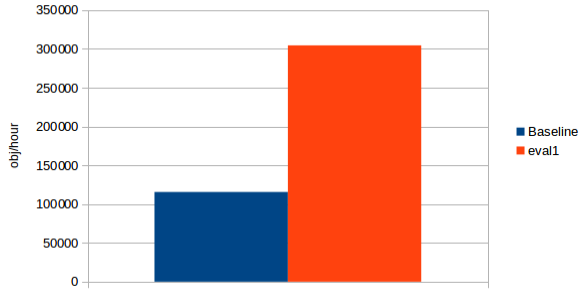


Figure 14 Scaling out to improve performance

As the chart shows, a significant improvement was gained. The final result of this evaluation, with a runtime of eight and a half hours[47], is a reasonable time for performing all those tasks.

3.1.4 Data Center Testbed

The user stories herein are about video and modelling, as well as the collaboration between institutions that produces data and institutions that stores the data.

Large scale video processing and interlinking

Researchers are dealing with SLAM (simultaneous localization and mapping) and visual geo localization and need to preserve results of large-scale scene reconstruction and rendering, together with related source objects and parameters of the computation process. The results will be available for (re-)use and further refinement in a long term.

The first experiment[48] was focused on measuring the computation time per item of different parts of a SLAM experimental application while varying the number of currently used computation nodes. Results of the metadata analysis will characterize the average computation time of each mentioned processing stage.

[47] For more information, see experiment in section 13.2 and evaluation in section 19.4
[48] Experiment and evaluation is found in section 14.1 and 20.1, respectively

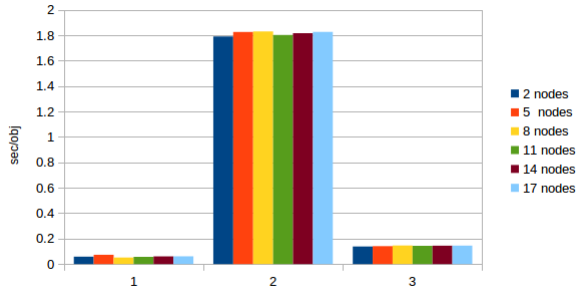


Figure 15 Timings for clustered processing

1. time of loading JPEG file in seconds
2. time of extracting features from images in seconds
3. time of matching extracted features in seconds

The results show that varying the number of computation nodes has no statistically significant effect on the item processing time.

This second experiment[49] explores relations among the frequency of edge sampling, the amount of memory and the time needed to compute the alignment with a certain sampling frequency and the corresponding alignment quality. The frequency of sampling is varied and the experiments are performed for each sampling value. The average value of memory and time consumption and the percentage of successfully aligned cases are computed.

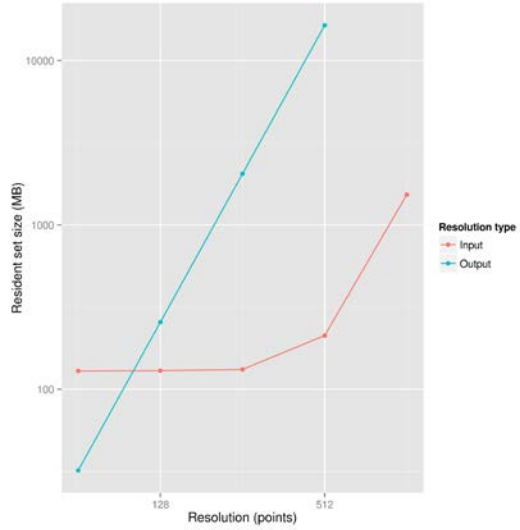


Figure 16 Memory consumption

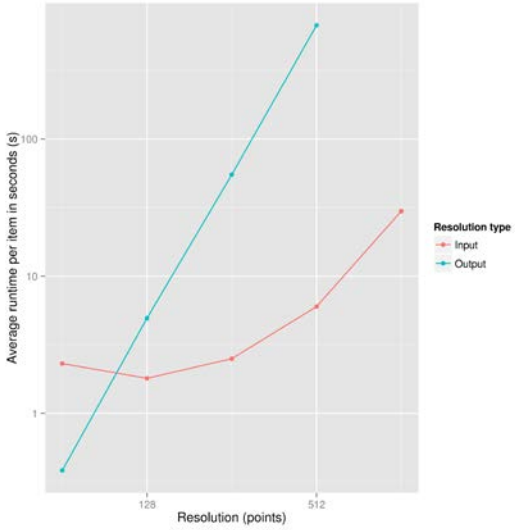


Figure 17 Performance

[49] Experiment is in section 14.2, and evaluations are in sections 20.2, 20.3 and 20.4

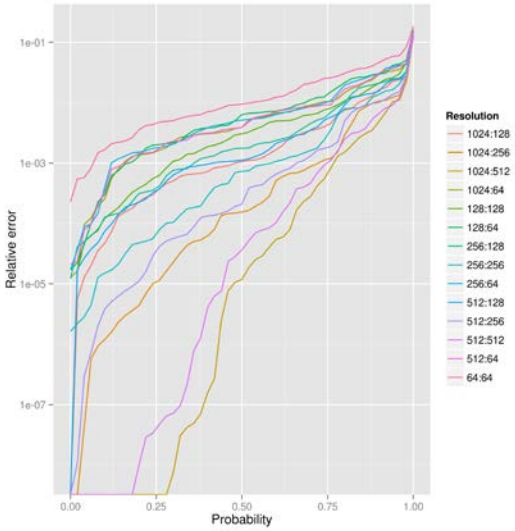


Figure 18 Precision of alignment

Memory consumption, computation time and quality of camera pose estimation were measured for many different combinations of input and output resolution of the spherical cross-correlation. The measurements indicate that the quality of results depends mainly on the output resolution, while the input resolution has much smaller effect. On the other hand, the consumption of computation time and system memory grows dramatically with the increase of output resolution.

The profiling information collected during the experiments will be used for further development of relevant applications.

A detailed presentation of the results can be found in Deliverable D23.2.

Large scale access to medical data at hospital (Medical Dataset)

The goal is to have access to preserved medical data, including access to the entire history of patient's treatment accessible directly from the Medical Data Center. The data is available via a dedicated API, and access is done at the hospital premises by the hospital staff.

The chart below shows the relation between download speed (in objects per second) and the number of download threads used in the test[50]. The first 5 tests show that a single client (named C1) computer can reach up to approx. 16 objects per second when using 20 concurrent download threads (which is maximum, because the performance of the client computer starts to significantly decrease, when the number of download threads is above 20). When using two client (named C1 and C2) computers, it was possible to retrieve around 27 objects per second (using 40 threads).

[50] Experiment is described in section 14.3, evaluation is in section 20.5

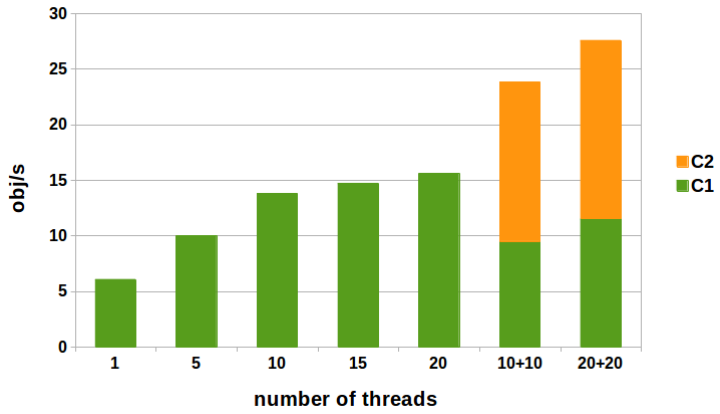


Figure 19 Performance of the access interface

Large scale access to medical data for educational purposes (Medical Dataset)

For this user story, the goal is to have access to data samples of various diseases, which can be used during university courses and also to showcase good practices from various treatment cases.

The evaluation[51] is related to the performance of the search function in the Medical Data Center access portal. The search function provides possibility to specify a set of criteria that will be used to search for medical cases. The evaluation investigated response time of the search function depending on the number of criteria applied in the search.

The table below presents statistics related to mean response time of the search function. From the statistics it is visible that the response time is significant, but still acceptable in the context of educational scenario (where a teacher presents and discusses medical cases). The reason for long response times is that the search function depends heavily on the HBase[52] tables available in Medical Data Center (during the search process HBase tables are analysed and also created). For the future work it can be considered to use dedicated solution for the search function, e.g. ElasticSearch or Solr. Nevertheless for the educational purposes, the current performance is sufficient.

Search criteria	Number of threads	Mean response time
Search medical cases by ICD10	1	2.4035
Search medical cases by ICD9 code		11.025
Search medical cases by patient's city		7.8035
Search medical cases by patient's sex		14.3115
Search medical cases by patient's age		5.527
Search medical cases by visit's dates		5.728
Search medical cases by laboratory tests		4.0595
Search medical cases by all of the above criteria		10.5015
Search medical cases by ICD10	10	4.556
Search medical cases by ICD9 code		11.6135
Search medical cases by patient's city		10.969
Search medical cases by patient's sex		16.2625
Search medical cases by patient's age		8.4365
Search medical cases by visit's dates		7.0335

[51] Experiment in described in section 14.4, evaluation is described in section 20.6

[52] http://hbase.apache.org/

Search medical cases by laboratory tests	4.7715
Search medical cases by all of the above criteria	13.743

Table 2 Search function response time

Large scale analysis (Medical Dataset)

Here the goal is about analysing large amounts of medical data related to patients' treatment; for research activities and to calculate statistics at hospitals. The following queries were implemented and evaluated:

- Age of patients treated in a given period
- Sex of patients treated in a given period
- Number of cases of a given disease in a given period
- Number of abnormal results in laboratory examinations for a given disease codes in a given period
- Average time of patient's visit for a given disease codes in a given time period

Results of evaluations[53] related to specific analysis are presented and discussed in the following part of this section.

The main goal of the first evaluation was to obtain statistics on the age of patients treated at WCPT in a given period. The age intervals are the input parameters for analysis algorithm. As the evaluation metric the number of objects per second has been selected (the object is defined as a single record in the HBase table, and each HBase table row stores information about the age of patient who visited WCPT hospital). The algorithm was executed three times with three different periods to be analysed. The table below presents statistics related to the evaluation.

Parameter	Test1	Test2	Test3
Number of objects processed per second	2592 [obj/s]	2417 [obj/s]	1465 [obj/s]
Analysed period	1.07.2012-1.07.2014	1.01.2013-31.12.2013	1.01.2014-1.05.2014
Processing time	65 [s]	61 [s]	7 [s]

Table 3 Summary of evaluation related to statistics on age of patients treated at WCPT

The chart below presents results of analysis for Test3. Colours indicate different age ranges for the patients who visited WCPT hospital (the exact age range is given in the middle-right part of the chart and on the chart itself). Each colour on the pie chart has related entry (note). Each entry is composed as follows: X-Y = Z [P], where X-Y is the age range (patients between age X and Y), Z is the number of patient's visits (indicated the number of visits for specified age range and analysed time period) and P is the percentage of the number of patient's visit in the overall context. An example can be an entry for yellow colour: 41-60 = 309 [35%] - it means that yellow represents percentage of patients (35%) in age between 41 and 60 (including) which visited WCPT hospital in the period 2014-01-01 - 2014-05-01.

[53] For more information, experiment in section 14.5 and evaluations in sections 20.7, 20.8, 20.9, 20.10, 20.11

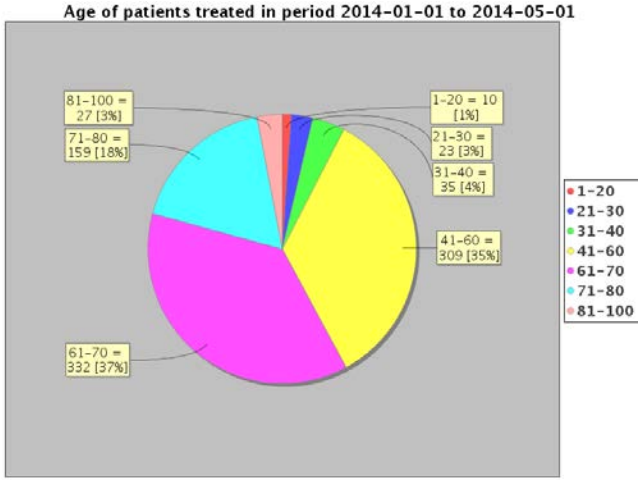


Figure 20 Age of patients treated in period 2014-01-01 to 2014-05-01

The main goal of the second evaluation was to obtain statistics on the average time of visit for patients treated at WCPT in a given period and because of a specific disease (indicated by ICD10 codes). The period of time and ICD10 codes are the input parameters for analysis algorithm. As the metric the number of objects per second was used (the number of records processed per second). The table below presents statistics related to the evaluation.

Parameter	Test1	Test2	Test3
Number of objects processed per second	2812 [obj/s]	2569 [obj/s]	1438 [obj/s]
Analysed period	1.07.2012-1.07.2014	1.01.2013-31.12.2013	1.01.2014-1.05.2014
Processing time	59 [s]	57 [s]	7 [s]

Table 4 Summary of evaluation related to statistics on average time of visit for patients treated at WCPT

The chart below presents results of the analysis for Test 2. Colours indicate different ICD10 disease codes. Test has been performed for the patients who visited WCPT hospital between 1-01-2013 and 31-12-2013. Each column indicates the average time of patients' visits. Descriptions of the ICD10 codes investigated in this analysis are as follows:

- A15.0 - Tuberculosis of lung, confirmed by sputum microscopy with or without culture
- A15.1 - Tuberculosis of lung, confirmed by culture only
- J85.1 - Abscess of lung with pneumonia

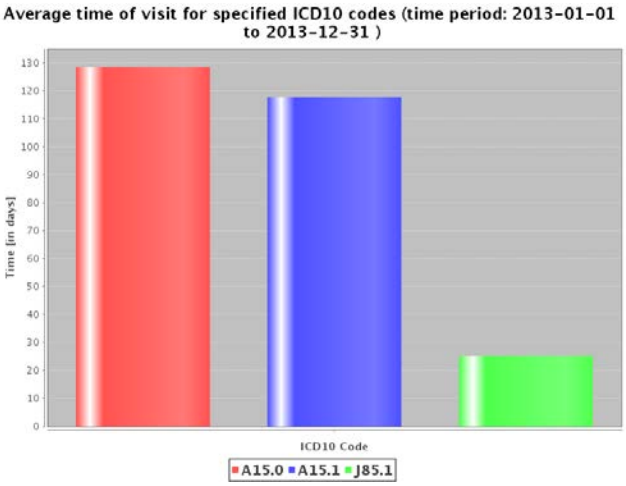


Figure 21 Average time of visit for specified ICD10 codes

The main goal of the third evaluation was to execute analysis on the number of abnormal laboratory examination results for a given disease codes in a given period. The investigated period and list of ICD10 codes are the input parameters for analysis algorithm. As the evaluation metric the number of objects per second has been selected (the object is defined as a single HL7 file stored in HDSF). The table below presents statistics related to the evaluation.

Parameter	Test1	Test2	Test3
Number of objects processed per second	4196 [obj/s]	4761 [obj/s]	4979 [obj/s]
Analysed period	1.07.2012-1.07.2014	1.01.2013-31.12.2013	1.01.2014-1.05.2014
Processing time	80 [m]	71 [m]	68 [m]

Table 5 Summary of evaluation related to statistics on abnormal results in laboratory examinations for WCPT patients

The chart below presents results of analysis for Test 2. Colours indicate different ICD10 disease codes. Test has been performed for patients who visited WCPT hospital between 1-01-2013 and 31-12-2013. Each column indicates the number of abnormal results in laboratory examinations for all patients. The ICD10 disease codes investigated in this analysis are as follows:

- A15.0 - Tuberculosis of lung, confirmed by sputum microscopy with or without culture
- A15.1 - Tuberculosis of lung, confirmed by culture only
- J85.1 - Abscess of lung with pneumonia

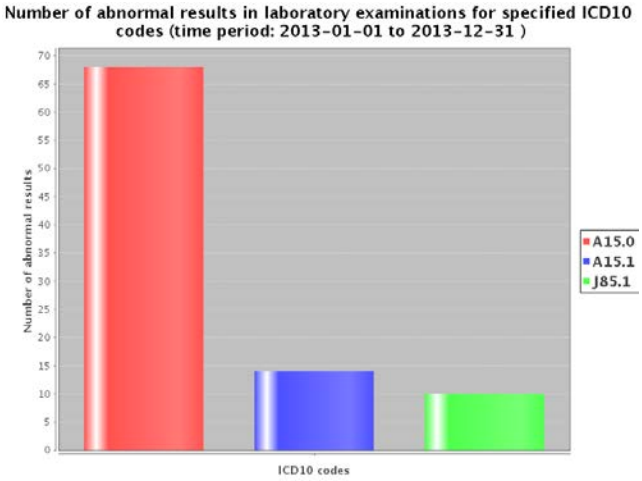


Figure 22 Number of abnormal results in laboratory examinations for specified ICD10 codes

The main goal of fourth evaluation was to obtain statistics on the number of medical cases related to a given ICD10 code in a given period. The analysed period of time is additionally split into a given number of sub-periods. The analysed period, number of sub-periods and the ICD10 code are the input parameters for analysis algorithm. As the metric the number of objects per second has been used (the number of records processed per second). The table below presents statistics related to the evaluation.

Parameter	Test1	Test2	Test3
Number of objects processed per second	2563 [obj/s]	3731 [obj/s]	2041 [obj/s]
Analysed period	1.07.2012-1.07.2014	1.01.2013-31.12.2013	1.01.2014-1.05.2014
Processing time	64 [s]	39 [s]	5 [s]

Table 6 Summary of evaluation related to statistics on the number of WCPT patients related to specified ICD10 codes

The chart below presents results of analysis for Test 2. Colours indicate different sub-periods of time. Test has been performed for the patients who visited WCPT hospital between 1-01-2013 and 31-12-2013. This period is split into 5 sub-periods as seen on the chart below (each sub-period corresponds

to one column). Each column indicates the number of patient visits for a given ICD10 code in a given sub-period. The ICD10 code in this test was set to J85.1 - Abscess of lung with pneumonia. The total number of cases found in a given period is presented on the chart as well (it is 33 in this particular case).

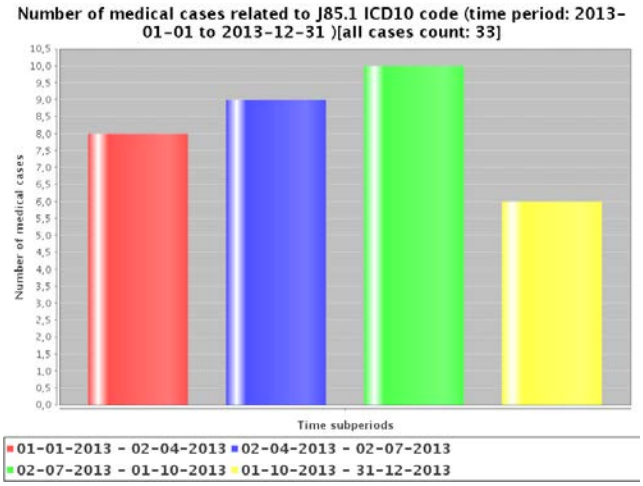


Figure 23 Number of medical cases related to specified ICD10 code

The goal of this evaluation was to compute statistics on the gender of patients treated in a given period. The period of time to analysis is given as the input parameter for the analysis algorithm. As the metric the number of objects per second (number of records processed per second) has been selected. The table below presents statistics related to the evaluation.

Parameter	Test1	Test2	Test3
Number of objects processed per second	1798 [obj/s]	1600 [obj/s]	984 [obj/s]
Analysed period	1.07.2012-1.07.2014	1.01.2013-31.12.2013	1.01.2014-1.05.2014
Processing time	95 [s]	94 [s]	10 [s]

Table 7 Summary of evaluation related to statistics on the gender of patients treated at WCPT

The chart below presents results of the analysis for Test 2. Colours indicate gender of the patients. Each colour on the pie chart has related entry (note). Each entry is composed as follows: Y = Z [P], where Y is the name of gender, Z is the number of patient's visits (indicates the number of visits for analysed time period) and P is the percentage of the patient's visit in the overall context.

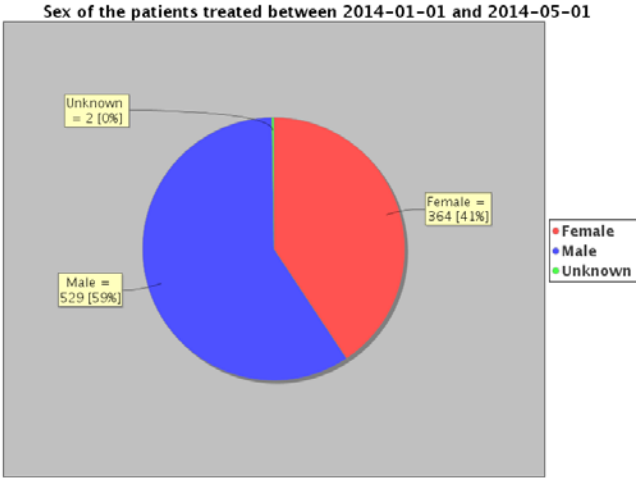


Figure 24 Sex of the patients treated in a given period

All of the above evaluations showcase that the medical data can be easily processed using large scale analysis tools. The performance of the analysis increases with the volume of data that need to be processed (the more data need to be processed the more data is processed per second). It confirms that parallelisation is more beneficial with large datasets.

Large scale ingest of medical data (Medical Dataset)

With this user story, the aim is a system for ingesting medical data for the purpose of archiving and processing. The reason is the lack of necessary resources (mainly storage space) due to the requirement (enforced by law) to store medical data for at least 20 years (30 years in some cases).

The first experiment[54] that includes copying data to archiving system, shows that a single client can reach up to approx. 1,2 objects per second, which is in fact limited not by the client computer but by the server (quite long time of storing files in the archiving system that uses hybrid storage space composed of hard drives and tapes). The charts below shows ingest using 1, 4 and 10 threads.

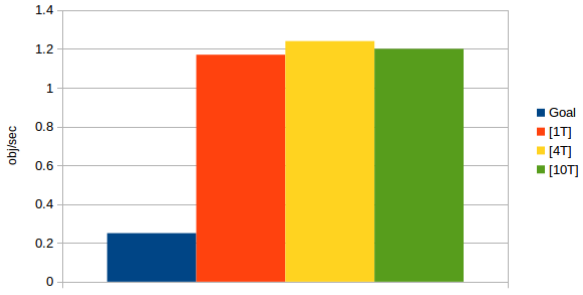


Figure 25 DICOM files ingest with archiving system

The second experiment, without copying data to archiving system, shows that a single client can reach up to approx. 45 objects per second, which is in fact the limit of the client computer. The chart shows ingest using 5, 10, 15 and 20 threads.

[54] Experiment and evaluations are described in sections 14.6, 20.12 and 20.13

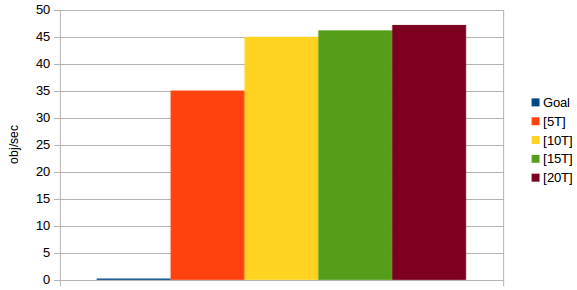


Figure 26 DICOM files ingest without archiving

For both experiments, the goal of 0.25 objects per second was reached. The goal was set by the fact that WCPT hospital produces 10GB of data per day and these data need to be transferred to the external archiving and processing system (in this case Medical Data Center). 10GB per days means minimum performance of 0.25 objects per second.

3.2 Concluding notes

In this chapter, we have looked at a very wide range of business areas, all of them being connected to digital preservation. We have given insight and a deeper understanding of the challenges that these institutions have and how they are searching for appropriate solutions.

One of the main issues that covers all of the described user stories is the vast amount of data, which needs to be processed and thereby also how this can be done. This often requires new ways of thinking about a potential solution, and the solution needs very much to consider the business area, data formats and infrastructure, among other things. Some of the other issues, found in the user stories, are the huge diversity of the data, as well as the number, size and complexity of the objects to be processed.

The list below summarises the topics that have been explored through the user stories with large scale handling in mind.

- Accessing content
- Analysis of medical data
- Validation of content against specification and/or institutional policy
- Ingest of content
- Feature extraction from content
- Identification of content
- Characterisation of content
- Migration of content
- Validation of action
- Quality assurance of content

3.2.1 Improvements of continued experiments

Some experiments have been running for the entire length of the project, with the purpose of better understanding the means available for getting the best performance and the best outcome. As examples from the figure below, the experiment named LSDRT2 is looking at image migration, while WCT3 is identifying and characterising web content.
To reach the best outcome, it has often required changes in the digital preservation tools in use, and thereafter re-running the experiments again.

The results of the second evaluation, which can be found in the figure below, have all been achieved using different parts of the SCAPE platform, running in a clustered environment.

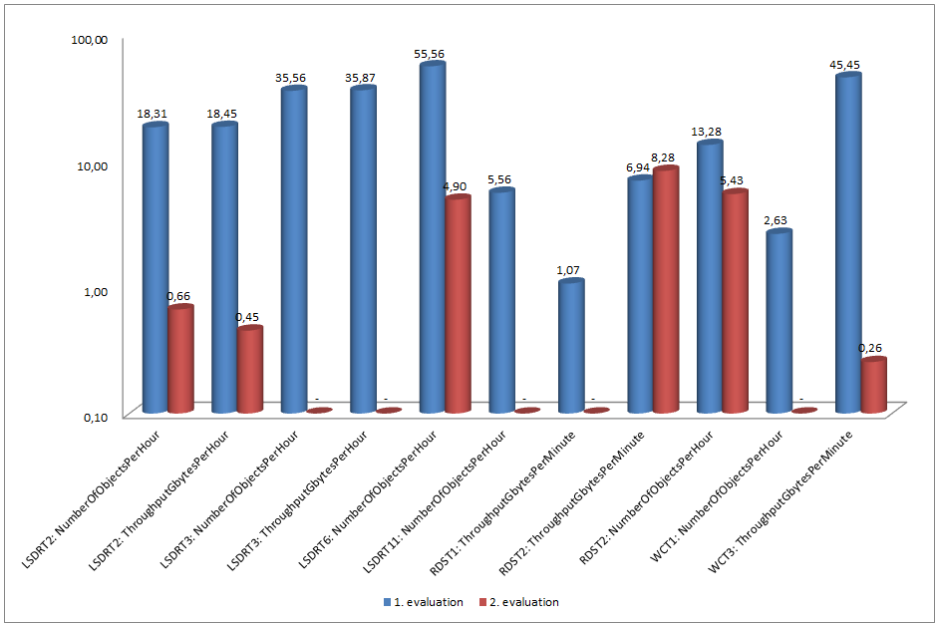


Table 8 Improvements from first evaluation (logarithmic scale in use)

The figure above shows what we have named an improvement factor for a number of experiments, where the blue bar shows the value from the first evaluation and the red bar shows the value from this, and second, evaluation. The value indicates how much more is needed to reach the goal; for instance the first blue bar (value of 18.31) indicates that the experiment needs to improve by a factor of 18.31.
To be able to get a nice overview, fitting the experiments and values into one chart, we have used a logarithmic scale. This was also the case for the first evaluation report, and it is appropriate to be consistent.
As we are using a logarithmic scale, when the value of the improvement factor is equal to or less than 1, this means that the goal has been reached.

For those experiments that have been continued, the transformation is as follows:

LSDRT2 → Large Scale Image Migration
LSDRT6 → Large Scale Audio Migration
RDST2 → Migration from local format to domain standard format
WCT3 → File Format Identification and Characterisation of Web Archives

Those columns that are without height and marked with a hyphen (-) for the second evaluation are experiments (LSDRT3, LSDR11, RDST1, WCT1), which have been discontinued or not completed and therefore no results are available.

The figure shows that two experiments have reached their goals, being LSDRT2 and WCT3.
The experiment LSDRT6 has improved performance, and would be able to reach the goal by scaling the clustered environment (see the associated user story in section 3.1.2).
As for the experiment RDST2, some aspects have improved while others have degraded; many of the issues are related to size of data (details can be found with associated user story in section 3.1.4, and sections 13.1, 19.1, 19.2, 19.3).

3.2.2 About testbed goals

The evaluation of some testbed goals as part of the work with user stories has for some experiments been easy, in particular for performance related goals. All experiments, no matter the associated user story, have evaluated some performance aspects, whether it is number of objects, throughput in bytes or sizes of objects.

As was previously described in section 2.2, it has been difficult to evaluate all testbed goals in all user stories, at least in direct connection to the work with experiments and testing.
The testbed goals *Reliability - Stability indicators* is included in those user stories, where tools have been developed to fulfil the tasks. More generally, guide lines and support have been handled in the XA.WP.2 work package, *Technical co-ordination*. Also, the joint continuous build site, http://projects.opf-labs.org/, shows status markers for individual SCAPE development projects.

Reliability - Runtime stability, the third testbed goal, is for most part of the experiments handled with different strategies and at different levels. Examples of strategies are exception handling and data cleansing to obtain the necessary stability; and this can be handled at levels such as for the tool, for the experiment that integrates the tool, or sometimes at the platform level.
The fact that experiments were able to finish successfully is a measure of reliability in itself.

Aiming for *Completeness* or *Correctness*, both categorised by *Functional suitability*, is considered, when further development has happened for the tools in use; for instance Flint, GeoLint, Hawarp, Jpylyzer, Nanite and xcorrSound.

4 Retrospective by SCAPE partners

In this section, we have collected thoughts, concerns and future ideas from many other of the partners involved in the testbed work. All partners have gathered some experience from the SCAPE project, and we would like to share these experiences for future projects.

4.1 Universitatea de Vest din Timișoara

West University of Timișoara, Romania (UVT)

UVT has been involved in three main activities: deploying the SCAPE platform in a Data Center, investigating portability of SCAPE Execution Platform and tools, and thirdly together with Brno University of Technology (BUT) we contributed to the large scale video processing and inter-linking user story. Firstly, we have deployed Apache Hadoop CDH4 distribution on a dedicated cluster of 8 physical servers and ran some experiments involving Taverna workflows. Same experiments were run on InfraGRID, an IBM cluster managed by LoadLeveler. We have also provisioned out of our virtualized infrastructure 4 VMs (2 Quadcode and 10GB RAM each) on which FIZ Karlsruhe - Leibniz Institute for Information Infrastructure (FIZ) team members were able to test the Fedora Directory deployment on IaaS environments.

Other important direction for us has been the investigation on the deployment of SCAPE platform on Cloud environments. To this end we developed a toolkit for Cloud deployment of SCAPE execution platform (Hadoop, Taverna server, plus various preservation components) that we tested against Eucalyptus, which offers an API compatible with Amazon Web Services. We are planning to make the most out of this toolkit, promoting it as standalone toolkit that can be re-used whenever complex systems are to be deployed.

On top of providing the aforementioned execution and deployment services, UVT has been providing support for customising existing services to meet the requirements of SCAPE partners. For instance, video processing tools often require a graphical interface, which is not usually provided in Data Center setups. Being able to execute OpenGL application on our GPU cluster required us to offer support for off-screen CUDA rendering to BUT partner, so that their tools are running on top of headless GPU systems.

Overall, we have driven our work based on the needs of our users, mostly FIZ and BUT partners, offering and tailoring our services to answer incoming requests from their side. We were able to test our e-Infrastructure (CPU and GPU clusters) on real-life scenarios related to digital preservation and video processing. In parallel, the toolkit developed for the deployment of SCAPE Execution Platform on hybrid Clouds[55] can be used as is, or extended by either SCAPE partners or other institutions to include additional software packages. We are planning to re-use the toolkit to generally handle the deployment of complex software systems on Cloud-based infrastructures.

[55] Daniel Pop, Marian Neagul and Dana Petcu. On Cloud Deployment of Digital Preservation Environments, ACM/IEEE Conference on Digital Libraries 2014, London

4.2 Instytut Chemii Bioorganicznej PAN

Institute of Bioorganic Chemistry, Poland (PSNC)

Medical data preservation was a primary focus of PSNC and Wielkopolskie Centrum Pulmonologii i Torakochirurgii (WCPT) cooperation in the SCAPE project. Work carried out during 13 months of the final project phase resulted in a very promising solution for hospitals that are focused on medical data preservation and long-term accessibility.

The prototype solution has been tested and validated by the WCPT and provided satisfactory results in terms of usefulness and required performance. PSNC was able to verify its High Performance Computing (HPC) infrastructure on a real-life scenario related to medical data preservation. Two server clusters were used for data processing and several approaches have been tested for data storage and access.

Because PSNC joined the project in the last year it was crucial to benefit from original SCAPE partners experiences, hence map-reduce approach was investigated in the medical scenario. Running medical data processing as well as storage and access services combined with large-scale analysis using map-reduce approach gave a deep understanding of the advantages and limitations of such platforms. Using a toolset composed of Hadoop, HBase, dcm4che[56] and PADI[57] it was possible to prepare a Medical Data Center (MDC) platform that can store, analyse and provide data in a large-scale manner.

The MDC has been integrated with the WCPT hospital environment for smooth data transfer, anonymization and preservation. The overall solution can be re-used in other medical environments, giving an opportunity to integrate data coming from various hospitals.

Beside medical aspects investigated in the context of SCAPE, PSNC was also adjusting its dArceo long-term preservation tool for cultural heritage institutions. Activities in this area, related to SCAPE data model, provided valuable findings. As a result dArceo has been supplemented with Data Connector API.

4.3 Österreichische Nationalbibliothek

Austrian National Library, Austria (ONB)

At the Austrian National Library, a dedicated experimental cluster has been set up for the SCAPE project. The Austrian National Library led the Web Content Testbed (TB.WP.1) and was partner in the Large Scale Digital Repositories Testbed (TB.WP.2).
In the first work package the user stories "ARC to WARC Migration"[58] and "File Format Identification and Characterisation of Web Archives"[59], and in the latter, the "Large Scale Ingest"[60] and "Quality

[56] http://www.dcm4che.org/
[57] https://github.com/openplanets/PADI
[58] http://wiki.opf-labs.org/display/SP/ARC+to+WARC+Migration
[59] http://wiki.opf-labs.org/display/SP/File+Format+Identification+and+Characterisation+of+Web+Archives
[60] http://wiki.opf-labs.org/display/SP/Large+Scale+Ingest

Assurance of Digitized Books"[61] user stories provided the institutional context for large scale workflow development.
During the first project year, the Austrian National Library focused on the setup and installation of the Hadoop environment as the basis of the SCAPE Platform and did experiments to find possible application scenarios for making use of Hadoop's MapReduce programming paradigm in the context of the "Quality Assurance of Digitized Books" user story.[62] On the one hand it turned out to be a useful approach to aggregate HTML, text, and metadata files aggregated in sequence files in order to make the textual and metadata information of large numbers of digital objects available in the Hadoop Distributed File System (HDFS). This approach allowed extracting singular information entities and statistical information from hundreds of thousands of books and millions of book page derivatives. It turned out that valid use cases for MapReduce existed, especially for aggregated statistical numbers, such as the average text block on book pages.

In the same period, there were also first experiments related to the "File Format Identification and Characterisation of Web Archives" and the main question was if the Hadoop framework could be used to unpack web archive container files in the ARC/WARC format in order to make individual files available for further processing, such as identifying the file type by assigning a MIME type or PUID (Pronom Unique Identifier). In this case the Hadoop framework proved to be useful for parallelising the unpacking process and it turned out that MapReduce can be used to process the identification information further to create statistical views about these results.

From the third project year on, the SCAPE Platform was released and subsequently deployed on the Austrian National Library's Cluster. Additional Apache components from the Apache Hadoop ecosystem, such as Apache Sqoop, Apache Hive and Apache Pig, gave new starting points for developing use cases and workflows that allowed for integration with existing SQL data bases (reading existing tables and writing result tables back), using Pig Scripts for creating complex workflows, and creating Hive tables to allow querying the information using SQL-like query language HiveQL. These components proved to be useful and are used in the day-to-day operation of the data processing in the context of the Google-Books-Project at the Austrian National Library.
For the "Large Scale Ingest" it was planned to use a set of 50000 METS files from the Austrian National Library's Google-Books collection to test a distributed ingest of these items into the Fedora 4 repository. The cluster deployment of the system on the cluster at the Austrian National Library was not completed due to technical issues which were documented in a series of experiments available on the SCAPE wiki.[63]

For the development of large-scale workflows for book metadata and web archive data processing, the SCAPE outcome ToMaR[64] proved to be a cornerstone because of the ability to leverage existing command line applications and making them available for building large-scale composite workflows. In terms of performance and stability the outcome of the evaluation gave good results for the composite workflows using SCAPE Preservation Components and the SCAPE Platform.

[61] http://wiki.opf-labs.org/display/SP/Quality+Assurance+of+Digitized+Books
[62] See workflows are described in deliverable D16.2 section 4.6, pp 27ff.
[63] http://wiki.opf-labs.org/display/SP/Ingest+of+digitized+book+METSs+into+Fedora+4
[64] http://tomar.openplanetsfoundation.org

Finally, outcomes of the Preservation Planning and Watch subproject, such as Plato 4 and C3PO were evaluated from a functional point of view. Plato 4 proved to be useful for gaining insights into the requirements that have to be considered when building preservation workflows, and C3PO helped creating statistical views on the characteristics of data contained in the web archive of the Austrian National Library.

4.4 Statsbiblioteket

State and University Library, Denmark (SB)

At SB we have run a number of experiments as part of the Testbed work package. The first tests were run on stand-alone machines, and were mostly focused on testing of tools and workflows. We used Taverna for some of the first tests, and we also use Taverna in a production TV and radio ingest workflow. This works in large scale as all processing is done in external components and only the workflow is handled by Taverna. SB will probably not use this approach in future projects as Taverna adds very little value when used like this.

We then moved on to the SB version of the scalable version of the SCAPE Platform, which is based on Hadoop. We have had a lot of experience with trying to fit Hadoop to our local storage solution. We have tried with different distribution; starting with Cloudera's Open Source Hadoop Platform (CDH) version 3, then Greenplum, and now CDH version 4 (we note that Cloudera is now on version 5.1). We conclude that we now have a Hadoop set up capable of large scale processing, and we also want to use this processing platform in the future.

Some of the experiments focused on the scalability of tools. We showed that the xcorrSound content comparison tool waveform-compare is scalable. We also experimented with a number of tools for file format identification and characterisation, and we would like to use Nanite for further web archive analysis. An experiment using Jpylyzer confirmed that this tool is not only scalable, but actually very fast. At SB Jpylyzer is used for property extraction for validation of archival content against an institutional policy in our production environment in the SB newspaper project[65].

As part of the platform SB has also developed tools to integrate the Fedora based DOMS metadata repository and the Bit Repository with Hadoop. SCAPE has devised a repository agnostic object format[66] based on METS and a generic repository REST interface[67]. SB has implemented the SCAPE data connector repository API for DOMS[68] and a stager-loader client[69] for this API. In this approach a number of records from the DOMS are "staged" as an archive file of METS records. The Hadoop job reads the records, work and writes new, updated records to the archive file, and the METS records are then "loaded" back into the DOMS repository. This has been evaluated as part of Testbed. The advantage in this approach is that the DOMS repository is only read once and updated once. The challenge is in concurrent processing jobs conflicting.
SB has simultaneously developed a DOMS/Bit Repository/Hadoop integration not using the SCAPE APIs as part of the SB newspaper project. In this integration, the work file list is first retrieved from

[65] http://en.statsbiblioteket.dk/national-library-division/newspaper-digitisation/newspaper-digitization
[66] github.com/openplanets/scape-platform-datamodel
[67] github.com/openplanets/scape-apis
[68] github.com/statsbiblioteket/scape-doms-data-connector
[69] github.com/statsbiblioteket/scape-stager-loader

the DOMS. The Hadoop job works on the file in the map step, and updates the record in DOMS in the reduce step. The advantage in this approach is that the DOMS repository handles any concurrency issues. The challenge here is that Hadoop accesses the repository directly in the reduce step, and it will be limited to the speed of the repository. This challenge has been mitigated by having only few simultaneous reducer tasks.

4.5 Science and Technologies Facilities Council

Science and Technologies Facilities Council, United Kingdom (STFC)

Migration from local to domain format

The experiments that have been run at STFC within the SCAPE project have been mainly based around the migration from a local based data format, which can vary depending on what instrument it was produced by, to a domain standard format (NeXus format). This has proved to be problematical as the NeXus format is based upon HDF5[70] which is not supported natively by Hadoop (HDFS). Some work has been done to incorporate the ability to use HDF5 format by Hadoop using NetCDF[71] but this is not mainstream yet and would still require the porting of NeXus format libraries to be able to use NetCDF and this was out of scope for the project.
ToMaR was used to work around this problem and this provides a mapper that copies files to be processed from HDFS to the local file system, processes them based on the specification provided and copies the results back to HDFS on completion.

Two Taverna workflows were developed, initially this was a stand-alone version not utilising Hadoop and then a version was developed that executed Hadoop jobs using ToMaR. The stand-alone version was executed on one of the Hadoop nodes over a small data set and this can be used as a direct comparison to the same dataset run using the Hadoop workflow with different Hadoop configurations.

This shows that, while Hadoop has overheads, the parallelisation of the workflow shows good performance increases over the stand-alone version and this proved the case when the small dataset was copied 1000 times to provide a dataset of 1.1TB. However, there were some worrying signs from Ganglia[72] that showed the Hadoop system was spending significant time in the wait state and therefore suggests that the process is IO bound[73].
A further dataset with a range of file sizes between 6Kb and 456Mb was then used and these had a similar execution profile, the experiment completed successfully but there was much more evidence that the process is IO bound. There were a number of initial tasks failing on the larger files and these were rescheduled by the Hadoop system and ran successfully on the second or third attempt.

The other major impact that is problematical is that the time taken to put the data onto the Hadoop system and then the time to move the migrated data off the Hadoop system is considerable. The design of the Hadoop system and HDFS is to execute the application near to the data so that data is not moved about. Once the data is on HDFS it should not be moved and then it becomes possible to

[70] http://www.hdfgroup.org/HDF5/
[71] http://www.unidata.ucar.edu/software/netcdf/
[72] http://ganglia.sourceforge.net/
[73] http://en.wikipedia.org/wiki/I/O_bound

more efficiently process and reprocess the data with the fault tolerance of the HDFS system coming in to its own. This is the model that is used here at RAL for the JASMIN[74] and SCARF[75] High Performance computing clusters. The atmospheric data that is processed on the JASMIN system took months to transfer onto the Jasmin data storage. Moving this on to a system and then off again once it has been processed is not a viable option.
A similar consideration has to be made with the ISIS and Diamond data, the cost in terms of time in moving the data onto HDFS and then off again is prohibitive and that coupled with the inability of HDFS to read and write directly to the NeXus and HDF5 file formats means that the Hadoop system is not suitable at present for the task of migrating from raw to NeXus formats at STFC.

4.6 The British Library

The British Library, United Kingdom (BL)

We have worked on five different user stories within Testbeds, all of which are based on data and workflows that the Library is tasked with preserving. The workflows, tools and information that have been derived from these scenarios have all been fed back into the organisation to ensure that the Library benefits from our involvement in the project.

Information from the image migration user story work was shared internally, and was pursued further in a conference paper[76] about differences in JPEG2000 codecs, submitted to iPres 2013. This is already proving of interest to collection managers, and it is likely that further investigations will build upon this work. The associated Jpylyzer and the Schematron profile checking are currently in use within the Library. Part of Nanite, a tool developed for the web archives identification and characterisation user story, is actively being used by the UK Web Archive at the Library, and will no doubt see further development. It has also been used by at least one external organisation who found it of use. The DRMlint/Flint tool was developed to target the Library's use case around policy validation of PDFs and EPUBs particularly with respect to DRM; it has been passed on to internal colleagues in consideration of its use within production workflows. The geospatial dataset used in testbeds is due for ingest into the Library's long term digital repository and GeoLint, the checksumming and validation software developed, along with information gathered, is already having a positive impact on our care of that collection. Finally, although the electoral register tabular data normalisation work was halted, the work that had been completed was documented, published and passed on to internal colleagues for reuse. This will provide a good base for future preservation planning work with respect to this collection.

The UK Web Archive at the Library currently have a large Hadoop installation, the base component of the SCAPE Platform, already in production use, with SCAPE work on Nanite actively built for this system. Beyond this, as already mentioned, it will be the tools and workflows that get the most use, with information about our experiences with the SCAPE Platform, and more generally Hadoop, serving to inform the Library's decisions on technology choices.

[74] http://www.jasmin.ac.uk/
[75] http://sct.esc.rl.ac.uk/SCARF/index.html
[76] http://purl.pt/24107/1/iPres2013_PDF/An%20Analysis%20of%20Contemporary%20JPEG2000%20Codecs%20for%20Image%20Format%20Migration.pdf

So, overall we have driven our work based on the needs of the Library, directing the knowledge we've gained from our work - the tools and workflows developed, and our experiences using the platform – back into the Library, so that we are in a position to consider and best implement tools, workflows and infrastructure of value. Ultimately, in this regard, through our work on the project we have demonstrated how Hadoop could potentially be used for the preservation of the Library's ever growing digital collections.

4.7 Vysoke uceni technicke v Brne

Brno University of Technology, Czech Republic (BUT)

BUT led work package 23 dealing with usability of the SCAPE platform for institutions that do not primarily focus on digital preservation. In particular, BUT investigated possibilities of large-scale experiments in external data centres in terms of their performance and repeatability. The latter was ensured by preserving key information about external High Performance Computing (HPC) infrastructure that can affect experimental results. This may relate to hardware and software settings of the external data centres, but also to their runtime properties, such as the actual consumption of available resources by all running processes.

The methodology for preserving relevant HPC parameters was identified and described in Deliverable D23.1. The document demonstrated specific aspects of the proposed methodology on experiments related to large-scale video semantic analysis. We paid attention mainly to distribution of algorithms across HPC computation nodes with the aim at maximizing the quality of results and minimizing the consumption of cluster resources.

The experiments were modelled as Taverna workflows. Although it would not be beneficial to employ the defined workflows directly in the implementation, they helped us to visualize and better explain all involved processes. Two applications were developed and used in preserving experiments. One aimed at reconstructing 3D scenes from large video data, another one focused on camera pose estimation and annotation of video captured in natural environments.

Large data sets were involved in experiments. For example, a huge amount of rendered mountain panoramic pictures, containing approximately 3TB of data, was collected for the second scenario. Experiments explored relations among the image resolution, the precision of camera pose estimation and the consumption of HPC resources. Acquired information was then used to optimize distribution of the computation across the nodes aiming at improving the pose estimation precision when the resources are limited. The experiments were performed at the UVT GPU cluster. Corresponding results are detailed in Deliverable D23.2. The cooperation between BUT and UVT in this field will continue beyond the SCAPE project.

4.8 The Internet Memory foundation

The Internet Memory Foundation, Netherlands (IMF)

The foundation participation within the project has been twofold. We actively contributed to the development of tools and workflows in relation to web archiving and lead the Platform Work Package (PT. WP1).

In relation to the web Testbed activities and as a one of the SCAPE content holders, we also participated to the user group tasks and initiated productization activities, with the aim of improving the understanding of the SCAPE outcomes and of contributing to their dissemination within the community. As most web archives, scalability issues are indeed critical for us, therefore, implementing robust and scalable automated tools within our infrastructure that could not only facilitate operations but would also reduce costs is crucial.

We proposed in that sense scenarios around quality assurance and deep characterisation at the beginning of the project. As the project progressed, we made the choice to focus on our Quality Assurance scenario We worked closely with another SCAPE partner, the University Pierre et Marie Curie (UPMC), leader of the QA work package, whom had developed a tool allowing to compare two screenshots of a web page. Our contribution first consisted in annotating hundreds of web pages to test and train the UPMC tool (Pagelyzer) and then on building a production workflow adapted to web archives. We also wrapped this tool so that it could run within our production workflow and on the SCAPE central instance.
Following several iterations of annotations, evaluations and developments, the UPMC team delivered its last version of the Pagelyzer tool, allowing one to compare two web pages visually and/or structurally (based on the page DOM). This last version was successfully integrated within the SCAPE central instance and the IMF infrastructure, as shown through our third testbed experiment that consisted in comparing a sample of 2.6 Millions of homepages taken from an IMF large scale crawl to their live version.

Improving characterisation tools so that they scale and developing QA tools designed for web archives, such as the Pagelyzer, are the most useful outcomes from our perspective. We were also strongly involved within the SCAPE platform work during the project, as outlined above and believe this platform is a useful example of how several preservation tools and systems can be integrated within one single infrastructure.

5 Commercial Readiness

Another characteristic for the SCAPE project is the ability to reach out to other communities and perhaps also commercial companies. One partner, Ex Libris, being a commercial company has given their view of the SCAPE project in this section.

5.1 Ex Libris

As described in the SCAPE report D16.2 - Large Scale Digital Repositories executable workflows for large-scale execution, Ex Libris experiments have been geared toward implementing SCAPE tools within the content of a commercial preservation system, namely Rosetta. For the purpose of this project, Rosetta functionality has been expanded to support loading SCAPE objects and accept RESTful API requests by the SCAPE Loader Application, while other SCAPE tools were integrated into the existing Rosetta extendible Plugin framework.

Due to external circumstances, the only possibility for a Rosetta Testbed instance was local environments, hosted by Ex Libris. This eliminated the possibility of testing large-scale datasets. It was decided, therefore, that scalability should be demonstrated by expanding a Rosetta environment and confirming an expected growth in throughput. The DRMLint experiments are exceptionally small – this is due to the files' size. So the metrics are based on running this experiment a number of times (deleting the files in between), and the average represents the overall results.

Results for these experiments are provided below. It is important to note that the tools' baseline performance itself (in fact, it's probably safe to assume that calling them from java wrappers – as Rosetta requires - will yield poorer performance than when run externally), while scalability results for each tool have been demonstrated on environments that support HDFS /MR - the platform for which the tools have been optimized.

Environments

	Single server	Multi server (3)
OS	RHEL 5.5	RHEL 6.3
CPU	4x Intel Xeon E5530 2.40GHz	24x Intel Xeon E5-2620 2.00GHz
RAM	16GB	32GB
JVM Xmx	4GB	16GB

Results

Tool	Jpylyzer	DRMLint	Jpylyzer	DRMLint
Number of files per load	1056	92	1056	92
Average file size	12.4GB	48GB	12.4GB	48GB
Average runtime (sec./file)	1.786	0.941	0.488	0.182

As a leading provider of solutions for academic, national, and research libraries, Ex Libris' commitment to its customers begins by participating in research projects, learning about new trends, and contributing from its rich experience to setting best practices and standards. From its inception, Rosetta, Ex Libris' digital preservation system, has been designed and constructed in close alliance with development partners representing the various stakeholders in the preservation community. Participation in the SCAPE project has contributed to Ex Libris' understanding of the evolving needs and emerging solutions in this community. Hearing from users who are not Rosetta customers was especially valuable, allowing a self-reflective perspective of our approach and methods. The adoption

of an object data model that is very similar to Rosetta's is an important validation to us. Other SCAPE components, less similar to Rosetta architecture, also offered valuable input by helping us think about the advantages and disadvantages of the solutions Rosetta offers and how these can be improved. While many of the SCAPE experiments demonstrated needs that have not yet been surfaced by Rosetta users, Ex Libris is now in a better place to anticipate these needs and provide timely solutions based on its experience in the project. As this deliverable demonstrates, some of the SCAPE tools have already be packaged to be used in Rosetta, while other fruits of this project are expected make their mark on the Rosetta roadmap in the future.

6 Appendix A – Testbeds

These are the testbed objectives, taken from the SCAPE Description of Work document.

```
THEME [ICT-2009.4.1]
[Digital libraries and digital preservation]
Grant agreement for: Collaborative project

Annex I - "Description of Work"
Project acronym: SCAPE
Project full title: "Scalable Preservation Environments"
Grant agreement no: 270137
Version date: 2013-08-30
```

6.1 Web Content Testbed

In order to demonstrate that the SCAPE approach provides the means for undertaking efficient and scalable long term preservation activities on archived web content data, a set of concrete web content preservation scenarios is necessary. This requires the definition of corpora and corresponding corporate policies on the one hand, as well as the identification and use of metadata extraction/characterisation, quality assurance, action, and preservation planning services on the other hand.

6.2 Large Scale Digital Repository Testbed

This work package will apply the SCAPE preservation services in a set of Large Scale Digital Repository (LSDR) scenarios in order to test, evaluate and tailor their applicability to the particular challenges of scale in the repository environment. These scenarios will inform requirements for preservation services developed in the Characterisation, Action and Quality Assurance work packages and provide the basis for applying, testing and evaluating the SCAPE services and infrastructure on the LSDR datasets.

6.3 Research Datasets Testbed

This work package will develop a test bed to test, evaluate and demonstrate the applicability of the evolving SCAPE preservation services to the complexity of the scientific research lifecycle. The services must address the variety of fundamental entities such as raw data, algorithms and their implementation as software, various stages of processed data, and publications based on that data which collectively make up the lifecycle of scientific endeavour.

The work package will have a dual focus on two aspects of preservation that are of particular importance for data from scientific facilities: migration of formats, and linking of (subsets of) data and ancillary resources to ensure traceability of the scientific process over time. This is part of an overall vision to achieve interoperability across different facilities (there is on-going work in this area) and across time (which is the preservation aspect). There are some key differences between the research datasets testbed and the other testbeds in the project. For research datasets, the idea of the process by which they were created is of great importance. It is expected that this work package will therefore provide opportunities to apply and test preservation actions some of which are common to the other testbeds (those concerned with migration) and others that are specific to the needs of the research datasets. This in itself will provide a validation of the SCAPE platform's ability to handle a diversity of preservation services.

6.4 Data Center Testbed

The main objective of this work package is the development of domain specific and scalable preservation workflows for data collections that are typically stored and processed in large data centers. The key distinguishing aspect of this testbed as compared to SCAPE's existing testbeds (TB.WP1, TB.WP2, TB.WP3) is a separation between the institution that produces and analyses the data sets (as for example domain scientists) and the institution that physically hosts the data sets (data center). Consequently, the organizations that own the data assets are different from the institutions that are responsible for storing, archiving, and preserving the data. The objective of this work package is to demonstrate the applicability of SCAPE solutions for academic and national data centers allowing them to provide scalable preservation services to their user communities (which are typically not primarily focused around digital long-term preservation).

7 Appendix B1 – User Stories in Web Content Testbed

7.1 ARC to WARC Migration

Status

Active

Contact

Sven Schlarb, ONB

User Story

As the owner of a number of legacy ARC files and a Web Archive currently harvesting WARCS, I need a digital preservation system that can migrate ARCs to WARCs in a timely fashion and ensure the completeness of the migration, so that I can more effectively manage Web Archive content by only having a single format to deal with. This also means I do not have to maintain two playback mechanisms for the long term.

User Requirements/Components

1. *I need a tool that can migrate ARC to WARC files.*
2. *I need a tool that can verify that the content of the ARC is the same as the content of the WARC.*

Experiments

- ARC2WARC Experiment at KB
- ARC2WARC Experiment at ONB

Developer Notes

A QA of the migration could use comparing snapshots of each of the sites, it could also take the approach of comparing all the files in each. There may be other aspects of ARCs and WARCs (header information, logs, etc.) that will need checking too. For example, has the log file format changed between the two? Is the WARC structurally sound?, etc.

Using JWAT

Related Documents

7.2 Comparison of Web Snapshots

Status

Active

Contact

Leila Medjkoune, IM

User Story

In order to be confident that we have preserved a website we need a digital preservation system that can automate the comparison of the two Web Snapshots. This could be a harvested copy and a previous harvested copy that has been manually verified as an accurate representation of the site or a harvested copy and its live version. *This will enable us to ensure Web content has been successfully harvested and inform harvesting policies.*

User Requirements/Components

I need a tool to generate an image of a Web page found in a WARC/ARC file so that I can compare this image with the live copy.

I need a tool to compare two web page screenshots and provide a similarity score so I can assess how closely the live site and the harvested site match visually.

1. *MUST be able to read WARC files*
2. *SHOULD be able to read ARC files*
3. *MUST continue after any network downtime*
4. *SHOULD be robots.txt aware*
5. *Similarity score MUST be normalised between 0 and 1, where 1 is identical and 0 is no similarity at all*
6. *WARC comparison MUST occur within the update frequency of the live website*

Experiments

- WCT2-EX1 Comparing newly archived Web sites against a verified copy (single node)
- WCT2-EX2 Comparing newly archived Web sites against a verified copy (multiple nodes)
- WCT2-EX3 Comparing newly archived Web sites against live web

Developer Notes

This is essentially the ARC to WARC migration without the migration. Makes sense to check out QA steps developed as part of that story.

Related Documents

7.3 File Format Identification and Characterisation of Web Archives

Status

Active

Contact

Per Møldrup-Dalum, SB
William Palmer, BL

User Story

As a Web Archive I need a Digital Preservation System that can process both ARC and WARC files and identify file formats/characterize of items contained so that I can assess preservation risks and plan which tools will be required for access to those formats.

User Requirements/Components

1. *A tool that can efficiently work through the content of an ARC file and identify the type of files found.*
    a. *Must provide a report in a usable format - where this is a large dataset, this could be a database of some sort*
    b. *Ideally the tool will also perform file format identification on files within container formats - media streams, zips and other compressed files, etc.*
2. *Look up of an appropriate access tool/software would be a bonus!*

Experiments

- WCT EX2 File ID at SB
- WCT EX3 File ID at BL
- WCT EX4 File ID and characterisation at SB
- Web Archive FITS Characterisation using ToMaR at ONB

Developer Notes

Related Documents

8 Appendix B2 – User Stories in Large Scale Digital Repository Testbed

8.1 Characterisation of Large Audio and Video Files

Status

Active

Contact

Bolette Jurik, SB

User Story

As the owner of a large collection of video files I need a digital preservation system that can characterise very large audio/video files to enable me (or a watch system) to evaluate the collection for preservation risks and ongoing risk management.

User Requirements/Components

1. *We need to be able to characterise very large video (8GB+) files*
    a. *This includes identifying container formats and contained streams*
    b. *Features extracted need to be decided - we need to consider what is useful here to extract and include here as requirements. See:*
        i. *http://www.jiscdigitalmedia.ac.uk/movingimages/advice/metadata-and-digital-video*
        ii. *http://www.jisc.ac.uk/media/documents/programmes/preservation/spmovimages_report.pdf*
2. *Needs to perform characterisation quickly and efficiently*
3. *Would be good to be able to validate video format compliance with specification*

Experiments

- Characterisation and validation of audio and video files during ingest

Developer Notes

The original scenario expressed some concern that existing characterisation tools (JHOVE for instance) do not seem to work well on large files.

This story provides the opportunity to compare tools over time, such as the Tika/DROID/etc. as performed by PC.CC. (See also Incubator).

Related Documents

8.2 Large Scale Audio Migration

Status

Active

Contact

Bolette Jurik, SB

User Story

As the owner of a large audio collection, I need a digital preservation system that can migrate large numbers of audio files from one format to another and ensure that the migration is a good and complete copy of the original.

User Requirements/Components

1. *We need a measure of similarity between two audio files based on how they 'sound'*
2. *We need to be able to compare properties of MP3s such as duration with those of the migrated WAV*
3. *We need to be able to migrate MP3 to WAV*

Experiments

- SB Experiment SO4 Audio mp3 to wav Migration and QA Workflow
- SB Experiment Audio mp3 to wav Migration and QA on Hadoop Cluster

Developer Notes

This scenario is closely related to the image migration user story, only it is for audio files - there may be scope to reuse workflows? The tools are however different.

Related Documents

8.3 Large scale document characterization and identification with Tika and DROID on SCAPE Azure platform

Status

Active

Contact

Ivan Vujic, MSR

User Story

As part of evaluating what platform user should be using to run characterization and identification tools, user should SCAPE Azure platform. We measured the speed of the Apache Tika Content Analysis Toolkit and the DROID File Format Identification Tool when they were run on a Microsoft Azure virtual machine. The results are compared to the speed of the same tools running on a traditional on-site server.

User Requirements/Components

Experiments

- Characterisation and Identification on SCAPE Azure Platform

Developer Notes

This story provides the opportunity to compare tools between platforms, tools such as the Tika and DROID as performed by PC.CC. (See also Incubator)

Related Documents

8.4 Large Scale Image Migration

Status

Active

Contact

William Palmer, BL

User Story

As a curator of image files, I need a digital preservation system that can migrate a large number of images from one format to another, ensuring that the migrated images conform to our institutional profile, that no image data is lost and that the migration is cost effective (saving storage for example).

User Requirements/Components

1. *We need to be able to migrate TIFFs to JP2Ks*
    a. *Ideally we can migrate TIFF to a JP2K conforming to any profile within the limits of the JP2K standard*
    b. *Migration must support the recommended JP2K profile*
2. *We need to compare a JP2K image file technical metadata and profile to the recommended profile.*
3. *We need to ensure that the JP2K contains all of the image data*
4. *We need to ensure that the JP2K is a good and complete copy of the TIFF*
5. *We need to be able to report on the storage saving and perhaps cost benefit of doing this*

Experiments

- KB Metamorfoze Image Migration & QA
- LSDRT2 EX1 BL Newspapers on the BL Platform
- TIFF to JPEG2000 Migration Experiment at ONB

Developer Notes

Related Documents

8.5 Large Scale Ingest

Status

Active

Contact

Sven Schlarb, ONB

User Story

As an institution we need a system that will enable us to ingest a large number of digital objects and associated metadata into our digital repository securely, correctly and with acceptable performance so that we can be ensure safe deposit of this data.

User Requirements/Components

Experiments

- Ingest of digitized book METSs into Fedora 4

Developer Notes

Related Documents

8.6 Policy-Driven Identification of Preservation Risks in Electronic Document Formats

Status

Active

Contact

Clemens Neudecker, KB; Johan van der Knijff, KB; William Palmer, BL

User Story

Digital repositories typically hold large numbers of electronic documents from various sources. Common document formats such as PDF and EPUB include features that are potential risks for long-term accessibility and preservation. Hence, in order to sustainably manage their collections, institutions may want to identify specific preservation risks, either at ingest or at some later stage.

User Requirements/Components

1. *We need to be able to identify "preservation risks" for a given document. These risks include, but are not limited to:*
    a. *password protection*
    b. *print protection*
    c. *copy protection*
    d. *other DRM*
    e. *embedded proprietary content such as commercial fonts JvdK: I think commercial fonts are only a problem if they are not embedded??*
    f. *missing or damaged fonts*
    g. *JavaScript (which may present several security risks)*
    h. *multimedia content*
    i. *other external dependencies*
2. *We need to be able to assess legacy files and deal with them appropriately*
3. *We need to be able to assess files prior to ingest and deal with them appropriately*
4. *We would ideally do 2 & 3 on the basis of some machine readable policy*

Experiments

- Validate PDF&EPUBs and check for DRM

Developer Notes

TBC, for PDF a possible approach would be to use the Apache Preflight PDF/A validator (part of PDFBox) to identify all potential risks, and then evaluate the output against a set of business rules that correspond to low-level (control) policies. This could be done with Schematron (requires development of XML output handler for Preflight!), resulting in an approach similar to the JPEG 2000 / Jpylyzer work. See also:
http://www.openplanetsfoundation.org/comment/385#comment-385

For EPUB something similar could be done using the EpubCheck tool.
Also this policy validation is something SCAPE's SCOUT should/could deal within.

Related Documents

8.7 Validation of Archival Content against an Institutional Policy

Status

Active

Contact

Bolette Jurik, SB

User Story

As a memory institution, I want to ensure that content in our repositories conforms both to its file format specification and (where appropriate) the profile of that format as specified by the institutional policies. This is to ensure that our content conforms to existing preservation policies and also that content we ingest is acceptable within the bounds of those policies.

User Requirements/Components

1. *We assume that the content file format is known - i.e. we know the collection is a number of MPEG-1 movies.*
2. *We need to be able to validate the file against its file format specification/structure.*
3. *We need to be able to define the expected file format profile to be machine readable.*
4. *We need to be able to compare the expected file format profile with the actual file format profile.*

User Requirements/Components

Experiments

- Validate JPEG2000 Newspapers Using Jpylyzer

Developer Notes

Related Documents

9 Appendix B3 – User Stories in Research Datasets Testbed

9.1 Migration from local format to domain standard format

Status

Active

Contact

Catherine Jones, STFC

User Story

As the content holder/manager of scientific data held in a local format, I wish to migrate this data into a domain standard format to reduce the risks of losing the ability to read/use and reuse the data contained within the file format.

User Requirements/Components

A tool/suite of tools to migrate from the local format to the domain standard. This would have the following functionality

- *The ability to identify the local format files (main file and associated log files)*
- *The ability to identify the type of instrument (schema changes depending on instrument)*
- *The ability to migrate the data*
- *The ability to do quality assurance, both structural and semantic*

Experiments

- raw2nexus Experiment at STFC

Developer Notes

Related Documents

9.2 Identification, validation and checksumming of a complex corpus

Status

Active

Contact

William Palmer, BL

User Story

As a holder of a large set of geospatial data, I want to ensure I can identify my files, create/check fixity and validate formats, where appropriate, so that I can be confident that the data formats I hold are valid and that the data does not change over time.

User Requirements/Components

1. *A tool to create/check fixity of files*
2. *A tool to identify the files in the corpus*
3. *A tool to validate files adhere to the format specifications, where appropriate*

Experiments

- GeoLint Experiment

Developer Notes

Related Documents

10 Appendix B4 – User Stories in Data Center Testbed

10.1 Large scale video processing and interlinking

Status

Active

Contact

Pavel Smrz, BUT

User Story

Today's digital cinematography, game industry, advanced robotics, and many other fields take advantage of data-intensive video content analysis and processing. One of key tasks involved consists in large-scale 3D reconstruction from photographic images/video sequences and special remote sensing devices such as the LiDAR. Many-core CPU and GPU clusters available in data centres provide a natural platform for such a task.

This story focuses on a preservation scenario dealing with large-scale video processing data and all related processes. It aims at preserving interlinks among raw and derived data, created models and metadata and it deals with information quality management procedures within the whole process. It also employs advanced workflows and preservation actions concerned with preserving contextual information of the processed datasets – enhancing capturing/harvesting information, meta-representation framework for the analysis models, data reuse models, etc.

Three levels of preservation components need to be combined to cope with user needs in this context:

- *consistency checking and quality assurance of resulting analysis results;*
- *preservation of the **static context** and pre-defined links among data, semantic relationships between the raw data and derived knowledge components;*
- *selection of characteristics profiling **actual runs** of particular tasks and influencing their results, preservation of log components and data-centre performance characteristics.*

Preservation strategies for data centres also need to take into account specific characteristics of the platform provider / customer setting. The centres usually operate independently of the particular application domain and they are accessed remotely to process a specific set of data. Thus, voluminous input data needs to be transferred first and results sent back to the task owner. The preservation has to reflect distinct roles of the data owner and the platform provider and pay attention to access rights and security in general.

A particular story that will define a base for specific preservation experiments aims at building detailed 3D models of a large area. Involved algorithms take advantage of the GPU cluster available at the Timisoara data centre. The quality of results (e.g., the coverage of an area in focus) generally depends on a task dispatching mechanism and actual performance characteristics of individual nodes and an overall load during the processing. Thus, these features need to be logged and taken as a part of preserved links between the raw data and the results.

The input data will be transferred from BUT servers to the UVT data centre first. It takes form of image and video files as well as LiDAR measurements. The input needs to be pre-processed first to enter the main 3D reconstruction and rendering component. For example, a video file needs to be split into individual shots. To preserve links between the input and the results in a consistent form, it is thus necessary to validate the data transfer, formats of input files and results of their pre-processing. These processes will take advantage of the map-reduce schema and its Hadoop implementation running at standard servers available in the data centre.

An initial setting of the large-scale experiment (referred to as the first phase in the DoW) will involve a simple distribution schema of the 3D reconstruction and rendering tasks over subsets of the data and individual nodes available. The processing will be defined as specific Taverna workflows that will be compared in terms of the effectiveness to reach a defined result.

The final version of the preservation experiment will involve more advanced workflows and preservation actions focusing of semantic-aware dynamic context of processed datasets, sophisticated dispatching techniques based on harvested information, and throughout analysis of results aiming at their low-barrier reusing.

Specific User Story Definition
As a researcher dealing with SLAM (simultaneous localization and mapping) and visual geo localization, I need to preserve results of large-scale scene reconstruction and rendering, together with related source objects and parameters of the computation process, so that the results will be available for (re-)use and further refinement in a long term.

User Requirements/Components

A suite of tools for preservable large-scale scenes reconstruction and annotation is needed:

- *Toolkit for preserving large-scale experiments providing functions divided into following groups:*
  1. *Functions for preserving data centre environment details*
  2. *Functions for creating and analysing interlinks*
  3. *Functions and tools for preserving input and output files*
  4. *Functions for checking metadata consistency*

- *Experimental video processing applications:*
  1. *Application for distributed 3D scene reconstruction equipped with preservation toolkit.*
  2. *Application for large-scale scene annotation and localization*

Experiments

- Scene reconstruction
- Video annotation and geo localization

Developer Notes

Related Documents

10.2 Large scale access at hospital (Medical Dataset)

Status

Active

Contact

Paweł Kominek, WCPT; Tomasz Parkoła, PSNC

User Story

As an employee at the hospital I need easy access to preserved medical data. Currently only two-year history of patient's treatment is held in the hospital system and available at hand (access to the current archiving system is limited). The goal is to have access to whole history of patient's treatment accessible directly from the archiving system.

User Requirements/Components

1. *Access to the data needs to be possible using the patient's identifier (PESEL or NIP) or Name/Surname pair.*

Experiments

- Performance tests for accessing medical data

Developer Notes

Related Documents

10.3 Large scale access for educational purposes (Medical Dataset)

Status

Active

Contact

Paweł Kominek, WCPT; Tomasz Parkoła, PSNC

User Story

As a university teacher I need an easy access to examples of various diseases. The examples will be used during university courses and will showcase good practices.

User Requirements/Components

1. *The access to the data should be done via Internet, preferably via web browser.*
2. *The retrieval should be possible with the use of ICD10 classification and the result set should be limited (e.g. several or a dozen of examples is sufficient).*

Experiments

- Performance tests of the search function in the MDC portal

Developer Notes

Related Documents

10.4 Large scale analysis (Medical Dataset)

Status

Active

Contact

Paweł Kominek, WCPT; Tomasz Parkoła, PSNC

User Story

As a researcher at the hospital I need a way of analysing large amounts of medical data related to patients treatment. The analysis will be helpful in my research activities or can be directly used by the hospital to calculate statistics.

User Requirements/Components

Experiments

- Analysis of epidemiological situation across WCPT patients

Developer Notes

Related Documents

10.5 Large scale ingest of medical data (Medical Dataset)

Status

Active

Contact

Paweł Kominek, WCPT; Tomasz Parkoła, PSNC

User Story

As an institution responsible for storing medical data I need a system for archiving the data. The reason is the lack of necessary resources (mainly storage space) due to the requirement (enforced by law) to store medical data for at least 20 years (30 years in some cases).

User Requirements/Components

1. *The ingestion process needs to be initiated by the content holder due to the high sensitivity of the content (personal data).*
2. *Only anonymised assets can be stored outside the content holder (sensitive personal data cannot be stored in external facilities).*
3. *Whenever possible well-known standards should be used (e.g. DICOM, HL7).*

Experiments

- WCPT to PSNC DICOM medical data ingest

Developer Notes

Related Documents

11 Appendix C1 – Experiments in Web Content Testbed

11.1 ARC2WARC Experiment at KB

User story: ARC to WARC Migration

Investigator(s)

Clemens Neudecker, KB

Dataset

KB Web Archive Dataset (sample batch) from the KB webarchief[77].

Platform

KB 1 Hadoop Platform

Workflow

The migration will be implemented in Java code and as a Taverna workflow.
We will make use of the Hawarp tool from ONB.

Requirements and Policies

TBD

Evaluations

- EVAL ARC2WARC with Hawarp

[77] http://www.kb.nl/en/expertise/e-depot-and-digital-preservation/web-archiving

11.2 ARC2WARC Experiment at ONB

User story: ARC to WARC Migration

Investigator(s)

Sven Schlarb, ONB

Dataset

Austrian National Library - Web Archive (ONB Web Archive Dataset)

Platform

ONB Hadoop Platform

Purpose of this experiment

The purpose of this experiment is to test the performance of two different approaches of implementing a large-scale ARC to WARC migration workflow.

The first approach is a native Map/Reduce application (ARC2WARC-HDP[78]) and is using a Hadoop-InputFormat for reading ARC files and a Hadoop-Output-Format for writing WARC files. And the second approach is a Java-based command line executable (ARC2WARC-TOMAR[79]) which directly transforms ARC to WARC files and uses the SCAPE tool ToMaR[80] to make this process scalable.

The main question which this experiment should help to answer is whether a native Map/Reduce job implementation has a significant performance advantage compared to using ToMaR with an underlying command line tool execution.

The Hadoop-version has an important limitation: In order to do the transformation based on a native Map/Reduce implementation it is required to use a Hadoop representation of a web archive record. This is the intermediate representation that is between reading the records from the ARC files and writing the records to WARC files. As it uses a byte array field to store web archive record payload content, there is a theoretical limit of around 2 GB due to the Integer length of the byte array which would be a value near Integer.MAXVALUE. Anyhow, the practical limitation of payload content size will be much lower depending on hardware setup and configuration of the cluster.

The performance advantage should be "significant" because the fact that using native Map/Reduce implementation would mean that it is not possible to process container files that have large record payload content. As a consequence, an alternative solution for these cases would be needed. And such a separation between "small" and "large" container would bring other difficulties, especially when it is required to include contextual information in the migration process, like crawl information that relates to a set of container files which must be available during the migration process.

The implementations used to do the migration are proof-of-concept tools, which means that they are not intended to be used to run a production migration at this stage. This means that there are the following limitations:

[78] https://github.com/openplanets/hawarp/tree/master/arc2warc-migration-hdp
[79] https://github.com/openplanets/hawarp/tree/master/arc2warc-migration-cli
[80] https://github.com/openplanets/tomar

1. As already mentioned, there is a file size limit regarding the in-memory representation of a web archive record, the largest ARC file in the data sets used in these experiments is around 300MG, therefore record-payload content can be easily stored as byte array fields.
2. Exceptions are caught and logged, but there is no gathering of processing errors or any other analytic results. As the focus lies here on the performance evaluation, any details regarding the record processing are not taken into consideration.
3. The current implementations do not include any quality assurance, like comparing digest information of payload content or doing rendering tests and taking snapshots which can be compared, for example.
4. Contextual information is not being taken into consideration. From a long-term preservation perspective, a real benefit of the ARC to WARC transformation would be to include contextual information, like information in the crawl log files, for example, so that as a result of the ARC to WARC migration, the WARC files would be the only files that need to be preserved.

Reading web archive content is a big effort in terms of the amount of data that must be read and maybe even transferred in a cluster or cloud environment first. While we are at it, we should therefore take the opportunity to run other processes as well. For that reason, the proof-of-concept implementations include Apache Tika as an example of a process which does payload content identification as an optional feature. All Hadoop job executions are tested with and without payload content identification enabled.

The proof-of-concept implementations use the Java Web Archive Toolkit[81] (JWAT) for reading web archive ARC container files and to iterate over the records.

Workflow

The baseline evaluation was done by executing the hawarp/arc2warc-migration-cli[82] java application from the command line on one worker node of the cluster and serves as a point of reference for the distributed processing (column "Metric baseline" in the evaluation tables).

Without Apache Tika payload identification the command used was:

```
java -jar hawarp/arc2warc-migration-cli/target/arc2warc-migration-1.0-SNAPSHOT-jar-
with-dependencies.jar -i /local/filysystem/input/dir
```

And including Apache Tika payload identification the command used was (flag -p):

```
java -jar hawarp/arc2warc-migration-cli/target/arc2warc-migration-1.0-SNAPSHOT-jar-
with-dependencies.jar -i /local/filysystem/input/dir -p
```

ARC2WARC-HDP Workflow

The Hadoop job evaluation was done using the hawarp/arc2warc-migration-hdp executable jar:

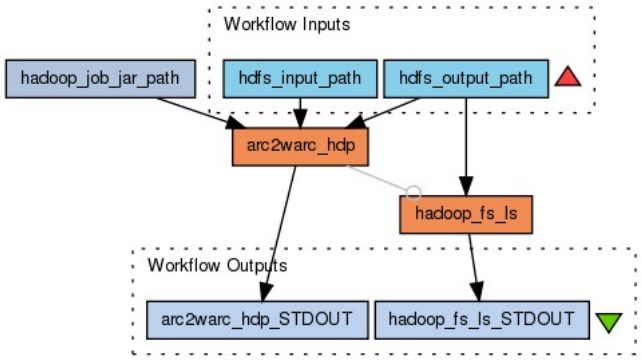
```
hadoop jar hawarp/arc2warc-migration-hdp/target/arc2warc-migration-hdp-1.0-jar-with-
dependencies.jar -i hdfs:///user/input/directory -o hdfs:///user/input/directory
```

[81] https://sbforge.org/display/JWAT/JWAT
[82] https://github.com/openplanets/hawarp/tree/master/arc2warc-migration-cli

To run the workflow with Apache Tika payload content identification, the "-p" flag is used:

```
hadoop jar hawarp/arc2warc-migration-hdp/target/arc2warc-migration-hdp-1.0-jar-with-
dependencies.jar -i hdfs:///user/input/directory -o hdfs:///user/input/directory -p
```

A simple wrapper workflow for the Hadoop job execution is available on myExperiment:



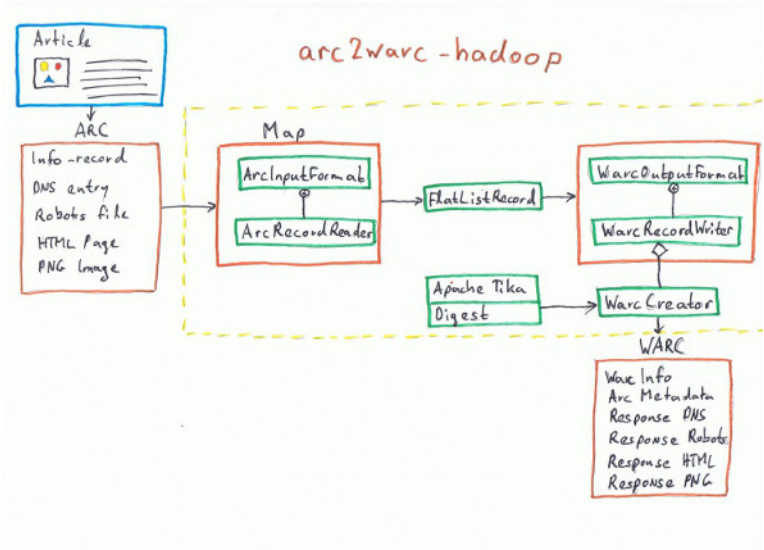
The ARC2WARC-HDP workflow is based on a Java-Implementation using a Hadoop-InputFormat for reading ARC files.

To iterate over all items inside the web archive ARC container files, the native JAVA map/reduce program uses a custom RecordReader based on the Hadoop 0.20 API. The custom RecordReader enables the program to read the records natively and iterate over the archive file record by record. One ARC file is processed per map and one WARC files is produced as output of this map phase. The implementation does not use a reducer, by that way, the WARC output files is not aggregated and one WARC file is created per ARC input file.

The custom record reader of the Hadoop input format used in the implementation uses the Java Web Archive Toolkit[83] (JWAT) for reading web archive ARC container files. The ARC records are converted to a Hadoop-record (FlatListRecord) which is the internal representation implementing the Writable or WritableComparable interface in order to provide a serializable key-value object.

Using a Hadoop output format the records are then written out to HDFS. The workflow is implemented as a native JAVA map/reduce application and allows to make use of Apache Tika™ 1.0 API (detector call) to detect the MIME type of the payload content of the records.

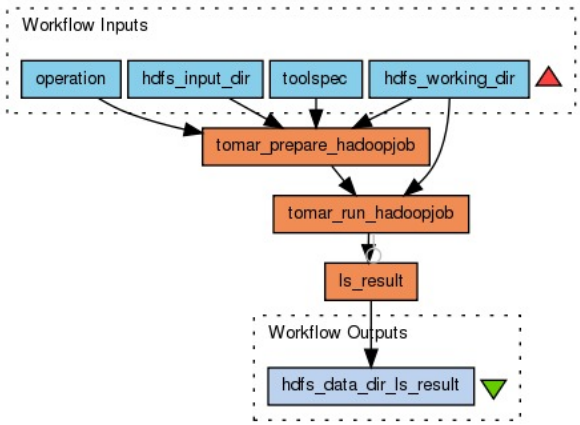
[83] https://sbforge.org/display/JWAT/JWAT



ARC2WARC-TOMAR Workflow

The ARC2WARC-TOMAR workflow is using the command line Java-Implementation hawarp/arc2warc-migration-cli[84].

The ToMaR evaluation was executed using the Taverna workflow ToMaR HDFS Input Directory Processing[85]:



The following ToMaR tool specification XML was used as input for the "toolspec" input port (the "-p" flag was added to the command to enable Apache Tika identification):

```
<?xml version="1.0" encoding="utf-8" ?>
<tool xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://scape-
project.eu/tool tool-1.0_draft.xsd" xmlns="http://scape-project.eu/tool"
xmlns:xlink="http://www.w3.org/1999/xlink" schemaVersion="1.0" name="bash">
  <operations>
    <operation name="migrate">
      <description>ARC to WARC migration using arc2warc-migration-cli</description>
```

[84] https://github.com/openplanets/hawarp/tree/master/arc2warc-migration-cli

[85] http://www.myexperiment.org/workflows/4144.html

```
      <command>java -jar /usr/local/java/arc2warc-migration-cli-1.0-jar-with-dependencies.jar
-i ${input} -o ${output}</command>
      <inputs>
        <input name="input" required="true">
          <description>Reference to input file</description>
        </input>
      </inputs>
      <outputs>
        <output name="output" required="true">
          <description>Reference to output file</description>
        </output>
      </outputs>
    </operation>
  </operations>
</tool>
```

Evaluation summary

		Objects/hour	Throughput(GB/min)	Avg. Runtime/item(s)
	Baseline	834	1,2727	4,32
Map/Reduce	1000 ARC files	4592	7,0089	0,78
	4924 ARC files	4645	7,0042	0,77
ToMaR	1000 ARC files	4250	6,4875	0,85
	4924 ARC files	4320	6,5143	0,83
	9856 ARC files	8300	12,4941	0,43
	Baseline	545	0,8321	6,60
Map/Reduce with Tika	1000 ARC files	2761	4,2139	1,30
	4924 ARC files	2813	4,2419	1,28
ToMaR with Tika	1000 ARC files	3318	5,0645	1,09
	4924 ARC files	2813	4,2419	1,28
	9856 ARC files	7007	10,5474	0,51

Evaluations

- EVAL ARC2WARC-HDP w.o. Tika
- EVAL ARC2WARC-HDP with Tika
- EVAL ARC2WARC-TOMAR w.o. Tika
- EVAL ARC2WARC-TOMAR with Tika

11.3 Comparing newly archived Web sites against a verified copy (single node)

User story: Comparison of Web Snapshots

Investigator(s)

Stanislav Barton, IMF

Dataset

Internet Memory Web Archive, 440 pairs of URLs annotated by IMF team.

Platform

Central instance at IMF

Purpose of the Experiment

From the tools developed in the scope of the project (in the preservation components sub-project), we selected the MarcAlizer tool, the first version of the Pagelyzer tool developed by UPMC, that performs the visual comparison between two web pages. The Markalyzer was then wrapped by the Internet Memory so that it can be used within its infrastructure and the SCAPE platform. In a second phase, the renderability analysis should also include the structural comparison of the pages, which is implemented by the Pagelyzer tool.

Since the core analysis for the renderability is thus performed by an external tool, the overall performance of the wrapped tool will be tight to this external dependency. We will keep integrating the latest releases issued from the MarcAlizer development, as well as the updates on the tool issued from a more specific training.

Workflow

The detection of the rendering issues is done in the following three steps:

Web pages screenshots automatically taken using Selenium framework, for different browser versions.

Visual comparison between pairs of screenshots using MarcAlizer tool (recently replaced by PageAlizer tool, to include also the structural comparison).

Automatically detect the rendering issues in the Web pages, based on the comparison results.

The wrapper application orchestrates the main building blocks (Selenium instances and MarcAlizer comparators) and performs large scale experiments on archived Web content.

The browser versions currently experienced and tested are: Firefox (for all the available releases), Chrome (only for the last version), Opera (for the official 11th and 12th versions) and Internet Explorer (still to be fixed).

The initial implementation is represented by several Python scripts, running on a Debian Squeeze (64 bits) platform. This version of the wrapped tool was released on GitHub and we received some valuable feedback from the sub-project partners:

https://github.com/crawler-IM/browser-shots-tool

The deployment and installation of the wrapped tool are rather easy, but strongly dependent on different other packages, since it uses "off-the-shelf" components that need to be already available on your system, such as:

Python 2.6 or higher

Selenium 2.24.1

MarcAlizer 0.9

In order to make all the tools run together, in a suitable environment, the following applications/packages need to be installed:
Selenium driver for the browsers: provided by Selenium in the Python Client on its official website (for example, the driver for Firefox is used in this project). Reference: http://pypi.python.org/pypi/selenium
If the Graphical User Interface (GUI) is not available on your system, you can use an X server (for example, we used Xvfb v 11). The packages to be installed in this case are: xvfb, xfonts-base, xfonts-75dpi, xfonts-100dpi, libgl1-mesa-dri, xfonts-scalable, xfonts-cyrillic, gnome-icon-theme-symbolic
Python: one can check the installed version by typing the command line: $ python
For the preliminary rounds of tests, we deployed the tool on three nodes of IM's cluster and we performed automated comparisons for around 440 pairs of URLs. The processing time registered in average was about 16 seconds per pair of Web pages. These results showed that the existing solution is suitable for small-scale analysis only. Most of the time in the process is actually represented by IO operations and disk access to the binary files for the snapshots. Taking the screenshots proven to be very time consuming and therefore if this solution is to be deployed on a large scale, the solution needed to be further optimized and parallelized.
These results showed also that a serious bottleneck for the performance of the tool is represented by the passage of intermediary parameters in between the modules. More precisely, the materialization of the screenshots in binary files on the disk is a very time consuming operation, especially when considering large scale experiments on a large number of Web pages.
We therefore have to move to a different implementation of the tool, which will use an optimized version of MarcAlizer. The Web pages screenshots taken with Selenium will be directly passed over to MarcAlizer comparator using streams and the new implementation of the browser-shots tool will be represented by a MapReduce job, running on a Hadoop cluster. Based on this framework, the current rounds of tests could be extended up to much higher number of pairs of URLs.

Requirements/Evaluation Criteria/Conditions of Satisfaction

- Must be able to compare 100 or more WARC images to their reference images per hour.
- Rate of comparison failure must be 0 - which is to say no WARCs should fail the comparison.

Notes/Context

Evaluations

EVAL - WCT2-EX1 Comparing newly archived Web sites against a verified copy (single node)

11.4 Comparing newly archived Web sites against a verified copy (multiple nodes)

User story: Comparison of Web Snapshots

Investigator(s)

Stanislav Barton, IMF

Dataset

Internet Memory Web Archive, a sample of 13 000 URLs

Platform

Central instance at IMF

Purpose of the Experiment

From the tools developed in the scope of the project (in the preservation components sub-project), we selected the MarcAlizer tool, the first version of the Pagelyzer tool developed by UPMC, that performs the visual comparison between two web pages. The Markalyzer was then wrapped by the Internet Memory so that it can be used within its infrastructure and the SCAPE platform. In a second phase, the renderability analysis should also include the structural comparison of the pages, which is implemented by the Pagelyzer tool.
Since the core analysis for the renderability is thus performed by an external tool, the overall performance of the wrapped tool will be tight to this external dependency. We will keep integrating the latest releases issued from the MarcAlizer development, as well as the updates on the tool issued from a more specific training.

Workflow

For this experiment the tool is implemented as a MapReduce job to parallelize the processing of the input. The input in this later case is a list of urls that together with a list of browser versions, that are used to render the screen shot - note the difference in comparison to the former version where the input where pairs of URLs that were rendered using one common browser version and these were compared.

Optimizations

In order to achieve acceptable running times of the tool newer version of the Marcalizer comparison tool was integrated into this tool. The major improvement brings the possibility of feeding to tool with in-memory objects instead of pointers to files on disk. This improvement and the elimination of the unnecessary IO operations lead into following average times got for the particular steps in the shot comparison:

- screenshot acquirement - 2s
- Markalyzer comparison 2s

Note that the time to take the render the screenshot using a browser mainly depends on the size of the rendered page, for instance capturing a wsj.com page takes about 15s on the IM machine where the resulting jpeg image is as heavy as 10MB.

MapReduce

As you can see, the operations on the operations on the screenshots are very expensive (remember that the list of the tested browsers can be very long and for each we need to spend one browser screen shot operation). Therefore we need to parallelize the tool to several machines working on the input list of urls. To facilitate this, we have employed Hadoop MapReduce, which is part of the SCAPEs platform.
The result of the comparisons is then materialized in a set of XML files where each file represents one pair of browser shots comparisons. In order to alleviate the problem of having big numbers of small files, these files are automatically bundled together into one ZIP file.

Evaluations

We have run preliminary tests on the currently supported browser versions - Firefox and Opera. The list of urls to test is about 13 000 entries long. We are using the IM central instance for these tests, currently having two worker nodes (thus we can cut the processing time to half in parallel execution).

Requirements/Evaluation Criteria/Conditions of Satisfaction

- Must be able to compare 100 or more WARC images to their reference images per hour.
- Rate of comparison failure must be 0 - which is to say no WARCs should fail the comparison.

Notes/Context

Evaluations

- EVAL - WCT2-EX2 Comparing newly archived Web sites against a verified copy (multiple nodes)

11.5 WCT2-EX3 - Visual automated QA at large scale

User story: Comparison of Web Snapshots

Dataset

Internet Memory Web Archive, sample of 2.6 million URLs

Workflow

The IMF takes into account the quality of archived web sites. The quality is assured by a visual inspection: comparing the site in Internet with the archived site in IMF servers.
In order to improve that process, IMF is trying to develop an application, using the Pagelyzer developed UPMC, which compares two images. These two images are produced by Selenium based framework (V.2.24.1) by taking two snapshots: ideally, one is taken from the archive access and the second from the live.

Workflow:

1. Load live page, take screen shot (Selenium + Firefox headless)
2. Load web page from archive, take screen shot (Selenium + Firefox headless)
3. Visual comparison of screenshots (Pagelyzer)
4. Produce the output result file (score of comparison)

The difference between the previous multi-node experiment is in the deployment of the selenium tool (previously on a separate cluster). Now, the Selenium + headless Firefox is run on every processing machine.

Requirements/Evaluation Criteria/Conditions of Satisfaction

The requirements are to be able to process large amount of URLs (comparisons) in a reasonable time (days) on the available infrastructure (mid-size cluster, see description of platform in the evaluation). The previous experiments showed that the comparison using Pagelyzer is very time consuming (2s), the rendering of a page as well (2s). So we use these values as a goal. Closer analysis showed that page rendering of a page consists of three components:

1. Getting the source of a page (depends on the speed of the connection, saturation of link, speed of remote web server)
2. Rendering the page (depends on the speed of the local machine running headless Firefox)
3. Getting the snapshot (depends on the size of the page)

Note that 1. cannot be addressed by any optimization.

The outcome of this experiment is the frequency of scores coming from Pagelyzer that helps assess the quality of the crawl.

Notes/Context

Evaluations

- EVAL-WCT2-EX3 – Large Input Large Infrastructure

11.6 WCT EX2 File ID at SB

User story: File Format Identification and Characterisation of Web Archives

Investigator(s)

Per Møldrup-Dalum, SB

Dataset

SB Web Archive Data

Platform

SB Test Platform

Workflow

Since November 2011 we have been running FITS (link to component @ myexperiment) on a selection of our web content spread over the years from 2005 up till 2011.

The data is stored in ARC files on a SAN. These ARC files are fetched from this SAN, unpacked and the FITS are run on each ARC record.

Running FITS on a ARC record produces an XML file. These XML files from a single ARC are packed into TGZ files and made available to the Planning and Watch subproject.

Requirements and Policies

ThroughputGbytesPerHour >= 60
OrganisationalFit = ?

FITS is already in use within this institution, so file format ID using FITS would be useful. However, other tools may be used.

Evaluations

- EVAL-WCT3-1

11.7 WCT EX3 File ID at BL

User story: File Format Identification and Characterisation of Web Archives

Investigator(s)

William Palmer, BL

Dataset

BL Web Archive SCAPE Testbed Dataset

Platform

BL Hadoop Platform

Workflow

The workflow has been implemented using a native Java/Hadoop application called Nanite, which was originally developed within SCAPE and has since seen further development. Nanite uses Tika & Droid and operates directly on the content of arc/warc files using a RecordReader.

Nanite code is here: https://github.com/openplanets/nanite

The arc/warc files are held in HDFS

- Nanite gives one arc/warc file to a mapper, which then executes map methods on the contents using an arc/warc RecordReader.
- The Mapper currently processed each file/record from the arc/warc as follows:
  - characterise using Tika
  - characterise using Droid
  - (optionally - off by default) characterise using libmagic: https://github.com/openplanets/libmagic-jna-wrapper
  - file extension extracted from URI (if available)
  - content-type given by the original web server (if available)
- The Reducer reduces the output and provides a count for each occurrence of the same set of information

Requirements and Policies

ReliableAndStableAssessment = Is the code reliable and robust and handles errors sensibly with good reporting?
NumberOfFailedFiles = 0

Evaluations

- EVAL-BL-WCT-01

11.8 WCT EX4 File ID and characterisation at SB

User story: File Format Identification and Characterisation of Web Archives

Investigator(s)

Per Møldrup-Dalum, SB

Dataset

8TB of non-ordered data from the Danish Web Archive

Platform

SB Hadoop Platform

Workflow

A set of file references as NFS paths were aggregated into a set of input files. The input files were, one at a time, fed to the pre-programmed Hadoop module of the Nanite[86] project. During the set-up phase of the project a few corrections were implemented in this module.

Requirements and Policies

Evaluations

- EVAL-SB-WCT-04

[86] https://github.com/openplanets/nanite

11.9 Web Archive FITS Characterisation using ToMaR at ONB

User story: File Format Identification and Characterisation of Web Archives

Investigator(s)

Sven Schlarb, ONB

Dataset

ONB Web Archive Dataset

Platform

ONB Hadoop Platform

Purpose of the Experiment

The purpose of the experiment is to evaluate the performance of characterising web archive data available in form of ARC container files using FITS[87] (File Information Toolset) executed by ToMaR[88] with subsequent ingest into a MongoDB in order to make it available to the SCAPE profiling tool C3PO[89].

The SCAPE Execution Platform leverages functionality of existing command line applications by using ToMaR, a Hadoop-based application, which, amongst other things, allows for the execution of command line applications in a distributed way using a computer cluster.

FITS (File Information Tool Set) produces “normalised” output of various file format identification and characterisation tools. In order to be able to use this tool with web archive data, it is necessary to unpack the files contained in ARC or WARC container files in a first step. In a second step the FITS file format characterisation process can then be applied to the individual files using ToMaR.
Workflow

To run over all items inside the ARC.GZ files, the native JAVA map/reduce program uses a custom RecordReader based on the Hadoop 0.20 API. The custom RecordReader enables the program to read the ARC.GZ files natively and iterate over the archive file record by record (content file by content file). Each record is processed by a single map method call to detect its MIME type.

The workflow below is an integrated example of using several SCAPE outcomes in order to create a profile of web archive content. It shows the complete process from unpacking a web archive container file to viewing aggregated statistics about the individual files it contains using the SCAPE profiling tool C3PO:

[87] https://code.google.com/p/fits/
[88] https://github.com/openplanets/tomar
[89] https://github.com/peshkira/c3po

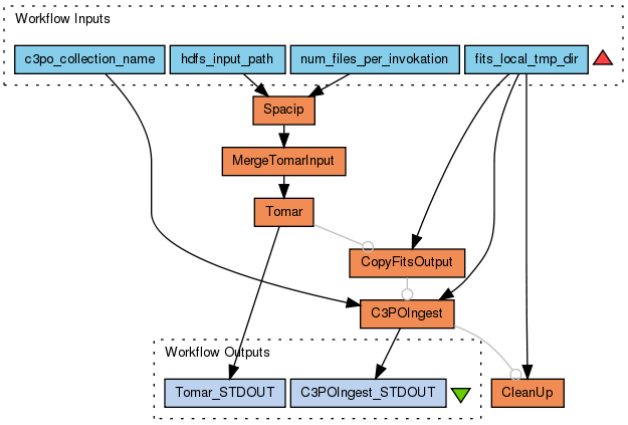


Figure 27 Web Archive FITS Characterisation using ToMaR, available on myExperiment[90]

The inputs in this worklow are defined as follows:

- "c3po_collection_name": The name of the C3P0[91] collection to be created.
- "hdfs_input_path", a Hadoop Distributed File System (HDFS) path to a directory which contains textfile(s) with absolute HDFS paths to ARC files.
- "num_files_per_invocation": Number of items to be processed per FITS invocation.
- "fits_local_tmp_dir": Local directory where the FITS output XML files will be stored

The workflow uses a Map-only Hadoop job[92] to unpackage the ARC container files into HDFS and creates input files which subsequently can be used by ToMaR. After merging the Mapper output files into one single file (MergeTomarInput), the FITS characterisation process is launched by ToMaR as a MapReduce job. ToMaR uses an XML tool specification[93] document which defines inputs, outputs and the execution of the tool. The tool specification document for FITS[94] used in this experiment defines two operations, one for single file invocation, and the other one for directory invocation.

FITS comes with a command line interface API that allows a single file to be used as input to produce the FITS XML characterisation result. But if the tool were to be started from the command line for each individual file in large a web archive, the start-up time of FITS including its sub-processes would accumulate and result in a poor performance. Therefore, it comes in handy that FITS allows the definition of a directory which is traversed recursively to process each file in the same JVM context. ToMaR permits making use of this functionality by defining an operation, which processes a set of input files and produces a set of output files.

The question how many files should be processed per FITS invocation was be addressed by setting up a Taverna[95] experiment like the one shown below.

[90] www.myexperiment.org/workflows/3933
[91] https://github.com/peshkira/c3po
[92] https://github.com/openplanets/hawarp/tree/master/tomar-prepare-inputdata
[93] https://github.com/openplanets/scape-toolspecs/blob/master/toolspec.xsd
[94] http://dl.dropboxusercontent.com/u/19171456/fits.xml
[95] http://www.taverna.org.uk/

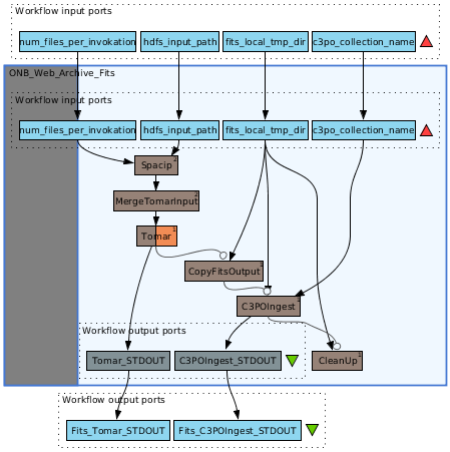


Figure 28 Wrapper workflow to produce a test series

The workflow presented above is embedded in a new workflow in order to generate a test series. A list of 40 values, ranging from 10 to 400 in steps of 10 files to be processed per invocation is given as input to the "num_files_per_invocation" parameter. Taverna will then automatically iterate over the list of input values by combining the input values as a cross product and launching 40 workflow runs for the embedded workflow.

5 ARC container files with a total size of 481 Megabytes and 42223 individual files were used as input for this experiment. The 40 workflow cycles were completed in around 24 hours and led to the result shown in figure 29.

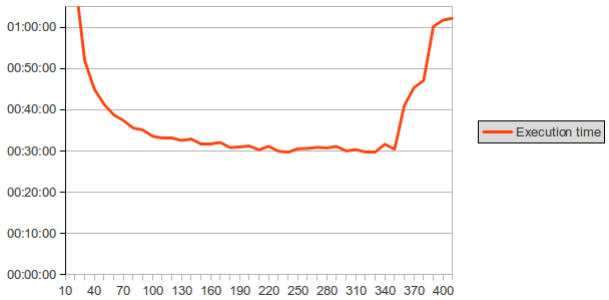


Figure 29 Execution time vs. number of files processed per invocation

The experiment shows a range of values with the execution time stabilising at about 30 minutes. Additionally, the evolution of the execution time of the average and worst performing task is illustrated in figure 30 and can be taken into consideration to choose the right parameter value.

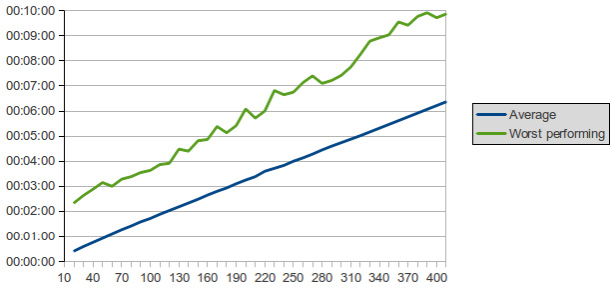


Figure 30 Average and worst performing tasks

Based on this preparatory experiment, the parameter to set the number of lines per ToMaR invocation was set to 250.

Evaluations

- EVAL Taverna-Fits-ToMaR-C3PO

12 Appendix C2 – Experiments in Large Scale Digital Repository Testbed

12.1 Characterisation and validation of audio and video files during ingest

User story: Characterisation of Large Audio and Video Files

Investigator(s)

Bolette Jurik, SB

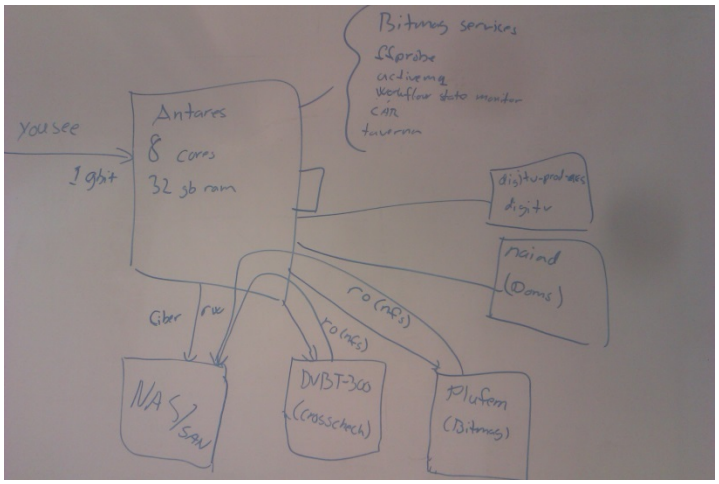
Dataset

Danish TV broadcasts, mpeg videos, Danish TV broadcasts, mpeg-2 transport stream and Danish Radio broadcast MPEG1-Layer 2.

This characterisation is done at ingest time, when new data is added to the collection. The daily Radio/TV broadcast ingest is around 800GB - 1TB.

Platform

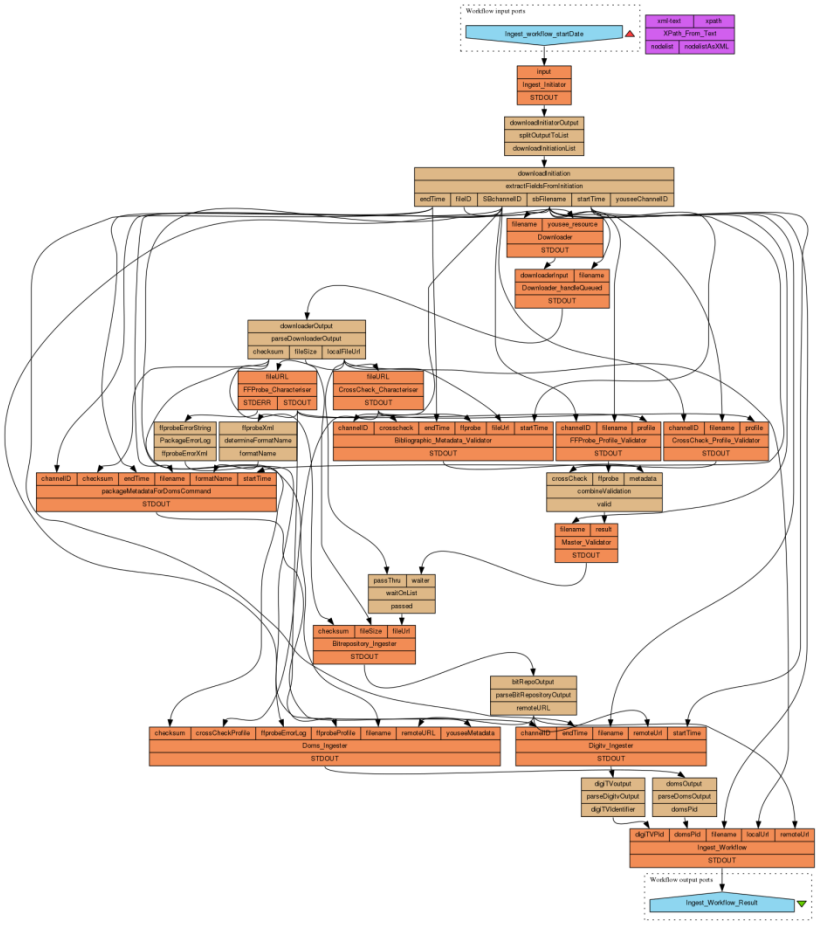
SB Video File Ingest Platform



Workflow

The Taverna workflow is part of the SB youseeingestworkflow Github repository[96].

[96] https://github.com/statsbiblioteket/youseeingestworkflow



Requirements and Policies

Evaluations

- EVAL Characterisation and validation of audio and video files during ingest

12.2 SB Experiment SO4 Audio mp3 to wav Migration and QA Workflow

User story: Large Scale Audio Migration

Investigator(s)

Bolette Jurik, SB

Dataset

Danish Radio broadcasts, mp3

Platform

SB Test Platform

This test used only one of the 5 servers in the SB Test Platform, see EVAL-LSDR6-1.

Workflow

myexperiment Workflow Entry: Mp3 To Wav Migrate QA CLI List Test[97]

Informally the workflow is as follows. Migration from Mp3 to Wav is done using FFmpeg[98] which is one of the SCAPE Action Services recommended tools. The QA is split into a number of steps. The first step is validation that the migrated file is a correct file in the wanted format. This is done using JHOVE2[99].The second step compares the header information properties of the original and the migrated files to see if they are 'close enough'. This is done using Ffprobe[100] to extract header information and Taverna Beanshells to compare the extracted properties. Another step could be to extract more properties by 'playing' the two files.

The third step uses an analysis tool comparing the sound waves. To do this we have to 'play' or interpret the mp3 file, just as a human needs to 'play' or interpret the file to hear the sound. The player used in this workflow is MPG321[101]. Note that MPG321 is an independent implementation of an mp3-decoder – thus independent from FFmpeg, which is used to actually migrate the file. The result of playing the file is a WAV file. The migrated file is already a WAV file, and we can compare the two files using the analysis tool xcorrSound[102] waveform-compare (earlier migrationQA).

Requirements and Policies

Evaluations

- EVAL-LSDR6-1

[97] http://www.myexperiment.org/workflows/3292.html
[98] http://wiki.opf-labs.org/display/TR/FFmpeg
[99] http://wiki.opf-labs.org/display/TR/JHOVE2
[100] http://wiki.opf-labs.org/display/TR/Ffprobe
[101] http://wiki.opf-labs.org/display/TR/MPG321
[102] http://wiki.opf-labs.org/display/TR/xcorrSound

12.3 SB Experiment Audio mp3 to wav Migration and QA on Hadoop Cluster

User story: Large Scale Audio Migration

Investigator(s)

Bolette Jurik, SB

Dataset

Danish Radio broadcasts, mp3

Platform

SB Test Platform

Workflow

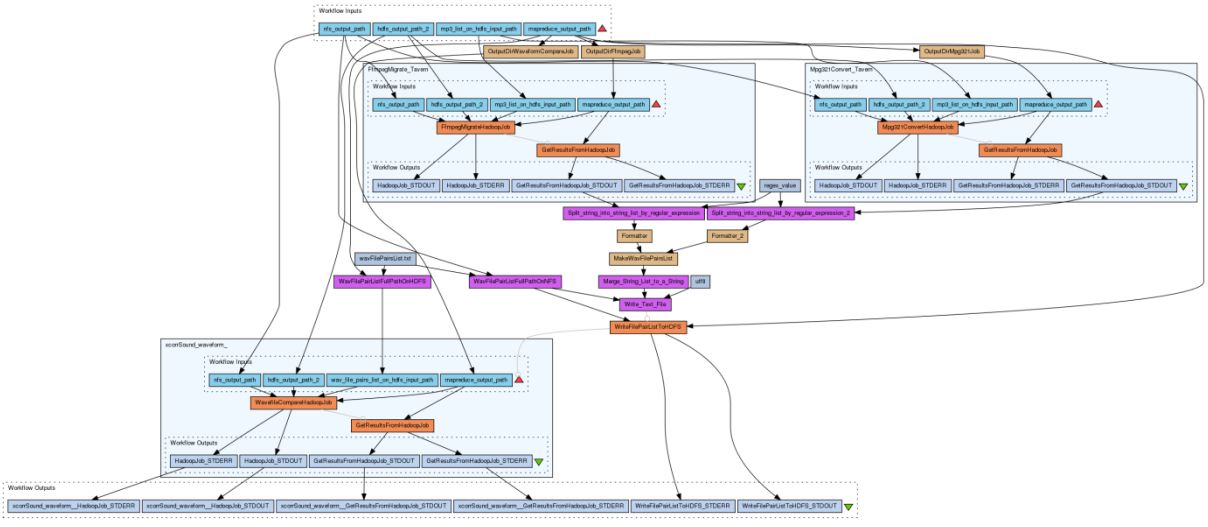
The workflow is the same as SB Experiment SO4 Audio mp3 to wav Migration and QA Workflow[103].

- Migration from Mp3 to Wav using FFmpeg
- Validation that the migrated file is a correct file in the wanted format using JHOVE2
- Extract and compare header information properties of the original and the migrated files using Ffprobe
- Convert the Mp3 file to Wav using MPG321
- Compare the two Wav files using xcorrSound waveform-compare (earlier migrationQA)

The difference is that the workflow is written as a number of Hadoop jobs / Hadoop mappers instead of a Taverna workflow.

The project is available from https://github.com/statsbiblioteket/scape-audio-qa.

In addition there now is a Taverna workflow combining three of these Hadoop jobs.



[103] http://wiki.opf-labs.org/display/SP/SB+Experiment+SO4+Audio+mp3+to+wav+Migration+and+QA+Workflow

To sum up what this workflow does, is migration, conversion and content comparison. The top left box (nested workflow) migrates a list of mp3s to wav files using a Hadoop[104] map-reduce job using the command line tool Ffmpeg[105], and outputs a list of migrated wav files. The top right box converts the same list of mp3s to wav files using another Hadoop map-reduce job using the command line tool mpg321[106], and outputs a list of converted wav files. The Taverna work flow then puts the two lists of wav files together and the bottom box receives a list of pairs of wav files to compare. The bottom box compares the content of the paired files using a Hadoop map-reduce job using the xcorrSound[107] waveform-compare command-line tool, and outputs the results of the comparisons.

Input/Output
The file containing the list of mp3 files to be migrated is available from HDFS. The mp3 files are stored on NFS and the resulting wav files are written to NFS. This has a number of reasons.

- The first is that the audio tools, we are using, were written to read from and write to NFS.
- Also at SB digitally preserved material does not reside on HDFS, which means that in order to migrate from and to HDFS, we would first need to copy the mp3s to HDFS and later copy the wavs from HDFS. These extra copy operations are expensive, when we are talking large-scale audio collections.
- Finally the SB Hadoop Platform is set up using network storage as local storage, which means that we do not exploit the HDFS locality property, and thus accessing the files on NFS rather than HDFS does not present a large overhead.

The preservation event and log files are all written to HDFS. This means we have a rather complex input/output model with input from both HDFS and NFS and also with output to both HDFS and NFS. And this is of course only an experiment! If this workflow is going to be used in production, we need to add the repository connection, such that data can be both retrieved from the repository and written to the repository.

Future Work
What we would like to do next is:

- Run an experiment using 1TB of mp3 files on the SB Hadoop cluster. This however requires some updates to the workflow. For 1TB input mp3 files, the workflow currently generates approximately 25TB of output and temporary wav files. Our test set-up is not suited for this, so we would like to delete these files along the way. Thus we would like the Taverna workflow to work on lists of lists of files. We can then limit the size of data written to eg. 2TB, then delete before continuing, as the only important output of the experiments is the comparison results and performance measurements. Also we can experiment with sending the Hadoop jobs lists of different sizes.
- Extend the workflow with property comparison. The waveform-compare tool only compares sound waves; it does not look at the header information. This should be part of a quality

[104] http://hadoop.apache.org/
[105] http://www.ffmpeg.org/
[106] http://mpg321.sourceforge.net/
[107] http://openplanets.github.io/scape-xcorrsound/

assurance of a migration. The reason this is not top priority is that FFprobe property extraction and comparison is very fast, and will probably not affect overall workflow performance much.

Requirements and Policies

Evaluations

- Evaluation - SB Experiment mp3 to wav Migration and QA on Hadoop Cluster

12.4 Characterisation and Identification on SCAPE Azure Platform

User story: Large scale document characterization and identification with Tika and DROID on SCAPE Azure platform

Introduction

This document describes a study that measured the speed of the Apache Tika Content Analysis Toolkit and the DROID File Format Identification Tool when they were run on a Microsoft Azure virtual machine. The results are compared to the speed of the same tools running on a traditional on-site server.

Background

The Apache Tika Content Analysis Toolkit (or "Tika," for brevity) is Java-based software that identifies the format of a file and extracts metadata from it. It's available at http://tika.apache.org/. The DROID File Format Identification Tool (or "DROID") is Java-based software that performs batch identification of file formats. It's available at http://www.nationalarchives.gov.uk/information-management/projects-and-work/droid.htm. Tika and DROID are important tools in the SCAPE digital content preservation project.

In an earlier study, SCAPE researchers evaluated these tools and measured their performance (Radtisch, May, Blekinge and Møldrup-Dalum, 2012). The ability of Tika and DROID to accurately identify file formats was determined by running them against a set of approximately one million files in the Govdocs1 corpus and comparing the results to a ground truth provided by Forensic Innovations, Inc. The speed of the tools was measured while running them on an on-site server.

This newer study measured the speed of these tools when they were run in a cloud environment, namely on a Microsoft Azure virtual machine (VM). (No effort was made to rerun the accuracy tests, as there is no reason to believe that the accuracy of the same tools would vary on different servers.) Tika and DROID are used today in an Azure-based SCAPE API Service implemented by Microsoft Research and might be used in future Azure-based projects, so the speed of the tools when running in the cloud is an important consideration.

Procedure

The procedure this study used to measure the speed of the tools adhered reasonably closely to what was done in the earlier study. The one concession to expediency was the use of a subset of about 12,000 randomly chosen documents instead of the entire corpus of one million.

Evaluations

- EVAL Characterisation and Identification on SCAPE Azure Platform

12.5 KB Metamorfoze Image Migration & QA

User story: Large Scale Image Migration

Investigator(s)

Clemens Neudecker, KB

Dataset

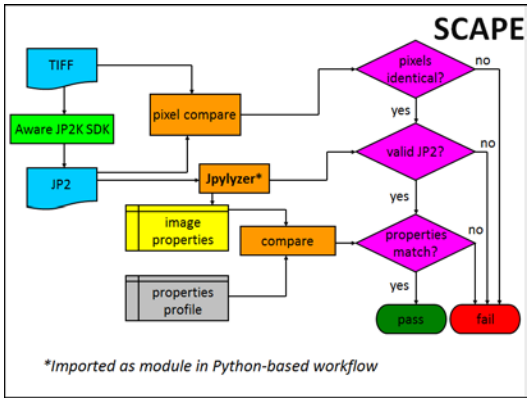
KB Metamorfoze Migration (sample batch) from the Metamorfoze[108] project.

Platform

KB Hadoop Platform

Workflow

The migration is implemented as a batch file[109], in Java code[110], and as a Taverna[111] workflow.



Latest Java code, workflow and batch files are available on GitHub[112].

The workflow comprises of the following steps:

- Recover TIFF file from storage (HDFS)
- Run Exiftool[113] to extract metadata from TIFF
- Migrate TIFF -> JP2 (using Aware JP2K SDK[114])
- Run Exiftool to extract metadata from JP2
- Run Jpylyzer[115] over the JP2
- Run Probatron validator[116] over Jpylyzer outputs to validate conformance of migrated image to the specified profile

[108] http://www.metamorfoze.nl/english
[109] https://github.com/KBNLresearch/hadoop-jp2-experiment/blob/master/run.sh
[110] https://github.com/KBNLresearch/hadoop-jp2-experiment
[111] http://www.taverna.org.uk/
[112] https://github.com/KBNLresearch/hadoop-jp2-experiment
[113] http://www.sno.phy.queensu.ca/~phil/exiftool/
[114] http://www.aware.com/imaging/jpeg2000sdk.html
[115] http://openplanets.github.io/jpylyzer/
[116] http://www.probatron.org/probatron4j.html

- Use GraphicsMagick[117] to compare TIFF and JP2
- Create report
- Create output package (JP2, results, etc.)
- Post files back to relevant storage

Requirements and Policies

- The JP2 files produced by the migration must be valid JP2s (checked with Jpylyzer)
- The JP2 files produced by the migration must adhere to a specific profile (checked with Probatron against Jpylyzer output)
- The Pixel-comparison with the original TIFF image must return identical results (checked with GraphicsMagick)

Evaluations

- EVAL KB Metamorfoze Image Migration & QA

[117] http://www.graphicsmagick.org/

12.6 LSDRT2 EX1 BL Newspapers on the BL Platform

User story: Large Scale Image Migration

Investigator(s)

William Palmer, BL

Dataset

BL 19th Century Digitized Newspapers

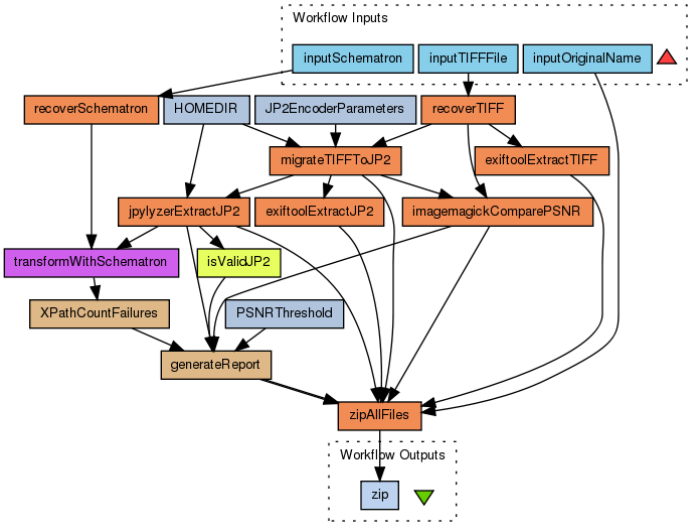
Platform

BL Hadoop Platform

Workflow

The workflow has been implemented as a Taverna workflow, in Java code and as a batch file.

The latest Taverna workflow is here: http://www.myexperiment.org/workflows/3401.html



Latest Java code, workflow and batch files are here: https://github.com/bl-dpt/chutney-hadoopwrapper/

The latest (Taverna/Java) workflow contains the following steps:

- *Recover TIFF file from storage (HDFS/Fedora/Webdav)
- Run Exiftool to extract metadata from TIFF
- Migrate TIFF->JP2 (using OpenJPEG/Kakadu)
- Run Exiftool to extract metadata from JP2
- Run Jpylyzer over the JP2
- *Run Schematron validator over Jpylyzer outputs to validate conformance of migrated image to the specified profile
- Use ImageMagick to compare TIFF and JP2
- *Create report

- Create output package (JP2, results, etc.)
- Post files back to relevant storage (see above)

Note that steps marked * are not performed in the batch workflow.

Requirements and Policies

NumberOfObjectsPerHour >= 1600 (This assumes we want to process the entire collection within 2 months).
ThroughputGbytesPerHour >= 25 (This assumes we want to process the entire collection within 2 months).
OrganisationalFit = "Can this workflow/solution/components be applied and used at the BL? Are the components using supported technology? etc."
NumberOfFailedFiles = 0 (We can probably lose speed, but we cannot without question lose files)

Evaluations

- http://wiki.opf-labs.org/display/SP/EVAL-LSDR3-1
- EVAL-BL-LSDRT-TIFFJP2-01

12.7 TIFF to JPEG2000 Migration Experiment at ONB

User story: Large Scale Image Migration

Investigator(s)

Sven Schlarb, ONB

Dataset

Austrian National Library Tresor Music Collection

Platform

ONB Hadoop Platform

Purpose of this experiment

The purpose of this experiment is to evaluate the performance of a scalable workflow for migrating TIFF images to images in the JPEG2000 format compared to an equivalent Taverna version of the workflow processing the data sequentially.

Evaluation method

A Taverna workflow for sequential processing serves as a reference point for the large-scale execution. Out of the full Austrian National Library Tresor Music Collection data set subsets of increasing size are selected by a random process.

The following bash statement is used to create a random sample from the full data set:

```
find . -type f -exec ls -l -sd {} + | grep ".tif" | awk 'BEGIN {srand()} {printf
"%05.0f %s \n",rand()*99999, $0; }' | sort -n |  awk '{print $10 "\t" $7}' | head -
$NUM > ~/tresormusicfilepaths${NUM}_withsize.csv
```

The statement prepends a random number to the file paths list and orders the list subsequently. Variable NUM is the desired size of the data set. The resulting file contains the local file paths and can be used as input for the Taverna workflow presented in the next section.

Additionally the files are uploaded to HDFS as input for the large-scale workflow execution.

By that way it is possible to compare the sequential execution time to the large-scale processing time.

Taverna workflow - sequential processing

The proof-of-concept version of the TIFF to JPEG2000 image migration workflow with quality assurance was created as a Taverna workflow illustrated by the following workflow diagram:

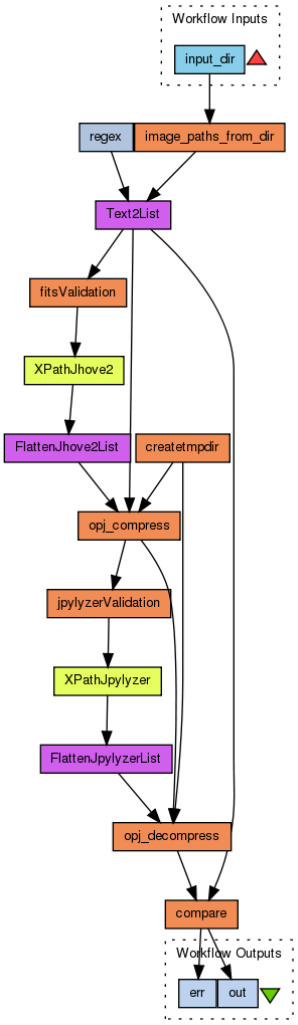


Figure 31 Taverna workflow

Diagram of the TIFF to JPEG2000 image migration workflow, Workflow available on MyExperiment at http://www.myexperiment.org/workflows/4276.html
The Taverna workflow reads a text file containing absolute paths to TIF image files and converts them to JP2 image files using OpenJPEG (https://code.google.com/p/openjpeg).

Based on the input text file, the workflow creates a Taverna list to be processed file by file. A temporary directory is created (createtmpdir) where the migrated image files and some temporary tool outputs are stored.

Before starting the actual migration, it is checked if the TIF input images are valid file format instances using Fits (https://code.google.com/p/fits, JHove2 under the hood, http://www.jhove2.org). An XPath service is used to extract the validity information from the XML-based Fits validation report.

If the images are valid TIF images, they are migrated to the JPEG2000 (JP2) image file format using OpenJPEG 2.0 (opj_compress).

Subsequently, it is again checked if the migrated images are valid JP2 images using SCAPE tool Jpylyzer (http://www.openplanetsfoundation.org/software/jpylyzer). An XPath service (XPathJpylyzer) is used to extract the validity information from the XML-based Jpylyzer validation report.

Finally, we verify if the migrated JP2 images are valid surrogates of the original TIF images by restoring the original TIF image from the converted JP2 image and comparing whether original and restored images are identical.

The sequential execution of this workflow is used as a reference point for measuring the parallelisation efficiency of the scalable version and it allows measuring how the processing times of the different components compare to each other.

The following diagram shows the average execution time of each component of the workflow in seconds and was created from a 1000 images sample of the Austrian National Library Tresor Music Collection:

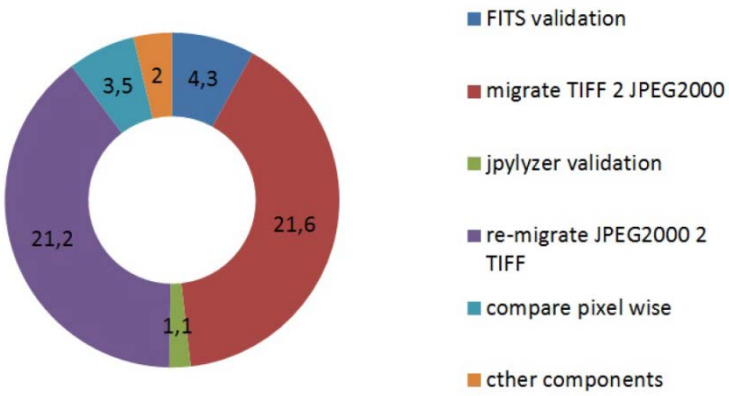


Figure 32 execution times of each of the workflows' steps

In the design phase this analysis is used to examine the average execution times for the individual tools. As a consequence of this experiment we might conclude, that over 4 seconds for the FITS-based TIF image validation takes too much time and that this processing step needs to be improved, while the Jpylyzer validation is acceptable taking only slightly more than 1 second per image file in average.

SCAPE Platform workflow - distributed processing

Apache Pig[118] was used to create a scalable version of this workflow. The different processing steps of the Taverna workflow for sequential processing are represented by Pig Latin statements.

The comments of each processing step in the script below indicate which the corresponding processing component in the Taverna workflow is.

[118] http://pig.apache.org/

```
/* file: tiff2jp2_migrate.pig */

/* Built from https://github.com/openplanets/tomar */
REGISTER /home/onbfue/ToMaR/target/tomar-1.5.2-SNAPSHOT.jar;

DEFINE ToMarService eu.scape_project.pt.udf.ControlLineUDF();
DEFINE XPathService eu.scape_project.pt.udf.XPathFunction();

SET job.name 'Tomar-Pig-Taverna-OpenJpeg';

/* make sure that one task per input file is created */
SET pig.noSplitCombination true;

SET mapred.task.timeout 420000

%DECLARE toolspecs_path '/hdfs/path/to/toolspecs';
%DECLARE xpath_exp1 '/fits/filestatus/valid';
%DECLARE xpath_exp2 '/fits/identification/identity/@mimetype';
%DECLARE xpath_exp3 '/jpylyzer/isValidJP2';

/* STEP 1: load image paths */
image_paths = LOAD '$image_paths' USING PigStorage() AS (image_path: chararray);

/* STEP 2: validation of tiff image files using fits  */
fits = FOREACH image_paths GENERATE image_path as image_path, ToMarService('$toolspecs_path',
CONCAT(CONCAT('fits stdxml --input="hdfs://', image_path), '")) as xml_text;

/* STEP 3: extract fits validity and mime-type using xpath */
fits_validation_list = FOREACH fits GENERATE image_path, XPathService('$xpath_exp1', xml_text)
AS node_list1, XPathService('$xpath_exp2', xml_text) AS node_list2;
fits_validation = FOREACH fits_validation_list GENERATE image_path, FLATTEN(node_list1) as
node1, FLATTEN(node_list2) as node2;
store fits into 'output/fits';
store fits_validation into 'output/fits_validation';

/* STEP 5: migration of tiff image files to jpeg2000 */
openjpeg = FOREACH fits_validation GENERATE image_path as image_path,
ToMarService('$toolspecs_path',CONCAT( CONCAT(  CONCAT('openjpeg image-to-j2k --
input="hdfs://', image_path), '" --output="'),  CONCAT(  CONCAT(   CONCAT('hdfs://',
image_path), '.jp2'),'"'))) as ret_str;
STORE openjpeg INTO 'output/openjpeg';

/* STEP 6: validation of migrated jpeg2000 files using jpylyzer */
jpylyzer = FOREACH fits_validation GENERATE image_path as image_path,
ToMarService('$toolspecs_path',CONCAT(CONCAT(CONCAT('jpylyzer validate --input="hdfs://',
CONCAT(image_path,'.jp2')), '" --output="'),CONCAT(CONCAT( CONCAT('hdfs://', image_path),
'.jp2.xml'),'"'))) as jpy_xml;
STORE jpylyzer INTO 'output/jpylyzer';

/* STEP 7: extract jpylyzer validity using xpath */
jpylyzer_validation_list = FOREACH jpylyzer GENERATE image_path, XPathService('$xpath_exp3',
jpy_xml) AS jpy_node_list;
jpylyzer_validation = FOREACH jpylyzer_validation_list GENERATE image_path,
FLATTEN(jpy_node_list) as node1;
store jpylyzer_validation into 'output/jpylyzer_validation';

/* STEP 8: migrate jpeg2000 image file back to tiff */
j2k_to_img = FOREACH fits_validation GENERATE image_path as image_path,
ToMarService('$toolspecs_path',CONCAT( CONCAT(  CONCAT('openjpeg j2k-to-image --
input="hdfs://', CONCAT(image_path,'.jp2')), '" --output="'),  CONCAT(  CONCAT(
CONCAT('hdfs://', image_path), '.jp2.tif'),'"'))) as j2k_to_img_ret_str;
STORE j2k_to_img INTO 'output/j2k_to_img';

/* STEP 9: compare orignal to restored image file */
imgcompare = FOREACH fits_validation GENERATE image_path as image_path,
ToMarService('$toolspecs_path',CONCAT( CONCAT(CONCAT('imagemagick compare-pixelwise --
inputfirst="hdfs://', image_path), CONCAT(CONCAT('" --
inputsecond="hdfs://',CONCAT(image_path,'.jp2.tif')),'" --
diffoutput="hdfs://')),CONCAT(image_path,'.cmp.txt"'))) as imgcompare_ret_str;
STORE imgcompare INTO 'output/imgcompare';
```

The following ToMaR tool specification files were used in this experiment:

- openjpeg.xml[119]
- imagemagick.xml[120] (compare.sh[121])
- fits.xml[122]
- jpylyzer.xml[123]

Note that these XML-based tool descriptions must be stored in the directory /hdfs/path/to/toolspecs which is declared as the toolspecs_path variable in the pig script above.

The script is then executed as follows:

```
pig -param image_paths=/hdfs/path/to/imagefiles/ tiff2jp2_migrate.pig
```

and produces the result files in the same directory where the input image files are located, for example, input image path /hdfs/path/to/imagefiles/imagefile.tif:

1. /hdfs/path/to/imagefiles/imagefile.tif.jp2 (result of the conversion to JP2)
2. /hdfs/path/to/imagefiles/imagefile.tif.jp2.tif (result of the re-conversion to TIF)
3. /hdfs/path/to/imagefiles/imagefile.tif.txt (result of the pixel-wise comparison between original and re-converted TIF files)

Evaluations

- EVAL TIFF to JPEG2000 Migration Experiment at ONB

[119] http://wiki.opf-labs.org/download/attachments/43515906/openjpeg.xml?version=1&modificationDate=1403515996000
[120] http://wiki.opf-labs.org/download/attachments/43515906/imagemagick.xml?version=1&modificationDate=1403516011000
[121] http://wiki.opf-labs.org/download/attachments/43515906/compare.sh?version=1&modificationDate=1403516057000
[122] http://wiki.opf-labs.org/download/attachments/43515906/fits.xml?version=1&modificationDate=1403516036000
[123] http://wiki.opf-labs.org/download/attachments/43515906/jpylyzer.xml?version=1&modificationDate=1403516196000

12.8 Ingest of digitized book METSs into Fedora 4

User story: Large Scale Ingest

ID

METSFedora4

Contact

Matthias Hahn, FIZ

User Story

Ingesting a huge amount of data into a repository could become a difficult task. The increasing amount of data that has to be ingested in a limited time, demands a repository that is able to scale in this respect.

We used the - still in development - Fedora 4 implementation based on an alpha release to measure the performance of ingest of data provided by the ONB and random generated data.

User Requirements/Components

1. I need a repository with an ingest throughput of 5000 Google Book Scans per month.

Experiments

We measured the ingest performance with Modeshape, as the underlying JCR repository implementation of Fedora 4, and measured the ingest performance with Fedora 4 without the SCAPE Connector API and with the SCAPE Connector API. All numbers have been distributed to Duraspace and have been discussed with the developers (including Modeshape developers). As a result the scalability and performance of a Fedora 4 cluster has been postponed to the Fedora 4.1 release and will not be part of Fedora 4.0 (the release date is still not known). All experiments are based on an alpha release of Fedora 4.0.

- Fedora 4 Ingest Throughput
- Modeshape Ingest Throughput
- SCAPE Fedora 4 Ingest Throughput

Related Documents

12.8.1 Fedora 4 Ingest Throughput

Investigator(s)

Frank Asseg
Matthias Hahn

Dataset

The Dataset used was a random set with different file sizes

Platform

We tested on diverse Cluster at the Steinbuch Center for Computing SCC, on a Amazon AWS instance, on a local Cluster at FIZ Karlsruhe and on a Cluster at Timisoara (a SCAPE Partner)
Workflow

We first deployed Fedora 4 in clustered mode on the Cluster Nodes with the help of Shell- and Puppet scripts to do this efficiently. We used a benchtool to measure the ingest performance of the datasets.

Evaluation Summaries

Experiment Fedora 4 at SCC Cluster:
https://wiki.duraspace.org/display/FF/Performance+evaluation+on+the+SCC+Cluster

Experiment Fedora 4 at AWS Cluster:
https://wiki.duraspace.org/display/FF/Performance+evaluation+on+AWS

Experiment Fedora 4 at FIZ Cluster:
https://wiki.duraspace.org/display/FF/Performance+evaluation+on+the+FIZ+cluster

Experiment Fedora 4 at Timisoar Cluster: for this test we do not have a Wiki Page at Duraspace, instead we provide the numbers here:

The cluster is currently running on these three nodes:

http://scape-fiz-1.info.uvt.ro:8080/fcrepo/rest
http://scape-fiz-3.info.uvt.ro:8080/fcrepo/rest
http://scape-fiz-4.info.uvt.ro:8080/fcrepo/rest

The first small ingest test of a single file with 10m yielded this result:

```
17:15  INFO  Found Fedora 4 at http://localhost:8080/fcrepo
16:17:15  INFO  Running 1 INGEST action(s) against FCREPO4 with a binary size of 10.0 MB using
1 thread(s)
16:17:16  INFO  The Fedora cluster has 3 node(s) before the benchmark
16:17:16  INFO  preparing 1 objects
16:17:16  INFO  creating 1 objects took 57 ms
16:37:16  INFO  scheduling 1 actions
17:08:55  INFO  purging 1 objects and datastreams
17:08:55  INFO  Completed 1 INGEST action(s) executed in 1898784 ms
17:08:55  INFO  The Fedora cluster has 3 node(s) after the benchmark
17:08:55  INFO  Throughput was 0.01 MB/sec
17:08:55  INFO  Condensed results:
17:08:55  INFO  1 10485760 1 INGEST 1898784 0.0052665286 no-tx
17:08:55  INFO  All operations completed in 3099875 ms
```

So it did work but the throughput with one file was only 0.1mb/s.
Running a larger test ingesting 100 Objects with 10mb each gives the same result.

The profile of the cluster is available here:
https://wiki.duraspace.org/pages/viewpage.action?pageId=50528479

12.8.2 Modeshape Ingest Throughput

Investigator(s)

Frank Asseg

Dataset

The same random dataset as for the test with Fedora 4 have been used: 50 x 1MB Random data, 3 JCR nodes werden pro 1MB file

Platform

We used the local cluster at FIZ Karlsruhe

Workflow

No real workflow here: deploy Modeshape on the Cluster and run the ingests test with the benchtool.

Evaluations Summary

Modeshape 3.7.1 - Local Cluster on two linux boxes

The result on a single node:

```
16:00:39  INFO  Benchmark finished.
16:00:39  INFO  ----------Overall results-------
16:00:39  INFO  Overall throughput 13.58 mb/sec
16:00:39  INFO  Overall size 50.0 MB
16:00:39  INFO  Overall duration 3.68 secs
```

Using a second modeshape node the performance drops significantly

```
16:03:15  INFO  Benchmark finished.
16:03:15  INFO  ----------Overall results-------
16:03:15  INFO  Overall throughput 0.54 mb/sec
16:03:15  INFO  Overall size 50.0 MB
16:03:15  INFO  Overall duration 92.76 secs
```

this result matches with the performance measured with Fedora 4.

12.8.3 SCAPE Fedora 4 Ingest Throughput

Investigator(s)

Matthias Hahn
Frank Asseg

Dataset

A random generated METS data set

Platform

We deployed Fedora 4 with the SCAPE Connector API Implementation on a Cluster at the Steinbuch Center for Computing[124].

Workflow

No real workflow here, but we used JMETER and a simple Loadbalancer based on Apache to perform the ingest.

Evaluations

A summary of the results is available at:
https://portal.ait.ac.at/sites/Scape/PT/Shared%20Documents/PT.WP.5%20Repository%20Integration/Performance_and_Scalability_Tests_with_Fedora_4.pdf

[124] https://wiki.duraspace.org/display/FF/Test+-+Platform+Profile%3A+Cluster+at+Karlsruhe+Institute+of+Technology+-+SCC

12.9 Validate PDF&EPUBs and check for DRM

User Story: Policy-Driven Identification of Preservation Risks in Electronic Document Formats

Investigator(s)

William Palmer, BL

Datasets

Govdocs1 Corpus

1. Using Govdocs1 corpus (231,683 PDFs/ 127.8GB) for initial testing - http://digitalcorpora.org/corpora/files/
2. ~~Seeking access to internal dataset of PDFs (~40k) (not currently tested)~~
3. Very small internal-only dataset of EPUBs (not currently tested)

Platform

BL Hadoop Platform

Workflow

Uses a simple Hadoop MapReduce program (FlintHadoop) to execute Flint over input files stored in HDFS. Using sequence files for input would require additional changes to the code and the benefit may be minimal.

Flint currently uses Apache PDFBox, iText, Jhove, EpubCheck, Tika and Calibre along with its own code, to determine if files are valid, and whether or not they contain DRM. Results from each test/tool are provided in the output XML.

For PDF files all the tools/libraries used for analysis are written in Java.

~~DRMLint: https://github.com/willp-bl/drmlint~~

Flint source code: https://github.com/openplanets/flint

The MapReduce steps are as follows:
Map: retrieve file from HDFS and run Flint on it. Flint checks for validity, DRM, validates against a policy, produces a report xml file, and additionally extracts text from the PDF. The report and extracted text are placed into a zip file that is stored in HDFS.

Reduce: process each of the outputs from Flint and produce a csv that has one line per file containing detailed results

Flint contains the following checks for PDF files (all Java code):

Validity:

- Check with Apache PDFBox
  - Runs Apache Preflight - if a PDF syntax error is detected then fails validation
  - Then tries to extract text from the PDF using Apache PDFBox
- Check with iText

- Try and extract text from each page of the PDF, fail validity checks if errors encountered

DRM:

- Check PDDocument.isEncrypted() with Apache PDFBox
- Manual scan for "/encrypt" keyword in the PDF
- Check PdfReader.isEncrypted() with iText
- NOTE: checks are not currently made against print/copy restrictions etc.

Ideally the current checks for validity and DRM will be validated against a set of files with a known ground-truth.

For the July 2014 evaluation:
DRMlint, now renamed Flint, has been further developed and extended to now check PDF files against an institutional policy. This work is based on Apache PDFBox Preflight/Schematron work from the KB. For example - JavaScript or embedded files can now be detected, and a policy can be defined that will "fail" such PDF files, allowing institutions to validate against their own institutional policies. Full results are in the Reducer output and can be easily post-processed.

The evaluation completed in July 2014 performs the same checks as before and adds the additional policy validation check.

Requirements and Policies

ReliableAndStableAssessment = Is the code reliable and robust and does it handle errors sensibly with good reporting?
NumberOfFailedFiles = 0

Evaluations

- EVAL-BL-LSDRT-PDFDRM-01

12.10 Validate JPEG2000 Newspapers Using Jpylyzer

User story: Validation of Archival Content against an Institutional Policy

Investigator(s)

Rune Bruun Ferneke-Nielsen, SB

Dataset

Danish newspaper - Morgenavisen Jyllandsposten

Platform

SB Hadoop Platform

Workflow

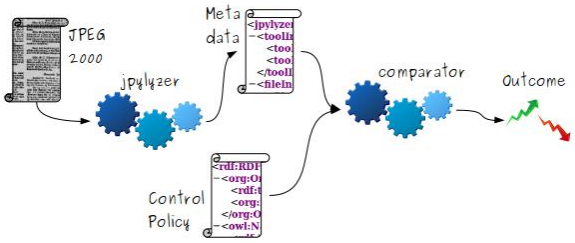
The idea behind this experiment is that you have a digital newspaper collection, in JPEG 2000[125] format, and you want to verify that certain properties hold true for every file in the collection. The properties that should hold true are specified in a control policy[126], which at least contains information about the digital newspaper collection.

Evaluation 1

The first iteration of the experiment will use a very simple setup and focus on processing the files using jpylyzer[127] - we would like to get a first indication of the performance without bringing extra complexity into the equation. Therefore, files will be read from local storage instead of using our repositories as would normally be the case. Moreover, output from the processing will be discarded - failing processes being the exception - instead of being stored in our repositories. The Hadoop configuration has not been altered, apart from the necessary settings that correspond to our cluster.

About the details

First step will extract metadata, using jpylyzer, from each file in the newspaper collection. Second step will compare the extracted meta-data against the control policy, and report on any differences.



Considerations

- Where are newspaper collection files stored - in a repository, on local storage (outside of Hadoop), hdfs storage?

[125] http://www.jpeg.org/jpeg2000/
[126] http://wiki.opf-labs.org/display/SP/Catalogue+of+Preservation+Policy+Elements
[127] https://github.com/openplanets/jpylyzer

- What is the appropriate number of concurrently running tasks? One way to handle this is by specifying split size, which will determine how many map tasks to start. Moreover, jpylyzer is able to handle several input paths, which is kind of a second level in handling concurrently running tasks. Specification of hardware should off course be taken into consideration in this discussion (available nodes, cpu cores & threads, memory, etc.).
- Should generated meta-data be stored, and where - in a repository, on local storage, hdfs storage, discard it?

Improvements & suggestions

- configure 'fair scheduler' for Hadoop cluster, thereby controlling the number of simultaneous running (map) tasks.
- as of February 2014, this experiment is missing a component/tool for converting tool-specific output into a scape-generic output - this has been mocked for the experiment.

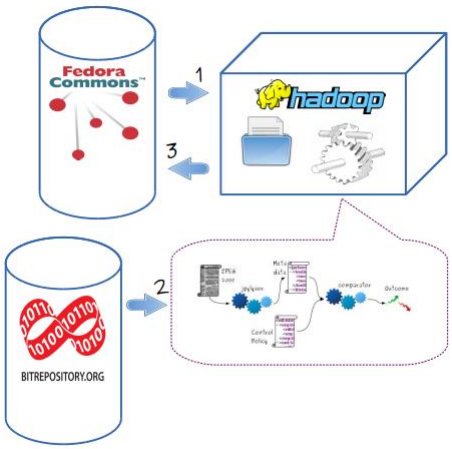
Evaluation 2

Building upon the results from the first iteration, we want the experiment to reflect our reality more. We extend the setup further by adding repositories, where data will be read from and written to. In details, we will use a Fedora[128]-based repository for reading and writing content meta-data, and a bit[129] repository for reading content. By adding these systems, we need to extend the experiment with components that can load and store data in an efficient manner.

About the details

The environment has been extending to also include the two repositories, containing metadata and content of the images.

1. Extracting metadata from Fedora-based repository
2. Performing quality assurance on Hadoop platform
3. Storing metadata into Fedora-based repository



[128] http://www.fedora-commons.org/
[129] http://digitalbevaring.dk/det-nationale-bitmagasin/

Step 1 can be split into:

- provide list of IDs for objects to extract from repository

```
uuid:723110d9-c76d-4e1b-870d-4968c3ebdf51
uuid:1c0194a3-c5af-4b40-b140-5ac64cfa43af
uuid:ef88a6b3-2ce3-4dd5-b0c7-8bebe1656aa5
uuid:d7183549-f108-40a3-81a8-704114667174
uuid:548a7a25-82d6-47b5-9e14-eae3adabd423
```

- transform objects into METS[130] documents; sample[131]
- store METS documents in a Hadoop sequence[132] file

Step 2 can be split into:

- read sequence file, and for each METS document
  - get file reference to locate image file on NFS mount, can be found under <mets:file> node
  - validate image using Jpylyzer and control policy
  - write image metadata and validation result into METS document
- store METS documents in a sequence file

Step 3 can be split into:

- read updated METS documents from sequence file, and for each METS document
  - update corresponding repository object with changes from METS document

The SCAPE Stager & Loader components

Extracting and storing data in the repository is handled by the SCAPE Stager and Loader components[133], which both interact with the repository through a SCAPE Data Connector[134].

Requirements and Policies

Evaluations

- Evaluation 1 - JPEG2000 validation

[130] http://www.loc.gov/standards/mets/

[131] http://wiki.opf-labs.org/download/attachments/36012173/uuid1c0194a3c5af4b40b1405ac64cfa43af.xml?version=1&modificationDate=1404290974000

[132] http://wiki.apache.org/hadoop/SequenceFile

[133] https://github.com/statsbiblioteket/scape-stager-loader

[134] https://github.com/statsbiblioteket/scape-doms-data-connector

13 Appendix C3 – Experiments in Research Datasets Testbed

13.1 raw2nexus Experiment at STFC

User story: Migration from local format to domain standard format

Investigator(s)

Alastair Duncan, STFC

Dataset

Platform

STFC Hadoop Platform

Workflow

raw2nexus migration[135]

Requirements and Policies

Evaluations

- raw2nexus migration large dataset big files
- raw2nexus migration large dataset copied from small dataset
- raw2nexus small dataset evaluation

[135] http://www.myexperiment.org/workflows/3954

13.2 GeoLint Experiment

User story: Identification, validation and checksumming of a complex corpus

Investigator(s)

William Palmer, BL

Dataset

A ~1.4TB set of geospatial data files from Ordnance Survey (GB & NI). Main data types are GML and NTF, but several other file types are included.

Within the dataset file sizes vary significantly; the mean filesizes for the four main filetypes (by total size) are:

Filetype	Mean size
NTF	214KB
GML (Gzipped)	3MB
TIF	7.2MB
ISO	3113MB

Platform

BL Hadoop Platform

Workflow

The workflow is implemented as a native MapReduce program (GeoLintHadoop), based on previous Flint/DRMLint work.

GeoLintHadoop is responsible for recovering the file from HDFS, to a local temporary directory for processing. This is necessary as GeoLint uses GDAL.OGR JNI libraries to read through NTF files and that requires a file to be available.

To reduce the time it takes GeoLintHadoop to process the files it generates a series of checksums (cksum CRC, CRC32, MD5, SHA-1 and SHA-256) at the same time as copying the data.

Once this is complete GeoLintHadoop calls GeoLint to process the file. The following steps are performed:

1. The file is identified using Apache Tika, using a custom-mimetypes.xml specifically relating to geospatial files that is included in GeoLint
2. If the file is a GML file it is validated against the Ordnance Survey XML Schema (http://www.ordnancesurvey.co.uk/xml/schema/)
3. If the file is an NTF file it is validated using internal code and using the GDAL/OGR library (http://www.gdal.org/drv_ntf.html)
4. The resulting XML is passed to the Reducer for collation
5. Any Exceptions are also reported in the Reducer outputs

The XML output from GeoLint can be used for post processing. The output for 1.03TB of data is a ~1.6GB XML file.

Checksum manifests can be verified against that data, or statistics can be derived from that data, such as mimetype, extension, number of files of a particular type etc.

GeoLint code is here: https://github.com/bl-dpt/geolint

Requirements and Policies

ReliableAndStableAssessment = Is the code reliable and robust and does it handle errors sensibly with good reporting?
NumberOfFailedFiles = 0

Evaluations

- GeoLint Evaluation

14 Appendix C4 – Experiments in Data Center Testbed

14.1 Scene reconstruction

User story: Large scale video processing and interlinking

Investigator(s)

Pavel Smrz, BUT
Ondrej Klima, BUT

Dataset

Malaga Urban Dataset

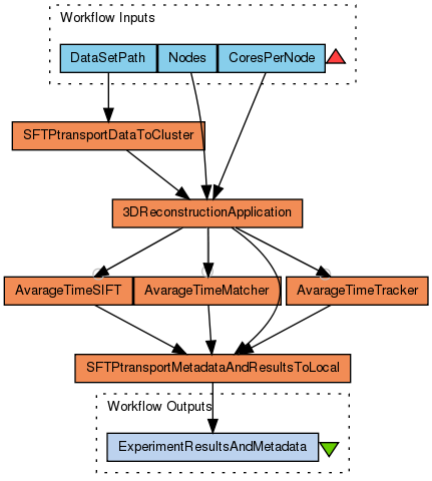
Platform

UVT Hadoop Platform

Purpose of this Experiment

This experiment aims at finding an optimal distribution of the reconstruction algorithm across the cluster of computation nodes. The experimental application will be executed many times with different numbers of nodes and processes running on each computation node. Time necessary for each step of the algorithm – the extraction of SIFT features, matching these features between corresponding stereo images and tracking them in neighbour images – will be preserved for each input picture in the XML metadata and then processed by Hadoop-based tools. Results of the metadata analysis will characterize the average computation time of each mentioned processing stage.

Workflow



Requirements and Policies

Evaluations

- Average time evaluation

14.2 Video annotation and geo localization

User story: Large scale video processing and interlinking

Investigator(s)

Ondrej Klima, BUT

Dataset

BUT Alp Mountains Dataset

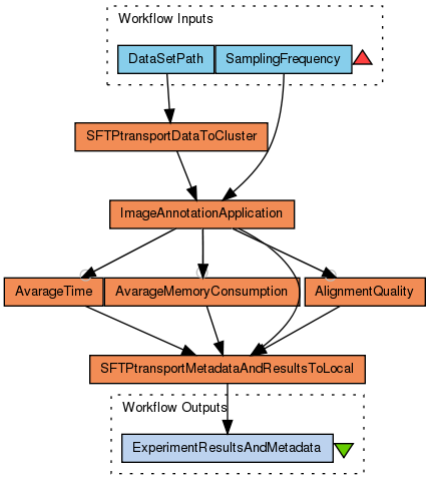
Platform

UVT Hadoop Platform

Purpose of this Experiment

This experiment explores relations among the frequency of edge sampling, the amount of memory and the time needed to compute the alignment with a certain sampling frequency and the corresponding alignment quality. This experiment produces a significant amount of metadata. Execution time needed for computing cross-correlations and the special metrics will be saved for each alignment, as well as the amount of memory consumed by the computation. This information will be preserved in the XML metadata files. The frequency of sampling will be varied and the experiments will be performed for each sampling value. The average value of memory and time consumption and the percentage of successfully aligned cases will be computed from metadata for each value of sampling frequency using the Hadoop tools.

Workflow



Requirements and Policies

Evaluations

- Evaluation of memory consumption
- Performance evaluation
- Precision of alignment evaluation

14.3 Performance tests for accessing medical data

User story: Large scale access at hospital (Medical Dataset)

Investigator(s)

Paweł Kominek, Michał Kozak, Aleksander Stroiński, Tomasz Parkoła

Dataset

The DICOM medical data for this experiment comes from the overall WCPT dataset described here: WCPT medical dataset.

Platform

PSNC Hadoop Platform

Purpose of this experiment

The main goal of this experiment is to test performance of a single instance of the DICOM data provider which is running on one of the nodes of the PSNC Hadoop cluster. The resulting statistics should show how many requests can be handled by a single DICOM data provider which is running on PSNC Hadoop cluster.

Workflow

The following steps will be undertaken and evaluated in this experiment:

1. Produce multiple requests for downloading random DICOM files
2. Locate corresponding DICOM file
3. Create response and send to the requesting client
4. Receive the response from the DICOM data provider

Requirements and Policies

It is required that only a single instance of the DICOM data provider is used. We do not envision additional instances which (e.g. load-balanced).

Evaluations

- Evaluation of DICOM data access

14.4 Performance tests of the search function in the MDC portal

User story: Large scale access for educational purposes (Medical Dataset)

Investigator(s)

Paweł Kominek, Michał Kozak, Aleksander Stroiński, Tomasz Parkoła

Dataset

The medical dataset for this experiment comes from the overall WCPT dataset described here: WCPT medical dataset.

Platform

PSNC Hadoop Platform

Purpose of this experiment

The main goal of this experiment is to test performance of the search functionality build-in into the MDC portal. The statistics should show how many concurrent users can use MDC portal.

Workflow

The following steps compose the search functionality in the MDC portal:

1. Request from the user (search in the MDC portal)
2. Search request interpretation and lookup in the Hadoop cluster for the best matches
3. Creation of the response, including data retrieval from HBase/HDFS
4. Response to the user with the search results

Requirements and Policies

The experiment should test only a single instance of MDC portal (e.g. no load balancing should be used).

Evaluations

- Performance depending on search criteria

14.5 Analysis of epidemiological situation across WCPT patients

User story: Large scale analysis (Medical Dataset)

Investigator(s)

Paweł Kominek, Michał Kozak, Aleksander Stroiński, Tomasz Parkoła

Dataset

The DICOM medical data for this experiment comes from the overall WCPT dataset described here: WCPT medical dataset. The data for this experiment will be composed of HL7 XML files.

Platform

PSNC Hadoop Platform

Purpose of this experiment

The goal of this experiment is to analyse the epidemiological situation across WCPT patients. It includes the following analysis:

- Age of patients treated in a given period
- Sex of patients treated in a given period
- Number of cases of a given disease in a given period
- Number of abnormal results in laboratory examinations for a given disease codes in a given period
- Average time of patient's visit for a given disease codes in a given time period

Workflow

The analysis will be done via means of Hadoop jobs which will be executed on PSNC Hadoop cluster. There will be no specific general workflow, as the algorithm implemented in Hadoop job will handle all of the processing. The main steps in the analysis will be:

1. Define input and output and criteria for the analysis
2. Implement Hadoop job
3. Execute Hadoop job and gather results
4. Prepare results in a human-readable way

Evaluations

- Evaluation of the age of patients treated in a given period
- Evaluation of the average time of patient's visit for a given disease codes in a given time period
- Evaluation of the number of abnormal results in laboratory examinations for a given disease codes in a given period
- Evaluation of the number of medical cases for a given period
- Evaluation of the patients gender for a given period

14.6 WCPT to PSNC DICOM medical data ingest

User story: Large scale ingest of medical data (Medical Dataset)

Investigator(s)

Paweł Kominek, Michał Kozak, Aleksander Stroiński, Tomasz Parkoła

Dataset

The DICOM medical data for this experiment comes from the overall WCPT dataset described here: WCPT medical dataset. For this particular experiment we envision data storage test of approx. 10GB of medical data, which is approx. the amount of data produced by the WCPT hospital in one day.

Platform

PSNC Hadoop Platform

Purpose of this experiment

The main goal of this experiment is to evaluate how fast the data center facilities are able to process the amount of data produced by WCPT in one day. The evaluation should measure time of storage on HDFS cluster, storage on country-wide cloud storage and metadata ingest into HBase. The evaluation should be a real case scenario, therefore it is to be executed via the network connection between WCPT and PSNC, which is limited by the asynchronous link with 100Mbps throughput.

Workflow

The ingestion process which will be investigated in this experiment is composed of the following steps:

1. Receive data from the WCPT endpoint and store them on HDFS
2. Validate received data and extract necessary information (validate required DICOM tags, extract information from certain DICOM tags)
3. Store extracted information from the DICOM tags into the HBase (this step is necessary for further processing)
4. Rename DICOM file
5. Store backup copy of the DICOM file in the cloud storage

Requirements and Policies

It is required that the experiment is executed in real case environment, it means that the experiment needs to involve WCPT and PSNC parts. It should be initiated from the WCPT environment, and the measurements should be done at PSNC side, where all of the investigated activities take place.

Evaluations

- Evaluation of DICOM data ingest (with copying data to archiving system)
- Evaluation of DICOM data ingest (without copying data to archiving system)

15 Appendix D – Platform

15.1 SB Video File Ingest Platform

SB Video File Ingest Platform

Field	Datatype	Value
Platform-ID	String	SB Video File Ingest Platform
Platform description	String	The Video File Ingest Platform at SB includes a central server (antares) which runs the ingest workflow. The central server has access to the scratch NAS/SAN, where the files are stored during ingest, and to 4-6 different servers running services used in the ingest workflow. The workflow itself is not distributed. It runs on the central server, which is described here.
Number of nodes	Integer	1
Total number of physical CPUs	Integer	2
CPU specs	String	Intel® Xeon® Processor X5355 (8M Cache, 2.66 GHz, 1333 MHz FSB)
Total number of CPU-cores	Integer	32
Total amount of RAM in GB	Integer	32
Average CPU-cores for nodes	Integer	32
Average RAM in GB for nodes	Integer	32
Operating System on nodes	String	Linux (CentOS release 6.3 (Final))
Storage system/layer	String	NAS/SAN + NFS
Network layer between nodes	String	3* 1 gbit network. One for download, One to storage (fibrechannel). One to certain services on other internal servers (ethernet).

Parallel Execution System

Field	Value
Installation description	-
Configuration notes	-

Benchmarks of Parallel Execution System

Metric	Benchmark type	Value
-	-	-

15.2 SB Test Platform

SB Test Platform

Field	Datatype	Value
Platform-ID	String	Platform SB 1
Platform description	String	We have five Blade servers located at SB
Number of nodes	Integer	5 physical servers
Total number of physical CPUs	Integer	10
CPU specs	String	Intel® Xeon® Processor X5670 (12M Cache, 2.93 GHz, 6.40 GT/s Intel® QPI)
Total number of CPU-cores	Integer	60
Total amount of RAM in GB	Integer	288
Average CPU-cores for nodes	Integer	6
Average RAM in GB for nodes	Integer	96
Operating System on nodes	String	Red Hat based Linux
Storage system/layer	String	SAN and EMC Isilon
Network layer between nodes	String	2 GB ethernet

Parallel Execution System

Field	Value
Installation description	-
Configuration notes	-

Benchmarks of Parallel Execution System

Metric	Benchmark type	Value
-	-	-

15.3 SB Hadoop Platform

SB Hadoop Platform

Field	Datatype	Value
Platform-ID	String	Platform SB Hadoop
Platform description	String	We have five Blade servers located at SB
Number of nodes	Integer	4 physical servers
Total number of physical CPUs	Integer	8
CPU specs	String	Intel® Xeon® Processor X5670 (12M Cache, 2.93 GHz, 6.40 GT/s Intel® QPI)
Total number of CPU-cores	Integer	48
Total amount of RAM in GB	Integer	348
Average CPU-cores for nodes	Integer	12
Average RAM in GB for nodes	Integer	96
Operating System on nodes	String	Red Hat based Linux
Storage system/layer	String	mounted NFS
Network layer between nodes	String	2 GB Ethernet

Parallel Execution System

Field	Value
Installation description	Cloudera CDH4, Hadoop 2.0.0-cdh4.5.0
Configuration notes	Replicationfactor is n/a. Using HDFS protocol for EMC Isilon Storage System 1 x Namenode (JobTracker) 4 x Datanode (TaskTracker)

Benchmarks of Parallel Execution System

Metric	Benchmark type	Value
Concurrent write throughput	TestDFSIO	2.6 Gb/s
Concurrent read throughput	TestDFSIO	4.4 Gb/s
Execution time	TeraSort of one billion 100 bytes index's (100GB)	10,07 minutes

15.4 MSR Azure Platform

SCAPE Azure Architecture

Architecture components

Following SCAPE Azure v.1.0 components are the building blocks of implemented architecture:

- Authentication is in charge of all the User Authentication (e.g. user profile and authentication)

With Service Authentication we want to ensure that external services can communicate securely with internal services currently running in SAZ.

- SCAPE Azure Execution Layer is responsible for running and managing all the operations and logging within SAZ.
- Content Representation Layer is metadata layer which is describing Data, Reports, Logs and Workflows. It maps stored data and metadata in SQL Azure.
- Tools and Resources Layer represents our Action services and tools we are using for Characterization, Conversion, Comparison and Reporting.
- Data store is virtually unlimited storage in BLOB.

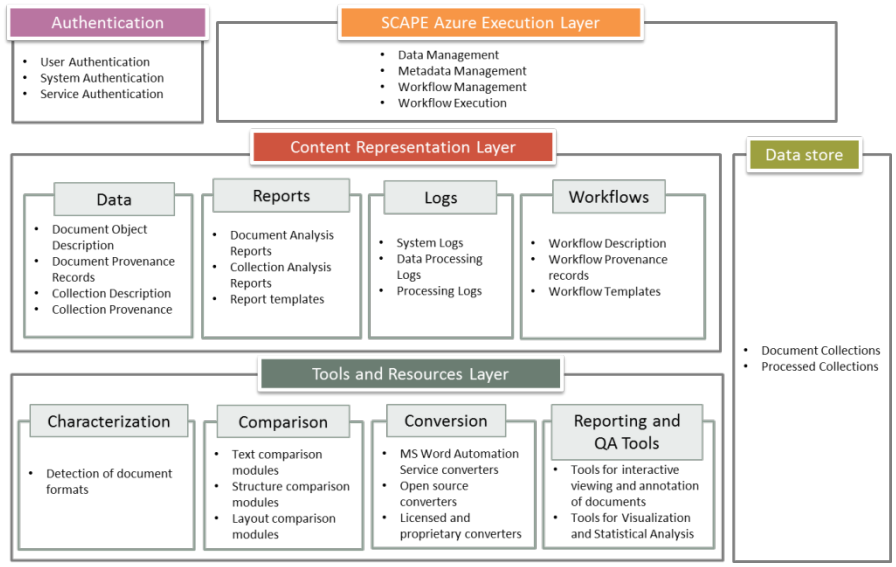


Figure 33 Architecture components of SCAPE Azure v.1.0

Implemented architecture

SCAPE Azure v.1.0 is implemented so that data is stored in the Azure BLOB storage, Tables, and SQL Azure. Conversion functions leverage SharePoint 2010 Word Automation Services. The service communication is facilitated through WCF Services. It is also possible to communicate directly with the BLOB storage via REST.

Figure 34 shows the details of the implemented architecture. Data is placed in the blob storage. Conversion and comparison functions are implemented as worker roles. SharePoint is placed in a VM environment and the Word Automation Services are leveraged to convert document formats. Reporting services are under development. They will aggregate processing information, ranging from

system performance related to ingest, conversion, and comparison, to qualitative data about the quality of the conversion, based on different techniques.

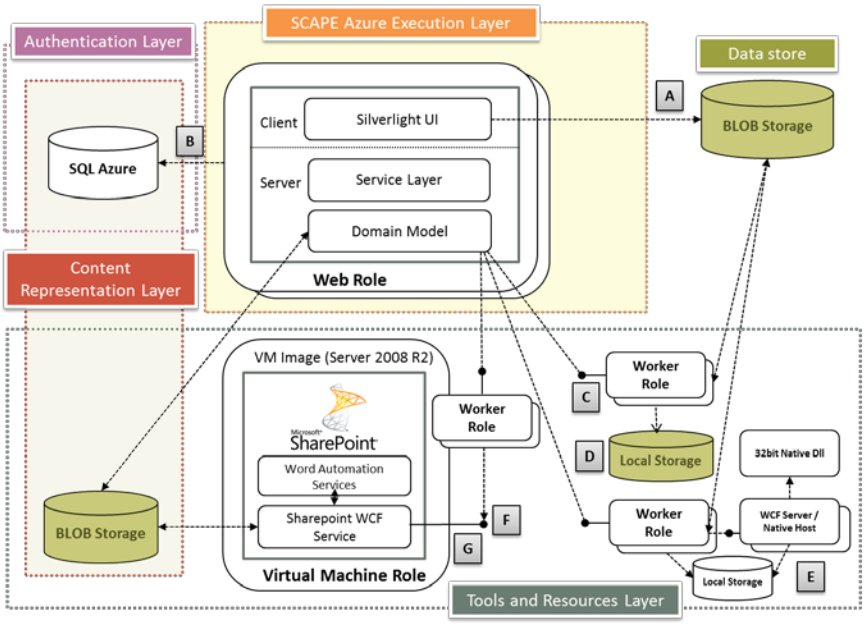


Figure 34 System architecture of SCAPE Azure v.1.0

Legend for the Figure 34, describing all the scenarios that are supported by SCAPE Azure Architecture:

A. Client has direct access to BLOB storage (virtually unlimited storage) via a REST API. This improves responsiveness since there is no need for access services hosted on the server
B. SQL Azure database stores user profiles and profile management information. Alternative database solutions can be employed either locally or within the cloud, e.g. MySQL)
    1. A worker role (processing node) encapsulates a number of discrete actions such as data processing functions, diagnostics, analysis, QA methods, etc.
    2. The actions of a worker role can be exposed externally and internally
    3. Scalability is attained by replicating Worker Roles. One can instantiate any number of processing nodes for conversion, analysis, comparison, QA or other operations on the data
C. External endpoints can make use of the Azure load balancer. For internal endpoints, the most applicable solution can be employed by the Domain Model
D. Temporary local storage within the worker roles can be employed when performing analysis or conversion. That eliminates the need for continuous communication with the Blob storage system and improves performance and reliability
E. Using a WCF endpoint as a proxy it is possible to run legacy 32 bit applications within worker roles enabling legacy software to be employed and scaled as necessary. Worker roles run within 64-bit environment by default
F. VM roles can be hosted with the same scaling and redundancy capabilities as other types of roles within Azure:
    1. SharePoint Word Automation Services (WAS) have been enabled within the SCAPE portal

2. Worker roles make calls to SharePoint service hosted on the VM. The exposed SharePoint service, with access to WAS, is only available via an internal endpoint although could be made external if necessary)
3. The SharePoint Service initiates WAS by retrieving document from the BLOB storage area and upon transformation transfers converted document back to BLOB storage

G. A Second VM hosts OmniPage (note this VM is not graphically represented on Figure 34) to perform the OCR duties before analysis is performed and results transferred to BLOB storage

Figure 34 glossary

- **SQL Azure** – cloud-based, scale-out version of MS SQL Server
- **Web Role** – we used it for frontend (Silverlight client) and overall logic of the system
- **Worker Role** – we used them for execution of Action services and tools (something like computation nodes in your system)
- **Word Automation Services** – SharePoint services for batch document conversion
- **SharePoint WCF Service** – collection of SharePoint Services accessible via Windows Communication Foundation (WCF)
- **Virtual Machine Role** – virtual machine within Worker Role

Hardware on VMs

This study was performed on an Azure "medium" virtual machine, with two CPU cores and 3.5 GB of memory. Microsoft doesn't specify the CPU it uses for virtual machines, but the system properties in Windows reported an "AMD Opteron 4171 HE 2.09 GHz 3.50 GB."

Operating System on VMs

The virtual machines are running Windows Server 2008 R2 SP1.

15.5 KB Hadoop Platform

KB Hadoop Platform

Field	Datatype	Value
Platform-ID	String	KB 1
Platform description	String	KB dev cluster This is a pseudo-distributed Cloudera Hadoop CDH4.5.0 instance running on VMWare with 4 vCPU cores, 16GB RAM and ~1.5TB HDD.
Number of nodes	Integer	4 (1 master, 3 worker nodes) Number of virtual hosts involved.
Total number of physical CPUs	Integer	1
CPU specs	String	Intel Xeon(TM) Processor 5150 2.66 Ghz Quad-Core
Total number of CPU-cores	Integer	4 (8 Hyper-threaded)
Total amount of RAM in GB	Integer	16
Average CPU-cores for nodes	Integer	1
Average RAM in GB for nodes	Integer	4
Operating System on nodes	String	Ubuntu 12.04 LTS x86_64 (Precise Pangolin)
Storage system/layer	String	1.52TB HDFS (HDFS on virtual disk (ext4))
Network layer between nodes	String	1 Gbit/s (Virtual network)

Parallel Execution System

Field	Value
Installation description	-
Configuration notes	-

Benchmarks of Parallel Execution System

Metric	Benchmark type	Value	Description
-	-	-	-

15.6 BL Hadoop Platform

BL Developer Platform

Field	Datatype	Value
Platform-ID	String	Platform BL 1
Platform description	String	This is a VMWare ESXi cluster with 32 CPUs, 224GB RAM and ~27TB HDD. It is currently configured to have 30 1CPU nodes; 1 manager, 1 namenode/jobtracker and 28 datanode/tasktrackers, each with 1CPU/6GB RAM/500GB HDD.
Number of nodes	Integer	29 (1 master/28 worker)
Total number of physical CPUs	Integer	2
CPU specs	String	AMD Opteron(TM) Processor 6276
Total number of CPU-cores	Integer	32
Total amount of RAM in GB	Integer	224
Average CPU-cores for nodes	Integer	1
Average RAM in GB for nodes	Integer	6
Operating System on nodes	String	Ubuntu 12.04 LTS
Storage system/layer	String	HDFS on virtual disk.
Network layer between nodes	String	Virtual network

Parallel Execution System

Field	Value
Installation description	-
Configuration notes	-

Benchmarks of Parallel Execution System

Metric	Benchmark type	Value	Description
-	-	-	-

15.7 ONB Hadoop Platform

ONB Hadoop Platform

Field	Datatype	Value
Platform-ID	String	ONB 1
Platform description	String	ONB Local Cluster
Number of nodes	Integer	5
Total number of physical CPUs	Integer	5
CPU specs	String	Quad-Core
Total number of CPU-cores	Integer	20 (40 Hyper-threaded)
Total amount of RAM in GB	Integer	80
Average CPU-cores for nodes	Integer	4
Average RAM in GB for nodes	Integer	16
Operating System on nodes	String	Ubuntu Server 10.04 LTS 64bit
Storage system/layer	String	8.68 TB (HDFS)
Network layer between nodes	String	10 Gbit

Parallel Execution System

Field	Value
Installation description	-
Configuration notes	-

Benchmarks of Parallel Execution System

Metric	Benchmark type	Value	Description
-	-	-	-

15.8 STFC Hadoop Platform

STFC Hadoop Platform

Field	Datatype	Value
Platform-ID	String	STFC hadoop
Platform description	String	STFC development cluster
Number of nodes	Integer	8 (2 x HX525T2i 2U Quad motherboard Compute Node)
Total number of physical CPUs	Integer	16
CPU specs	String	E5-2620v2 (6 Core, 2.1GHz) xeon_e5-2620v2
Total number of CPU-cores	Integer	96
Total amount of RAM in GB	Integer	1024
Average CPU-cores for nodes	Integer	12
Average RAM in GB for nodes	Integer	128
Operating System on nodes	String	Scientific Linux 6
Storage system/layer	String	HDFS (8 x 4TB HDD)
Network layer between nodes	String	1 Gbit/s

Parallel Execution System

Field	Value
Installation description	Cloudera CDH4, Hadoop 2.0.0-cdh4.5.0
Configuration notes	1 name node, 1 job tracker on one of the nodes

Benchmarks of Parallel Execution System

Metric	Benchmark type	Value	Description
-	-	-	-

15.9 UVT Hadoop Platform

UVT Hadoop Platform

Field	Datatype	Value
Platform-ID	String	UVT Hadoop platform
Platform description	String	This Hadoop platform is setup for preservation experiments
Number of nodes	Integer	8 HP ProLiant DL-385 servers
Total number of physical CPUs	Integer	16
CPU specs	String	CPU AMD Opteron 2.4 GHz, dual core, 1 MB L2 cache per core
Total number of CPU-cores	Integer	32 cores
Total amount of RAM in GB	Integer	4 GB RAM / node x 8 = 32GB (total)
Average CPU-cores for nodes	Integer	4
Average RAM in GB for nodes	Integer	4
Operating System on nodes	String	CentOS 6
Storage system/layer	String	• NFS Staging: max 512GB • HDFS Storage: ~400GB
Network layer between nodes	String	Each node is equipped with 2 NICs 1 Gb/s • 1 NIC for NAS access • 1 NIC for Hadoop Operation

Parallel Execution System

Field	Value
Installation description	InfraGRID Cluster - an IBM LoadLeveler based system
Configuration notes	Configuration[136]

Benchmarks of Parallel Execution System

Metric	Benchmark type	Value	Description
-	-	-	-

[136] http://hpc.uvt.ro/infrastructure/infragrid/

15.10 PSNC Hadoop Platform

PSNC Hadoop Platform

Field	Datatype	Value
Platform-ID	String	PSNC Hadoop Platform
Platform description	String	6 physical servers Fujitsu® RX300 S4
Number of nodes	Integer	6
Total number of physical CPUs	Integer	12
CPU specs	String	Intel® Xeon® Processor 2.83 GHz
Total number of CPU-cores	Integer	48
Total amount of RAM in GB	Integer	48
Average CPU-cores for nodes	Integer	8
Average RAM in GB for nodes	Integer	8
Operating System on nodes	String	Ubuntu 12.04 LTS
Storage system/layer	String	Fujitsu® FibreCAT SX40 Storage Subsystem (~30 TB for HDFS)
Network layer between nodes	String	1 GB ethernet

Parallel Execution System

Field	Value
Installation description	-
Configuration notes	-

Benchmarks of Parallel Execution System

Metric	Benchmark type	Value	Description
-	-	-	-

15.11 Platform IMF 1

Platform IMF 1

Field	Datatype	Value
Platform-ID	String	IMF Cluster
Platform description	String	Cloudera CDH3u2. 3 dual-core low consumption nodes
Number of nodes	Integer	3
Total number of physical CPUs	Integer	3
CPU specs	String	Dual core AMD G-T56N on 1600MHz
Total number of CPU-cores	Integer	6 Cores (3 * 2 Cores)
Total amount of RAM in GB	Integer	24GB (3 * 8GB)
Average CPU-cores for nodes	Integer	2
Average RAM in GB for nodes	Integer	8
Operating System on nodes	String	Debian 6 squeeze (64bit)
Storage system/layer	String	HDFS
Network layer between nodes	String	Local copy between two nodes : 80 MB/s 640 Mbps

Platform IMF 2

Field	Data type	Value
Platform-ID	String	IMF Cluster 2
Platform description	**String**	**Cloudera CDH4.6** **43 nodes**
Number of nodes	integer	43
Total number of physical CPUs	**integer**	**43**
CPU specs	string	15 * Dual core AMD G-T56N on 1600MHz, 28 * Intel(R) Core(TM) i5-3470S CPU @ 2.90GHz
Total number of CPU-cores	**integer**	**142 Cores (15 * 2 Cores + 28 * 4 Cores)**
Total amount of RAM in Gbytes	integer	568GB (15 * 8GB + 28 * 16)
average CPU-cores for nodes	**integer**	**3.3**
average RAM in Gbytes for nodes	integer	13.2
Operating System on nodes	**String**	**Debian 6 squeeze (64bit)**
Storage system/layer	String	HDFS
Network layer between nodes	**String**	**Local copy between two nodes : 80 MB/s 640 Mbps**

Parallel Execution System

Field	Value
Installation description	-
Configuration notes	-

Benchmarks of Parallel Execution System

Metric	Benchmark type	Value	Description

-	-	-	-

16 Appendix E - Dataset

16.1 Danish TV broadcasts, mpeg videos

Title	Danish TV broadcasts, mpeg video
Description	sample of a 430 TB collection In the sample • 500 Gbytes mpeg-2 video with Danish TV broadcasts o number of files: 18 o average file size: 27.7 Gb o largest file: 54.2 Gb • 200 Gbytes mpeg-1 video with Danish TV broadcasts o number of files: 48 o average file size: 4.2 Gb o largest file: 10 Gb
Licensing	Sample only available to SCAPE partners that sign licence available at: SCAPE Agreement Sharing testdata with Statsbiblioteket[137]
Owner	The State and University Library, Aarhus (SB)
Dataset location	https://scape.statsbiblioteket.dk/data/mpeg2/ https://scape.statsbiblioteket.dk/data/mpeg1/
Collection expert	Bjarne Andersen, SB
Issues brainstorm	• Validation? Do the files conform to standard? • identification + properties (both sound and video) for preservation; large collection; distributed platform
List of issues	IS22 Characterise and Validate very large mpeg-1 and mpeg-2 files[138] IS45 Audio and Video Recordings have unreliable broadcast time information[139]

[137] https://portal.ait.ac.at/sites/Scape/TB/Lists/Data%20sets/Attachments/5/SCAPE%20Agreement%20Sharing%20testdata%20with%20Statsbiblioteket.pdf

[138] http://wiki.opf-labs.org/display/SP/IS22+Characterise+and+Validate+very+large+mpeg-1+and+mpeg-2+files

[139] http://wiki.opf-labs.org/display/SP/IS45+Audio+and+Video+Recordings+have+unreliable+broadcast+time+information

16.2 Danish TV broadcasts, mpeg-2 transport stream

Title	Danish TV broadcasts, mpeg-2 transport stream
Description	sample of a 280 TB collection In the sample • 200 Gbytes mpeg-2 transport stream with embedded mpeg-4 (MUX2) with Danish TV broadcasts o number of files: 24 o average file size: 8.9 Gb o largest file: 8.9 Gb • 200 Gbytes mpeg-2 transport stream with embedded mpeg-2 (MUX1) with Danish TV broadcasts o number of files: 24 o average file size: 8.9 Gb o largest file: 8.9 Gb
Licensing	Sample only available to SCAPE partners that sign licence available at: SCAPE Agreement Sharing testdata with Statsbiblioteket[140]
Owner	The State and University Library, Aarhus (SB)
Dataset location	https://scape.statsbiblioteket.dk/data/mux2/ https://scape.statsbiblioteket.dk/data/mux1/
Collection expert	Bjarne Andersen, SB
Issues brainstorm	
List of issues	IS3 Large media files are difficult to characterise without mass processing + We cannot identify preservation risks in uncharacterised files[141]

[140] https://portal.ait.ac.at/sites/Scape/TB/Lists/Data%20sets/Attachments/5/SCAPE%20Agreement%20Sharing%20testdata%20with%20Statsbiblioteket.pdf

[141] http://wiki.opf-labs.org/pages/viewpage.action?pageId=5701678

16.3 Danish Radio broadcasts, mp3

Title	Danish Radio broadcasts, mp3
Description	sample of a 20 TB collection In the sample • 200 Gbytes mp3 (128kbit) with Danish Radio broadcasts o number of files: 1688 o average file size: 118Mb o largest file: 124Mb
Licensing	Sample only available to SCAPE partners that sign licence available at: SCAPE Agreement Sharing testdata with Statsbiblioteket[142]
Owner	The State and University Library, Aarhus (SB)
Dataset location	https://scape.statsbiblioteket.dk/data/mp3/
Collection expert	Bjarne Andersen, SB
Issues brainstorm	• This mp3-collection is known to have files with very bad sound (ex: file P1_1400_1600_940103_001.mp3) • mp3 to wav conversion + QA
List of issues	IS21 Migration of mp3 to wav[143] IS20 Detect audio files with very bad sound quality[144]

[142] https://portal.ait.ac.at/sites/Scape/TB/Lists/Data%20sets/Attachments/5/SCAPE%20Agreement%20Sharing%20testdata%20with%20Statsbiblioteket.pdf
[143] http://wiki.opf-labs.org/display/SP/IS21+Migration+of+mp3+to+wav
[144] http://wiki.opf-labs.org/display/SP/IS20+Detect+audio+files+with+very+bad+sound+quality

16.4 KB Metamorfoze Migration (sample batch)

Title	KB Metamorfoze Migration (sample batch)
Description	The collection consists of 4.7 million pages of digitised documents from the Metamorfoze[145] Programme. Content includes: • TIFF page masters (colour, uncompressed), • XML metadata (DMD), • plain text (log files). The complete collection in TIFF form is ~146 TB. Samples of migrated JPEG2000 files are also available, as well as log files with statistics from the ongoing TIFF -> JP2 conversion. The sample batch consists of 8047 pages of TIFF images with metadata and log files and is ~169 GB large.
Licensing	The sample is not generally available, thus restricting use to research within SCAPE only. Upon request, however it can be made available to SCAPE partners for the duration of the project.
Owner	Koninklijke Bibliotheek
Dataset location	Koninklijke Bibliotheek
Collection expert	Reinier Deinum, KB
Issues brainstorm	-
List of issues	-

[145] http://www.metamorfoze.nl/english

16.5 BL 19th Century Digitized Newspapers

Title	19th Century Digitised Newspapers
Description	The collection consists of 2.2million pages of digitised 19th Century Newspapers. Content includes: • TIFF page masters, • TIFF page service copies, • TIFF article service copies, • XML METS metadata, • XML. Samples of migrated JPEG2000 files are also available, including some truncated JPEG2000s that were damaged during a faulty migration process. The complete collection in TIFF form is ~80TB.
Licensing	The collection sample is available for use under a BL licence, restricting usage for research only. Otherwise it is not restricted to SCAPE Project partners. See full licence[146]
Owner	British Library
Dataset location	TBC
Collection expert	TBC
Issues brainstorm	-
List of issues	-

[146] http://wiki.opf-labs.org/download/attachments/8356134/license.pdf

16.6 Austrian National Library Tresor Music Collection

Title	Tresor project music collection
Description	The collection consists of 2.2million pages of digitised 19th Century Newspapers. Content includes: • TIFF page masters, • JPEG2000 derivatives partly created. The collection consists of 15472 pages summing up to a total of about 1TB.
Licensing	Restricted
Owner	Austrian National Library
Dataset location	Austrian National Library
Collection expert	-
Issues brainstorm	-
List of issues	-

16.7 Govdocs1 Corpus

Title	Govdocs1 Open Corpus
Description	A corpus of 1 million documents that is freely available for research, drawn from US government web sites, of various formats. This dataset contains 231,683 PDFs which total 127.8GB
Licensing	None. Free to used and distribute.
Owner	N/A
Dataset location	http://digitalcorpora.org/corpora/files
Collection expert	N/A
Issues brainstorm	-
List of issues	-

16.8 Danish newspaper - Morgenavisen Jyllandsposten

Title	Danish newspaper collection
Description	The collection consists of 17978 pages of digitised newspapers. Content includes: • 167 GB jp2 files (179,304,678,919 bytes) • smallest size: 1.4 MB • largest size: 17 MB
Licensing	Sample only available to SCAPE partners that sign licence available at: SCAPE Agreement Sharing testdata with Statsbiblioteket[147]
Owner	The State and University Library, Aarhus (SB)
Dataset location	The State and University Library, Aarhus (SB)
Collection expert	Bjarne Andersen, SB
Issues brainstorm	-
List of issues	-

[147] https://portal.ait.ac.at/sites/Scape/TB/Lists/Data%20sets/Attachments/5/SCAPE%20Agreement%20Sharing%20testdata%20with%20Statsbiblioteket.pdf

16.9 KB Web Archive Dataset (sample batch)

Title	KB - Web Archive Data
Description	5TB of web archive content from the .nl TLD in ARC format (sample batch: ~37GB, 406 ARC files)
Licensing	This is a closed archive only accessible by Dutch researchers
Owner	Koninklijke Bibliotheek
Dataset location	SurfSara
Collection expert	Rene Voorburg, KB
Issues brainstorm	-
List of issues	-

16.10 ONB Web Archive Dataset

Title	Austrian National Library - Web Archive
Description	The Austrian National Library uses a representative datasets from their web archive • events selective crawls: during an event frequently harvested sites, e.g. EU election 2009, Olympia 2010, • domain crawls 2009 from about 1 million domains. The web archive data is available in the ARC.GZ format. The size of the ARC.GZ data set is 1377GB. The metadata log file produced during the crawl process is available as txt file and has a size of 197GB. Sub-Sample: waa-full-arcs-1-sample1000
Licensing	Sample only available to SCAPE partners.
Owner	Austrian National Library (ONB)
Dataset location	-
Collection expert	Sven Schlarb, ONB
Issues brainstorm	-
List of issues	-

16.11 Internet Memory Web Archive

Title	Internet Memory Web collections
Description	The data consists in web content crawled, stored and hosted by the Internet Memory Foundation (W)ARC format (approx. 300TB) Using this content, IM can also use its taskforce (QA team) to provide annotated data such as pairs of annotated snapshots for quality assurance scenarios. 1000 annotated pairs of web pages (similar/dissimilar) were produced as part of PC.WP3: Quality Assurance Components.
Licensing	Web collections crawled on behalf of partner institutions will require institutions agreement to be used by SCAPE partners
Owner	Internet Memory
Dataset location	Provided upon request
Collection expert	Leïla Medjkoune, IM
Issues brainstorm	-
List of issues	-

16.12 SB Web Archive Data

Title	State and University Library Denmark - Web Archive Data
Description	220 TB of web archive content in ARC format
Licensing	This is a closed archive only accessible by Danish researchers
Owner	The State and University Library, Aarhus (SB)
Dataset location	Currently not available
Collection expert	Bjarne Andersen, SB
Issues brainstorm	-
List of issues	-

16.13 BL Web Archive SCAPE Testbed Dataset

Title	BL Web Archive SCAPE Testbed Dataset
Description	~1TB of ARC and WARC files from UK Web Archive (~14k files)
Licensing	This content cannot be shared
Owner	British Library
Dataset location	Not available
Collection expert	William Palmer, BL
Issues brainstorm	-
List of issues	-

16.14 Malaga Urban Dataset

Title	Malaga Urban Dataset
Description	227 thousand JPEG images (60 GB) captured from stereo camera mounted on a car during a 36.8 long drive through a town environment in the town of Málaga.
Licensing	This content is publicly available
Owner	University of Málaga
Dataset location	http://www.mrpt.org/MalagaUrbanDataset
Collection expert	Ondrej Klima, BUT
Issues brainstorm	-
List of issues	-

16.15 BUT Alp Mountains Dataset

Title	BUT Alp Mountains Dataset
Description	Data set comprising two parts: • The first part contains geo-localized real images of mountain scenery of Alps, which were taken from top of prominent summits. • The second part contains synthetically rendered panorama images taken from automatically detected significant summits. This part contains tens of thousands images.
Licensing	This data set is not shared
Owner	BUT
Dataset location	Not available
Collection expert	Ondrej Klima, BUT
Issues brainstorm	
List of issues	

16.16 WCPT medical dataset

Title	WCPT medical dataset
Description	This dataset consists of approx. 20-30TB of data including: • RTG images (DICOM files) • CT images (DICOM files) • bronchoscopy examination recordings (DICOM files) • HIS records (text files)
Licensing	This content cannot be shared. After anonymisation it is available to SCAPE partners only.
Owner	WCPT
Dataset location	Currently not available.
Collection expert	Paweł Kominek, WCPT
Issues brainstorm	
List of issues	

17 Appendix F1 – Evaluations in Web Content Testbed

17.1 EVAL ARC2WARC with Hawarp

Experiment: ARC2WARC Experiment at KB

Functional Evaluation

A functional evaluation was performed at the SCAPE developers workshop on 23-25 April at the KB in The Hague.

Version 45cc6bc[148] of the Hawarp[149] tool was used to migrate a batch of 406 ARC files from the KB's web archive to the WARC format, using the KB SCAPE Platform. The following steps were executed:

1. Build the main project.
Some adaptations were done to the pom.xml[150] as to satisfy the fact that a different version of the CDH is used at KB (CDH4) than at ONB (CDH3). Other than indicating the correct version of the Hadoop libraries in the dependencies, no further changes had to be made.

2. Build the module arc2warc-migration-cli[151], which performs ARC to WARC migration in local mode. In this step, a minor test failure was detected that was quickly fixed[152] by the developer of the tool. Once the module was built successfully, it was executed against the 406 ARC files of KB. This revealed there are invalid (incorrect payload size) ARC files in the dataset, which causes the tool to exit in local mode. The invalid ARC files will need to be investigated further.

3. Build the module arc2warc-migration-hdp[153], which performs ARC to WARC migration using Hadoop with files in HDFS.
With this module, it was possible to migrate the full batch of 406 ARC files to WARC, in a single run. Some issues in the documentation regarding the expected input format for the tool were clarified with the tool developer, and the documentation updated accordingly.

4. Build the module droid-identify[154], to characterise the sample set of ARC files.

5. Build the module tomar-prepare-inputdata[155], to prepare the ARC files stored on HDFS for processing with the SCAPE ToMaR[156] tool.

6. In a last step, results from FITS identification shall be ingested into the c3po[157] tool for visualization and presentation purposes (not done yet).

---

[148] https://github.com/openplanets/hawarp/commit/45cc6bcc0abda596768dbe3c00144acd1bc842bd
[149] https://github.com/openplanets/hawarp/
[150] https://github.com/cneud/hawarp/blob/master/pom.xml
[151] https://github.com/openplanets/hawarp/blob/master/arc2warc-migration-cli/README.md
[152] https://github.com/openplanets/hawarp/issues/4
[153] https://github.com/openplanets/hawarp/blob/master/arc2warc-migration-hdp
[154] https://github.com/openplanets/hawarp/blob/master/droid-identify/
[155] https://github.com/openplanets/hawarp/tree/master/tomar-prepare-inputdata
[156] https://github.com/openplanets/tomar
[157] https://github.com/openplanets/c3po

17.2 EVAL ARC2WARC-HDP w.o. Tika

Experiment: ARC2WARC Experiment at ONB

Evaluator(s)

Sven Schlarb, ONB

Evaluation points

Assessment of measurable points

Metric	Description	Metric baseline	Metric goal	March 04, 2014 (1000)	March 04, 2014 (4924)
NumberOfObjectsPerHour	Number of objects processed in one hour	833,8098788		4591,836735	4645,283019
MinObjectSizeHandledInGbytes	Smallest ARC file in sample	0,001638618		0,001638618	0,0001516
MaxObjectSizeHandledInGbytes	Biggest ARC file in sample	0,295765739		0,295765739	0,295765739
ThroughputGbytesPerMinute	The throughput of data measured in Gybtes per minute	1,272703878		7,008850061	7,004187217
ThroughputGbytesPerHour	The throughput of data measured in Gbytes per hour	76,36223269		420,5310036	420,251233 (*)
ReliableAndStableAssessment	Manual assessment on if the experiment performed reliable and stable	true		true	true
NumberOfFailedFiles	Number of files that failed in the workflow	0		0	0
AverageRuntimePerItemInSeconds	The average processing time in seconds per item	4,32		0,78	0,774979691

Technical details

The different evaluation points in the table above refer to data sets of different size and parameter variations, the Hadoop Job-ID links to details about the job execution:

March 04, 2014 (1000): waa-full-arcs-1 (subset 1000)[158], job_201401221447_0079[159]
March 04, 2014 (4924): waa-full-arcs-1 (4924 arc files)[160], job_201401221447_0078[161]
These data samples are subsets of the ONB web archive crawl ONB Web Archive Dataset.

[158] http://wiki.opf-labs.org/display/SP/waa-full-arcs-1+%28subset+1000%29
[159] http://fue-l/scape-tb-evaluation/onb/arc2warc/arc2warc-hdp-at-onb/eval-arc2warc-hdp-wo-tika/march-04-2014-1000/Hadoop%20job_201401221447_0079%20on%20fue-hdc01.html
[160] http://wiki.opf-labs.org/display/SP/waa-full-arcs-1+%284924+arc+files%29
[161] http://fue-l/scape-tb-evaluation/onb/arc2warc/arc2warc-hdp-at-onb/eval-arc2warc-hdp-wo-tika/march-04-2014-4924/Hadoop%20job_201401221447_0078%20on%20fue-hdc01.html

17.3 EVAL ARC2WARC-HDP with Tika

Experiment: ARC2WARC Experiment at ONB

Evaluator(s)

Sven Schlarb, ONB

Evaluation points

Assessment of measurable points

Metric	Description	Metric baseline	Metric goal	March 04, 2014 (1000)	March 04, 2014 (4924)
NumberOfObjectsPerHour	Number of objects processed in one hour	833,8098788		2760,736196	2813,267735
MinObjectSizeHandledInGbytes	Smallest ARC file in sample	0,001638618		0,001638618	0,0001516
MaxObjectSizeHandledInGbytes	Biggest ARC file in sample	0,295765739		0,295765739	0,295765739
ThroughputGbytesPerMinute	The throughput of data measured in Gybtes per minute	1,272703878		4,213909852	4,241862946
ThroughputGbytesPerHour	The throughput of data measured in Gbytes per hour	76,36223269		252,8345911	254,5117767
ReliableAndStableAssessment	Manual assessment on if the experiment performed reliable and stable	true		true	true
NumberOfFailedFiles	Number of files that failed in the workflow	0		0	0
AverageRuntimePerItemInSeconds	The average processing time in seconds per item	4,32		1,30	1,27965069

Technical details

The different evaluation points in the table above refer to data sets of different size and parameter variations, the Hadoop Job-ID links to details about the job execution:

March 04, 2014 (1000): waa-full-arcs-1 (subset 1000)[162], job_201401221447_0079[163]
March 04, 2014 (4924): waa-full-arcs-1 (4924 arc files)[164], job_201401221447_0078[165]
These data samples are subsets of the ONB web archive crawl ONB Web Archive Dataset.

[162] http://wiki.opf-labs.org/display/SP/waa-full-arcs-1+%28subset+1000%29
[163] http://fue-l/scape-tb-evaluation/onb/arc2warc/arc2warc-hdp-at-onb/eval-arc2warc-hdp-wo-tika/march-04-2014-1000/Hadoop%20job_201401221447_0079%20on%20fue-hdc01.html
[164] http://wiki.opf-labs.org/display/SP/waa-full-arcs-1+%284924+arc+files%29
[165] http://fue-l/scape-tb-evaluation/onb/arc2warc/arc2warc-hdp-at-onb/eval-arc2warc-hdp-wo-tika/march-04-2014-4924/Hadoop%20job_201401221447_0078%20on%20fue-hdc01.html

17.4 EVAL ARC2WARC-TOMAR w.o. Tika

Experiment: ARC2WARC Experiment at ONB

Evaluator(s)

Sven Schlarb, ONB

Evaluation points

Assessment of measurable points

Metric	Description	Metric baseline	Metric goal	March 04, 2014 (1000)	March 04, 2014 (4924)
NumberOfObjectsPerHour	Number of objects processed in one hour	833,8098788		4250,295159	4320,350963
MinObjectSizeHandledInGbytes	Smallest ARC file in sample	0,001638618		0,001638618	0,0001516
MaxObjectSizeHandledInGbytes	Biggest ARC file in sample	0,295765739		0,295765739	0,295765739
ThroughputGbytesPerMinute	The throughput of data measured in Gybtes per minute	1,272703878		6,487530635	6,514252601
ThroughputGbytesPerHour	The throughput of data measured in Gbytes per hour	76,36223269		389,2518381	390,8551561
ReliableAndStableAssessment	Manual assessment on if the experiment performed reliable and stable	true		true	true
NumberOfFailedFiles	Number of files that failed in the workflow	0		0	0
AverageRuntimePerItemInSeconds	The average processing time in seconds per item	4,32		0,847	0,833265638

Technical details

The different evaluation points in the table above refer to data sets of different size and parameter variations, the Hadoop Job-ID links to details about the job execution:

March 04, 2014 (1000): waa-full-arcs-1 (subset 1000)[166], job_201401221447_0056[167]
March 04, 2014 (4924): waa-full-arcs-1 (4924 arc files)[168], job_201401221447_0059[169]
These data samples are subsets of the ONB web archive crawl ONB Web Archive Dataset.

[166] http://wiki.opf-labs.org/display/SP/waa-full-arcs-1+%28subset+1000%29
[167] http://fue.onb.ac.at/scape-tb-evaluation/onb/arc2warc/arc2warc-tomar-at-onb/eval-arc2warc-tomar-wo-tika/march-04-2014-1000/
[168] http://wiki.opf-labs.org/display/SP/waa-full-arcs-1+%284924+arc+files%29
[169] http://fue.onb.ac.at/scape-tb-evaluation/onb/arc2warc/arc2warc-tomar-at-onb/eval-arc2warc-tomar-wo-tika/march-04-2014-4924/

17.5 EVAL ARC2WARC-TOMAR with Tika

Experiment: ARC2WARC Experiment at ONB

Evaluator(s)

Sven Schlarb, ONB

Evaluation points

Assessment of measurable points

Metric	Description	Metric baseline	Metric goal	March 04, 2014 (1000)	March 04, 2014 (4924)	March 04, 2014 (9856)
NumberOfObjectsPerHour	Number of objects processed in one hour	545,17246		3317,97235	2813,267735	7006,635071
MinObjectSizeHandledInGbytes	Smallest ARC file in sample	0,001638618		0,001638618	0,0001516	0,000151632
MaxObjectSizeHandledInGbytes	Biggest ARC file in sample	0,295765739		0,295765739	0,295765739	0,295765739
ThroughputGbytesPerMinute	The throughput of data measured in Gybtes per minute	0,832135864		5,064459399	4,241862946	10,54745844
ThroughputGbytesPerHour	The throughput of data measured in Gbytes per hour	49,92815185		303,8675639	254,5117767	632,8475062
ReliableAndStableAssessment	Manual assessment on if the experiment performed reliable and stable	true		true	true	true
NumberOfFailedFiles	Number of files that failed in the workflow	0		0	0	0
AverageRuntimePerItemInSeconds	The average processing time in seconds per item	6,60		1,09	1,27965069	0,513798701

Technical details

The different evaluation points in the table above refer to data sets of different size, the Hadoop Job-ID links to details about the job execution:

March 04, 2014 (1000): waa-full-arcs-1 (subset 1000)[170], job_201401221447_0057[171]
March 04, 2014 (4924): waa-full-arcs-1 (4924 arc files)[172]
These data samples are subsets of the ONB web archive crawl ONB Web Archive Dataset.

[170] http://wiki.opf-labs.org/display/SP/waa-full-arcs-1+%28subset+1000%29
[171] http://fue.onb.ac.at/scape-tb-evaluation/onb/arc2warc/arc2warc-tomar-at-onb/eval-arc2warc-tomar-with-tika/march-04-2014-1000/
[172] http://wiki.opf-labs.org/display/SP/waa-full-arcs-1+%284924+arc+files%29

17.6 EVAL-WCT2-EX1 Comparing newly archived Web sites against a verified copy (single node)

Experiment: WCT2-EX1 Comparing newly archived Web sites against a verified copy (single node)

Evaluation specs platform/system level

Field	Data type	Value
Evaluation seq. num.	int	1
Evaluator-ID	email	radu.pop@internetmemory.net
Evaluation description	text	The IMF takes into account the quality of archived web sites. The quality is assured by a visual inspection: comparing the site in Internet with the archived site in IMF servers. In order to improve that process, IMF is trying to develop an application, using the Markalizer developed UPMC, which compares two images. These two images are produced by Selenium based framework (V.2.24.1) by taking two snapshots: ideally, one is taken from the archive access and the second from the live. This evaluation uses screenshots taken from the IMF Web Archive at two different dates in time. Note also that for this specific test, only one node of the platform was used. Workflow: 1° Loading a pair of Web Archive pages (2 urls given) 2° Take screenshots (Selenium) 3° Visual comparison of screenshots (Markalizer) 4° Produce the output result file (score of comparison) **Goal / Sub-goal:** **Performance efficiency / Throughput** • Loading webpages can take time and depends on different factors such as the complexity of the page, the Internet connection, the browser and browser version used and/or the status of remote servers. • Taking the screenshot using Selenium Compare with Markalizer Overhead (preparation of next comparison) **Reliability / Stability Indicators** The external tools needed are : • Selenium Firefox (for this evaluation) • Xvfb (A graphical server, needed to run Firefox in virtual screen) • Markalizer The application is developed in Python All needed components are installed separately (dependencies of packages) **Reliability / Runtime stability** • The result has been measured as a float number that can measure and detect the differences between two images
Evaluation-Date		01/11/2012
Platform-ID	string	Platform IMF 1
Dataset(s)	string	Pairs of urls from IMF web archive
Workflow method	string	Python application wrapping and managing Selenium and the Markalizer tool
Workflow(s) involved	URL(s)	
Tool(s) involved	URL(s)	
Link(s) to Scenario(s)	URL(s)	WCT1

Platform IMF 1

Field	Data type	Value
Platform-ID	String	IMF Cluster
Platform description	String	Cloudera CDH3u2. 3 dual-core low consumption nodes
Number of nodes	integer	3
Total number of physical CPUs	integer	3
CPU specs	string	Dual core AMD G-T56N on 1600MHz
Total number of CPU-cores	integer	6 Cores (3 * 2 Cores)
Total amount of RAM in Gbytes	integer	24GB (3 * 8GB)
average CPU-cores for nodes	integer	2
average RAM in Gbytes for nodes	integer	8
Operating System on nodes	String	Debian 6 squeeze (64bit)
Storage system/layer	String	HDFS
Network layer between nodes	String	Local copy between two nodes : 80 MB/s 640 Mbps

Evaluation points

Metric	Baseline definition	Baseline value	Goal	Evaluation 1 (01/11/2012)
NumberOfObjectsPerHour	Number of comparisons made per hour	0	100	38
NumberOfFailedFiles	Number of images screenshots that failed in the workflow	0	0	0

17.7 EVAL-WCT2-EX2 Comparing newly archived Web sites against a verified copy (multiple nodes)

Experiment: WCT2-EX2 Comparing newly archived Web sites against a verified copy (multiple nodes)

Evaluation specs platform/system level

Field	Data type	Value
Evaluation seq. num.	int	1
Evaluator-ID	email	stanislav.barton@internetmemory.net
Evaluation description	text	Workflow: 1° Take list of Web Archive pages (1 URL given) and a list of web browsers (one reference and at least one on to compare with list) 2° Take screenshots (Selenium) for each web browser of each page 3° Visual comparison of screenshots (Marcalizer) between reference and to compare list 4° Produce the output result file (score of comparison - one pair per comparison) **Goal / Sub-goal:** • Integration: parallelized solution works with similar performance figures as the single node approach • Communication of results via C3PO to SCOUT
Evaluation-Date	DD/MM/YY	01/05/2013
Platform-ID	string	Platform IMF 1
Dataset(s)	string	Pairs of URLs from IMF web archive
Workflow method	string	Python application wrapping and managing Selenium and the Marcalizer tool
Workflow(s) involved	URL(s)	
Tool(s) involved	URL(s)	
Link(s) to Scenario(s)	URL(s)	WCT1

Platform IMF 1

Field	Data type	Value
Platform-ID	String	IMF Cluster
Platform description	String	Cloudera CDH3u2. 3 dual-core low consumption nodes
Number of nodes	integer	3
Total number of physical CPUs	integer	3
CPU specs	string	Dual core AMD G-T56N on 1600MHz
Total number of CPU-cores	integer	6 Cores (3 * 2 Cores)
Total amount of RAM in Gbytes	integer	24GB (3 * 8GB)
average CPU-cores for nodes	integer	2
average RAM in Gbytes for nodes	integer	8
Operating System on nodes	String	Debian 6 squeeze (64bit)
Storage system/layer	String	HDFS
Network layer between nodes	String	Local copy between two nodes : 80 MB/s 640 Mbps

Evaluation points

Metric	Baseline definition	Baseline value	Goal	Evaluation 1 (01/05/2013)
TimeToTakeSnapshot	Time to take snapshot (seconds)	0	0	2
TimeToCompare	Time to compare snapshots using marcalizer (seconds)	0	0	2

17.8 EVAL-WCT2-EX3 – Large Input Large Infrastructure

Experiment: WCT2-EX3 Visual automated QA at large scale

Evaluation specs platform/system level

Field	Data type	Value
Evaluation seq. num.	int	1
Evaluator-ID	email	stanislav.barton@internetmemory.net
Evaluation description	text	The IMF takes into account the quality of archived web sites. The quality is assured by a visual inspection: comparing the site in Internet with the archived site in IMF servers. In order to improve that process, IMF is trying to develop an application, using the Pagelyzer developed UPMC, which compares two images. These two images are produced by Selenium based framework (V.2.24.1) by taking two snapshots: ideally, one is taken from the archive access and the second from the live. Workflow: 1° Load live page, take screen shot (Selenium + Firefox headless) 2° Load web page from archive, take screen shot(Selenium + Firefox headless) 3° Visual comparison of screenshots (Pagelyzer) 4° Produce the output result file (score of comparison) **Goal / Sub-goal:** **Performance efficiency / Throughput** • Loading webpages can take time and depends on different factors such as the complexity of the page, the Internet connection, the browser and browser version used and/or the status of remote servers. • Taking the screenshot using Selenium Compare with Pagelyzer overhead (preparation of next comparison) **Reliability / Stability Indicators** The external tools needed are : • Selenium Firefox (for this evaluation) • Xvfb (A graphical server, needed to run Firefox in virtual screen) • Pagelyzer The application is developed in Java/Ruby All needed components are installed separately (dependencies of packages) **Reliability / Runtime stability** • The result has been measured as a float number that can measure and detect the differences between two images
Evaluation-Date	DD/MM/YY	01/09/2014
Platform-ID	string	Platform IMF 2
Dataset(s)	string	2.6 Millions urls from IMF web archive
Workflow method	string	MapReduce job using selenium and Pagelyzer internally
Workflow(s) involved	URL(s)	
Tool(s) involved	URL(s)	
Link(s) to Scenario(s)	URL(s)	WCT1

Platform IMF 2

Field	Data type	Value
Platform-ID	String	IMF Cluster 2
Platform description	String	Cloudera CDH4.6 43 nodes
Number of nodes	integer	43
Total number of physical CPUs	integer	43
CPU specs	string	15 * Dual core AMD G-T56N on 1600MHz, 28 * Intel(R) Core(TM) i5-3470S CPU @ 2.90GHz
Total number of CPU-cores	integer	142 Cores (15 * 2 Cores + 28 * 4 Cores)
Total amount of RAM in Gbytes	integer	568GB (15 * 8GB + 28 * 16)
average CPU-cores for nodes	integer	3.3
average RAM in Gbytes for nodes	integer	13.2
Operating System on nodes	String	Debian 6 squeeze (64bit)
Storage system/layer	String	HDFS
Network layer between nodes	String	Local copy between two nodes : 80 MB/s 640 Mbps

Evaluation points

Metric	Baseline definition	Baseline value	Goal	Evaluation 1 (01/09/2014)
NumberOfObjectsPerSecond	Number of comparisons made per hour	0	3	4
ScoresAchieved	Frequency of similarity scores assessed by Pagelyzer	0	0	0
TotalNumberOfURLsProcessed	Total number of URLs used for comparison	0	2,600,000	2,600,000
AverageGetTimeFromArchive	Average time spent getting page from web archive in seconds	0	2	1.7
AverageGetTimeFromLive	Average time spent getting page from live web	0	2	2
AveragePagelyzerTime	Average time spent comparing snapshots	0	2	1.7

17.9 EVAL-WCT3-1

Experiment: WCT EX2 File ID at SB

Evaluation specs platform/system level

Field	Data type	Value
Evaluation seq. num.	int	1
Evaluator-ID	email	pmd@statsbiblioteket.dk
Evaluation description	text	Since November 2011 we have been running FITS on a selection of our web content spread over the years from 2005 up till 2011. The data is stored in ARC files on a SAN. These ARC files are fetched from this SAN, unpacked and the FITS are run on each ARC record. Running FITS on an ARC record produces an XML file. These XML files from a single ARC are packed into TGZ files and made available to the Planning and Watch subproject. To evaluate this job we extract information on the timing of the FITS jobs together with information from the ARC files.
Evaluation-Date	DD/MM/YY	25th of November 2011 till 8th of November 2012
Platform-ID	string	Platform SB 1
Dataset(s)	string	http://wiki.opf-labs.org/display/SP/State+and+University+Library+Denmark+-+Web+Archive+Data
Workflow method	string	Command line
Workflow(s) involved	URL(s)	None
Tool(s) involved	URL(s)	fits 0.6.0, arc-unpacker 0.2
Link(s) to Scenario(s)	URL(s)	WCT3

Platform SB 1

Field	Data type	Value
Platform-ID	String	Platform SB 1
Platform description	String	We have five Blade servers located at SB
Number of nodes	integer	5 physical servers
Total number of physical CPUs	integer	10
CPU specs	string	Intel® Xeon® Processor X5670 (12M Cache, 2.93 GHz, 6.40 GT/s Intel® QPI)
Total number of CPU-cores	integer	60
Total amount of RAM in Gbytes	integer	288 GB
average CPU-cores for nodes	integer	6
average RAM in Gbytes for nodes	integer	4 with 48 GB and one with 96 GB
Operating System on nodes	String	Red Hat based Linux
Storage system/layer	String	Only SAN storage
Network layer between nodes	String	1 GB Ethernet

Evaluation points

The motivation behind the goal is as follows: we want to be able to run a FITS-like characterisation on a complete snap-shot of the Danish TLD within weeks. Such a snap-shot harvest amounts to 25 TB. This gives a throughput in the order of 1GB/minute. "FITS-like" is here defined as a characterisation using multiple tools combined with a comparison of the output of these tools.

Even though the base line is calculated based on one thread on one CPU, we did the actual assessment on a five machine cluster where each process was allowed to use up to 4 threads. This experiment is our first evaluation.

Metric	Baseline definition	Baseline value	Goal	Evaluation 1 (8/11 2012)
ThroughputGbytesPerHour	Measurement of the running time of the FITS jobs assuming one thread on one machine. During the last year the job has actual run on one to five servers using one to four threads but that job distribution is not represented in the metadata.	0.162	60	1.32
OrganisationalFit		N/A	true	true

17.10 EVAL-BL-WCT-01

Experiment: WCT EX3 File ID at BL

Evaluator(s)

William Palmer, BL

Evaluation points

Assessment of measurable points

Metric	Metric Goal	Early Jan 2014	Late Jan 2014 (incl CloseShield patch)	July 2014
TotalRuntime		31:33:00	17:32:00	54:11:15 [0]
TotalSize		1024GB	1024GB	1024GB
NumberOfObjectsPerHour (object = file in warc)		2956292.8[1]	5319641.0 [1]	1923994 [2]
NumberOfObjectsPerHour (object = warc file)		457.8	823.9	266.57
ThroughputGbytesPerHour		32.4[3]	58.4[3]	18.89 [3]
ReliableAndStableAssessment	TRUE	TRUE	TRUE	TRUE
NumberOfFailedFiles	0	0	0	0
NumberOfFailedFilesAcceptable	-	TRUE	TRUE	TRUE
Options enabled:				INCLUDE_SERVERTYPE USE_DROID USE_TIKADETECT USE_TIKAPARSER GENERATE_C3PO_ZIP
		This test run was identification (i.e.mimetype detection only)	This test run was identification (i.e.mimetype detection only)	This test run was identification and characterisation (i.e.metadata extraction)

The test in late January was following work to increase the speed of Nanite. The test in July extended the previous work, and added characterisation and a series of other options and outputs.

- [0] This run was longer than the previous one, because it also produced characterization metadata for each file, unlike earlier tests which only produced a mime type.
- [1] In total, 93,271,038 map input records were processed
- [2] In total, 104,256,430 map input records were processed (for the same ARC input files) This is most likely due to a version upgrade of the record readers used by Nanite, but requires further investigation. See the results in the table below
- [3] Note that this is throughput in relation to the compressed sizes of ARC files

For the July 2014 characterisation run, the following exceptions/errors were recorded in the logs:

Type	Count
Potentially malformed records	3
Java (non-IO)Exceptions thrown	4779
Java IOExceptions thrown	1865

Technical details

Jan evaluations: patches upstream now:
https://github.com/openplanets/nanite/releases/tag/nanite-1.0.72.2 (need to re-enable nanite-hadoop) - this code is for the Late Jan test (includes CloseShield patch)

July evaluations: code is here:
https://github.com/openplanets/nanite/tree/7ed4d5536c42ff77367a10fb9671d5fab2a6935d

Evaluation notes

Some warc files were truncated/zero length, probably due to issues when being copied - a small Hadoop program was written to identify these files so they could be excluded from the full runs. Runtime of the check is very quick. This can be chained before the FormatProfiler MapReduce program in Nanite, but it is turned off and not included in these evaluation runs as we have already identified the problematic files and runtime is very short (see: https://github.com/willp-bl/nanite/tree/master/nanite-hadoop/src/main/java/uk/bl/wap/hadoop/gzchecker)

For the characterization run in July 2014, more files were processed from within the ARC files than in previous evaluation runs. This is most likely due to an upgrade of dependencies, for example the warc-hadoop-recordreaders, amongst others. However, a number of exceptions/errors remained and this issue should be looked in to, so we can ensure that all files within the ARC files are processed in future.

Conclusion

The decision to use and develop Nanite further for this experiment has proved to have been a sound one. Nanite benefits greatly due to being tightly coupled with Hadoop, and making use of pure-Java libraries so no external applications are called. After initially reducing the runtime by almost 50%, further work was undertaken to add in full characterisation of the input files, which proved to be very performant and compared favourably to other methods of characterisation at scale. Nanite is a good base for future work on gleaning more information from web archives and can be easily extended further. An example of this is the c3po compatible outputs for exploring the characterisation information of ones archives. Additional options for storing files that Tika cannot process are already included and will potentially be useful for improving Tika.

One of our web archive collections totals 30TB of compressed (W)ARC files, and using Nanite to characterise that data on the same test cluster would be expected to take 68 days, which is acceptable.

17.11 EVAL-SB-WCT-04

Experiment: WCT EX4 File ID and characterisation at SB

Evaluator(s)

Per Møldrup-Dalum, SB

Evaluation points

Assessment of measurable points

Metric	Description	Metric baseline	Metric goal	8/11/2012	28/5/2014
ThroughputGbytesPerHour	Goal, objective, baseline notes	0.162	60	1.32	235.2
OrganisationalFit			true	true	true

Motivation behind the metric goal

As in the previous evaluation from 8/11/2012, we still want to be able to run a characterisation on a complete snap-shot of the Danish TLD within weeks. Such a snap-shot harvest amounts to 25 TB. As can be seen from the above metrics, this goal has been achieved through the SCAPE project using SCAPE tools and methodology.

Technical details

Unfortunately this experiment and evaluation is done on test data that we are unable to make public. On top of that, the Hadoop module of Nanite that was used had local changes done on a development fork of the main Nanite project. All relevant changes will be pushed to the main Nanite project and any local improvements will be available in a separate fork.

The same restrictions as on the original data are imposed on the results of the experiment. This is due to URLs being present in the results and such data is subject to protection by our national privacy act.

Evaluation notes

The experiment and the development leading up to it is described in detail in the A Weekend With Nanite[173] blog post on the Open Planet Foundation blog.

[173] http://www.openplanetsfoundation.org/blogs/2014-05-28-weekend-nanite

17.12 EVAL Taverna-Fits-ToMaR-C3PO

Experiment: Web Archive FITS Characterisation using ToMaR at ONB

Evaluator(s)

Sven Schlarb, ONB

Evaluation points

Assessment of measurable points

Metric	Description	Metric baseline	Metric goal	May 13, 2014 (100)
NumberOfObjectsPerHour	Number of objects processed in one hour	0,5605381166		9,7328863415
MinObjectSizeHandledInGbytes	Smallest ARC file in sample	0,0001516324		0,0001516324
MaxObjectSizeHandledInGbytes	Biggest ARC file in sample	0,0010629473		1,2779601114
ThroughputGbytesPerMinute	The throughput of data measured in Gybtes per minute	0,00000495		0,0020730401
ThroughputGbytesPerHour	The throughput of data measured in Gbytes per hour	0,0002967503		0,1243824051
ReliableAndStableAssessment	Manual assessment on if the experiment performed reliable and stable	true		true
NumberOfFailedFiles	Number of files that failed in the workflow	0		0
AverageRuntimePerItemInSeconds	The average processing time in seconds per item	6422,4		369,88

Given these numbers, the experimental platform available at ONB would not be sufficient to process the web archive data summing up to a total of over 40 Terabytes at the time of running this experiment.

Technical details

May 13, 2014 (100): 100 arc files, Processing time 10:16:28 (hh:mm:ss), 1,28 GB (1308,63 MB,1372199221 Bytes)

These data samples are subsets of the ONB web archive crawl ONB Web Archive Dataset.

18 Appendix F2 – Evaluations in Large Scale Repository Testbed

18.1 EVAL Characterisation and validation of audio and video files during ingest

Experiment: Characterisation and validation of audio and video files during ingest

Evaluation specs component level

Field	Data type	Value
Evaluation seq. num.	int	1
Evaluator-ID	email	
Evaluation description	text	The overall goals is first to evaluate the stability of Taverna as the in-production workflow engine. Secondly we want to test Ffprobe as characterisation, file format validation and property validation tool (in combination with schematron) for 'Large Video Files'. Thirdly we would like to test the Isilon Storage performance, when the scratch storage is moved from NAS/SAN to Isilon Storage.
Evaluation-Date	DD/MM/YY	29/07/13
Dataset(s)	string	Danish TV broadcasts, mpeg-2 transport stream[174] Danish TV broadcast, H.264/MPEG-4 AVC Danish Radio broadcast, MPEG1-Layer 2
Workflow method	string	Taverna
Workflow(s) involved	URL(s)	https://github.com/statsbiblioteket/youseeingestworkflow
Tool(s) involved	URL(s)	Ffprobe[175] (a large number of tools is involved, but the focus in this evaluation is Ffprobe)
Link(s) to Scenario(s)	URL(s)	Characterisation and validation of audio and video files during ingest

Technical setup

Field	Data type	Value
Description	String	SB Video File Ingest Platform
Total number of physical CPUs	integer	8
CPU specs	string	Intel® Xeon® Processor X5355 (8M Cache, 2.66 GHz, 1333 MHz FSB)
Total number of CPU-cores	integer	32
Total amount of RAM in Gbytes	integer	32
Operating System	String	Linux (CentOS release 6.3 (Final))
Storage system/layer	String	NAS/SAN + NFS

Evaluation points

Metric	Baseline definition	Baseline value (24/7/2013)	Goal	Evaluation 1
ReliableAndStableAssessment	**Reliability - Runtime stability** (focus: Taverna workflow) Manual assessment. The workflow is set up with a workflow monitor in which each step of the workflow is recorded. If a file does not complete the workflow, it is added to the download list again and is run through the workflow again. It is also marked as 'failed' and can be viewed on the workflow monitor GUI, and a mail is sent to the digital content manager. If it later completes it is simply marked 'completed'. If the workflow cannot complete due to an inconsistency (e.g. wrong file format), the content provider is contacted.	true	true	
NumberOfFailedFilesAcceptable	**Reliability - Runtime stability** (focus: Ffprobe component) Manual assessment. Ffprobe returns errors on broken files. This is acceptable as these files should not continue through the workflow, but rather be re-downloaded. Ffprobe may also miss audiotracks that are not present from the beginning of the video file, but added at a later	true	true	

[174] http://wiki.opf-labs.org/display/SP/Danish+TV+broadcasts%2C+mpeg-2+transport+stream

[175] http://wiki.opf-labs.org/display/TR/Ffprobe

	time stamp. This means that the characterisation information from Ffprobe is not complete. We however also have characterisation information from a different tool, so we consider this acceptable.		
MaxObjectSizeHandledInGbytes	**Performance efficiency - Capacity** (focus: Taverna workflow - all components)	3.91Gb	5
MinObjectSizeHandledInMbytes	**Performance efficiency - Capacity** (focus: Taverna workflow - all components)	113.79Mb	100
NumberOfObjectsPerHour	**Performance efficiency - Capacity** / Time behaviour (focus: Taverna workflow)	79	100

18.2 EVAL-LSDR6-1

Experiment: SB Experiment SO4 Audio mp3 to wav Migration and QA Workflow

Evaluation specs component level

Field	Data type	Value
Evaluation seq. num.	int	1
Evaluator-ID	email	bam@statsbiblioteket.dk
Evaluation description	text	The evaluation of the *mp3 to wav migration and QA workflow* has three overall goals: • **Scalability** The workflow must be able to process a large collection within reasonable time. That is we want to be able to migrate and QA a large collection of radio broadcast mp3-files (20 Tbytes - 175.000 files) within weeks rather than years. • **Reliability** The workflow must run reliably without failing on a large number of files, and it must be possible to restart the workflow without losing work. • **Correctness** We must believe to some extent that the QA is correct. When a migrated file passes the QA, we should be able to say that we are y% certain that the migration was correct. This depends on the individual tools in the workflow.
Evaluation-Date	DD/MM/YY	13/11/12
Dataset(s)	string	mp3 (128kbit) with Danish Radio broadcasts
Workflow method	string	Taverna
Workflow(s) involved	URL(s)	MyExperiment Workflow Entry: Mp3 To Wav Migrate QA CLI List Test
Tool(s) involved	URL(s)	The workflow uses the following tools • FFmpeg • Ffprobe • JHOVE2 • MPG321 • xcorrSound
Link(s) to Scenario(s)	URL(s)	LSDRT6 Large scale migration from mp3 to wav

Technical setup

Field	Data type	Value
Description	String	iapetus.statsbiblioteket.dk
Total number of physical CPUs	integer	2
CPU specs	string	Intel® Xeon® Processor X5670 (12M Cache, 2.93 GHz, 6.40 GT/s Intel® QPI)
Total number of CPU-cores	integer	12
Total amount of RAM in Gbytes	integer	96
Operating System	String	Linux
Storage system/layer	String	NFS mounted files

Evaluation points

Metric	Baseline definition	Baseline value (2-16/10 2012)	Goal	Evaluation 1 (9-13/11 2012)
NumberOfObjectsPerHour	**Performance efficiency - Capacity / Time behaviour** Number of mp3 files migrated and QA'ed (no manual spot checks). The QA performed as part of the workflow at the time of the baseline test is Ffprobe Property Comparison, JHove2 File Format Validation and XcorrSound migrationQA content comparison. The mp3 files are 118Mb on average, and the two wav produced as part of the workflow are 1.4Gb on average. Thus a baseline value of 10 objects per hour means that we process 1.18Gb per hour and we produce 28Gb per hour (+ some property and log files). The collection that we are targeting is 20 Tbytes or 175.000 files. With baseline value we would be able to process this collection in a little over 2 years. The goal value is set so we would be able to process the collection in a week. Evaluation 1 (9th-13th November 2012). Simple parallelisation. Started two parallel workflows using separate jhove2 installations. Both on the same machine. Processed 879+877 = 1756 files in 4 days, 1 hour and 12 minutes.	10	1000	18
ReliableAndStableAssessment	**Reliability - Runtime stability** Manual assessment: the experiment performed reliably and stably for 13 days, but then Taverna failed with java.lang.OutOfMemoryError: Java heap space due to /tmp/ being filled up. All results were however saved, and the workflow could simply be restarted with a new starting point in the input list.	true (assessment October 16th 2012)	true	
NumberOfFailedFiles	**Reliability - Runtime stability** Files that fail are currently not handled consistently by the workflow, but we have so far not experienced any failed files.	0 (test 2nd-16th October 2012)	0	
QAFalseDifferentPercent	**Functional suitability - Correctness** This is a measure of how many content comparisons result in *original and migrated different*, even though the two files sound the same to the human ear. The parallel measure *QAFalseSimilarPercent* is how many content comparisons result in *original and migrated similar*, even though the two files sound different to the human ear. We have **not** experienced this - and we do not expect it to happen. We note that this measure is not improved by Testbed improvements, but rather by improvements to the XcorrSound migrationQA content comparison tool in the PC.QA work package. The goal value is set to make manual checking feasible. The collection that we are targeting is 20 Tbytes or 175.000 files. With *QAFalseDifferentPercent* at .5%, we would still need to check 175 2-hour files manually... Evaluation 1 (5th-9th November 2012). Processed 728 files in 3 days, 21 hours and 17 minutes = 5597 minutes, which is 5597/728 = 7.7 minutes pr. file in average. The number of files which returned Failure (original and migrated different) is 3 in 728 or 0.412 % of the files. We still need to check the failed files to see why they failed.	161 in 3190 ~= 5% (test 2nd-16th October 2012)	.1%	0.412 % (5th-9th November 2012)

We note that we would like to measure *QAConfidenceInPercent* - how sure are we of the QA? (Functional suitability - Correctness) This evaluation requires a *ground truth* that is not currently established.

18.3 Evaluation - SB Experiment mp3 to wav Migration and QA on Hadoop Cluster

Experiment: SB Experiment Audio mp3 to wav Migration and QA on Hadoop Cluster

Evaluator(s)

Bolette Jurik, SB

Evaluation points

In this testbed experiment we focus on performance. The earlier experiment EVAL-LSDR6-1 on mp3 to wav migration and QA using xcorrSound also focused on correctness. Moving the workflow to Hadoop to prove scalability, should not affect correctness of the tool.

- **Scalability** The workflow must be able to process a large collection within reasonable time. That is we want to be able to migrate and QA a large collection of radio broadcast mp3-files (20 Tbytes - 175.000 files) within weeks rather than years. The goal of 1000 for Number Of Objects Per Hour (or 0.28 for number of objects per second) would mean that we can migrate the 20TB radio broadcast mp3 collection in a week.
- **Reliability** The workflow must run reliably without failing on a large number of files, and it must be possible to restart the workflow without losing work.
- **Correctness/Scalability** We must believe to some extent that the automatic QA correctly identifies the "questionable" migrations, such that these can be checked in a manual QA process. We must however also insist that the number of migrations to check manually is minimal, as this is a very resource demanding process. The goal for QAFalseDifferentPercent has been changed to 2%. This means that we would have to check 3500 migrated 2 hour wav files manually. This is already too resource demanding. However the poor quality of the original files is a great challenge for the content comparison tool, and it turns out this is also too much to ask!

This table shows the results of the evaluation of the mp3 to wav migration and QA performed on the SB Hadoop Cluster.

Assessment of measurable points

Metric	Description	Metric baseline	Metric goal	Evaluation 2014 April 8th*	Evaluation 2014 June 17th-23rd**
number of objects per second	Performance efficiency - Capacity / Time behaviour	0.005	0.28	0.0567	0.0619
Number Of Objects Per Hour***	Performance efficiency - Capacity / Time behaviour Number of objects that can be processed per second	18 (9th-13th November 2012)	1000	204	223
QAFalseDifferentPercent	Functional suitability - Correctness Ratio of 'QA decided different'/'human judged same', that is ratio of content comparisons resulting in original and migrated different, even though human evaluation define original and migrated similar 0.412 %	0.412 % (5th-9th November 2012)	2%		~8.7%

*Based on the small experiment from April with max split size 128 below.
**Based on the large scale experiment from June below.
***This measure is not defined in the Metrics Catalogue, but we have kept it as a more readable extra supplement to number of objects per second.

Discussion and Conclusion
The conclusion is that the workflow does scale! We will not be able to migrate the collection in 1 week, but we will be able to do it in one month on the SB Hadoop cluster, which is considerably better than the one year needed without Hadoop (last evaluation).

The conclusion on correctness is really more of a discussion... The measure QAFalseDifferentPercent is defined as "Ratio of 'QA decided different'/'human judged same', that is ratio of content comparisons resulting in original and migrated different, even though human evaluation define original and migrated similar". As the large scale evaluations were performed on .5TB input, comprising 4998 2-hour mp3 files or 416.5 days, that is over a year of audio, we did not annotate the input, which means that ~8.7% is the number of files, where our audio content comparison tool reported that content of the original and the migrated files were not similar. Some of these may be actual migration errors. Most of them we however believe are due to the poor quality of the original material. Some of the original mp3 files have long periods of "nothing recorded" or silence. Silence or "almost silence" is very difficult to compare, and the tool will report not similar on these files. A better output would probably be "too much silence to perform content comparison". I would like to refer to the correctness evaluation of the xcorrSound waveform-compare tool done last year instead in section 2.4 Correctness based Benchmarks and Validation Tests of deliverable D11.2 Quality Assurance Workflow, Release 2 + Release Report[176].

Small Experiments April 2014
All run on a file list of 58 files (7.2Gb in total).

max-split-size	duration	launched map tasks on the three Hadoop jobs
1024	37m, 58.593s = 2278.593s	3,3,7
512	24m, 1.9s = 1441.9s	6,6,14
256	18m, 17.917 = 1097.917	12,12,28
128	17m, 3.176 = 1023.176	24,24,57
64	16m, 54.703s = 1014.703s	47,47,113
32	17m, 29.96s = 1049.96	93,93,225

The small experiments were mainly run to decide an optimal max-split-size (or an optimal number of map-reduce map tasks). A split is a part of an input file that one map task is working on. Picking the appropriate size for the tasks for your job can radically change the performance of Hadoop. When working on text files, the default of letting the number of map tasks depend on the number of DFS blocks in the input files works well. We are not working on text files. The input to our map-reduce jobs are text files, but very small text files only containing lists of paths to the audio files, we actually want to work on. We thus want much smaller splits of only a few lines each. The max-split-size (mapred.max.split.size) is the maximum size of such a split in bytes.

The exact number of MR map tasks seems not to have a big influence on performance, as long as we have more than 2*12. That is as long as max split size is at most 256 on an input file list of 58 files.

[176] http://www.scape-project.eu/deliverable/d11-2-quality-assurance-workflow-release-2-release-report

We note that we get approximately twice as many launched maps for the waveform-compare Hadoop job, simply because the input list is approximately twice as big, as it is a list of pairs. We can of course adjust this to get approximately the same number of jobs, but as the two first jobs FfmpegMigrate and Mpg321Convert run simultaneously, and the WaveformCompare job runs alone, we actually have approximately the same number of map tasks throughout the workflow.

The large scale experiments were held up for a while, due to too few connections to storage. Remember we are using the SB Hadoop Platform. As this job writes very much data, the Isilon disk I/O and CPU use were being maxed out, even though we were trying to "play nice" and only run 28 maps concurrently. The number of connections to the 16 nodes Isilon storage solution at SB was 2 when the small scale experiments were run. It was then set up to five connections before we ran the large scale experiments.

Large Scale Experiments June 2014
This line of tests will focus on scalability. If max-split-size=256 (bytes) gives us 2*12 maps on an input-txt-file with 58 files, this means 256 bytes is approx 58/12=4,8333 files, so one file is approx 256/4.8333=52.9655 bytes. Then if we want approx 2*12 maps on an input-txt-file with 1000 files, we want max-split-size to be approx 1000/12*52.9655 = 4413.7931 bytes.

The jobs were run on file-lists of approximately 1000 files (129GB); the max.split.size was set to 4414; and each job writes approximately 3.1TB of intermediate and output wav files (+ some small log files).

date	size:#mp3s	total size	duration	total duration	NumberOfObjectsPerHour	failure	total failure	QAFalseDifferentPercent
2014 Jun 17	1000	1000 (129GB)	4h, 33m	4h, 33m	220	63	63	6.3
2014 Jun 18	1000	2000 (258GB)	4h, 23m	8h, 56m	224	111	174	8.7
2014 Jun 19	999	2999 (387GB)	4h, 20m	13h, 29m	222	52	226	~7.5
2014 Jun 20	1000	3999 (516GB)	4h, 27m	17h, 56m	223	142	368	~9.2
2014 Jun 23	999	4998 (645GB)	4h, 28m	22h, 24m	223	67	435	~8.7

Assessment of non-measurable points
In the last evaluation, we did include ReliableAndStableAssessment Reliability - Runtime stability in the evaluation points, and we wrote true both in goal and in baseline value (Manual assessment: the experiment performed reliably and stably for 13 days, but then Taverna failed with java.lang.OutOfMemoryError: Java heap space due to /tmp/ being filled up. All results were however saved, and the workflow could simply be restarted with a new starting point in the input list). This measure is not a part of the scape metrics catalogue, but stability judgement is and an evaluation follows here.

The experiment performed reliable and stably for around 4 hours. I will however note that this experiment was not focused on reliability, and all intermediate results are potentially lost if the workflow is killed. I will also note that we partitioned the input to the workflow, so it worked on only 1000 files at a time. This was done as the test environment had on upper limit on available storage, and the workflow produces approximately 3.1TB of output files for each 1000 input files. The workflow will fail if it does not have enough output storage. Working on only 1000 files at a time of

course has the benefit, that only 1000 results can be lost at a time, and as the workflow seems to run stably for this size input it is reliable and stable in this configuration. Using this configuration however means that for a 20TB 175000 file collection, I need 175 input files and a script that starts the workflow 175 times sequentially (and roughly .5 Petabyte available storage).
A note about goals-objectives omitted, and why

This evaluation covers performance, reliability and functional suitability to some extent. We did not look at the metrics max object size handled in bytes and min object size handled in bytes. These measures would certainly contribute to the evaluation. Our collection (Danish Radio broadcasts, mp3) has mp3 files varying very little in size (approx. 2 hours, average file size 118Mb, largest file: 135Mb) and the workflow thus produces wav files varying very little in size (2 wav files of around 1.4Gb for one 118Mb mp3 file). The test mp3 files used under development were of course considerably smaller (around 7Mb) and produced smaller output (around 50Mb*2 per mp3 file). We think that the workflow can handle larger files as well, but this was not tested. We can report that for input min object size handled in bytes is around 7Mb (7000000 bytes) and max object size handled in bytes is around 135Mb (135000000 bytes). For output min object size handled in bytes is around 50Mb (50000000 bytes) and max object size handled in bytes is around 1.4Gb (1400000000 bytes). This would be an interesting measure to experiment further with.

We did also not look at the metrics throughput in bytes per second. This measure can be computed from number of objects per second or Number Of Objects Per Hour. The evaluation 2014 June 17th-23rd gave us Number Of Objects Per Hour=223. To compute throughput in bytes per second, we need the throughput size. Our question here is what throughput means. We wrote that the 1000 files in input were only approximately 129GB but they produced 3.1TB of intermediate and output wav files. Half of these (1.55TB) is output, and we will use this as the throughput size. Then for Number Of Objects Per Hour=223, we get 1.55/1000*223 = 0.34565TB or 0.34565×1024 = 353.9456Gb of throughput per hour, that is 353945600000 / 60 / 60 = 98318222 bytes or 98 MB of throughput per second.

The evaluation does not cover

- Organisational maturity
- Maintainability
- Planning and monitoring efficiency
- Commercial readiness

This experiment was focused on tools and platform and performance, and we will keep the evaluations to the specific experiment.

Technical details

The workflow that was used is version 4 of the Slim Migrate And QA mp3 to Wav Using Hadoop Jobs workflow available from http://www.myexperiment.org/workflows/4080.html

The Hadoop jobs that were used are from commit e1ec47d of the https://github.com/statsbiblioteket/scape-audio-qa-experiments project.

The waveform-compare tool that was used was from xcorrSound release v2.0.2 https://github.com/openplanets/scape-xcorrsound/releases/tag/v2.0.2.

The ffmpeg used was version 0.10 Copyright (c) 2000-2012 the FFmpeg developers built on Mar 9 2012 09:32:12 with gcc 4.4.6 20110731 (Red Hat 4.4.6-3).

The mpg321 used was version 0.2.10. Copyright (C) 2001, 2002 Joe Drew.

The cluster set up that was used was the June 2014 version of the SB Hadoop Platform.

WebDAV
The Taverna logs and outputs of the June experiment are stored on http://fue.onb.ac.at/scape-tb-evaluation/sb/LargeScaleAudioMigration/Mp3ToWavMigrationOnHadoop/ along with the SB scape Hadoop Cluster map-reduce client configuration.

Evaluation notes

- We have (stubbornly) kept the old measure Number Of Objects Per Hour in our evaluation, as it is simply easier to read when the processing time is as long as in this experiment.
- QAFalseDifferentPercent was introduced as a measure, when we were working on smaller annotated datasets. When we are working on large scale real life datasets it is problematic. A better idea would probably be to have a Dissimilar in Percent measure along with a Correctness judgement based on the Dissimilar in Percent measure along with prior correctness evaluations on annotated data. We would then also need a discussion of the adequacy of the solution when taken into account the level of automation and the human resources still needed.

Conclusion

The conclusion is that we are able to migrate our 20TB mp3 collection to wav including quality assurance in one month on the SB Hadoop Platform. We however need roughly 0.5 Petabyte available storage, which is not feasible, and we will not do this migration. The xcorrSound waveform-compare tool has proven robust and easy to integrate in a larger workflow, and we will continue maintenance and maybe further development on xcorrSound.

18.4 EVAL Characterisation and Identification on SCAPE Azure Platform

Experiment: Characterisation and Identification on SCAPE Azure Platform

Results

The speed at which Tika and DROID identified file formats is shown in Table 1 below, along with the speed of an MD5 calculation utility operating on the same files. Calculating a file's MD5 checksum involves reading the entire file while performing very few calculations, so the MD5 numbers effectively compare file access times between the two servers.

The results are in files per second, so larger numbers are better.

(The on-site server numbers were read from a bar chart in the earlier report and are approximate.)

	On-Site Server	Azure VM
Tika	61	659
DROID	47	65
MD5	42	426

Files per second

Evaluation points

Assessment of measurable points Tika

Metric	Description	February 04, 2014	May 21, 2014
NumberOfObjectsPerHour	Number of objects processed in one hour	2371809	2447712
MinObjectSizeHandledInGbytes	Smallest ARC file in sample	0.000000026	0.000000026
MaxObjectSizeHandledInGbytes	Biggest ARC file in sample	0.347421616	0.347421616
ThroughputGbytesPerMinute	The throughput of data measured in Gybtes per minute	20.0	20.7
ThroughputGbytesPerHour	The throughput of data measured in Gbytes per hour	1201.8	1240.2
ReliableAndStableAssessment	Manual assessment on if the experiment performed reliable and stable	true	true
NumberOfFailedFiles	Number of files that failed in the workflow	N/A	0
AverageRuntimePerItemInSeconds	The average processing time in seconds per item	0.001517829	0.001470761

Assessment of measurable points DROID

Metric	Description	February 04, 2014	May 21, 2014
NumberOfObjectsPerHour	Number of objects processed in one hour	234472	217809
MinObjectSizeHandledInGbytes	Smallest ARC file in sample	0.000000026	0.000000026
MaxObjectSizeHandledInGbytes	Biggest ARC file in sample	0.347421616	0.347421616
ThroughputGbytesPerMinute	The throughput of data measured in Gybtes per minute	20.0	20.7
ThroughputGbytesPerHour	The throughput of data measured in Gbytes per hour	118.8	110.4
ReliableAndStableAssessment	Manual assessment on if the experiment performed reliable and stable	true	true
NumberOfFailedFiles	Number of files that failed in the workflow	N/A	1727
AverageRuntimePerItemInSeconds	The average processing time in seconds per item	0.015353637	0.016528232

Assessment of measurable points MD5

Metric	Description	February 04, 2014	May 21, 2014
NumberOfObjectsPerHour	Number of objects processed in one hour	1532443	1340888
MinObjectSizeHandledInGbytes	Smallest ARC file in sample	0.000000026	0.000000026
MaxObjectSizeHandledInGbytes	Biggest ARC file in sample	0.347421616	0.347421616
ThroughputGbytesPerMinute	The throughput of data measured in Gybtes per minute	12.9	11.3
ThroughputGbytesPerHour	The throughput of data measured in Gbytes per hour	776.5	679.4
ReliableAndStableAssessment	Manual assessment on if the experiment performed reliable and stable	true	true
NumberOfFailedFiles	Number of files that failed in the workflow	N/A	0
AverageRuntimePerItemInSeconds	The average processing time in seconds per item	0.00234919	0.002684789

Interpretation

The huge difference in the results for the MD5 utility more than an order of magnitude indicates that a much faster file system was used on the Azure VM. Indeed, the Govdocs1 files were put directly on the VM's hard drive, whereas the files in the on-site study were mounted on a separate Network File System server accessed over a network. If the files in the earlier study had been put directly on the server (which probably wasn't possible), the on-site performance would likely have been much better.

(And it must be noted that if one of the other storage options for Azure had been used BLOB storage or SQL Server database, for example the Azure performance would likely have been much worse.)

This suggests that if a very large collection of documents needs to be identified quickly, the location of the files is important.

It's likely that the difference in Tika performance is also due primarily to storage differences. However, the DROID numbers seem odd, at least at first glance. DROID ran faster on the Azure VM, but certainly not an order of magnitude faster. Further investigation would be required to prove this, but a possible explanation is that MD5 and Tika are IO-bound, while DROID is CPU-bound. In other words, DROID spends more time calculating than reading, so much so that even a 10X increase in reading speed results in only a moderate increase in overall speed.

Note that different CPUs were used in the two servers: The on-site server used dual Xeon X5670 CPUs running at 2.93 GHz, while the Azure VM used two cores of a six-core AMD Opteron 4171 HE running at 2.09 GHz. It's not clear which should be faster, but in any case the storage differences in the two studies clearly had a much larger effect that any CPU speed differences.

Conclusion

The results can be easily summarized: Tika and DROID run fast on a Microsoft Azure VM, provided that the files that need to be identified are local to the machine.

A few additional facts became apparent during the study. First, Tika and DROID can be easily used in the Microsoft Azure environment. They were run on an Azure VM for this study, and they are currently used in an Azure Web Role in the SCAPE API Service. (VMs are part of Azure's

infrastructure as a service offering, while Web Roles and Worker Roles are part of Azure's platform as a service offering.) In both cases, no serious problems were encountered while deploying the tools.

Second, the scalability inherent in Azure makes it easy to allocate hardware as needed. If the VM's hard disk isn't large enough to hold the files that need to be identified, a larger disk can easily be attached. Or if one VM is inadequate, duplicate VMs can be easily created. Those are manual steps, but scaling can also be done dynamically in response to changing demands by utilizing Azure's auto-scaling features. This is all much easier than trying to scale up physical hardware in an on-site environment.

18.5 EVAL KB Metamorfoze Image Migration & QA

Experiment: KB Metamorfoze Image Migration & QA

Evaluations

Metric	Datatype	Description	Value
NumberOfObjectsPerHour	integer	Number of objects that can be processed per hour	1341
MigrationCorrectnessInPercent	integer	Defining a statistical measure for binary evaluations	100 %
ThroughputGbytesPerMinute	float	The throughput of data measured in Gybtes per minute	0,47
ThroughputGbytesPerHour	float	The throughput of data measured in Gbytes per hour	28,16
ReliableAndStableAssessment	boolean	Manual assessment on if the experiment performed reliable and stable	true
NumberOfFailedFiles	integer	Number of files that failed in the workflow	0
NumberOfFailedFilesAcceptable	boolean	Manual assessment of whether the number of files that fail in the workflow is acceptable	true
QAFalseDifferentPercent	integer	Number of content comparisons resulting in original and migrated different, even though human spot checking says original and migrated similar.	0 %
AverageRuntimePerItemInHours	float	The average processing time in hours per item	0.00099
AwareCompressRuntimeAvg	integer	Average running time of Aware jp2_compress in milliseconds	2377
JpylyzerCheckRuntimeAvg	integer	Average running time of Jpylyzer validation in milliseconds	212
ProbatronCheckRuntimeAvg	integer	Average running time of Probatron profile validation in milliseconds	1628
KakaduExpandRuntimeAvg	integer	Average running time of Kakadu kdu_expand in milliseconds	2087
GMCompareRuntimeAvg	integer	Average running time of GraphicsMagick pixel comparison in milliseconds	381
AwareCompressRuntimeFull		Total running time of Aware jp2_compress	5:18,46
JpylyzerCheckRuntimeFull		Total running time of Jpylyzer validation	0:28,25
ProbatronCheckRuntimeFull		Total running time of Probatron profile validation	3:38,18
KakaduExpandRuntimeFull		Total running time of Kakadu kdu_expand	4:39,54
GMCompareRuntimeFull		Total running time of GraphicsMagick pixel comparison	0:51,07
AwareCompressSuccess		Success rate of Aware jp2_compress	100 %
JpylyzerCheckSuccess		Success rate of Jpylyzer validation	100 %
ProbatronCheckSuccess		Success rate of Probatron profile validation	100 %
KakaduExpandSuccess		Success rate of Kakadu kdu_expand	100 %
GMCompareSuccess		Success rate of GraphicsMagick pixel comparison	100 %

18.6 EVAL-BL-LSDRT-TIFFJP2-01

Experiment: LSDRT2 EX1 BL Newspapers on the BL Platform

Evaluator(s)

William Palmer, BL

Evaluation points

Assessment of measurable points

There are various runs of the migration workflow entered here:

The first part of title details the storage type:

- HDFS: in HDFS on the Hadoop cluster
- Webdav: stored on a NAS local to the Hadoop cluster
- Fedora: stored in/via a Fedora Commons Repository, with object storage on the same NAS as for the Webdav code

The second part of the title described the execution method of the workflow (all using OpenJPEG unless explicitly labelled):

- CommandLineJob: a Java controlled workflow i.e. native MapReduce calling out to external programs as required
- CommandLineJob-Kakadu: same as CommandLineJob, but replacing OpenJPEG for Kakadu
- Taverna: a Taverna workflow, called via the Taverna command line application, calling out to external programs as required

Metric	Metric goal	Metric baseline - Batch on one processing node	Fedora-CommandLineJob	Webdav-CommandLineJob
TotalRuntime	40hours	38:08:30	57:50:00	57:58:00
NumberOfObjectsPerHour	1600	26.2	725.6	723.9
ThroughputGbytesPerHour	25	0.8	16.6	16.6
ReliableAndStableAssessment	TRUE	-	TRUE	TRUE
NumberOfFailedFiles	0	-	3*	3*
NumberOfFailedFilesAcceptable	-	-	TRUE	TRUE
		1000 files only	41963 files	41963 files
			See notes 0 & 2	See notes 0 & 4

Metric	Metric goal	HDFS-CommandLineJob	Fedora-Taverna	HDFS-Taverna	HDFS-CommandLineJob-Kakadu
TotalRuntime	40hours	57:02:00	68:05:00	67:11:00	17:25:00
NumberOfObjectsPerHour	1600	735.8	616.3	624.6	2409.4
ThroughputGbytesPerHour	25	16.9	14.1	14.3	55.3
ReliableAndStableAssessment	TRUE	TRUE	TRUE	TRUE	TRUE
NumberOfFailedFiles	0	3*	3*	3*	3*
NumberOfFailedFilesAcceptable	-	TRUE	TRUE	TRUE	TRUE
		41963 files	41963 files	41963 files	41963 files
		See note 1	See notes 0 & 2	See note 1	See notes 1 & 3

- Note 0: For the Fedora and Webdav runs, the runtime includes recovering the file across the network and posting the migrated file back across the network
- Note 1: Copying data from NAS to HDFS took 08:03 (hh:mm). Copying processed data from HDFS to the NAS will also take time but was not measured. None of the copying time is included in TotalRuntime
- Note 2: Fedora Commons hosted on a VM, retrieving files from the NAS and serving them to the Hadoop job
- Note 3: When using Kakadu, the migrated JP2 files have slightly lower PSNR values, thus threshold was lowered from 50 for OpenJPEG, to 48, so files would pass
- Note 4: Creating a directory in a webdav folder is expensive, therefor, all output files are put in to one directory (same as for HDFS)
- Note 5: One of the files failed after going through tifftopnm, Kakadu didn't like the input - "Image file for component 0 terminated prematurely!"

We can meet, and exceed our 40 hour target, by using CommandLineJob-Kakadu as the workflow. That workflow is twice as fast as our metric goal, meaning we could process our entire collection in less than a month on our Hadoop instance.

* The three failed files either failed to migrate or failed the QA step and this was correctly reported by each workflow. Those files can now be investigated as to why they did not migrate successfully. They did not stop the execution of the workflow.

Technical details

Taverna workflow: http://www.myexperiment.org/workflows/3401.html
Source code: https://github.com/bl-dpt/chutney-hadoopwrapper/commit/73378803e9838ff7a17fc49b5407231a48ac99a7
Platform: http://wiki.opf-labs.org/display/SP/BL+Hadoop+Platform
Fedora version used: 3.6(?)

Evaluation notes

Conclusion

The various ways in which Taverna and Hadoop could be used together were investigated within this experiment. It is interesting to note that execution speed of the workflow when recovering and storing files remotely from the cluster only took a fraction longer than when the files were stored in HDFS. Since the files took 8 hours to be copied in, it may not make sense to cache files in HDFS before processing in this instance, unless the files are already stored there. Due to the execution speed of the various migration workflows we can deduce that we could migrate files from a remote repository, using CommandLineJob/Kakadu, within roughly half the total time goal we were aiming for. In that case we would not need to copy files to HDFS for processing in advance of executing the workflow. Should the files be smaller, or on a slower network, then the results will differ and consideration should be given to caching the files in HDFS, or storage that is more local.

18.7 EVAL TIFF to JPEG2000 Migration Experiment at ONB

Experiment: TIFF to JPEG2000 Migration Experiment at ONB

Evaluation summary

Files := Size of random sample
Total GB := Total size in Gigabytes
Secs := Processing time in seconds
Mins := Processing time in minutes
Hrs := Processing time in hours
Afg.p.f. := Average processing time per file in seconds
Obj/h := Number of objects processed per hour
GB/min := Throughput in Gigabytes per minute
GB/min := Throughput in Gigabytes per hour
Err := Number of processing errors

Taverna Workflow - Sequential execution

Files	Total GB	Secs	Mins	Hrs	Avg.p.f.	Obj/h	GB/min	GB/h	Err
5	0,31 GB	179	2,98	0,05	35,80	101	0,10	6,22	0
7	0,89 GB	438	7,30	0,12	62,57	58	0,12	7,29	0
10	0,90 GB	478	7,97	0,13	47,80	75	0,11	6,8	0
20	2,23 GB	1150	19,17	0,32	57,50	63	0,12	6,98	0
30	2,99 GB	1541	25,68	0,43	51,37	70	0,12	6,98	0
40	3,60 GB	1900	31,67	0,53	47,50	76	0,11	6,81	0
50	3,46 GB	2039	33,98	0,57	40,78	88	0,10	6,1	0
75	6,05 GB	3425	57,08	0,95	45,67	79	0,11	6,36	0
100	8,30 GB	4693	78,22	1,30	46,93	77	0,11	6,37	0
200	15,19 GB	9246	154,10	2,57	46,23	78	0,10	5,91	0
300	19,07 GB	11773	196,22	3,27	39,24	92	0,10	5,83	0
400	24,78 GB	15644	260,73	4,35	39,11	92	0,10	5,70	0
500	34,55 GB	21345	355,75	5,93	42,69	84	0,10	5,82	0
750	63,07 GB	37397	623,28	10,39	49,86	72	0,10	6,07	0
1000	71,82 GB	42376	706,27	11,77	42,38	85	0,10	6,10	0
2000	139,00 GB	84938	1415,63	23,59	42,47	85	0,10	5,89	0
3000	211,85 GB	128959	2149,32	35,82	42,99	84	0,10	5,91	0

Pig Workflow - Distributed Execution

Files	Total GB	Secs	Mins	Hrs	Avg.p.f.	Obj/h	GB/min	GB/h	Err
5	0,31 GB	96	1,60	0,03	19,20	188	0,19	11,60	0
7	0,89 GB	101	1,68	0,03	14,43	250	0,53	31,64	0
10	0,90 GB	103	1,72	0,03	10,30	350	0,53	31,56	0
20	2,23 GB	114	1,90	0,03	5,70	632	1,17	70,45	0
30	2,99 GB	138	2,30	0,04	4,60	783	1,30	77,99	0
40	3,60 GB	161	2,68	0,04	4,03	894	1,34	80,41	0
50	3,46 GB	183	3,05	0,05	3,66	984	1,13	68,01	0
75	6,05 GB	272	4,53	0,08	3,63	993	1,34	80,11	0
100	8,30 GB	373	6,22	0,10	3,73	965	1,34	80,15	0
200	15,19 GB	669	11,15	0,19	3,35	1076	1,36	81,73	0
300	19,07 GB	808	13,47	0,22	2,69	1337	1,42	84,95	0
400	24,78 GB	1091	18,18	0,30	2,73	1320	1,36	81,77	0
500	34,55 GB	1397	23,28	0,39	2,79	1288	1,48	89,03	0
750	63,07 GB	2399	39,98	0,67	3,20	1125	1,58	94,64	0
1000	71,82 GB	2746	45,77	0,76	2,75	1311	1,57	94,16	0
2000	139,00 GB	5450	90,83	1,51	2,73	1321	1,53	91,82	0
3000	211,85 GB	8328	138,80	2,31	2,78	1297	1,53	91,58	0

The following diagram shows the comparison of wall clock times in seconds (y-axis) of the Taverna workflow and the Pig workflow using an increasing number of files (x-axis).

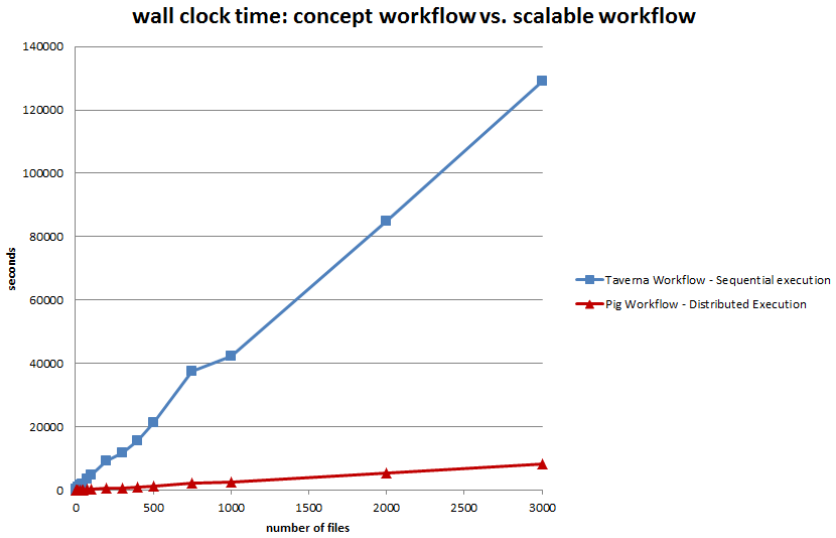


Figure 35 Wallclock times of concept workflow and scalable workflow

However, the throughput we can reach using this[177] cluster and the chosen pig/hadoop job configuration is limited; as figure 36 shows, the throughput (measured in Gigabytes per hour – GB/h) is rapidly growing when the number of files being processed is increased, and then stabilises at a value around slightly more than 90 Gigabytes per hour (GB/h) when processing more than 750 image files.

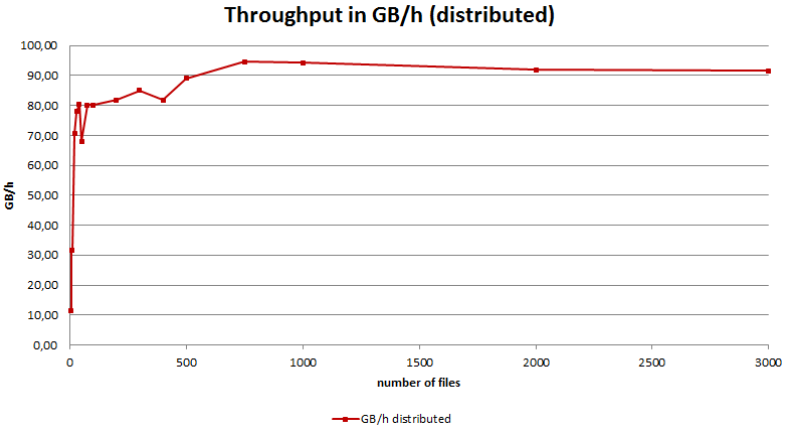


Figure 36 Throughput of the distributed execution measured in Gigabytes per hour (GB/h) against the number of files processed

[177] http://wiki.opf-labs.org/display/SP/ONB+Hadoop+Platform

18.8 EVAL-BL-LSDRT-PDFDRM-01

Experiment: Validate PDF&EPUBs and check for DRM

Evaluator(s)

William Palmer, BL

Evaluation points

Assessment of measurable points

Metric	Metric goal	January 2014 (Baseline)	July 2014
TotalRuntime		4:28:00 (hh:mm:ss)	13:22:42 (hh:mm:ss) [0]
TotalObjects		231683	231671 [1]
NumberOfObjectsPerHour		**51869.32836**	**17316**
ThroughputGbytesPerHour		28.61674477	9.55
ReliableAndStableAssessment	TRUE	TRUE	TRUE [2]
NumberOfFailedFiles	0	0	12 [2] [3] + 660 Exceptions [4]
NumberOfFailedFilesAcceptable	-	TRUE	TRUE [1]
		Note this is just DRM and validity checks	For this run the tool was renamed Flint and it contained DRM and validity checks, along with a policy check for PDF files

Note [0] Note that this runtime includes additional policy checks (see the selected policy validation (PV) results in the table below)

Note [1] twelve fewer files were used, as those files were found to crash or hang the JVM (in the policy check stage). This is due to a combination of factors, including the use of a relatively old JVM on the Hadoop cluster, using in-development software and the files potentially being corrupt - further investigation is required. Upgrading the JVMs on the entire cluster may well solve these issues. Removing these files from the input data meant that the test run could be completed successfully.

Note [2] the run completed successfully, however, twelve files had to be excluded from the final run (see [1]). Additionally, for 26184 files, policy validation execution failed. We have found issues with corrupt/broken files in the Govdocs1 corpus due to reporting these issues to Apache PDFBox (see https://issues.apache.org/jira/browse/PDFBOX-1756, https://issues.apache.org/jira/browse/PDFBOX-1757, https://issues.apache.org/jira/browse/PDFBOX-1761, https://issues.apache.org/jira/browse/PDFBOX-1762, https://issues.apache.org/jira/browse/PDFBOX-1769, https://issues.apache.org/jira/browse/PDFBOX-1774, https://issues.apache.org/jira/browse/PDFBOX-1795)

Note [3] The excluded failed files were: 020087.pdf 165487.pdf 289451.pdf 289452.pdf 383325.pdf 299694.pdf 375118.pdf 451665.pdf 451675.pdf 526572.pdf 924677.pdf 870521.pdf

Note [4] 660 exceptions were present in the output. 626 were due to a failure after the text extraction step of PDF validation (as no output file was present), 31 were null pointer exceptions, with 3 other miscellaneous exceptions. Due to the nature of the dataset, these sorts of errors are

expected, and given the small number of issues, they are fixable. The files that failed are identifiable so testing of further development could be run against those in particular.

Selected results from Flint output:

Test	Number of files	
DRM detected NOTE: checks are not currently made against print/copy restrictions etc	9791	4.2%
Well-formed failure	19166	8.3%
Policy Validation execution failure	26184	11.3%
PV encryption check (present)	6609	2.9%
PV damaged fonts present	89	0.04%
PV javascript present	221	0.1%
PV embedded files present	4873	2.1%
PV multimedia present	66	0.03%

As Flint provides results as a table, different policies can be checked against the Flint output as various times. For example - whether or not files contain multimedia may become an issue and this can then be determined from the results.

Technical details

Platform: http://wiki.opf-labs.org/display/SP/BL+Hadoop+Platform

Source code: [Jan 14]
https://github.com/bl-dpt/drmlint/commit/ecca9a28fe095bed6b770e59046d17d7e595fd09

[July 2014] https://github.com/openplanets/flint

Evaluation notes

Files are kept in HDFS, it's possible that sequence files might help in this instance as the files themselves are relatively small.

There are issues with the JVM but it was not possible to easily upgrade the JVM on the cluster

Some really broken files crashed the JVM/libraries and it is not possible to protect against JVM crashes. There are known to be issues with some of the input files in Govdocs1 (see above issue reports).

Conclusion

Testing with the policy checks takes approximately three times as long as the basic checks. Extrapolating from the test dataset for this evaluation, it would be possible to process 1TB of PDF files, with policy checks, in less than 4.5 days on our Hadoop cluster. This is acceptable for using on a routine basis, should that be necessary. Although PDF files can be relatively small, Flint's execution speed does not appear to suffer from the small files problem - its processing is CPU bound, not I/O bound, as evidenced by the 9.55GB/h processing speed (i.e. 2.7MB/s read speed).

18.9 Evaluation 1 - JPEG2000 validation

Experiment: Validate JPEG2000 Newspapers Using Jpylyzer

Evaluator(s)

Rune Ferneke-Nielsen, SB

Purpose

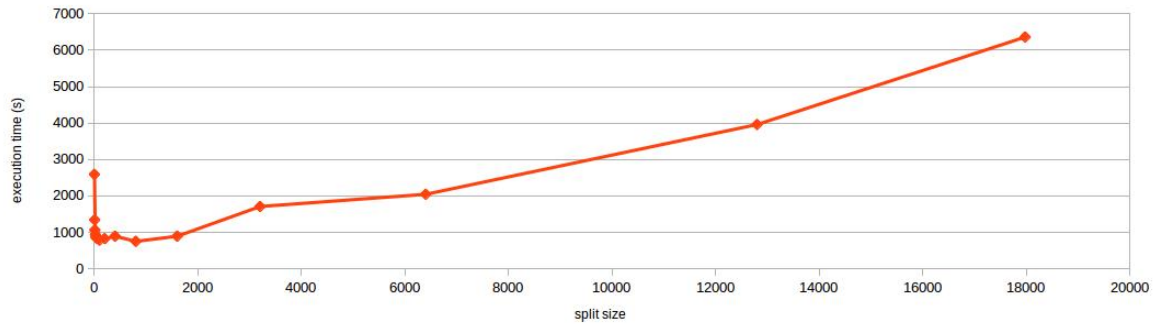
In this evaluation, Evaluation 1, we would like to get an indication of how fast we can validate a set of JPEG2000 files against our institutional policy. The validation step is only one part of a larger workflow, but the result should still give us a good indication.

The image files are accessed via a NFS mount, which is a valid environment configuration at SB. How the image files appear on the NFS mount (e.g. from a tape storage) before being accessed is not in scope for this experiment, and it may require several manual steps to be performed. Also, another important aspect is that the provenance data from the validation should be stored in a repository; this is not part of this evaluation.

Evaluation points

Assessment of measurable points

Metric	Description	Metric baseline	Metric goal	February 24, 2014
NumberOfObjectsPerHour	Performance efficiency - Capacity / Time behaviour Number of newspaper pages (i.e. meta data) being validated against an organisational policy	10175 (split size: 20000)	5000	20000+ (approx 65000)
NumberOfFailedFiles	Reliability - Runtime stability Number of files failed, in jpylyzer step and/or policy comparison step	N/A	0	0



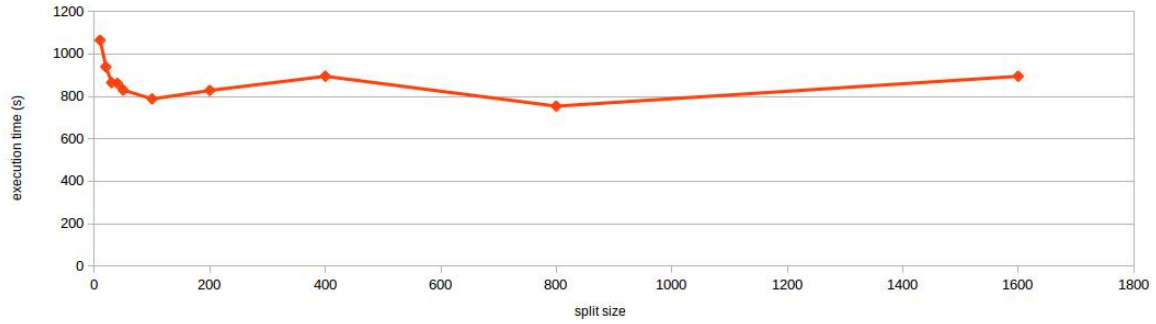


Figure 37 Zoomed version

The above graph is generated from data:

split size	1	5	10	20	30	40	50	100	200	400	800	1600	3200	6400	12800	17978
execution time (s)	2592	1344	1065	939	865	861	830	788	828	895	754	895	1710	2045	3960	6360

When using a spilt size between 10 and 1600, it takes approximately 1000 seconds to process the entire data set (17978 files). This means that we can process 20000+ (the number is closer to 65000) files within an hour, and the metric goal has therefore been reached in the first iteration of this experiment.
In the worst case scenario, where one process will do all the work, it takes 6360 seconds to process the entire data set. This means that we can process approximately 10000 files within an hour, which is also enough to reach the metric goal.

Assessment of non-measurable points

Reliability - Stability indicators

- A Scape component for converting tool-specific output into a scape-generic format is under development, and could therefore not be used in this experiment. Instead a minimal implementation was created as part of the experiment, so it was possible to execute and evaluated the experiment (concern).
- A Scape component for comparing the scape-generic format with an organisational policy is under development, a simple implementation was used. Also, this component is somewhat dependent of the above described tool-specific conversion component (concern).
- It is uncertain whether the Scape modules (for jpylyzer and organisational policy) have an active community (concern).
- Java and Hadoop both have proved usable as real-life systems, and have an active community (no concern).

Functional suitability - Correctness

- Software packages / modules handling organisational policy are under development and as such correctness cannot be verified (concern).

Organisational maturity - Dimensions of maturity: Awareness and Communication; Policies, Plans and Procedures; Tools and Automation; Skills and Expertise; Responsibility and Accountability; Goal Setting and Measurement

- No planning or monitoring is present.

Maintainability - Reusability

- The Jpylyzer tool is placed in a repository, making it easily accessible and reusable. Other Scape packages / modules are still to be made reusable. The OPF organisation is working towards a solution for handling the lifecycle of preservation tools (no concern).
- The policy validation can be handled in a number of different ways, but to make it automatic and machine-readable it still requires technical staff (no concern).

Maintainability - Organisational fit

- Given that the organisation have suited technical staff, such as software developers, the approach is viable (no concern).

Commercial readiness

- The Jpylyzer tool is in use at the State and University Library, being used actively in a quality assurance process.

Functional suitability - Completeness

- Only one input format is in play and no plans to expand.

Planning and monitoring efficiency - Information gathering and decision making effort

- No planning or monitoring is present.

Technical details

Implementation

The implementation can be found at github: statsbiblioteket/scape-jp2-qa[178]
Use the tag scape_evaluation_1 for actual code point: git checkout scape_evaluation_1

Provenance Data

- Hadoop job implementation[179]
- Control Policy[180]

- Output from the Hadoop job can be found here[181]
- Timings for running the jpylyzer tool as a single process can be found here[182]

[178] https://github.com/statsbiblioteket/scape-jp2-qa
[179] http://fue.onb.ac.at/scape-tb-evaluation/sb/ValidationOfArchivalContentAgainstAnInstitutionalPolicy/ValidateJPEG2000NewspapersUsingJpylyzer/scape-jp2-qa-1.0-SNAPSHOT-jar-with-dependencies.jar
[180] http://fue.onb.ac.at/scape-tb-evaluation/sb/ValidationOfArchivalContentAgainstAnInstitutionalPolicy/ValidateJPEG2000NewspapersUsingJpylyzer/statsbiblioteket_control_policy_jpeg2000.rdf
[181] http://fue.onb.ac.at/scape-tb-evaluation/sb/ValidationOfArchivalContentAgainstAnInstitutionalPolicy/ValidateJPEG2000NewspapersUsingJpylyzer/out1-20000.log
[182] http://fue.onb.ac.at/scape-tb-evaluation/sb/ValidationOfArchivalContentAgainstAnInstitutionalPolicy/ValidateJPEG2000NewspapersUsingJpylyzer/jpylyzer_timings.txt

- Timings for running the md5sum tool as a single process can be found here[183]

Conclusion

From this first evaluation, we have found that the solution for policy driven validation is appropriate and can easily handle the forecasted load. We have seen that the processing handles around 65000 image files every hour, which is much more than the metric goal of 5000 image file every hour. This is a good result, and we can move forward towards an integrated solution that incorporates extracting and storing data via our repository. Further, the large span between 5000 and 65000 is very positive, as it will take time to extract and store data; and we still need to reach the metric goal.

[183] http://fue.onb.ac.at/scape-tb-evaluation/sb/ValidationOfArchivalContentAgainstAnInstitutionalPolicy/ValidateJPEG2000NewspapersUsingJpylyzer/md5sum_timings.txt

19 Appendix F3 – Evaluations in Research Datasets Testbed

19.1 raw2nexus migration large dataset big files

Experiment: raw2nexus Experiment at STFC

Evaluator(s)

Alastair Duncan, STFC

Evaluation points

Assessment of measurable points

Metric	Description	02/07/2014 maps 1 split 1	03/07/2014 maps 4 split 1	08/07/2014 maps 8 split 1	08/07/2014 maps 8 split 4
NumberOfObjectsPerHour	Number of objects processed in one hour	1711.29	2261.80	2522.29	
MaxObjectSizeHandledInGbytes	Max size of raw files	0.433	0.433	0.433	0.433
MinObjectSizeHandledInGbytes	Min size of raw files	6.19888e-6	6.19888e-6	6.19888e-6	6.19888e-6
ThroughputGbytesPerMinute	The throughput of data measured in Gybtes per minute	0.001	0.002	0.002	
ThroughputGbytesPerHour	The throughput of data measured in Gybtes per hour	0.082	0.109	0.122	
ReliableAndStableAssessment	Manual assessment on if the experiment performed reliable and stable	true	true	true	false
NumberOfFailedFiles	Number of files that failed in the workflow	1	1	1	16810
AverageRuntimePerItemInSeconds	The average processing time in seconds per item	0.475	0.630	0.700	
throughput in bytes per second	The throughput of data measured in bytes per second	24587665.126	32497311.333	32640084.059	
number of objects per second	Number of objects that can be processed per second	2.10	1.59	1.43	
max object size handled in bytes	Max size of raw files	465826304	465826304	465826304	465826304
min object size handled in bytes	Min size of raw files	6656	6656	6656	6656

Technical details

No taverna workflow just raw2nexus migration of 20130 files ranging in size from 6Kb up to 454.1Mb The single failure was for a raw file which was corrupt.

WebDAV

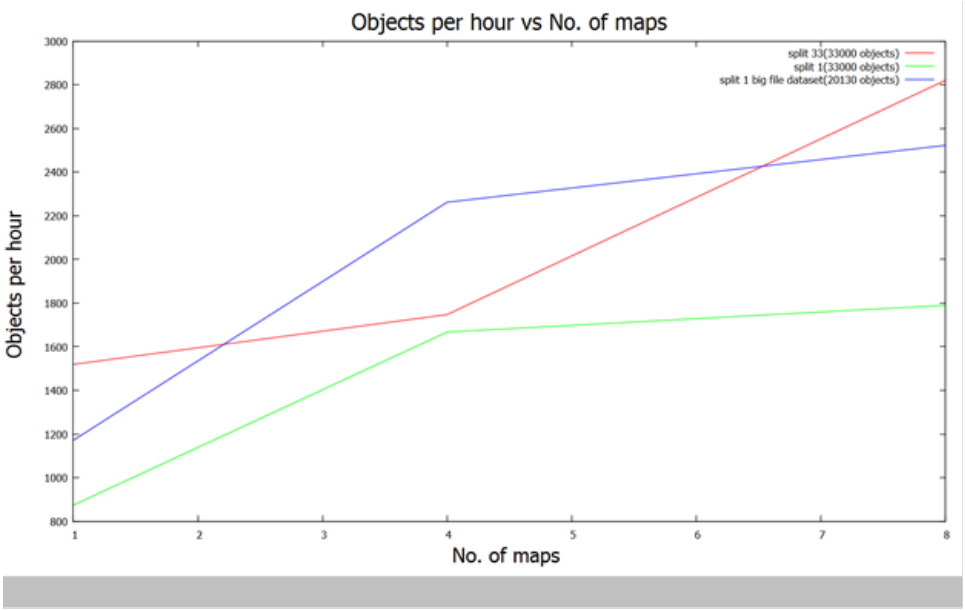
http://fue.onb.ac.at/scape-tb-evaluation/stfc/raw2nexus/largedatasetbigmap1split1
http://fue.onb.ac.at/scape-tb-evaluation/stfc/raw2nexus/largedatasetbigmap4split1/
http://fue.onb.ac.at/scape-tb-evaluation/stfc/raw2nexus/largedatasetbigmap8split1/
http://fue.onb.ac.at/scape-tb-evaluation/stfc/raw2nexus/largedatasetbigmap8split4/

Evaluation notes

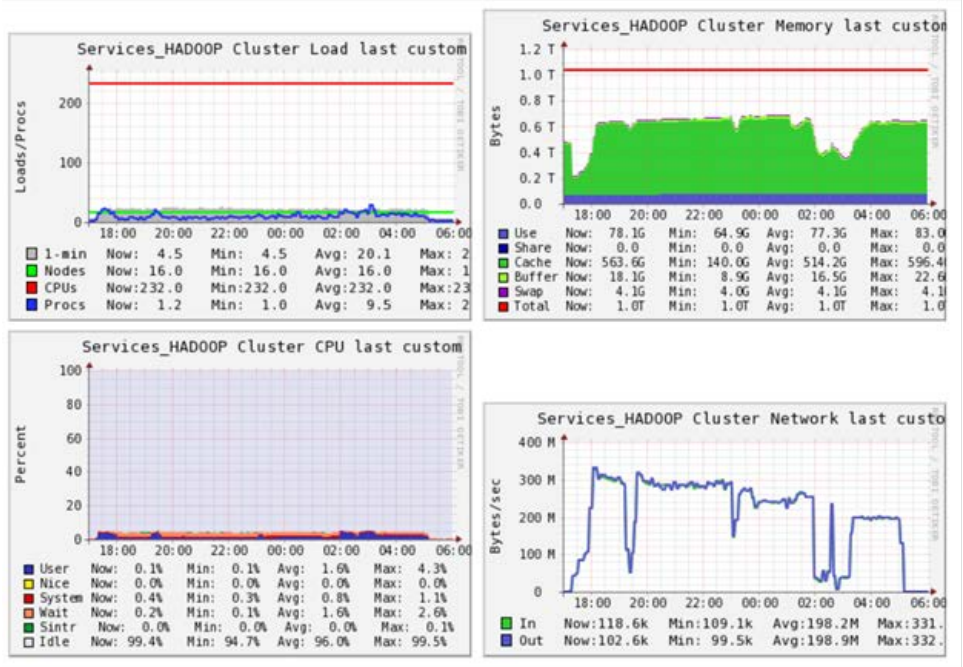
The larger files failed with Task attempt_201407021401_0001_m_000064_0 failed to report status for 600 seconds. Killing! These were then retried and were all successful except for the test

08/07/2014 maps 8 split 4. The failures were tried 4 times, after 4 failures and then killed after a number of such failures the whole job was killed with 830 completed tasks out of the 5033 tasks.

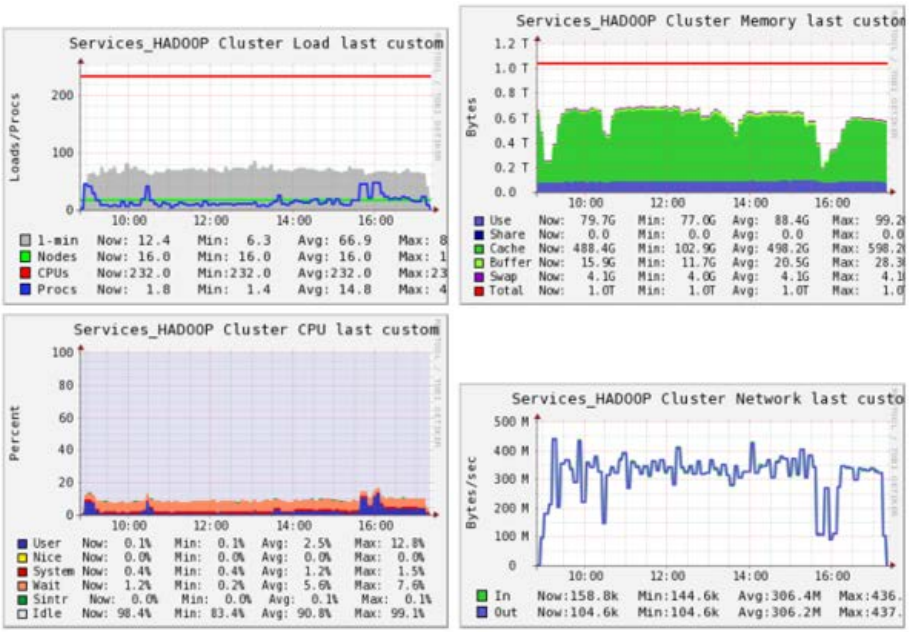
The blue trace on the graph below shows the executions of the large dataset 0.97Tb in size made up from 20130 raw files ranging in size from 6Kb to 456Mb. It does not show the execution of 8 maps with a split of 4 as this did not complete successfully.



The Ganglia screenshot below shows the experiments with the Maps 1 split 1. It can be seen that the machine was not overloaded although there is some orange showing on the CPU screen indicating that the CPU is undergoing some wait time which shows that Input Output (IO) is blocking somewhere.



This next graph shows the same dataset but with 4 maps and a split of 1. The system is not overloaded but there is a little more orange wait time showing and the 1 minute load has increased.

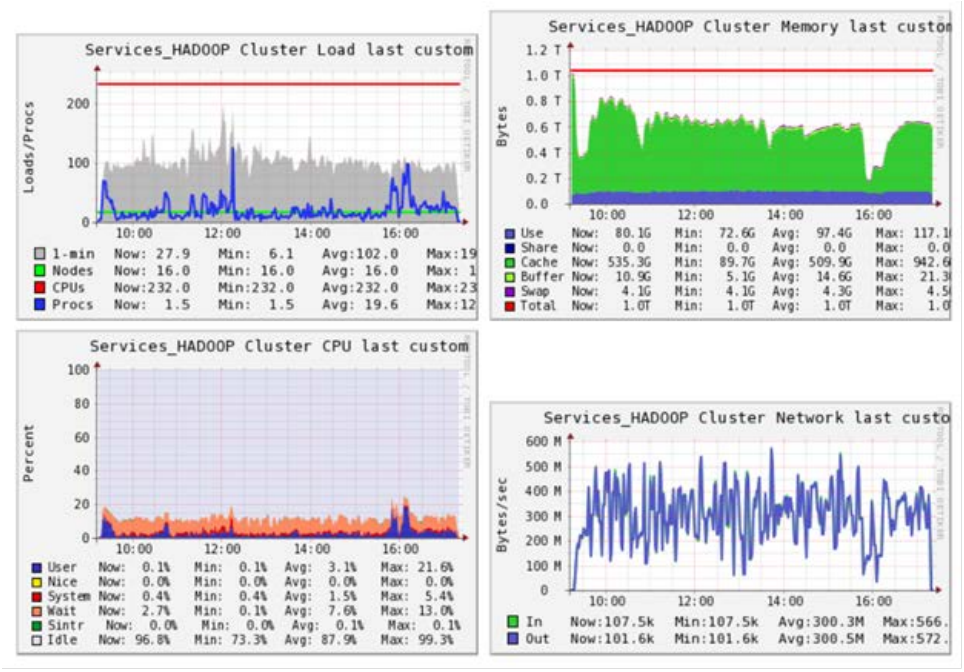


Around 100 of the tasks failed due to overrunning the 10 minute cap on run time but they were rescheduled and ran successfully to completion. An example of this is below.

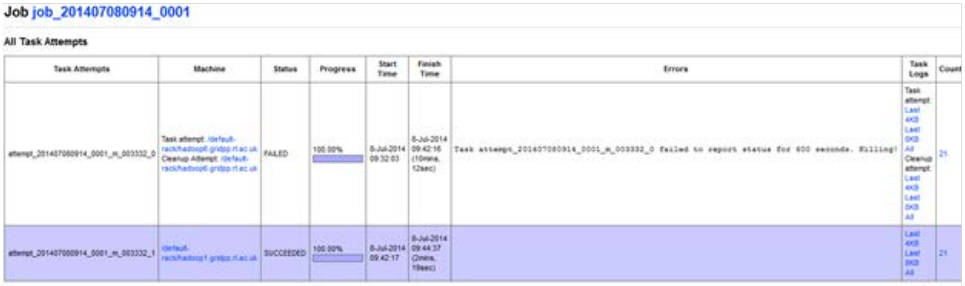

task_201407030853_0001_m_005695 attempts for 0001

Task Id	Start Time	Finish Time	Host	Error	Task Logs
attempt_201407030853_0001_m_005695_0	3/07 11:08:52	3/07 11:19:07 (10mins, 15sec)	hadoop7.gridpp.rl.ac.uk	Task attempt_201407030853_0001_m_005695_0 failed to report status for 600 seconds. Killing!	Last 4KB Last 8KB All
attempt_201407030853_0001_m_005695_1	3/07 11:19:54	3/07 11:22:46 (2mins, 52sec)	/default-rack/hadoop8.gridpp.rl.ac.uk		Last 4KB Last 8KB All

With the same data set and settings of 8 maps and a split of one, it can be seen that there is significantly higher 1 minute load and much more wait time even though the system is not anywhere close to capacity, the amount of CPU being used is really very small.



Over 1000 of the tasks failed due to overrunning but were rescheduled and ran successfully.



With a map of 8 and a split of 4 on the same dataset only 830 of the 5033 tasks completed successfully.

Hadoop job_201407080914_0002 on hadoop2

User: ad43
Job Name: tomar-1.4.1-SNAPSHOT-jar-with-dependencies.jar
Job File: hdfs://hadoop2.gridpp.rl.ac.uk/dfs/tmp/mapred/staging/ad43/.staging/job_201407080914_0002/job.xml
Submit Host: hadoop8.gridpp.rl.ac.uk
Submit Host Address: 130.246.181.179
Job-ACLs: All users are allowed
Job Setup: Successful
Status: Failed
Failure Info:NA
Started at: Tue Jul 08 17:21:37 BST 2014
Failed at: Tue Jul 08 18:29:25 BST 2014
Failed in: 1hrs, 7mins, 48sec
Job Cleanup: Successful
Black-listed TaskTrackers: 2

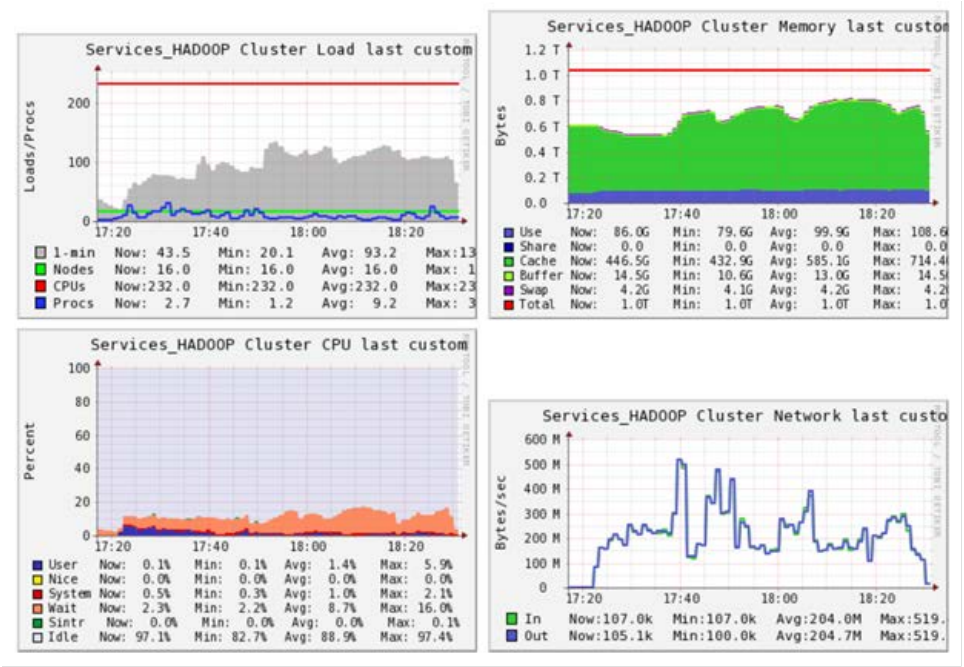
Kind	% Complete	Num Tasks	Pending	Running	Complete	Killed	Failed/Killed Task Attempts
map	100.00%	5033	0	0	830	4203	192 / 63
reduce	100.00%	1	0	0	0	1	0 / 1

Job job_201407080914_0002

All Task Attempts

Task Attempts	Machine	Status	Progress	Start Time	Finish Time	Errors	Task Logs	Counters
attempt_201407080914_0002_m_000833_0	/default-rack/hadoop3.gridpp.rl.ac.uk Cleanup Attempt: /default-rack/hadoop3.gridpp.rl.ac.uk	FAILED	100.00%	8-Jul-2014 17:39:58	8-Jul-2014 17:46:11 (10mins, 12sec)	Task attempt_201407080914_0002_m_000833_0 failed to report status for 600 seconds. Killing!	Task attempt: Last 4KB Last 8KB All Cleanup attempt: Last 4KB Last 8KB All	21
attempt_201407080914_0002_m_000833_1	/default-rack/hadoop2.gridpp.rl.ac.uk Cleanup Attempt: /default-rack/hadoop2.gridpp.rl.ac.uk	FAILED	100.00%	8-Jul-2014 17:46:15	8-Jul-2014 17:57:43 (11mins, 27sec)	Task attempt_201407080914_0002_m_000833_1 failed to report status for 600 seconds. Killing!	Task attempt: Last 4KB Last 8KB All Cleanup attempt: Last 4KB Last 8KB All	21
attempt_201407080914_0002_m_000833_2	/default-rack/hadoop8.gridpp.rl.ac.uk Cleanup Attempt: /default-rack/hadoop8.gridpp.rl.ac.uk	FAILED	100.00%	8-Jul-2014 17:57:46	8-Jul-2014 18:13:17 (15mins, 30sec)	Task attempt_201407080914_0002_m_000833_2 failed to report status for 600 seconds. Killing!	Task attempt: Last 4KB Last 8KB All Cleanup attempt: Last 4KB Last 8KB All	21
attempt_201407080914_0002_m_000833_3	/default-rack/hadoop6.gridpp.rl.ac.uk	KILLED	100.00%	8-Jul-2014 18:14:38	8-Jul-2014 18:29:19 (14mins, 41sec)		Last 4KB Last 8KB All	21

After 3 attempts the task is killed off and not rescheduled. There is a similar loading on the Hadoop system while the task was running but it runs for a much shorter time due to the majority of tasks being killed off.


Services_HADOOP Cluster Load last custom
Loads/Procs
200
100
0
17:20
17:40
18:00
18:20
1-min Now: 43.5 Min: 20.1 Avg: 93.2 Max:13
Nodes Now: 16.0 Min: 16.0 Avg: 16.0 Max: 1
CPUs Now:232.0 Min:232.0 Avg:232.0 Max:23
Procs Now: 2.7 Min: 1.2 Avg: 9.2 Max: 3
Services_HADOOP Cluster Memory last custom
Bytes
1.2 T
1.0 T
0.8 T
0.6 T
0.4 T
0.2 T
0.0
17:20
17:40
18:00
18:20
Use Now: 86.0G Min: 79.6G Avg: 99.9G Max: 108.6
Share Now: 0.0 Min: 0.0 Avg: 0.0 Max: 0.0
Cache Now: 446.5G Min: 432.9G Avg: 585.1G Max: 714.4
Buffer Now: 14.5G Min: 10.6G Avg: 13.0G Max: 14.5
Swap Now: 4.2G Min: 4.1G Avg: 4.2G Max: 4.2
Total Now: 1.0T Min: 1.0T Avg: 1.0T Max: 1.0
Services_HADOOP Cluster CPU last custom
Percent
100
80
60
40
20
0
17:20
17:40
18:00
18:20
User Now: 0.1% Min: 0.1% Avg: 1.4% Max: 5.9%
Nice Now: 0.0% Min: 0.0% Avg: 0.0% Max: 0.0%
System Now: 0.5% Min: 0.3% Avg: 1.0% Max: 2.1%
Wait Now: 2.3% Min: 2.2% Avg: 8.7% Max: 16.0%
Sintr Now: 0.0% Min: 0.0% Avg: 0.0% Max: 0.1%
Idle Now: 97.1% Min: 82.7% Avg: 88.9% Max: 97.4%
Services_HADOOP Cluster Network last custo
Bytes/sec
600 M
500 M
400 M
300 M
200 M
100 M
0
17:20
17:40
18:00
18:20
In Now:107.0k Min:107.0k Avg:204.0M Max:519.
Out Now:105.1k Min:100.0k Avg:204.7M Max:519.

19.2 raw2nexus migration large dataset copied from small dataset

Experiment: raw2nexus Experiment at STFC

Evaluator(s)

Alastair Duncan, STFC

Evaluation points

Metric	Description	19/06/2014 Maps per node 4 Split 1	20/06/2014 Maps per node 8 Split 1	23/24/06/2014 Maps per Node 1 split 1	25/06/2014 Maps per Node 1 split 33	26/06/2014 Maps per Node 4 split 33	27/06/2014 Maps per Node 8 split 33
NumberOfObjectsPerHour	Number of objects processed in one hour	1667.09	1789.48	874.16	1518.95	1746.88	2820.24
MaxObjectSizeHandledInGbytes	Max size of raw files	0.16689453	0.16689453	0.16689453	0.16689453	0.16689453	0.16689453
MinObjectSizeHandledInGbytes	Min size of raw files	2.24113e-5	2.24113e-5	2.24113e-5	2.24113e-5	2.24113e-5	2.24113e-5
ThroughputGbytesPerMinute	The throughput of data measured in Gybtes per minute	0.856	0.919	0.449	0.780	0.897	1.45
ThroughputGbytesPerHour	The throughput of data measured in Gybtes per hour	51.379	55.151	26.94	46.81	53.84	86.91
ReliableAndStableAssessment	Manual assessment on if the experiment performed reliable and stable	true	true	true	true	true	true
NumberOfFailedFiles	Number of files that failed in the workflow	1000	1000	1000	1000	1000	1000
AverageRuntimePerItemInSeconds	The average processing time in seconds per item	2.16	2.01	4.12	2.37	2.06	1.28
throughput in bytes per second	The throughput of data measured in bytes per second	15324467.037	16449541.634	8035585.716	13962718.892	16057937.712	25924702.545
number of objects per second	Number of objects that can be processed per second	0.46	0.49	0.24	0.42	0.48	0.78
max object size handled in bytes	Max size of raw files	160312320	160312320	160312320	160312320	160312320	160312320
min object size handled in bytes	Min size of raw files	24064	24064	24064	24064	24064	24064

WebDAV logs and configuration

http://fue.onb.ac.at/scape-tb-evaluation/stfc/raw2nexus/largedatasetmap8split1/
http://fue.onb.ac.at/scape-tb-evaluation/stfc/raw2nexus/largedatasetmap4split1/

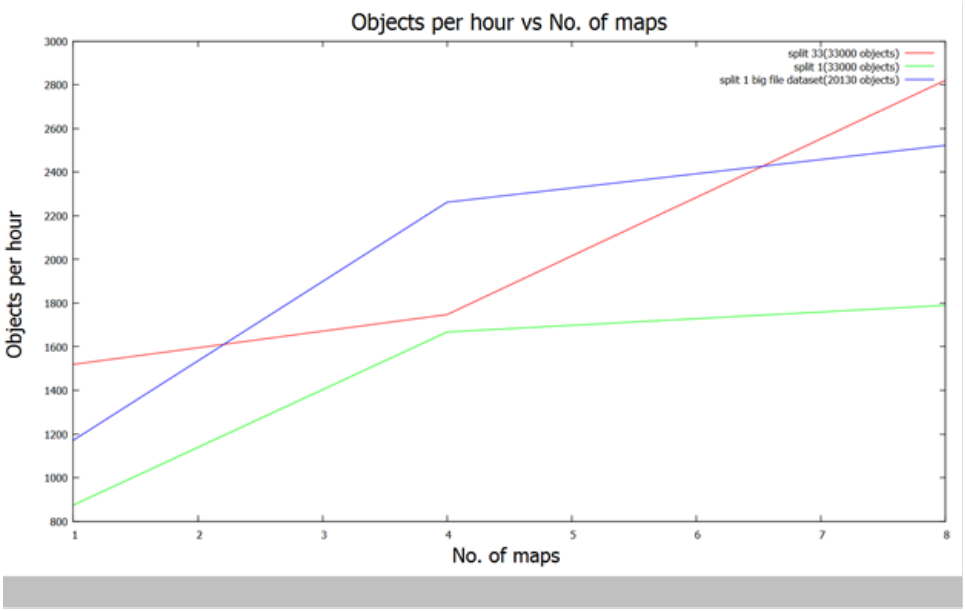
http://fue.onb.ac.at/scape-tb-evaluation/stfc/raw2nexus/largedatasetmap1split1/
http://fue.onb.ac.at/scape-tb-evaluation/stfc/raw2nexus/largedatasetmap1split33/
http://fue.onb.ac.at/scape-tb-evaluation/stfc/raw2nexus/largedatasetmap4split33/
http://fue.onb.ac.at/scape-tb-evaluation/stfc/raw2nexus/largedatasetmap8split33/

Evaluation notes

Timings for creating the copied dataset and generating the input files for ToMaR are not included in the results above.

There are 1000 failed executions because one of the test set (33 files) of files has a metadata file missing. The data set is 1000 copies of the 33 files.

In the graph below a direct comparison can be made between the red and green experiment results these were executed over the same data set but with a split of either 33 or 1. Three runs were done with the number of maps being 1, 4 and 8. This shows much larger gains in performance when the split is increased and the number of maps is 8. The blue graph shows the execution times of the other large dataset with a split of 1 which is a similar shape to the green graph. When the split was increased to 4 on this dataset then the experiment did not complete successfully.



19.3 raw2nexus small dataset evaluation

Experiment: raw2nexus Experiment at STFC

Evaluator(s)

Alastair Duncan, STFC

Evaluation points

Baseline

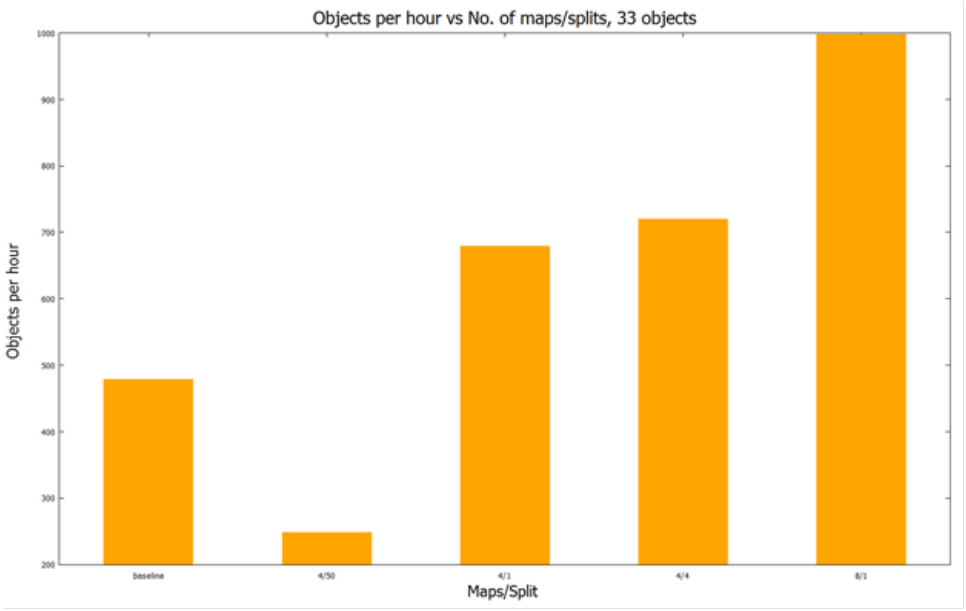
Metric	Description	Metric baseline	20/06/2014 Maps per node 8 Split 1	20/06/2014 Maps per node 4 Split 4	20/06/2014 Maps per node 4 Split 50
NumberOfObjectsPerHour	Number of objects processed in one hour	479.3	998.32	720	238.55
MaxObjectSizeHandledInGbytes	Max size of raw files	0.16689453	0.16689453	0.16689453	0.16689453
MinObjectSizeHandledInGbytes	Min size of raw files	2.24113e-5	2.24113e-5	2.24113e-5	2.24113e-5
ThroughputGbytesPerMinute	The throughput of data measured in Gybtes per minute	0.246	0.513	0.370	0.123
ThroughputGbytesPerHour	The throughput of data measured in Gybtes per hour	14.764	30.768	22.190	7.352
ReliableAndStableAssessment	Manual assessment on if the experiment performed reliable and stable	true	true	true	true
NumberOfFailedFiles	Number of files that failed in the workflow	1	1	1	1
AverageRuntimePerItemInSeconds	The average processing time in seconds per item	7.51	3.60	5.0	15.09
throughput in bytes per second	The throughput of data measured in bytes per second	4403436.169	9176908.992	6618498.000	2191875.843
number of objects per second	Number of objects that can be processed per second	0.13	0.28	0.20	0.07
max object size handled in bytes	Max size of raw files	24064	24064	24064	24064
min object size handled in bytes	Min size of raw files	160312320	160312320	160312320	160312320

WebDAV

http://fue.onb.ac.at/scape-tb-evaluation/stfc/raw2nexus/smalldatasetmap4split50/
http://fue.onb.ac.at/scape-tb-evaluation/stfc/raw2nexus/smalldatasetmap4split4/
http://fue.onb.ac.at/scape-tb-evaluation/stfc/raw2nexus/smalldatasetmap8split1/

Evaluation notes

Timings for moving the data onto HDFS and generating the input files for ToMaR are not included in any of the evaluations. There was one failure due to missing metadata file for one of the test files.



In the graph above the baseline used Taverna running on a single node of the Hadoop cluster but not using Hadoop. As the dataset comprised 33 objects (raw files) having a split of 50 ensures that all of the files will be run on a single node within a single mapper, it can be compared directly to the base line and it can be seen that the Hadoop system has overheads. The other maps/splits were experiments to see what effect adjusting the configuration parameters would have to the overall execution time.

Discussion

The experiment results and monitoring data from Ganglia shows that the migration process is Input Output (IO) bound. Processing of files which are small < 129Mb is possible but when larger files are processed the system struggles if the tasks are run in parallel. The users of the Hadoop system have control over the number of splits that are made for a job but only a systems administrator can modify the number of mappers that are available per node. The file size of the test datasets are quite modest with the largest at < 500Mb and these are relatively small files that can be produced by the systems at ISIS and Diamond. The size of files being produced are often in the 10s of Gb. Moving the files onto the specialised HDFS and then off again after processing is very time consuming. The design of the Hadoop system and HDFS is to execute the application near to the data so that data is not moved about. Once the data is on HDFS is should not be moved and then it becomes possible to more efficiently process and reprocess the data with the fault tolerance of the HDFS system coming in to its own. This is the model that is used here at RAL for the JASMIN and SCARF clusters. The atmospheric data that is processed on the JASMIN system took months to transfer onto the Jasmin data storage. Moving this on to a system and then off again once it has been processed is not a viable option. A similar consideration has to be made with the ISIS and Diamond data, the cost in terms of time in moving the data onto HDFS is prohibitive and that coupled with the inability of HDFS to read and write directly to the NeXus and HDF5 file formats means that the Hadoop system is not suitable for the task of migrating from raw to NeXus formats at STFC.

19.4 GeoLint Evaluation

Experiment: GeoLint Experiment

Evaluator(s)

William Palmer, BL

Evaluation points

Assessment of measurable points

Metric	Metric goal	Metric baseline (14th August 2014)	8th August 2014
TotalRuntime		22:31:31 (hh:mm:ss)	08:32:53 (hh:mm:ss)
TotalObjects		2,602,737 [1]	2,602,737 [1]
NumberOfObjectsPerHour		115547	304482
ThroughputGbytesPerHour		46.83	123.4 [2]
ReliableAndStableAssessment	TRUE	TRUE	TRUE
NumberOfFailedFiles	0	0	0
NumberOfFailedFilesAcceptable	-	TRUE	TRUE
		[4]	[3]
		Using 8 simultaneous map slots/nodes	Using 28 simultaneous map slots/nodes

Note 1: This is a subset of the main geospatial dataset, totalling approximately 1055GB. These files took 24h37m to be copied from a NAS into HDFS

Note 2: This is 35MB/s (below the benchmarked max I/O rate of between 74-146MB/s - Benchmarking Hadoop installations[184])

Note 3: Problems were detected in 2371 GML and NTF files, 2214 of which were false positives and GeoLint has subsequently been modified to correctly parse those files. Of the remaining 157 files there were some GML files that failed validation and four NTF files that need to be checked. A positive result, with 157 files to review instead of 2.6 million, some of the GML files may have shared issues.

Note 4: The software was modified to reduce false positives for this run. The same 157 files as in note 3 were identified, along with 602 of the previous 2214 "false positives" that require further investigation.

Technical details

The full path to the input files is listed in a text file, in HDFS. 2000 lines are passed to each mapper.

Evaluation notes

An attempt was made to use SequenceFiles to ensure data locality, an issue was encountered due to the variation in sizes of data - see NTF/ISO sizes, which was resolved. However, when creating the SequenceFile there was heap size issues/exception after a very long execution. As the creation time for the SequenceFile was significantly longer than that of just copying the data into HDFS and processing it, that approach was abandoned. Additionally, the data will not be stored for long term preservation in SequenceFiles.

[184] http://wiki.opf-labs.org/display/SP/Benchmarking+Hadoop+installations

Conclusion

This experiment has a heterogeneous dataset of many different file types, with all files identified (mimetype) and checksummed, with the NTF and GML files also being validated. The final result of this evaluation, with a runtime of eight and a half hours, is a reasonable time for performing all those tasks. However, the files are not stored permanently in HDFS and it took 24 hours for them to be copied into HDFS. It is worth noting that although a number of the files are small (~214kb average), we were still not hitting the I/O limits of the cluster - see note 2 above. When there are more mappers processing data we do not see linear growth, although we still see a decrease in wall-clock time (8 mappers @ 46.84GB/h would scale to 163.94GB/h if it was linear growth, instead of the recorded 123.4GB/h)

20 Appendix F4 – Evaluations in Data Center Testbed

20.1 Average time evaluation

Experiment: Scene reconstruction

Evaluator(s)

Ondrej Klima, BUT

Evaluation points

Assessment of measurable points

- Dependency of average time on a number of used nodes from May 8 - May 16, 2014

Metric	Description	2 nodes	5 nodes	8 nodes	11 nodes	14 nodes	17 nodes
average JPEG time per item	time of loading JPEG file from FS in seconds	0,0578872	0,07268144	0,05151228	0,05662498	0,06022247	0,06061896
average SIFT time per item	time of extracting features from images in seconds	1,791039	1,826097	1,830387	1,802353	1,816661	1,826933
average matching time per item	time of matching extracted features in seconds	0,1381299	0,1410788	0,1461478	0,1433206	0,144336	0,1452223

Technical details

Evaluation notes

20.2 Evaluation of memory consumption

Experiment: Video annotation and geo localization

Evaluator(s)

Ondrej Klima, BUT

Evaluation points

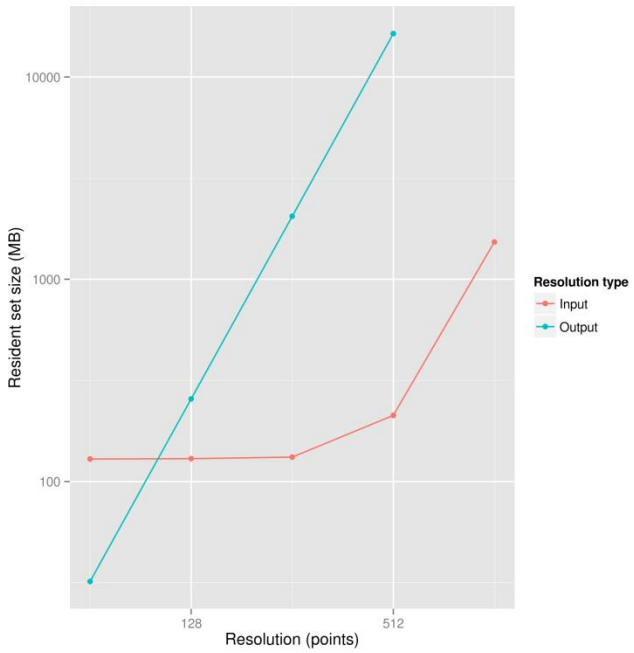
Assessment of measurable points

Dependency of memory consumption on **input** resolution from **July 7 - July 25, 2014**

Metric	Description	resolution 64 points	128 points	256 points	512 points	1024 points
maximum peak memory used	Maximum of the peak memory used in processing, measured in megabytes.	129.180	129.727	132.007	212.263	129.180
max object size handled in bytes	Maximum size of PFM file, measured in megabytes.	33.82	33.82	33.82	33.82	33.82
min object size handled in bytes	Minimum size of input file in megabytes.	0.28	0.28	0.28	0.28	0.28

Dependency of memory consumption on **output** resolution from **July 7 - July 25, 2014**

Metric	Description	resolution 64 points	128 points	256 points	512 points
maximum peak memory used	Maximum of the peak memory used in processing, measured in megabytes.	32.058	256.005	2048.005	16384.006
max object size handled in bytes	Maximum size of PFM file, measured in megabytes.	33.82	33.82	33.82	33.82
min object size handled in bytes	Minimum size of input file in megabytes.	0.28	0.28	0.28	0.28



Technical details

Values of "maximum peak memory used" metric were extracted from /proc/self/status file, concretely the field "RSS" was used. The resident set size is the amount of a process's memory that is actually held in RAM.

Evaluation notes

20.3 Performance evaluation

Experiment: Video annotation and geo localization

Evaluator(s)

Ondrej Klima, BUT

Evaluation points

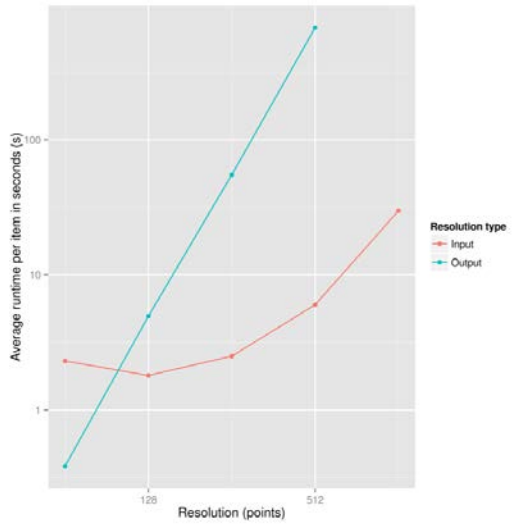
Assessment of measurable points

Dependency of memory consumption on **input** resolution from **July 7 - July 25, 2014**

Metric	Description	resolution 64 points	128 points	256 points	512 points	1024 points
average runtime per item in seconds	The average processing time in seconds per item.	2.3	1.8	2.5	6.0	29.8
reliable and stable assessment	Manual assessment on if the experiment performed reliable and stable.	true	true	true	true	true
number of failed files	Number of files that failed in the workflow.	0	0	0	0	0

Dependency of memory consumption on **output** resolution from **July 7 - July 25, 2014**

Metric	Description	resolution 64 points	128 points	256 points	512 points
average runtime per item in seconds	The average processing time in seconds per item.	0.4	4.9	55.0	673.5
reliable and stable assessment	Manual assessment on if the experiment performed reliable and stable.	true	true	true	true
number of failed files	Number of files that failed in the workflow.	0	0	0	0



Technical details

The actual wall time was measured. The **boost** library was used for time measurement.

Evaluation notes

20.4 Precision of alignment evaluation

Experiment: Video annotation and geo localization

Evaluator(s)

Ondrej Klima, BUT

Evaluation points

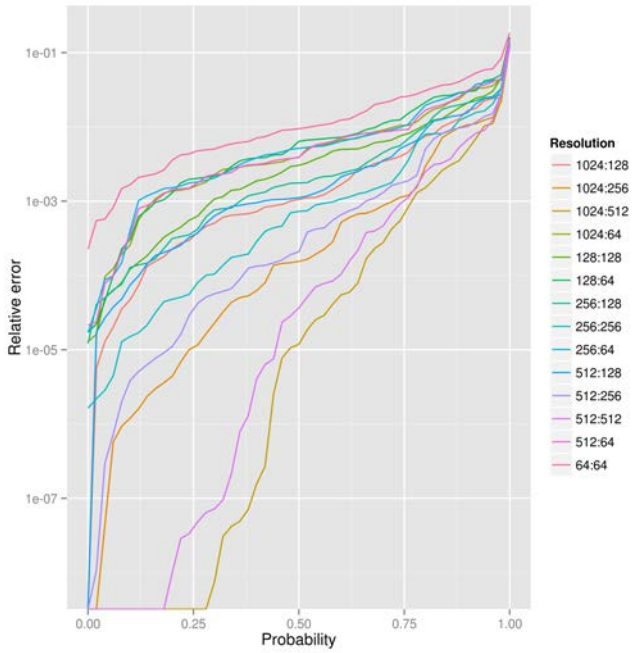
Assessment of measurable points

- Dependency of relative error on both input and output resolution from July 7 - July 25, 2014

Metric	Description	resolution 64:64 points	128:64 points	256:64	512:64 points	1024:64 points	128:128 points	256:128 points
median relative error	Alignment relative error of median measurement	0.009438515	0.00638008	0.0051446	0.00387144	0.00390434	0.00303662	0.00175625
reliable and stable assessment	Manual assessment on if the experiment performed reliable and stable.	true	true	true	true	true	true	true

- Dependency of memory consumption on output resolution from July 7 - July 25, 2014

Metric	Description	resolution 512:128 points	1024:128 points	256:256 points	512:256 points	1024:256 points	512:512 points	1024:512
median relative error	Alignment relative error of median measurement	0.001114905	0.00105649	0.00072851	0.000208288	0.000154354	3.64948e-05	1.182685e-05
reliable and stable assessment	Manual assessment on if the experiment performed reliable and stable.	true	true	true	true	true	true	true


1e-01
1e-03
1e-05
1e-07
Relative error
0.00
0.25
0.50
0.75
1.00
Probability
Resolution
1024:128
1024:256
1024:512
1024:64
128:128
128:64
256:128
256:256
256:64
512:128
512:256
512:512
512:64
64:64

20.5 Evaluation of DICOM data access

Experiment: Performance tests for accessing medical data

Evaluator(s)

Tomasz Hoffmann, PSNC

Evaluation points

The objective of this evaluation task was to test the efficiency of one instance of the DICOM data provider available within the Medical Data Center platform at PSNC. In order to measure download performance the metric named number of objects per second has been used. Other metrics gathered during the tests provide additional information and include throughput in bytes per second, max object size handled in bytes and min object size handled in bytes. The metric goal has not been determined because there are currently no related requirements defined by the users. This evaluation is composed of 7 tests from which the first 5 are executed using one computer (client) and the last two are executed using two computers (the reason for split into two computers is that one computer can efficiently test MDC platform with maximum 20 concurrent threads).

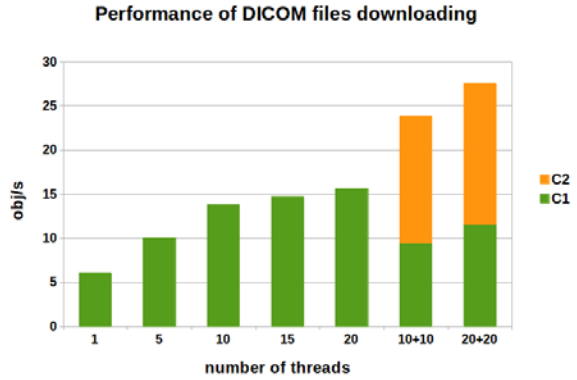
Assessment of measurable points

Metric	Description	July 1, 2014 [Test 1]	July 4, 2014 [Test 2]	July 4, 2014 [Test 3]	July 4, 2014 [Test 4]	July 4, 2014 [Test 5]	July 7, 2014 [Test 6]	July 7, 2014 [Test 7]
number of objects per second	number of downloaded DICOM files* per second	6.08 [obj/s]	10.01 [obj/s]	13.82 [obj/s]	14.74 [obj/s]	15.59 [obj/s]	9.50 + 14.36 [obj/s]	11.53 + 16.06 [obj/s]
throughput in bytes per second	_bytes per second_	2 854 804 [bytes/s]	4 644 810 [bytes/s]	6 413 019 [bytes/s]	6 839 388 [bytes/s]	7 233 589 [bytes/s]	4 410 289 + 6 684 607 [bytes/s]	5 350 649 + 7 489 928 [bytes/s]
max object size handled in bytes	max size of DICOM file	1 827 084 [bytes]	1 827 084 [bytes]	1 827 084 [bytes]	1 827 084 [bytes]	1 827 084 [bytes]	1 827 084 [bytes]	1 827 084 [bytes]
min object size handled in bytes	min size of DICOM file	44 090 [bytes]	44 090 [bytes]	44 090 [bytes]	44 090 [bytes]	44 090 [bytes]	44 090 [bytes]	44 090 [bytes]

Note: *an object is a single DICOM file

Visualisation of results

The chart below shows relation between download speed (in objects per second) and the number of download threads used in the test. First 5 tests show that a single client can reach up to approx. 16 objects per second, which is in fact the limit of the client computer, as when using two client computers it was possible to retrieve around 27 objects per second (with 40 threads).



Tables below provide additional statistics. Table 1. presents a summary of all files and series used in the test (please note that a single series is composed of multiple DICOM files related to a single patient's examination, e.g. CT scan, RTG). Table 2. and Table 3 presents overall statistics related to series. Table 4. and Table 5. presents overall statistics related to files (objects).

Table 1. Overall statistics of the tests

Parameter	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7
Number of threads	1	5	10	15	20	10 + 10	20 + 20
Number of series	263	263	263	263	263	262 + 262	262 + 262
Number of DICOM files	27 053	27 053	27 053	27 053	27 053	27 051 + 25 881	27 051 + 25 761
Total size of DICOM files	12 550 [MB]	12 550 [MB]	12 550 [MB]	12 550 [MB]	12 550 [MB]	12 547 + 12 105 [MB]	12 547 + 12 013 [MB]
Total download time	4 448 [s]	2 702 [s]	1 957 [s]	1 835 [s]	1 735 [s]	2 845 + 1 802 [s]	2 345 + 1 604 [s]
Average serie download speed (series per second)	0.05 [series/s]	0.09 [series/s]	0.13 [series/s]	0.14 [series/s]	0.15 [series/s]	0.09 + 0.14 [series/s]	0.11 + 0.16 [series/s]

Table 2. Overall statistics related to series (maximum values)

Parameter	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7
Maximum series size	233 [MB]	233 [MB]	233 [MB]	233 [MB]	233 [MB]	233 + 233 [MB]	233 + 233 [MB]
Maximum series download time	73 [s]	165 [s]	223 [s]	303 [s]	373 [s]	291 + 233 [s]	463 + 383 [s]

Table 3. Overall statistics related to series (minimum values)

Parameter	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7
Minimum series size	0.04 [MB]	0,04 [MB]	0,04 [MB]	0,04 [MB]	0,04 [MB]	0.04 + 0.04 [MB]	0.04 + 0.04 [MB]
Minimum series download time	0.09 [s]	0.11 [s]	0.12 [s]	0.13 [s]	0.19 [s]	0.17 + 0.10 [s]	0.22 + 0.14 [s]

Table 4. Overall statistics related to files (maximum values)

Parameter	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7
Maximum file size	1.82 [MB]	1.82 [MB]	1.82 [MB]	1.82 [MB]	1.82 [MB]	1.82 + 1.82 [MB]	1.82 + 1.82 [MB]
Maximum file download time	5.24 [s]	5.63 [s]	5.53 [s]	5.59 [s]	7.36 [s]	2.21 + 5.27 [s]	4.32 + 5.49 [s]

Table 5. Overall statistics related to files (minimum values)

Parameter	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7
Minimum file size	0.04 [MB]	0.04 [MB]	0.04 [MB]	0.04 [MB]	0.04 [MB]	0.04 + 0.04 [MB]	0.04 + 0.04 [MB]
Minimum file download time	0.09 [s]	0.09 [s]	0.10 [s]	0.10 [s]	0.11 [s]	0.12 + 0.10 [s]	0.16 + 0.10 [s]

Raw log files

All log files created during the evaluation are available online at:

https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/dicom-test-results/access/test01_01_07_2014.log
https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/dicom-test-results/access/test02_04_07_2014.log
https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/dicom-test-results/access/test03_04_07_2014.log
https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/dicom-test-results/access/test04_04_07_2014.log
https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/dicom-test-results/access/test05_04_07_2014.log
https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/dicom-test-results/access/test06_07_07_2014.log
https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/dicom-test-results/access/test06_07_07_2014.log

Technical details

Workflow

The experiment was composed of the following steps:

1. prepare data set of DICOM files (over ~10GB, which is amount of data produced by WCPT hospital in one day),
2. set the number of downloading threads [dicomClient.py]
3. download data from MDC server, and log on information about: [dicomClient.py]
    a. size of received DICOM file
    b. time of receiving data for each DICOM file
4. parse a log file in order to evaluate metrics (for metrics description please see previous point) [dicomClientResultParser.py]

Scripts used to execute evaluation

https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/python/DICOM-tests/dicomClient.py
https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/python/DICOM-tests/dicomClientResultParser.py

Execution commands

```
python dicomClient.py
python dicomClientResultParser.py dicomClientStats.txt dicomClientTime.txt
test08_results.txt
```

Note: Scripts are prepared for Python in version 2.7.

20.6 Performance depending on search criteria

Experiment: Performance tests of the search function in the MDC portal

Evaluator(s)

Michał Kozak, PSNC

Evaluation points

The main goal of this evaluation was to test the performance of the education portal search function. The test was executed on the PSNC Hadoop cluster and WCPT medical dataset.

Assessment of measurable points

The mean query time is the mean of 10 iterations.

MDC handled one request at a time.

Metric	Description	July 24, 2014	July 29, 2014
MeanQueryTimeInSeconds	medical cases by ICD10 code	2.095	2.712
MeanQueryTimeInSeconds	medical cases by ICD9 code	9.345	12.660
MeanQueryTimeInSeconds	medical cases by patient's city	7.247	8.360
MeanQueryTimeInSeconds	medical cases by patient's sex	13.809	14.814
MeanQueryTimeInSeconds	medical cases by patient's age	5.320	5.734
MeanQueryTimeInSeconds	medical cases by visit's dates	5.160	6.296
MeanQueryTimeInSeconds	medical cases by laboratory tests	3.996	4.123
MeanQueryTimeInSeconds	medical cases by all of the above criteria	10.023	10.980

MDC handled ten requests at a time.

Metric	Description	July 24, 2014	July 29, 2014
MeanQueryTimeInSeconds	medical cases by ICD10 code	4.350	4.762
MeanQueryTimeInSeconds	medical cases by ICD9 code	11.219	12.008
MeanQueryTimeInSeconds	medical cases by patient's city	10.083	11.855
MeanQueryTimeInSeconds	medical cases by patient's sex	16.016	16.509
MeanQueryTimeInSeconds	medical cases by patient's age	8.205	8.688
MeanQueryTimeInSeconds	medical cases by visit's dates	5.836	8.231
MeanQueryTimeInSeconds	medical cases by laboratory tests	4.316	5.227
MeanQueryTimeInSeconds	medical cases by all of the above criteria	13.467	14.019

Technical details

A medical case is a list of hospital visits of one patient. A medical case satisfies criteria of a query when at least one visit of a patient satisfies them. MDC provides the following search options and an arbitrary conjunction of them:

- by ICD10 code - a visit satisfies this criteria when a patient has the ICD10 disease as an underlying or a concurrent during the visit
- by ICD9 code - a visit satisfies this criteria when a patient underwent the ICD9 medical procedure during the visit
- by patient’s city - a patient satisfies this criteria when the given city matches to the post code of the patient
- by patient’s sex - a patient satisfies this criteria when the given sex matches to the sex of the patient
- by patient’s age - a patient satisfies this criteria when he or she was in the given age at the time of discharge from the hospital (the age can be specified as an interval)
- by visit's dates - a visit satisfies this criteria when the given period intersects with the visit’s time

- by laboratory tests - a visit satisfies this criteria when all the given laboratory tests were carried out during the visit
- by all of the above criteria - conjunction of all above criteria

Please note that searching by patients' attributes is limited to those that remained after anonymization, namely: post code, sex and birth date.

In order to find and compose visits in medical cases two map-reduce jobs are executed. Afterwards, found medical cases are serialized to JSON and returned to the client. The execution time mainly depends on the amount of matching visits and patients. For example ICD9 codes of medical procedures are repeated for many visits. Naturally, the sex is repeated for patients. In the case of more than one criterion, conjunction takes time as well.

The first test (July 24, 2014) was performed when MDC contained information about 16 000 hospital visits.

The second test (July 29, 2014) was performed when MDC contained information about 19 000 hospital visits.

Link to the software that was used to test MDC: https://git.man.poznan.pl/stash/scm/scap/test-scripts/medd-provider-test

20.7 Evaluation of the age of patients treated in a given period

Experiment: Analysis of epidemiological situation across WCPT patients

Evaluator(s)

Tomasz Hofmann, PSNC

Evaluation points

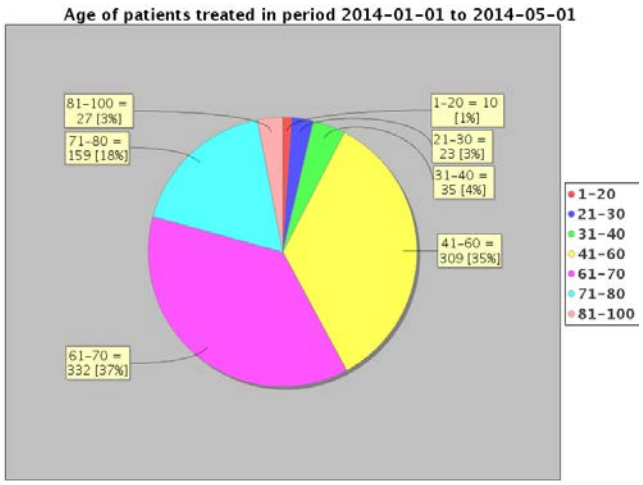
The main goal of this evaluation was to execute analysis on the medical data stored at Medical Data Center and obtain statistics on the age of patients treated at WCPT in a given period. The age intervals are the input parameters for analysis algorithm. Statistics were gathered using PSNC Hadoop Platform using the map-reduce approach. As the evaluation metric the number of objects per second has been selected (the object is defined as a single record in the HBase table, and each HBase table row stores information about the age of patient who visited WCPT hospital).

Assessment of measurable points

Metric	Description	July 21, 2014 [Test 1]	July 28, 2014 [Test 2]	July 30, 2014 [Test 3]
number of objects per second	number of records processed per second	2592 [obj/s]	2417 [obj/s]	1465 [obj/s]

Visualisation of results

The chart below presents results of analysis for Test 2. Colours indicate different age ranges for the patients who visited WCPT hospital (the exact age range is given in the middle-right part of the chart and on the chart itself). Each colour on the pie chart has related entry (note). Each entry is composed as follows: X-Y = Z [P], where X-Y is the age range (patients between age X and Y), Z is the number of patient's visits (indicated the number of visits for specified age range and analysed time period) and P is the percentage of the number of patient's visit in the overall context. An example can be an entry for yellow colour: 41-60 = 309 [35%] - it means that yellow represents percentage of patients (35%) in age between 41 and 60 (including) which visited WCPT hospital in the period 2014-01-01 - 2014-05-01.



Additional information

Table 1 presents processing time of each test executed in the evaluation. From the statistics in the table and measurable points it is visible that: a) the processing time depends on the number of records to be processed b) the more records to process the better performance is achieved (more

rows per second are processed). Table 2 and 3 provides additional statistics: processing times for map and reduce tasks respectively. It is visible that map task consumes most of the processing time, as the mappers are responsible for processing (reduce tasks calculate summary).

Table 1. Overall statistics

Parameter	Test 1	Test 2	Test 3
Analysed period	1.07.2012-1.07.2014	1.01.2013-31.12.2013	1.01.2014-1.05.2014
Processing time	65 [s]	61 [s]	7 [s]

Table 2. Statistics for map task

Parameter	Test 1	Test 2	Test 3
Processing time (for all records)	65 [s]	61 [s]	6 [s]
Number of records	167 893	148 207	9 462

Table 3. Statistics for reduce task

Parameter	Test 1	Test 2	Test 3
Processing time (for all records)	0,36 [s]	0,34 [s]	0,15 [s]
Number of records	8 259	7 744	592

Technical details

Workflow

The experiment is composed of the following steps (accordingly to MapReduce schema):

1. the map task [age.sh]:
    a. for each tuple in visits table:
        i. if visit belong to the given period, then calculate patients age and add into the context pair: Key=age, Value=visit_id
2. the reduce task [age.sh]:
    a. for each value of age aggregate all visits ids in hash set - in order to find out the number of different visits
    b. produce pair Key=age, Value=number of different visits (size of the hash set)
3. chart generation [age.sh]
4. additional statistics (see additional information section) are gathered by downloading and parsing log files [test.sh]

Scripts used to execute evaluation

https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/epidemic-jobs-tests/age.sh
https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/epidemic-jobs-tests/test.sh

Execution commands

```
./age.sh -hospital wcpit -admission 20140101 -discharge 20140501 -destination
tomek/tmp/age61 -intervals "1-20;21-30;31-40;41-60;61-70;71-80;81-100" -width 800 -
height 600
./test.sh age

where:
-admission : date of patient admission to hospital
-discharge : date of patient discharge from hospital
-destination : folder for hadoop job results (only one per job execution)
```

```
-width : width of the chart in pixels
-height : height of the chart in pixels
-intervals : patietnts age intervals

Important note: please change the -destination for each job execution.
```

Hadoop job

https://git.man.poznan.pl/stash/projects/SCAP/repos/mr-jobs/browse/epidemic-jobs/age

20.8 Evaluation of the average time of patient's visit for a given disease codes in a given time period

Experiment: Analysis of epidemiological situation across WCPT patients

Evaluator(s)

Tomasz Hofmann, PSNC

Evaluation points

The main goal of this evaluation was to execute analysis on the medical data stored at Medical Data Center and obtain statistics on the average time of visit for patients treated at WCPT in a given period and because of a specific disease (indicated by ICD10 codes). The period of time and ICD10 codes are the input parameters for analysis algorithm. Statistics were gathered using PSNC Hadoop Platform and the map-reduce approach. As the metric the number of objects per second was used (the number of records processed per second).

Assessment of measurable points

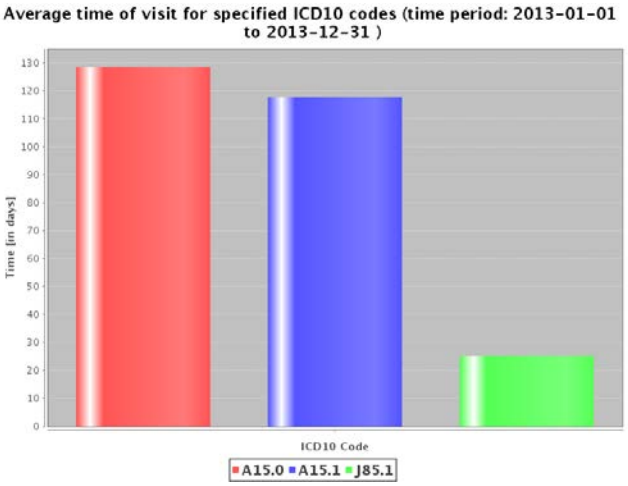
Metric	Description	July 21, 2014 [Test 1]	July 28, 2014 [Test 2]	July 31, 2014 [Test 3]
number of objects per second	number of records processed per second	2812 [obj/s]	2569 [obj/s]	1438 [obj/s]

Note: *as an object we proposed to use one scanned cell in Hbase table (one record)

Visualisation of results

The chart below presents results of the analysis for Test 2. Colours indicate different ICD10 disease codes that. Test has been performed for the patients who visited WCPT hospital between 1-01-2013 and 31-12-2013. Each column indicated the average time of patients visits. Descriptions of the ICD10 codes investigated in this analysis are as follows:

- A15.0 - Tuberculosis of lung, confirmed by sputum microscopy with or without culture
- A15.1 - Tuberculosis of lung, confirmed by culture only
- J85.1 - Abscess of lung with pneumonia



Additional information

Table 1 presents processing time of the whole job per test. Tables 2 and 3 provide information on the execution time and number of processed rows related to map and reduce tasks respectively. From

the statistics and measurable points it is visible that: a) the processing time depends on the number of records to be processed b) the more records to process the better performance is achieved (more rows per second are processed).

Table 1. Overall statistics

Parameter	Test 1	Test 2	Test 3
Analysed period	1.07.2012-1.07.2014	1.01.2013-31.12.2013	1.01.2014-01.05.2014
Processing time	59 [s]	57 [s]	7 [s]

Table 2. Statistics for map task

Parameter	Test 1	Test 2	Test 3
Processing time (for all records)	59 [s]	57 [s]	7 [s]
Number of records	165 903	146 748	9 294

Table 3. Statistics for reduce task

Parameter	Test 1	Test 2	Test 3
Processing time (for all records)	0,17 [s]	0,18 [s]	0,04 [s]
Number of records	898	642	83

Technical details

Workflow

The experiment is composed of the following steps (accordingly to the MapReduce schema) [casesaverage.sh]:

1. the map task,
    a. for each tuple in visits table:
        i. if icd10 code is in the set of given icd10 codes and if visit belong to the given period, then calculate the time of patients visits and add into the context pair: Key=icd10 code, Value=the length of the visit
2. the reduce task [casesaverage.sh]:
    a. for each icd10 code accumulate the length of visits and then divide by the number of visits
    b. produce pair Key=icd10 code, Value=average length of the visit
3. statistics are gathered by downloading and parsing log files [test.sh]

Scripts used to execute evaluation

https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/epidemic-jobs-tests/jobs-scripts/casesaverage.sh
https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/epidemic-jobs-tests/jobs-scripts/test.sh

Execution commands

```
./casesaverage.sh -admission 20140101 -destination tomek/tmp/casesaverage01 -discharge
20140501 -hospital wcpit -icd10s J85.1 -icd10s A15.0 -icd10s A15.1 -width 800 -height
600
./test.sh caseAvr

where:
-admission : date of patient admission to hospital
```

```
-discharge : date of patient discharge from hospital
-destination : folder for hadoop job results (only one per job execution)
-width : width of the chart in pixels
-height : height of the chart in pixels
-icd10s : list of idc10 codes

Important note: please change the -destination for each job execution.
```

Hadoop job

https://git.man.poznan.pl/stash/projects/SCAP/repos/mr-jobs/browse/epidemic-jobs/casesaverage

20.9 Evaluation of the number of abnormal results in laboratory examinations for a given disease codes in a given period

Experiment: Analysis of epidemiological situation across WCPT patients

Evaluator(s)

Tomasz Hofmann, PSNC

Evaluation points

The main goal of this evaluation was to execute analysis on the number of abnormal laboratory examination results for a given disease codes in a given period. The investigated period and list of ICD10 codes are the input parameters for analysis algorithm. Statistics were gathered using PSNC Hadoop Platform and the map-reduce approach. As the evaluation metric the number of objects per second has been selected (the object is defined as a single HL7 file stored in HDSF).

Assessment of measurable points

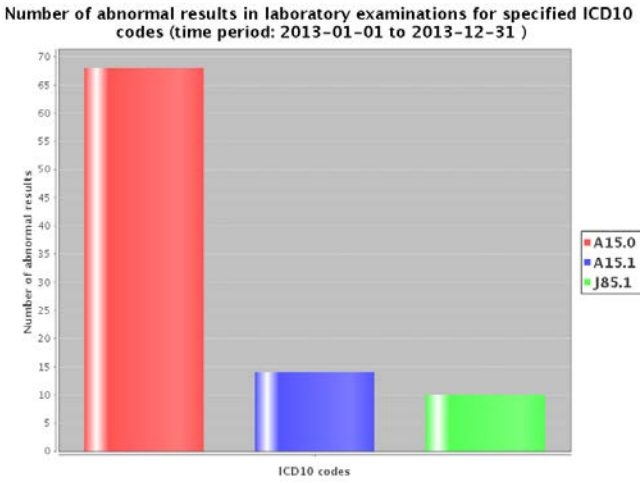
Metric	Description	July 21, 2014 [Test 1]	July 28, 2014 [Test 2]	July 30, 2014 [Test 3]
number of objects per second	number of HL7 files processed per second	4.196 [obj/s]	4,761 [obj/s]	4,979 [obj/s]

Note: *as an object we proposed to use one HL7 file

Visualisation of results

The chart below presents results of analysis for Test 2. Colours indicate different ICD10 disease codes. Test has been performed for patients who visited WCPT hospital between 1-01-2013 and 31-12-2013. Each column indicates the number of abnormal results in laboratory examinations for all patients. The ICD10 disease codes investigated in this analysis are as follows:

- A15.0 - Tuberculosis of lung, confirmed by sputum microscopy with or without culture
- A15.1 - Tuberculosis of lung, confirmed by culture only
- J85.1 - Abscess of lung with pneumonia



Additional information

Table 1 presents processing time of the whole job per test. Tables 2 and 3 provide information on the execution time and number of processed rows related to map and reduce tasks respectively. The

execution times (and performance) for three tests are similar because regardless of the analysed period it is necessary to process all HL7 files stored in the cluster.

Table 1. Overall statistics

Parameter	Test 1	Test 2	Test 3
Analysed period	1.07.2012-1.07.2014	1.01.2013-31.12.2013	1.01.2014-1.05.2014
Processing time	80 [m]	71 [m]	68 [m]

Table 2. Statistics for map task

Parameter	Test 1	Test 2	Test 3
Processing time (for all records)	80 [m]	71 [m]	68 [m]
Number of records	20 141	20 285	20 315

Table 3. Statistics for reduce task

Parameter	Test 1	Test 2	Test 3
Processing time (for all records)	30 [s]	26 [s]	24 [s]
Number of records	-	-	-

Technical details

Workflow

The experiment is composed of the following steps (accordingly to the MapReduce schema):

1. the map task [laboratory.sh]:
    a. for each HL7 file saved on HDFS do:
        i. parse document in order to find out abnormal laboratory results - count them and next add into the context the following pair: Key=icd10 code, Value=the count of the abnormal results
2. the reduce task [laboratory.sh]:
    a. for each icd10 code accumulate the count of the abnormal results
    b. produce the result pair Key=icd10 code, Value=the number of abnormal results

1. statistics are gathered by downloading and parsing log files [test.sh]

Scripts used to execute evaluation

https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/epidemic-jobs-tests/jobs-scripts/laboratory.sh
https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/epidemic-jobs-tests/jobs-scripts/test.sh

Execution commands

```
./laboratory.sh -admission 20090601 -destination tomek/tmp/laboratory11 -discharge
20140710 -hospital wcpit -icd10s J85.1 -icd10s A15.0 -icd10s A15.1 -laboratory RDW -
width 800 -height 600
./test.sh laboratory

where:
-admission : date of patient admission to hospital
-discharge : date of patient discharge from hospital
-destination : folder for hadoop job results (only one per job execution)
-width : width of the chart in pixels
-height : height of the chart in pixels
-icd10s : list of idc10 codes
```

```
Important note: please change the -destination for each job execution.
```

Hadoop job
https://git.man.poznan.pl/stash/projects/SCAP/repos/mr-jobs/browse/epidemic-jobs/laboratory

20.10 Evaluation of the number of medical cases for a given period

Experiment: Analysis of epidemiological situation across WCPT patients

Evaluator(s)

Tomasz Hofmann, PSNC

Evaluation points

The main goal of this evaluation was to execute analysis on the medical data stored at Medical Data Center and obtain statistics on the number of medical cases related to a given ICD10 code in a given period. The analysed period of time is additionally split into a given number of sub-periods. The analysed period, number of sub-periods and the ICD10 code are the input parameters for analysis algorithm. Statistics were gathered using PSNC Hadoop Platform and the map-reduce approach. As the metric the number of objects per second has been used (the number of records processed per second).

Assessment of measurable points

Metric	Description	July 21, 2014 [Test 1]	July 28, 2014 [Test 2]	July 30, 20144 [Test 3]
number of objects per second	number of records processed per second	2563 [obj/s]	3731 [obj/s]	2041 [obj/s]

Note: *as an object we proposed to use one scanned cell in Hbase table (one record)

Visualisation of results

The chart below presents results of analysis for Test 2. Colours indicate different sub-periods of time. Test has been performed for the patients who visited WCPT hospital between 1-01-2013 and 31-12-2013. This period is split into 5 sub-periods as seen on the chart below (each sub-period corresponds to one column). Each column indicates the number of patient visits for a given ICD10 code in a given sub-period. The ICD10 code in this test was set to J85.1 - Abscess of lung with pneumonia. The total number of cases found in a given period is presented on the chart as well (it is 33 in this particular case).

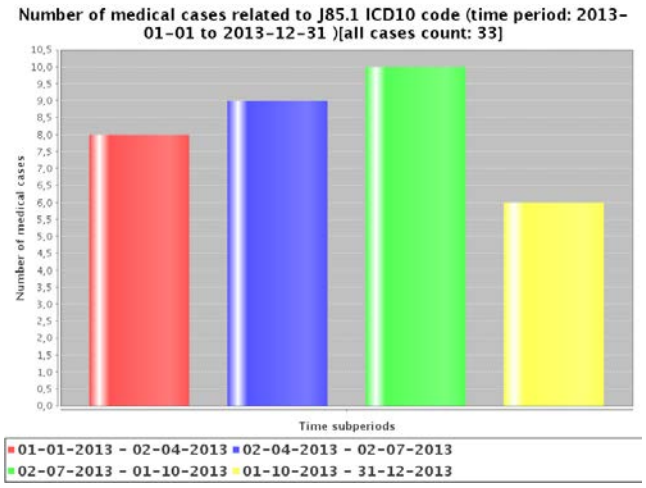


Table 1. Overall statistics

Parameter	Test 1	Test 2	Test 3
Analysed period	1.07.2012-1.07.2014	1.01.2013-31.12.2013	1.01.2014-1.05.2014

Processing time	64 [s]	39 [s]	5 [s]

Table 2. Statistics for map task

Parameter	Test 1	Test 2	Test 3
Processing time (for all records)	64 [s]	39 [s]	5 [s]
Number of records	165 518	146 608	9 287

Table 3. Statistics for reduce task

Parameter	Test 1	Test 2	Test 3
Processing time (for all records)	0,031 [s]	0,022 [s]	0,003 [s]
Number of records	894	640	83

Technical details

Workflow

1. The experiment is composed of the following steps (accordingly to the MapReduce schema) [casecounter.sh]:
    a. the map task,
        i. for each tuple in visits table:
            1. if icd10 code is in the set of given icd10 codes and if visit belong to the given period, then find the subperiod Id and add into the context pair: Key=subperiod_id, Value=visit_id
    b. the reduce task [casecounter.sh]:
        i. for each subperiod_id aggregate all visits ids in hash set - in order to find out the number of different visits
        ii. produce pair Key=superiod_id, Value=number of different visits (size of the hash set)
2. statistics are gathered by downloading and parsing log files [test.sh]

Scripts used to execute evaluation

https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/epidemic-jobs-tests/jobs-scripts/casecounter.sh
https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/epidemic-jobs-tests/jobs-scripts/test.sh

Execution commands

```
./casecounter.sh -admission 20130101 -destination tomek/tmp/casescounter08 -discharge
20131231 -hospital wcpit -icd10 J85.1 -periods 4 -width 800 -height 600
./test.sh casesCounter

where:
-admission : date of patient admission to hospital
-discharge : date of patient discharge from hospital
-destination : folder for hadoop job results (only one per job execution)
-width : width of the chart in pixels
-height : height of the chart in pixels
-periods : number of subperiods

Important note: please change the -destination for each job execution.
```

Hadoop job

https://git.man.poznan.pl/stash/projects/SCAP/repos/mr-jobs/browse/epidemic-jobs/casescounter

20.11 Evaluation of the patients gender for a given period

Experiment: Analysis of epidemiological situation across WCPT patients

Evaluator(s)

Tomasz Hofmann, PSNC

Evaluation points

The goal of this evaluation was to execute analysis on the Medical Data Center dataset and compute statistics on the gender of patients treated in a given period. The period of time to analysis is given as the input parameter for the analysis algorithm. As the metric the number of objects per second (number of records processed per second) has been selected.

Assessment of measurable points

Metric	Description	July 21, 2014 [Test 1]	July 22, 2014 [Test 2]	July 31, 2014 [Test 3]
number of objects per second	number of records per second	1798 [obj/s]	1600 [obj/s]	984 [obj/s]

Visualisation of results

The chart below presents results of the analysis for Test 2. Colours indicate gender of the patients. Each colour on the pie chart has related entry (note). Each entry is composed as follows: Y = Z [P], where Y is the name of gender, Z is the number of patient's visits (indicates the number of visits for analysed time period) and P is the percentage of the patient's visit in the overall context.

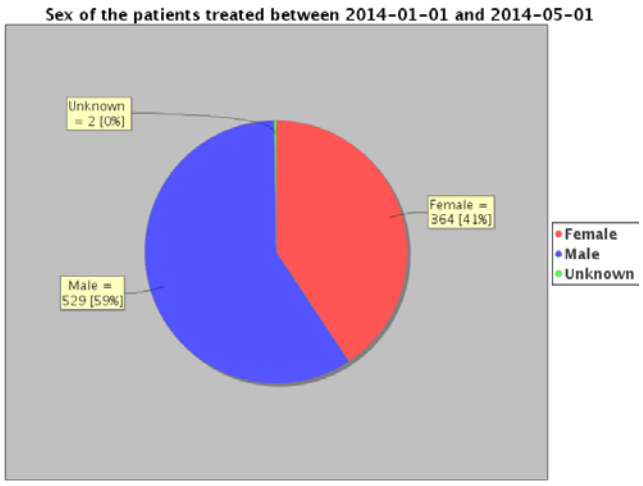


Table 1 presents processing time of the whole job per test. Tables 2 and 3 provide information on the execution time and number of processed rows related to map and reduce tasks respectively. From the statistics in the table and measurable points it is visible that: a) the processing time depends on the number of records to be processed b) the more records to process the better performance is achieved (more rows per second are processed).

Table 1. Overall statistics

Parameter	Test 1	Test 2	
Analyzed period	1.07.2012-01.07.2014	1.01.2013-31.12.2013	1.01.2014-1.05.2014
Processing time	95 [s]	94[s]	10[s]

Table 2. Statistics for map task

Parameter	Test 1	Test 2	Test 3
Processing time (for all records)	93 [s]	92 [s]	9,6 [s]
Number of records	168 021	148 333	9 462

Table 3. Statistics for reduce task

Parameter	Test 1	Test 2	Test 3
Processing time (for all records)	2,17 [s]	2,01 [s]	0,46 [s]
Number of records	15 331	14 164	895

Technical details

Workflow

The experiment is composed of the following steps (accordingly to MapReduce algorithm schema) [gender.sh]:

1. the map task [gender.sh]:
    a. for each tuple in visits table:
        i. if visit belong to the given period, then check patients sex and add into the context pair: Key=sex_id, Value=visit_id
2. the reduce task [gender.sh]:
    a. for each value of sex_id aggregate all visits ids in hash set - in order to find out the number of different visits
    b. produce pair Key=sex_id, Value=number of different visits (size of the hash set)
3. statistics are gathered by downloading and parsing log files [test.sh]

Scripts used to execute evaluation

https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/epidemic-jobs-tests/jobs-scripts/gender.sh
https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/epidemic-jobs-tests/test.sh

Execution commands

```
./gender.sh -hospital wcpit -destination tomek/tmp/gender09 -admission 20140101 -
discharge 20140701 -width 800 -height 600
./test.sh gender

where:
-admission : date of patient admission to hospital
-discharge : date of patient discharge from hospital
-destination : folder for hadoop job results (only one per job execution)
-width : width of the chart in pixels
-height : height of the chart in pixels

Important note: please change the -destination for each job execution.
```

Hadoop job

https://git.man.poznan.pl/stash/projects/SCAP/repos/mr-jobs/browse/epidemic-jobs/gender

20.12 Evaluation of DICOM data ingest (with copying data to archiving system)

Experiment: WCPT to PSNC DICOM medical data ingest

Evaluator(s)

Tomasz Hoffmann, PSNC

Evaluation points

The objective of this evaluation task was to test the efficiency of one instance of the DICOM data receiver available within the Medical Data Center platform at PSNC. In order to measure ingest performance the metric named number of objects per second has been used. Other metrics gathered during the tests provide additional information and include throughput in bytes per second, max object size handled in bytes and min object size handled in bytes. The metric goal was set to 0.25 objects per second to make sure that it is possible to ingest 10GB of data per day, which is the approx. amount of data produced by WCPT hospital each day. This evaluation is composed of 3 tests for 1, 4 and 10 concurrent threads sending DICOM files from one computer (client). This evaluation included copying data to archiving system.

Assessment of measurable points

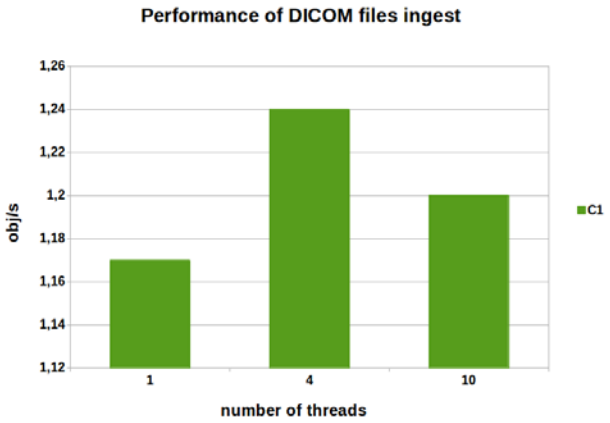
Metric	Description	Metric goal	June 6, 2014 [1T]	June 25, 2014 [4T]	June 25, 2014 [10T]
number of objects per second	number of ingested DICOM files* per second	0.25 [obj/s]	1 x 1.17 = 1.17 [obj/s]	4 x 0.31 = 1.24 [obj/s]	10 x 0.12 = 1.2 [obj/s]
throughput in bytes per second	_bytes per second_	116 000 [bytes/s]	514 483 [bytes/s]	145 763 [bytes/s]	54 645 [bytes/s]
max object size handled in bytes	max size of DICOM file	-	1 502 102 [bytes]	1 827 084 [bytes]	1 502 102 [bytes]
min object size handled in bytes	min size of DICOM file	-	201 554 [bytes]	44 090 [bytes]	44 406 [bytes]

Notes:

- an object is a single DICOM file
- xT - means x sending threads
- results in table presents statistics per one sending thread that are multiplied by the number of threads used in the test to get overall statistic

Visualisation of results

The chart below shows relation between ingest speed (in objects per second) and the number of sending threads used in the test. Tests show that a single client can reach up to approx. 1,2 objects per second, which is in fact limited not by the client computer but by the server (quite long time of storing files in the archiving system).



Tables below provide additional statistics. Table 1. presents a summary of all files and series used in the test (please note that a single series is composed of multiple DICOM files related to a single patient's examination, e.g. CT scan, RTG). Table 2. and Table 3 presents overall statistics related to series. Table 4. and Table 5. presents overall statistics related to files (objects).

Table 1. All files and series stats

Parameter	Test 1	Test 2	Test 3
No sending threads	1	4	10
All series number	33	262	66
All DICOM files number	3793	27076	6479
All DICOMs size	1658 [MB]	12564 [MB]	2901 [MB]
Saving time	3223 [s]	86196 [s]	53084 [s]
HDFS saving time	382 [s]	3285 [s]	1182 [s]
Hbase saving time	49 [s]	28 [s]	410 [s]
Sftp saving time	2722 [s]	81973 [s]	51402 [s]

Table 2. Maximum series stats

Parameter	Test 1	Test 2	Test 3
No sending thread	1	4	10
Max series size	309 [MB]	233 [MB]	211 [MB]
Max series sending time	704 [s]	1495 [s]	3816 [s]
Max series hdfs saving time	60 [s]	70 [s]	84 [s]
Max series hbase saving time	10 [s]	18 [s]	36 [s]
Max series sftp saving time	620 [s]	1424 [s]	3694 [s]
Max series items number	587	444	401

Table 3. Minimum series stats

Parameter	Test 1	Test 2	Test 3
No sending thread	1	4	10
Min series size	0.24 [MB]	0.04 [MB]	0.04 [MB]
Min series sending time	0.39 [s]	2.27 [s]	3.80 [s]
Min series hdfs saving time	0.08 [s]	0.06 [s]	0.06 [s]
Min series hbase saving time	0.01 [s]	0.006 [s]	0.01 [s]
Min series sftp saving time	0.24 [s]	2.13 [s]	3.50 [s]
Min series items number	1	1	1

Table 4. Maximum files stats

Parameter	Test 1	Test 2	Test 3
No sending thread	1	4	10
Max file size	1.50 [MB]	1.82 [MB]	1.50 [MB]
Max file sending time	24 [s]	14[s]	36 [s]
Max file hdfs saving time	1 [s]	5 [s]	2 [s]

Max file hbase saving time	1 [s]	1 [s]	1 [s]
Max file sftp saving time	23 [s]	13 [s]	35 [s]

Table 5. Minimum files stats

Parameter	Test 1	Test 2	Test 3
No sending thread	1	4	10
Min file size	0.20 [MB]	0.04 [MB]	0.04 [MB]
Min file sending time	0.16 [s]	0.98 [s]	1.10 [s]
Min file hdfs saving time	0.06 [s]	0.04 [s]	0.05 [s]
Min file hbase saving time	0.01 [s]	0.01 [s]	0.01 [s]
Min file sftp saving time	0.06 [s]	0.85 [s]	0.90 [s]

Raw log files

All log files created during the evaluation are available online at:
https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/dicom-test-results/ingest/with_sftp/test01_6_06_2014.log
https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/dicom-test-results/ingest/with_sftp/test02_25_06_2014.log
https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/dicom-test-results/ingest/with_sftp/test03_25_06_2014.log

Technical details

Workflow

The experiment is composed of the following steps:

1. prepare data set of DICOM files (over ~10GB, which is amount of data produced by WCPT hospital in one day),
2. send data to MDC server ([dicomSender.py]), and log on the server information about:
    a. size of received DICOM file
    b. time of saving data in:
        i. HDFS
        ii. Hbase
        iii. SFTP
3. parse server log file in order to evaluate metrics [resultsParser.py] (for metrics description please see previous point)

Scripts used to execute evaluation

https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/python/DICOM-tests/dicomSender.py
https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/python/DICOM-tests/resultsParser.py

Execution commands

```
python dicomSender.py ~/tmp/dicom/
python resultsParser.py ~/tests/dicom_tests/1_07_2014/dcmreceiverStats.log
test01_1_07_2014.txt

Notes:
    • scripts are prepared for Python in version 2.7.
    • dcmreceiverStats.log is a dcmrcv tool log file
```

20.13 Evaluation of DICOM data ingest (without copying data to archiving system)

Experiment: WCPT to PSNC DICOM medical data ingest

Evaluator(s)

Tomasz Hoffmann, PSNC

Evaluation points

The objective of this evaluation task was to test the efficiency of one instance of the DICOM data receiver available within the Medical Data Center platform at PSNC. In order to measure ingest performance the metric named number of objects per second has been used. Other metrics gathered during the tests provide additional information and include throughput in bytes per second, max object size handled in bytes and min object size handled in bytes. The metric goal was set to 0.25 objects per second to make sure that it is possible to ingest 10GB of data per day, which is the approx. amount of data produced by WCPT hospital each day. This evaluation is composed of 4 tests for 5, 10, 15, 20 concurrent threads sending DICOM files from one computer (client)._ _This evaluation did not include the process of copying data to the archiving system (cloud data storage).

Assessment of measurable points

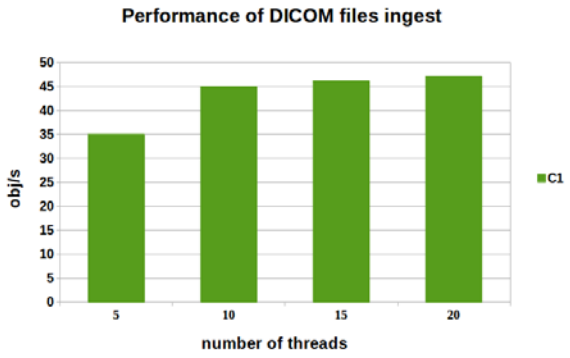
Metric	Description	Metric goal	June 30, 2014 [5T]	July 1, 2014 [10T]	July 1, 2014 [15T]	June 27, 2014 [20T]
number of objects per second	number of ingested DICOM files* per second	0.25 [obj/s]	5 x 7.01 = 35,05 [obj/s]	10 x 4.50 = 45 [obj/s]	15 x 3.08 = 46,2 [obj/s]	20 x 2.36 = 47,2 [obj/s]
throughput in bytes per second	_bytes per second_	116 000 [bytes/s]	3 252 209 [bytes/s]	2 088 492 [bytes/s]	1 429 771 [bytes/s]	1 091 593 [bytes/s]
max object size handled in bytes	max size of DICOM file	-	1 827 084 [bytes]	1 827 084 [bytes]	1 827 084 [bytes]	1 827 084 [bytes]
min object size handled in bytes	min size of DICOM file	-	44 090 [bytes]	44 090 [bytes]	44 090 [bytes]	44 090 [bytes]

Notes:

- an object is a single DICOM file
- xT - means x sending threads
- results in table presents statistics per one sending thread that are multiplied by the number of threads to get overall statistic

Visualisation of results

The chart below shows relation between ingest speed (in objects per second) and the number of sending threads used in the test. Tests show that a single client can reach up to approx. 45 objects per second, which is in fact the limit of the client computer.



Tables below provide additional statistics. Table 1. presents a summary of all files and series used in the test (please note that a single series is composed of multiple DICOM files related to a single patient's examination, e.g. CT scan, RTG). Table 2. and Table 3 presents overall statistics related to series. Table 4. and Table 5. presents overall statistics related to files (objects).

Table 1. All files and series stats

Parameter	Test 1	Test 2	Test 3	Test 4
No sending threads	5	10	15	20
All series number	263	263	263	254
All DICOM files number	27 053	27 053	27 053	26 300
All DICOMs size	12 550 [MB]	12 550 [MB]	12 550 [MB]	12 153 [MB]
Saving time	3 859 [s]	6 009 [s]	8 778 [s]	11 134 [s]
HDFS saving time	3 086 [s]	3 285 [s]	7 730 [s]	9 959 [s]
Hbase saving time	457 [s]	500 [s]	544 [s]	646 [s]

Table 2. Maximum series stats

Parameter	Test 1	Test 2	Test 3	Test 4
No sending threads	5	10	15	20
Max series size	233 [MB]	233 [MB]	233 [MB]	233 [MB]
Max series sending time	67 [s]	112 [s]	163 [s]	226 [s]
Max series hdfs saving time	52 [s]	91 [s]	143 [s]	198 [s]
Max series hbase saving time	10 [s]	14 [s]	14 [s]	28 [s]
Max series items number	444	444	444	444

Table 3. Minimum series stats

Parameter	Test 1	Test 2	Test 3	Test 4
No sending thread	5	10	15	20
Min series size	0.04 [MB]	0.04 [MB]	0.04 [MB]	0.04 [MB]
Min series sending time	0.06 [s]	0.06 [s]	0.08 [s]	0.09 [s]
Min series hdfs saving time	0.04 [s]	0.04 [s]	0.07 [s]	0.07 [s]
Min series hbase saving time	0.01 [s]	0.01 [s]	0.01 [s]	0.01 [s]
Min series items number	1	1	1	1

Table 4. Maximum files stats

Parameter	Test 1	Test 2	Test 3	Test 4
No sending thread	5	10	15	20
Max file size	1.82[MB]	1.82 [MB]	1.82 [MB]	1.82 [MB]
Max file sending time	3.55 [s]	3.05 [s]	2.61 [s]	3.74 [s]
Max file hdfs saving time	2.58 [s]	1.92 [s]	1.42 [s]	2.35 [s]
Max file hbase saving time	1.10 [s]	1.20 [s]	1 [s]	1.49 [s]

Table 5. Minimum files stats

Parameter	Test 1	Test 2	Test 3	Test 4
No sending thread	5	10	15	20
Min file size	0.04 [MB]	0.04 [MB]	0.04 [MB]	0.04 [MB]
Min file sending time	0.06 [s]	0.06 [s]	0.08 [s]	0.07 [s]
Min file hdfs saving time	0.04 [s]	0.04 [s]	0.07 [s]	0.05 [s]
Min file hbase saving time	0.01 [s]	0.01 [s]	0.01 [s]	0.01 [s]

Raw log files

All log files created during the evaluation are available online at:
https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/dicom-test-results/ingest/without_sftp/test01_30_06_2014.log
https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/dicom-test-results/ingest/without_sftp/test02_1_07_2014.log
https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/dicom-test-results/ingest/without_sftp/test03_1_07_2014.log
https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/dicom-test-results/ingest/without_sftp/test04_27_06_2014.log

Technical details

Workflow

The experiment is composed of the following steps:

1. prepare data set of DICOM files (over ~10GB, which is amount of data produced by WCPT hospital in one day),
2. send data to MDC server ([dicomSender.py]), and log on the server information about:
    a. size of received DICOM file
    b. time of saving data in:
        i. HDFS
        ii. Hbase
3. parse server log file in order to evaluate metrics [resultsParser.py] (for metrics description please see previous point)

Scripts used to execute evaluation

https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/python/DICOM-tests/dicomSender.py
https://git.man.poznan.pl/stash/projects/SCAP/repos/test-scripts/browse/python/DICOM-tests/resultsParser.py

Execution commands

```
python dicomSender.py ~/tmp/dicom/
python resultsParser.py ~/tests/dicom_tests/1_07_2014/dcmreceiverStats.log
test01_1_07_2014.txt

Notes:
    • scripts are prepared for Python in version 2.7.
    • dcmreceiverStats.log is a dcmrcv tool log file
```

21 Appendix G – Templates

21.1 User Story Template

Status

Single line status: one of Active, Started or Dormant - use tip macro

Contact

Contact information - name, institution and email - for this story

User Story

Brief statement of the issue of the form "As a <type of user>, I want <some goal> so that <some reason>"

User Requirements/Components

Precise statements of what is needed – e.g. "A tool to validate a WARC". Further detail can be added as sub-bullets and discussion with PC sub-project encouraged!

Experiments

Create experiments as child pages and they should appear automatically here

Developer Notes

Space for discussion, suggested solutions, links to other user stories

Related Documents

Scenarios, case studies, etc. that provide background to this story

21.2 Experiment Template

Investigator(s)

Names, links if available and emails

Dataset

Name and link to existing dataset with additional notes if required

Platform

Name and link to the experiment platform

Workflow

Description and ideally a link to a Taverna workflow

Requirements and Policies

Policy statements that relate to this experiment and any evaluation criteria taken from SCAPE metrics

Evaluations

Links to results of the experiment using the evaluation template

21.3 Platform template

[name]

Field	Datatype	Value	Description
Platform-ID	String		Unique string that identifies this specific platform. Use the platform name
Platform description	String		Human readable description of the platform. Where is it located, contact info, etc.
Number of nodes	Integer		Number of hosts involved - could be both physical hosts as well as virtual hosts
Total number of physical CPUs	Integer		Number of CPU's involved
CPU specs	String		Specification of CPUs
Total number of CPU-cores	Integer		Number of CPU-cores involved
Total amount of RAM in GB	Integer		Total amount of RAM on all nodes
Average CPU-cores for nodes	Integer		Number of CPU-cores in average across all nodes
Average RAM in GB for nodes	Integer		Amount of memory in average across all nodes
Operating System on nodes	String		Linux (specific distribution), Windows (specific distribution), other?
Storage system/layer	String		NFS, HDFS, local files, other?
Network layer between nodes	String		Speed of network interfaces, general network speed

Parallel Execution System

Field	Value
Installation description	
Configuration notes	

Benchmarks of Parallel Execution System

Metric	Benchmark type	Value	Description

21.4 Dataset template

Title	The name or short description of the dataset. This should be the same text as the title of the page
Description	Description of the dataset, including details of the file formats
Licensing	Details of any licencing restrictions. State as clearly and concisely as possible who can use the dataset. Reference any licences by URL or attachments
Owner	The institution that owns the dataset
Dataset location	A link to the dataset or instructions on how to obtain a copy of the dataset for testing/development purposes
Collection expert	The contact name for the collection. Add details of other collection experts and/or curators if appropriate. Identify the party with a link to their contact page on the SCAPE SharePoint site, as well as identifying their institution in brackets. E.g. Schlarb Sven (ONB)
Issues brainstorm	A bulleted list of possible preservation or business driven Issues. This is useful for describing ideas that might be turned into detailed Issues at a later date
List of issues	A list of links to detailed Issue pages relevant to this dataset

21.5 Evaluation template

Evaluator(s)

Contact information - name, institution - for this evaluation

Evaluation points

Assessment of measurable points

Metric	Description	Metric baseline	Metric goal	*2014-12-18*	*Evaluation*	*Evaluation*
	Goal, objective, baseline notes	10	1000	115		

Note: Metrics must be registered in the metrics catalogue

Assessment of non-measurable points

For some evaluation points it makes most sense to have a textual description/explanation. Also, please include a note about goals-objectives omitted, and why.

Technical details

Remember to include relevant information, links, versions about workflow, tools, APIs (e.g. Taverna, command line, Hadoop, links to MyExperiment, link to tools or SCAPE name, links to distinct versions of specific components/tools in the component registry)

WebDAV

We would like to store sufficient information about an experiment (Hadoop program, configuration, etc.), so we are able to rerun it. For this purpose, ONB is providing a WebDAV - if you have questions and need more information, please contact Sven or Reinhard at ONB.
Taverna workflows will still be stored on www.myexperiment.org.

Link: http://fue.onb.ac.at/scape-tb-evaluation

Please use the following structure for storing experiment results

http://fue.onb.ac.at/scape-tb-evaluation/{institutionid}/{storyid}/{experimentid}/{timestamp}/

Example:
http://fue.onb.ac.at/scape-tb-evaluation/onb/arc2warc/jwat/1374526050/

where institutionid = onb, storyid = arc2warc, experimentid = jwat, timestamp = 1374526050

Conclusion