SCAPE

Scalable Preservation Environments

First evaluation report

Authors

Bjarne Andersen (State and University Library), Catherine Jones (Science and Technologies Facilities Council), Kresimir Duretec (Vienna University of Technology), Peter May (The British Library), Yair Brama (Ex Libris)

January 2013

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

This report presents the results of the Evaluation of Results work package (TB.WP.4). In this work package, we have been developing a structured and systematic evaluation methodology based on goals, objectives, metrics and evaluation through experiments coupled to SCAPE-defined scenarios.

Using the defined methodology, a first round of evaluations of SCAPE was carried out during month 20 to month 22 (from a project period of 42 months).

The evaluations fall into 4 areas

1. Evaluation of results directly linked to Testbed scenarios
2. Evaluation of SCAPE functional review and development guidelines
3. Evaluation of a planning case study
4. Evaluation of SCAPE from a commercial point of view

The evaluation of Testbed scenarios has proven good results so far. The current status is that **performance** for some solutions has already improved with over a factor 200 by running experiments on a distributed platform based on Hadoop.

A small number of goals originally selected for evaluations were not found suitable for formal evaluation at this point of the project but these will all be included in future evaluations.

All in all the project is quite satisfied with the current state of developments based on the evaluations carried out. We feel that evaluations demonstrate good results but also indicate those areas that require improvements. This matches the expectations we have for a project that is approximately halfway through the planned work.

Table of Contents

1 Introduction

This report is the first evaluation report from SCAPE and TB.WP4. It presents the evaluation methodology developed in the project as well as the results from the first round of evaluations done in the project between M20 and M22. The methodology developed has been applied and 10 goals/objectives were initially selected for evaluation:

- Performance efficiency - capacity resource utilization and time behaviour
- Reliability – stability indicators
- Reliability – runtime stability
- Functional suitability – completeness
- Functional suitability – correctness
- Organisational maturity
- Maintainability – reusability
- Maintainability – organisational fit
- Planning and Monitoring efficiency - Information gathering and decision making effort
- Commercial Readiness

After selecting these top-10 goals the WP-members had to select specific SCAPE scenarios to evaluate looking at these goals. During this process and the following evaluations we chose to skip the evaluation of three of these goals (number 4, 6 and 7) for different concrete reasons described in chapter 3 (Top-10 goals and objectives). It turned out that it was not possible or meaningful to evaluate these three goals at this stage of the project but they will all be included in future evaluations already planned as a part of the future work in TB.WP4

Results from all Testbeds have been put together in this document because results are quite identical across the three application areas (Large Scale Digital Repositories, Web Content and Research Data Sets). Individual results for individual scenarios and thereby individual Testbeds can still be tracked and give input to the further developments in all areas of SCAPE.

Work during the evaluations has been carried out on the OPF-labs WIKI on SCAPE internal pages but all necessary information has been pulled out of the WIKI and put into this document including appendices with all detailed evaluation information as well as the WIKI-templates used during the process.

The following chapter presents the draft evaluation methodology (also known as milestone 76, MS76). After that we present the selected top-10 goals and objectives and their relation to specific scenarios.

Chapter 4 presents the evaluations done in relation to specific Testbed scenarios. 10 scenarios were selected for evaluation and evaluated in detail using the evaluation methodology. Chapter 5 presents the current status of the SCAPE functional review and development guidelines that will serve as the basis for evaluations of the goal “reliability – stability indicators” in future evaluations.

Chapter 6 presents the first results of a planning case study evaluating planning efficiency and chapter 7 presents the current thoughts and status from Ex Libris (EXL) being a commercial partner in the project.

Chapter 8 rounds up conclusions across the entire evaluation part of the report and following that the rest of the document is appendices that are only needed if you need all the detailed information about specific evaluations e.g. including technical specifications of the platforms used to carry out the individual experiments.

2 Draft Evaluation Methodology

The first milestone of the evaluation work package (TB.WP4) was to define the methodology to use for evaluating results in SCAPE. Having a common methodology across all evaluations and across the three different Testbeds ensures that results are comparable and meaningful to the rest of the project as well as the outside world. The agreed methodology is presented here.

Evaluation in SCAPE is done through the following steps

1. Define **top-10 goals and objectives** that should be evaluated - done initially at the SCAPE WIKI[1] and presented in this report in chapter 3
2. For each **goal/objective pair** select how to evaluate - in one or more of 4 possible ways (each instance called an **evaluation-point**)
    1. System/Platform level using a WIKI-template (see 10.1) - used when evaluating things on a distributed system - e.g. for performance metrics. The used platform is described individually using a WIKI-template (see 10.4)
    2. Component level using a WIKI-template (see 10.2) - used when evaluating things on a single machine - e.g. for accuracy metrics
    3. Registering the evaluation with a WIKI-template (see 10.3) and using the Plato Tool built for evaluation - used primarily for Action Components
    4. Writing up a summary with findings and results where no hard core uniquely defined metrics can be used - e.g. for organisational goals and objectives

Evaluations should be linked to SCAPE scenarios[1] where appropriate - see the templates in the appendices for detailed explanation. In the first round of evaluations we have tried not to define too many evaluation-points but for each goal/objective pair selected for evaluation there should be at least one. Some scenarios might be used to evaluate multiple goal/objective pairs.

Each evaluation will use a basic evaluation scheme

1. Setup evaluation (see template in Appendix B)
2. Define metrics (using the Metrics Catalogue – see Appendix A – metrics catalogue)
3. Define baseline (ground truth) - e.g. current state with a tool running on a single machine before SCAPE began
4. Define metric goal - what result do we want to achieve during SCAPE
5. Result of a given evaluation

For some objectives (e.g. commercial readiness) it might be hard or even impossible to define precise and measurable measures and goals. For these a more qualitative human understandable (e.g. in form of a short report / statement) evaluation will be done.

For many of the evaluations we foresee that they will get evaluated multiple times during the project - ultimately at least until the defined metric goal (4) is reached. The iterative process will end up with showing the progress of SCAPE developments with multiple values for (5) over time.

[1] http://wiki.opf-labs.org/pages/viewpage.action?pageId=14352645

The first round of evaluations is to be carried out in M20-M22 (and was carried out in this period by the time of writing this report) to be able to write up the first evaluation report as this deliverable for M24. At a later stage results (metrics) to be used in the evaluations will be queried from REF (Results Evaluation Framework – an RDF-based data store), but REF itself and integration between components/workflows are still under development and will not be ready for evaluation in the first round. REF will be integrated into the evaluation methodology in year-3.

Thus results used for the first round of evaluation will mostly be manually gathered and entered into the evaluation-pages on the WIKI.

3 Top-10 goals and objectives

Top-10 goals and objectives have been defined by Testbed WP-leads and reviewed by SP-leads.
The following two documents were used in the process of defining both the over all goals/objectives and selecting one or more evaluation-points for each

- An overview of scenarios and how they correspond/relate to work packages
- An overview of goals, objectives and suggested metrics defined by TB.WP4 and reviewed by SP-leads

Goals and objectives on the components and platform level have been mapped to the SQUARE software quality model. The following diagram (Figure 1) is taken from D14.1:

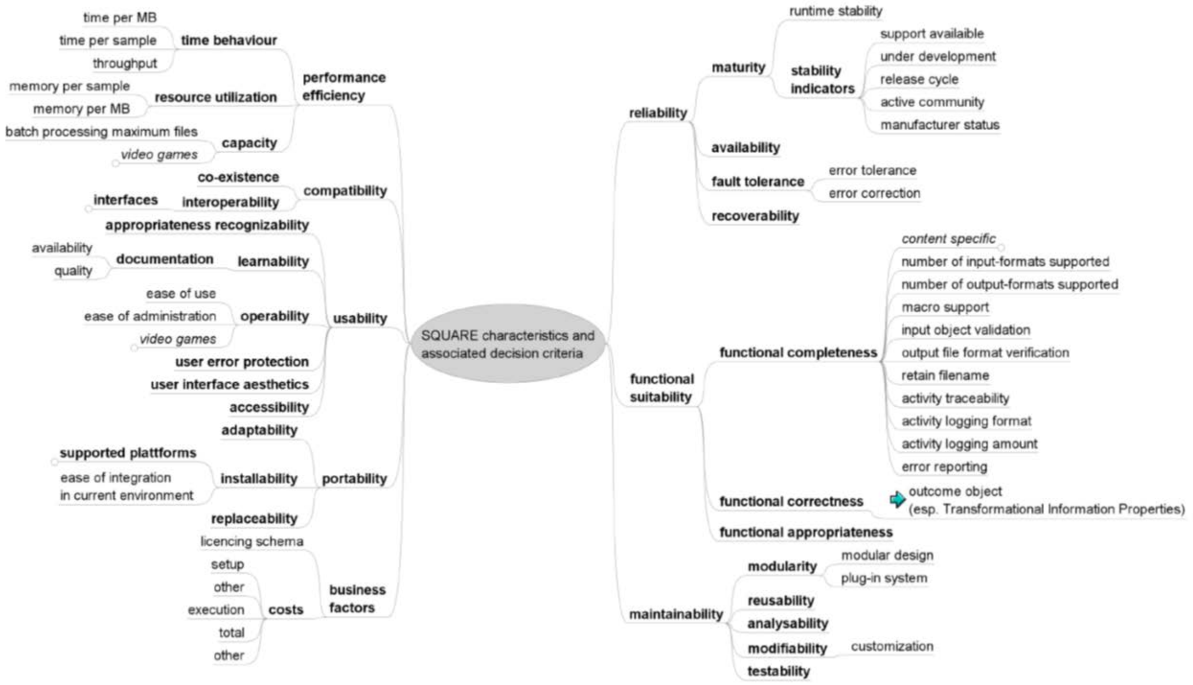


Figure 1

A number of project wide objectives are defined in the Description of Work (DoW) Part-B page 7-8:

- DoW-1: Addressing the problem of scalability in four dimensions: number of objects, size of objects, complexity of objects, and heterogeneity of collections
- DoW-2: Introducing automation and scalability in the areas of (2a) Preservation actions, (2b) Quality assurance, (2c) Technical watch, and (2d) Preservation planning
- DoW-3: Answering the question, what tools and technologies are optimal for scalable preservation actions, given a defined set of institutional policies?
- DoW-4: Providing a methodology and tools for capturing contextual information across the entire digital object lifecycle
- DoW-5: Producing a reliable, robust integrated preservation system prototype within the timeframe of the project
- DoW-6: Validating and demonstrating the scalability and reliability of this system against large collections from three different Testbeds
- DoW-7: Developing a skills base through training
- DoW-8: 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
- DoW-9: Providing insight into remaining barriers to take-up, clarifying the business cases for preservation, and investigating models for the provision of scalable preservation services

The specific evaluations in the Testbed evaluation methodology has been linked to these where appropriate

Top-10 goals and objectives (year-2)

In the table below (Table 1) the 10 selected goals and objectives are defined. These have been selected to cover a broad range of activities within the project as well as covering aspects across work packages and Testbeds.
For each goal/objective a number of SCAPE scenarios have been selected to make the actual evaluations where it makes sense to couple a goal/objective directly to scenarios. Some scenarios are evaluated with a more global perspective (not directly coupled to individual scenarios) and these have their own chapters following in this report.

SCAPE scenarios are based on a combination of a dataset, an issue and a solution and are designed to give the project a view on the progress made within SCAPE and to give a focus to the evaluation of the technical progress. 10 scenarios have been selected for evaluation. The scenarios and corresponding datasets come from one of the three Testbeds: Web Content (WC), Large Scale Digital Repositories (LSDR) and Research Data (RDST).

Web content scenarios

- WCT1: Comparison of web pages for quality assurance (Internet Memory Foundation)
- 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)
- WCT8 Huge text file analysis using hadoop (ONB)

Large Scale Digital Repositories

- LSDR2 Validating files migrated from TIFF to JPEG2000 (BL)
- LSDR6 Large scale migration from mp3 to wav (SB)
- LSDR11 Duplicate image detection within one book (ONB)

Research data

Both of the RDST scenarios are evaluated with data from the STFC.

- RDST1: General scientific data handling: looking at validation on ingestion
- RDST2: Format migration from RAW to NeXus: moving from a local format to the domain support standard.

Looking at the purpose of the scenarios for the evaluations, they can be categorised in the following way:

Characterisation of the content	Migration of the content	Validation of an actions	Quality Assurance	Specific action: Stats generated through text mining
WCT3, WCT4	LSDR6, RDST2	LSDR2, LSDR6, RDST1	WCT1, LSDR11	WCT8

This proves that the scenarios selected for evaluation covers a broad sweep of the project's scope and thus should give a good indication about the status of the project across the project – still seen from the Testbed work packages as the primary perspective.

A couple of goal/objects proved to be unsuitable to evaluate at this point of the project – this is briefly explained below and evaluations of these areas have not been carried out for this round of evaluations but will all be included in already planned (according to the DoW) future evaluations.

Table 1

No	Goal	Sub-goal	Objective	Relates to DoW objective	Evaluations
1	Performance efficiency	Capacity Resource utilization Time behaviour	Improve DP technology to handle large preservation actions within a reasonable amount of time on a multi node cluster	DoW-1: Number of objects Dow-1: Heterogeneity of collections Dow-5: Producing a reliable, robust integrated preservation system prototype	WCT1 WCT3 WCT4 WCT8 LSDR2 LSDR6 LSDR11 RDST1 RDST2 Evaluation results in chapter 0

				Dow-6: Validating and demonstrating the scalability and reliability against large collections	
2	Reliability	Stability Indicators	Package tools with known methods and run development with good open source practices	DoW-8: Ensuring a viable future for the results	This goal/objective is covered in chapter 5 about SCAPE functional review and development guidelines
3	Reliability	Runtime stability	Improve DP technology (platform and tools) to run automated with proper error handling and fault tolerance	DoW-2: Introducing automation and scalability in Preservation Action, Quality Assurance Dow-5: Producing a reliable, robust integrated preservation system prototype Dow-6: Validating and demonstrating the scalability and reliability against large collections	WCT1 WCT3 WCT4 WCT8 LSDR2 LSDR6 LSDR11 Evaluation results in chapter 4.2
4	Functional suitability	Completeness	Improve number of file formats correctly identified within a heterogeneous corpus		This goal was originally selected for evaluation but it proved unsuitable to evaluate this in relation to scenarios because not much work has yet happened on scenarios directly related to

					correctness of identification. This will be included in future evaluations when it gets relevant and possible.
5	Functional suitability	Correctness	Develop and improve components to do preservation actions more correctly	Dow-6: Validating and demonstrating the scalability and reliability against large collections	LSDR6 LSDR11 RDST1 RDST2 Evaluation results in chapter 4.3
6	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	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		Much work on this has been carried out within the PW sub project. A questionnaire has been built and tested within a couple of institutions but it is still too early to formally evaluate this aspect of the project. Will be included in future evaluations.
7	Maintainability	Reusability	Increase number of tools registered in components catalogue making them discoverable		Components Catalogue is slightly delayed – unable to evaluate at this point. Will be part of future evaluations.
8	Maintainability	Organisational fit	Ensure SCAPE technology fits organisational needs, competences and technical capabilities	DoW-8: Ensuring a viable future for the results	WCT3 LSDR2 Evaluation results available in chapter 4.4
9	Planning and Monitoring efficiency	Information gathering and decision making effort	Drastically reduce the effort required to create and maintain a	DoW-2: Introducing automation and scalability in	Case studies and assessments Quantitative metrics on numbers

			preservation plan	Preservation Planning	of decision making steps Potentially: an effort-aware planning component that tracks how much time people spend in certain decision making steps Evaluation results in chapter 6
10	Commercial readiness		Evaluate to what extent SCAPE technology is going in a direction that makes it ready for commercial exploitation		status-chapter by EXL – see chapter 7

4 Evaluation of TB-scenarios

Four of the goals have been evaluated in direct connection to specific scenarios worked on in the three Testbed work packages. Several scenarios have evaluated multiple of the goals since they are not surprisingly relevant for much of the work going on across work packages, institutions and preservation challenges.

This also means that several goals have been evaluated across several scenarios and that should give the evaluation results more weight when e.g. 10 different scenarios have been evaluating performance efficiency.

As part of the methodology described previously, evaluators had to set specific goals – e.g. a specific number for a given metric to "reach" by the end of the SCAPE project. This turned out to be rather hard even given the fact that scenarios are directly linked to practical preservation challenges on partner institutions.

Defining the exact number of items that the institution wants to process in one calendar month for example would require extensive business analysis as well as human overlap between SCAPE participants and digital preservation practitioners in the individual partner institutions. This is not always the case and having the developers (and evaluators) working for SCAPE defining precise institutional goals might lead to arbitrary goals and the risk of SCAPE failing to meet them. We have tried to tackle this situation with having SCAPE evaluators talk as much as possible to their home institutions as well as defining the goals as realistic as possible. Some of the goals set have been explained more detailed in textual form to also reveal what they actually mean and how they were set – see the actual evaluation templates in appendix 11 for more information.

4.1 Performance efficiency

Performance has been quite natural (since SCAPE is about scalability) evaluated in all 10 Testbed scenarios with the following results

Table 2

metric	baseline	Goal	eval-1
LSDR-2: NumberOfObjectsPerHour	50	1600	87,4
LSDR-2: ThroughputGbytesPerHour	0,766	25	1,355
LSDR-3: NumberOfObjectsPerHour	50	1600	45
LSDR-3: ThroughputGbytesPerHour	0,766	25	0,697
LSDR-6: NumberOfObjectsPerHour	10	1000	18
LSDR-11: NumberOfObjectsPerHour	0,05	1	0,18
LSDR-11: AverageRuntimePerItemInHours	N/A	1	5,4
RDST-1: ThroughputGbytesPerMinute	65,7	70	65,7
RDST-2: ThroughputGbytesPerMinute	1,73	12	1,73
RDST-2: NumberOfObjectsPerHour	1152	15300	1152
WCT-1: NumberOfObjectsPerHour	38	100	38
WCT-3: ThroughputGbytesPerMinute	0,162	60	1,32
WCT-4: ThroughputGbytesPerMinute	0,08	5	16,17
WCT-8: ThroughputGbytesPerMinute	0,35	5	11,93

The first column of Table 2 is the name of the metric. The next column is the measured baseline value ("situation before SCAPE"). The third column represents the goal set by the scenario-owners ("how fast do I need this solution to be on a given dataset") whereas the last column represents the actual evaluation carried out between September and December 2012 (current status).

Since the evaluations has been done on many different platforms and at the component level (not primarily evaluating performance but measuring the baseline and evaluating other aspects) the numbers are quite hard to compare because they range from below 1 to above 15.000

Some evaluations (e.g. WCT1, RSDT1) have not yet shown much improvement (difference between baseline and evalu-1) since the status of the solutions of those scenarios has primarily been to develop a working solution for a given preservation issue. So for some evaluations the baseline figures are the same as these initial evaluations.

Other scenarios (e.g. WCT3, WCT4 and WCT8) have already been tested on a distributed experimental platform (based on hadoop as prescribed by the Platform sub project) and these have already shown good results. A matter of fact two scenarios has already reached their goals within the timeframe of SCAPE.

To be able to present the performance results in a comparable way we have translated all the performance metrics goals in to 100 (set the goal to 100 and calibrate baseline and evaluation-1 accordingly). This gives the following histogram figure.

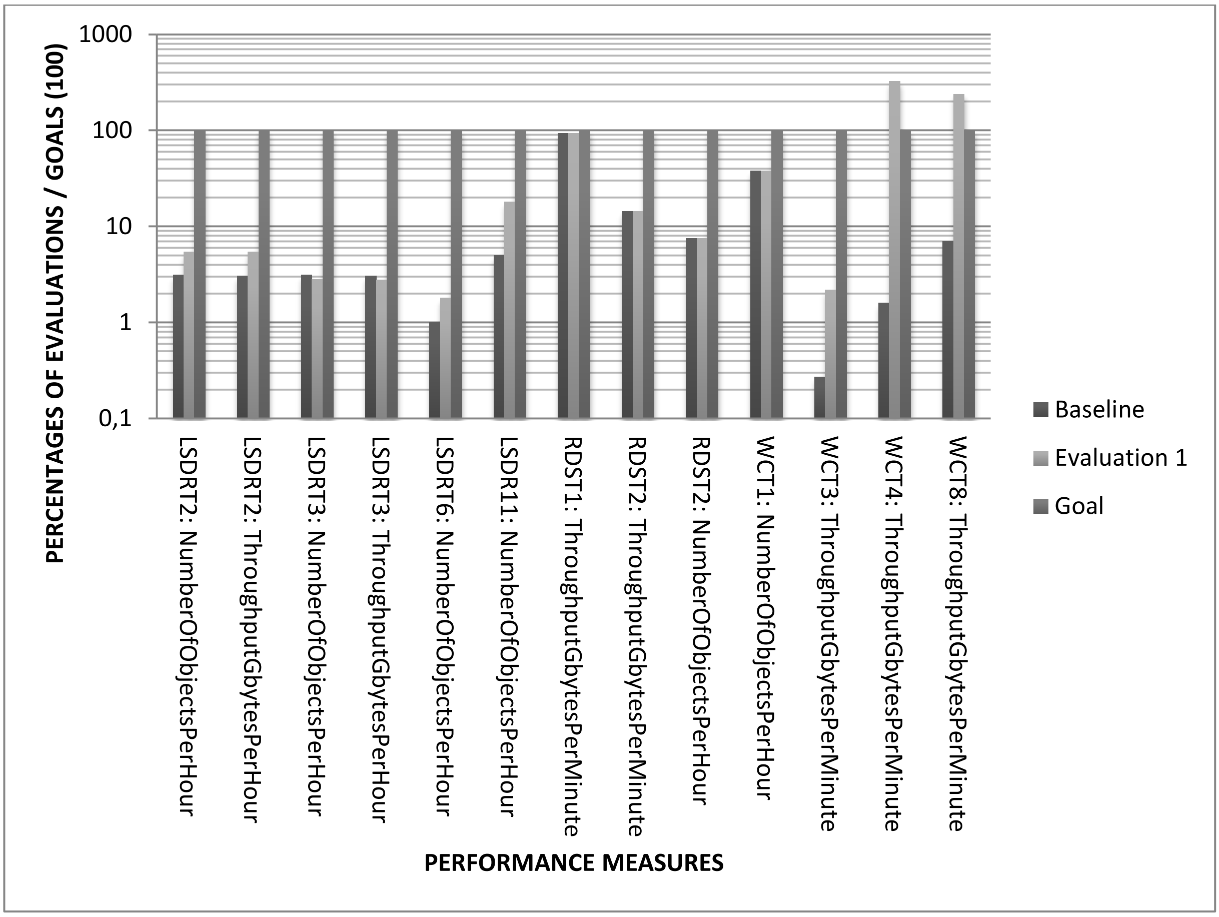


Figure 2

Please note that the results in Figure 2 are presented on a *logarithmic scale* because, even after the recalibration they vary between 2 and over 200 and thus not really presentable on a linear scale.

As the above diagram shows WCT-4 and WCT-8 are already above the goal (the middle bar higher than the right bar) within SCAPE when it comes to performance where as most others lie between 2% to 50% of the performance set in the goals.

The current improvements from the measured base line performance range from no improvements in speed (not surprising since several experiments have not been carried out on a distributed platform) to over a factor of 200. The current improvements can be visualised as follows (Figure 3):

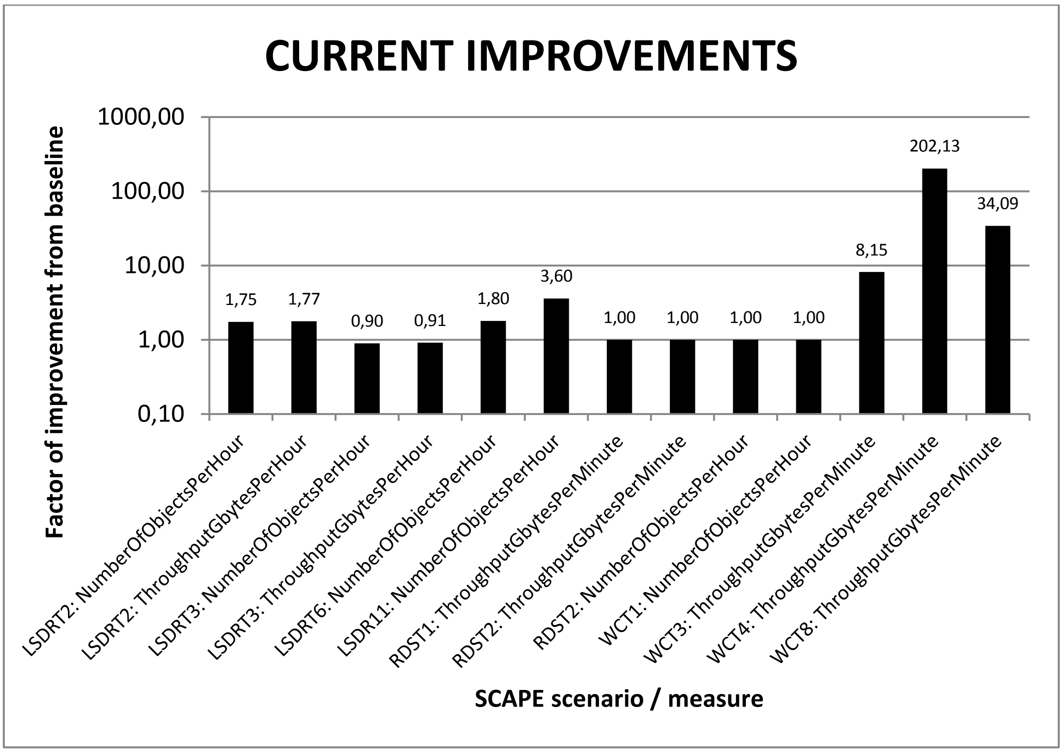


Figure 3

The most performance gain has not surprisingly been obtained in the three experiments done in real distributed environments with several physical servers involved. WCT4 is the top score with a performance improvement of over 200. This is gained on a platform with only 5 nodes (and 40 CPU cores) and the great improvement is thus also a result of implementing the workflow that runs the experiment to take fully advantage of the HDFS file system beneath hadoop and the preparation of the data to be processed into suitable chunks of data to improve disk I/O on HDFS and thus the overall performance significantly.

The still outstanding performance improvements to hopefully reach by the end of the project can be illustrated in the following figure (Figure 4)

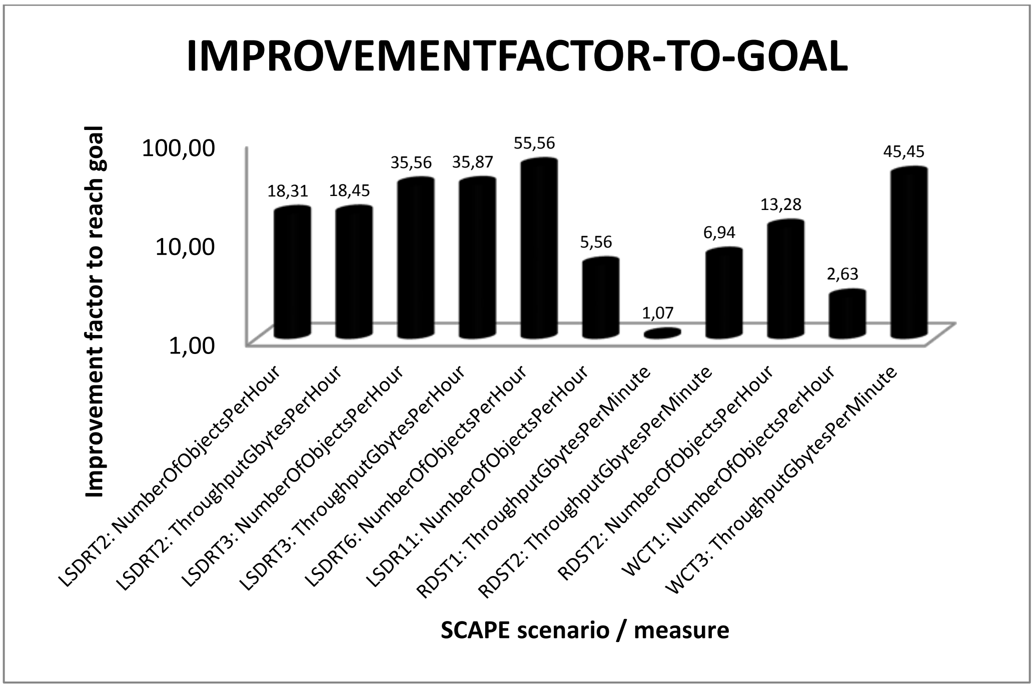


Figure 4

The above diagram shows that the average improvement in performance needed is around 15 times. The implementation of the solutions on distributed "real" SCAPE platforms is planned to give this performance boost – of cause requiring a platform with multiple nodes to reach the goals – most likely a platform with a bit more than 15 nodes to reach a performance factor of 15 because of a potential overhead with running workflows in a distributed environment. Performance could also require that individual tools and/or workflows should be improved to gain the required speed requirements.

Performance (scalability) is one of the key objectives of SCAPE and the status after close to two years into the project is that the technological solutions are starting to show good results in this area. There are still outstanding requirements of further performance improvements but there is quite clear indications that these should exactly been tackled in the already planned activities for the coming year in SCAPE.

4.2 Reliability – Runtime stability

A key requirement for having workflows and tools running in a distributed environment on millions of digital objects is that the technology is able to behave – even when e.g. tools fail on single individual objects. So Runtime stability is also an important objective for SCAPE.

4 scenarios have evaluated the overall reliability and stability of the implemented workflows and solutions. They have been evaluated manually by looking at the runtime results as well as investigating error logs and the like that comes out of the individual experiment. The evaluation has thus been carried out quite manually (which is OK according to the first draft evaluation methodology) and on a true/false scale.

The results are as follows (Table 3):

Table 3			
metric	**baseline**	**goal**	**eval-1**
LSDR-3: ReliableAndStableAssessment	False	True	True
LSDR-6: ReliableAndStableAssessment	False	True	True
WCT-4: ReliableAndStableAssessment	False	True	True
WCT-8: ReliableAndStableAssessment	False	True	True

All evaluations were positive in the sense that all four experiments were run fully as planned with no major incidents. The reason for all evaluations indicating "False" as the baseline is because all evaluated experiments represent solutions that did not exist before SCAPE, meaning that is not possible to carry out that same preservation task in a reliable and stable manner without the SCAPE specific developments.

Another precise measure of the success of a solution in the experiments evaluated is the number of failed files. This metric has been evaluated in 7 different scenarios and only one file in one scenario failed with all scenarios in total running on hundreds of thousands of files.

Table 4			
metric	**baseline**	**goal**	**eval-1**
LSDR-2: NumberOfFailedFiles	0	0	0
LSDR-3: NumberOfFailedFiles	0	0	1
LSDR-6: NumberOfFailedFiles	0	0	0
LSDR-11: NumberOfFailedFiles	0	0	0
WCT-1: NumberOfFailedFiles	0	0	0
WCT-4: NumberOfFailedFiles	0	0	0
WCT-8: NumberOfFailedFiles	0	0	0

All in all this (Table 4) shows that the evaluated solutions all have proved to be very stable and reliable. The one file failing in evaluation of LSDR3 has been reported by the evaluator in the following way

> *"One file failed during the migration. However, this did not stop the rest of the migration from completing and the failure was clearly identified in the outputs."*[2]

4.3 Functional suitability – Correctness

This objective has been evaluated in much fewer scenarios because correctness has not been a key focus for all the evaluated scenarios.

The correctness was evaluated in two totally different solutions (see Table 5). The first one deals with quality control of audio migration (LSDR6) and second one is about detecting duplicates in a corpus of scanned books (LSDR11).

[2] See evaluation of LSDR3 in chapter 11.1.2 for more details.

Table 5			
metric	**baseline**	**Goal**	**eval-1**
LSDR-6: QAFalseDifferentPercent	5	0,1	0,412
LSDR-11: IdentificationCorrectnessInPercent	N/A	98	96

In LSDR-6 the percentage of FalseDifferences was measured to 5% when the first version of this QA-tool was released in SCAPE. This has been improved several times during testing and development and the evaluation documented in this report shows that the tools is in the current version still resulting in 0,41% false difference files where the tools rejects a migration that is actually correct (false negatives).

The goal is to be able to have the tool running at a correctness level of 0.1% meaning that we still need a factor 4 more preciseness. Since this has nothing directly to do with scalability of the workflow the only obvious place to obtain this improvement will be in the tool itself – thus giving feedback into the development cycle of the QA components work package.

The other scenario evaluated for correctness is LSDR-11. This one is about detecting duplicates in a corpus of scanned books. The baseline is as the table shows not available since there were no existing solution at all for this preservation issue and SCAPE has been developing the needed technology from scratch. The goal for this scenario is to be able to find these duplicates with a correctness of 98% and as the evaluation shows the current result is that the correctness obtained in this first formal evaluation is 96% - quite close to the goal.

This evaluation (LSDR-11) was made on an annotated corpus of ground truth and you can read more about the solution and its evaluation here:
https://github.com/shsdev/bookpagetuples-detect-eval-lib/blob/master/README.md

The RDST evaluations did not directly measure correctness, however there were no errors detected during evaluation which is one measure of correctness. The RDST scenarios at present do not implement any checking for correctness in the results; however it would be possible to introduce this into the workflows. RDST1 is concerned with extracting the content of files so that validation could be performed. The next step in the process is to validate this against the schema for the file and this would provide information on correctness. RDST2 which performs format migration could be extended to include quality assurance for correctness.

4.4 Maintainability – Organisational Fit

Table 6			
metric	**baseline**	**goal**	**eval-1**
LSDR-2: OrganisationalFit	N/A	True	True
LSDR-3: OrganisationalFit	N/A	True	True
WCT-3: OrganisationalFit	N/A	True	True

The evaluation of “Organisational Fit” has been done looking at the following criteria

- Fits organisational needs

- Matches organisational competences
- Matches organisations technical capabilities

As shown in the table (Table 6) all three scenarios evaluated in this area were evaluated “True”. The two first scenarios were evaluated by British Library (BL) and the third by Statsbiblioteket (SB). The “True” means that the solutions developed for the specific scenarios matched the organisation that had the original digital preservation problem (issue). For future evaluations it will be fruitful to test solutions at other institutions and evaluate whether the solutions also match their needs, competences and technical capabilities.

5 SCAPE functional review and development guidelines

The SCAPE project aims to ensure the software products it produces are being developed following best practice, to meet agreed functionality and that it has in place all necessary procedures for integration and functional testing.

To enable this, functional review audits are scheduled for every eight months (the next being at M24, M32, and M40). These check main criteria: the development process, code quality, documentation, functional evaluation and installation & deployment.

An up-to-date copy of the current SCAPE Functional Requirements can be found on the wiki: http://wiki.opf-labs.org/display/SP/The+SCAPE+Functional+Review+Process. It is expected that these guidelines will evolve over time, through feedback from developers. An overview of the current guidelines is described below:

Coding guidelines:

- New projects should ideally be written in Java, as this is the preferred platform language.
- Code should be in a public repository (Open Planets Foundation GitHub).
- Code should adhere to the relevant language development guidelines (see wiki).
- Code should be well documented (i.e. JavaDoc)
- Unit tests must exist.

Documentation guidelines:

- There should be a README describing how to use the project.
- A functional specification should exist for the project. For newly started projects this may be the README.
- There should be documentation about building, installing, running etc.
- Dependencies and their licences should be clearly documented.
- Licensing of the project should be clearly documented.
- Official supported installations, e.g. the SCAPE central instance, should be clearly documented along with details on support/sustainability arrangements.
- The primary developer(s) and any other developers on the project should be named in appropriate locations i.e. Maven POM file & source code.

Packaging guidelines:

- A downloadable binary package should be provided for at least one OS. For example, a GitHub binary download.
- There should be Debian packaging for the project.

Since these guidelines are relatively new in their current state and thus not fully implemented in all SCAPE development and since the next official functional review is in M24 (January 2013) is has not been feasible to evaluate all the solutions with this perspective but this will be part of following evaluations.

6 Evaluation of planning case study

A three day preservation planning case study has been conducted by Technical University of Vienna (TUW) at Statsbiblioteket (SB) in Aarhus (Denmark). Participants included staff from the Digital Preservation Technology Group and from the National Library Division.

The two parts of the case study were:

1. tutorial on preservation planning in Plato 3
2. building a preservation plan

In the tutorial a preservation plan was created for an image collection SB is responsible for. On the basis of this scenario participants went through the steps required for a solid, well-documented preservation plan. In total, roughly one full day was spent on using and working with the approach and planning tool Plato 3.

The knowledge acquired in the tutorial served as the basis for a controlled case study aiming at the creation of a preservation plan in a structured way for a real-life scenario.

SB has a collection of radio broadcasts recordings. Parts of the collection are in MP3 format and parts in WAV. In order to harmonize file formats within the digital preservation repository, the migration of the MP3 collection to WAV was evaluated. A preservation plan was created to decide what should be done with MP3 samples: should they be migrated to WAV and if yes how or should they remain as they are.

The preservation plan was created following the 14 steps preservation planning workflow defined by Plato 3. For each step number of participants and time was measured. Results are presented in the Figure 5 below. Values are expressed as person hours. The total time required to create a preservation plan was 34:40 person hours.

As it can be seen from the Figure 5 most of the time was spent in identifying requirements, evaluating experiments and transforming measured values. The big amount of time spent in analysing results can be explained with all participants being included in the discussion of the results. Even though it required a lot of time this step should not be considered as an effort intensive step.

Besides using Plato things like quality assurance and collection analysis were done in a rather manual way. The collection analysis and selecting a sub collection took another 2 person hours.

In total the effort spent to create a preservation plan in Plato 3 was 36:40 person hours.

These measures will be used for a later comparison of efforts required to build the same plan in Plato 4 and the measures are serving as kind of the baseline for future evaluations of planning tasks.

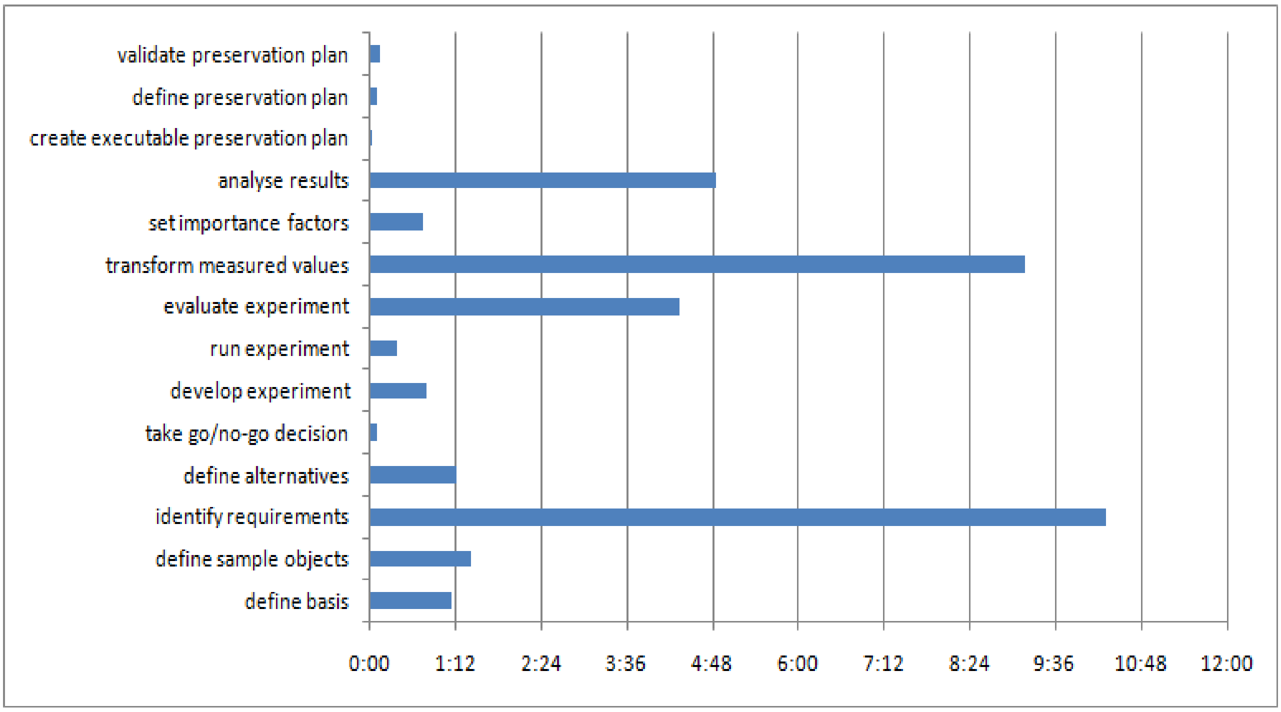


Figure 5 Required time for each Plato workflow step

7 Evaluation of results from a commercial point of view

Ex Libris involvement in SCAPE

Ex Libris is involved in SCAPE in the following ways:

1. As a partner in developing tools and APIs that can be used by other repositories/applications.
2. As a platform for testing tools and scenarios.
   a. Installed locally in institutions
   b. Installed on Ex Libris servers, hosting data from various partners of SCAPE

Rosetta[3] Part in Test-beds

Rosetta can be part of the TB effort by acting as a repository installed in one (or more) of the institutions that are part of the TB.

In addition (or instead), Ex Libris offers a hosted environment of Rosetta that can be used for tests and integrations with SCAPE developments.

Rosetta could for example be tested and evaluated in the following scenarios:

[3] Rosetta is a commercially available digital preservation system developed by SCAPE partner Ex Libris: http://www.exlibrisgroup.com/category/RosettaOverview

1. LSDR2 – Migrating TIFF to JPEG2000.
2. LSDR6 - The mp3 to wav migration.
3. RDST1 – Characterization of Nexus files
4. RDST2 - Raw to Nexus format migration

Tests Requirements

In order to use Rosetta in the suggested scenarios, regardless of where Rosetta is installed (locally or hosted) the following steps should take place:

1. Loading data into Rosetta – The files that need to be migrated or characterized should be packaged in a way Rosetta can upload them. Rosetta offers a variety of tools for loading the files manually one by one or automatically in a bulk. In addition, Rosetta offers a set of web-services for creating SIPs and loading them from an FTP server or a local file system.
2. Installing Plug-ins – The tools that are used for migration and characterization need to be installed in Rosetta as plug-ins. Rosetta can integrate such tools if they are in Java and wrapped according to Rosetta standards.

Rosetta as a Platform

Rosetta should be handled as an independent SCAPE platform with capabilities of processes management and load distribution of its own. Therefore tests that rely on using Taverna and Hadoop must be replicated and be done separately in Rosetta.

Current status and future work

No SCAPE components have been integrated and tested through Rosetta yet but a small number of solutions is planned to be implemented as a proof-of-concept on the Rosetta platform in cooperation between Ex Libris and one or more SCAPE partners. This work will be started within the next year of the project to allow for results to be evaluated and presented in the final evaluation report in month 42 of the project.

8 Conclusions from first evaluations

This first round of evaluations of SCAPE results has been carried out in month 20 to month 22 of the project period of 42 months.

The evaluations fall into 4 areas

- Evaluation of results directly linked to Testbed scenarios
- Evaluation of SCAPE functional review and development guidelines
- Evaluation of a planning case study
- Evaluation of SCAPE from a commercial point of view

The evaluation of Testbed scenarios has proven good results. The current status is that **performance** has already improved with over a factor 200 running experiments on a distributed platform based on hadoop. There is still room for improvement especially in the scenarios currently only evaluated on a component level on single machines. The average still outstanding performance improvement is around a factor of 15. This seems like a very descent and obtainable number when experiments start to run on distributed platforms as planned in the coming year of the project.

Runtime stability has already proven to be fantastic. All scenarios that were evaluated according to this specific goal proved to work out nicely without any major incidents. 7 scenarios evaluated the "number of failed files" and only one scenario had one single file that failed. This compared to the fact that experiments were carried out on hundreds of thousands of files proves that the developed components and workflows works flawlessly. The singled failed file was reported as failed by the workflow and it didn't stop the rest of the files to run through the entire workflow.

Correctness has been evaluated in two different scenarios that works specifically with correctness of quality assurance evaluating components and workflows developed in the QA work package. Both evaluations proved good results. The first evaluation has already improved the QA component for audio migration going from performing with 5% false positives to now only giving 0.4% false positives. The goal for this specific scenario is to reach a level of only 0.1% false positives reported by the workflow so there is still room for at bit of improvement in the development of the components being a part of that solution. The other solution evaluated has currently reached a number of 96% correct identification and the goal is to reach 98%. Compared to the fact that this solution solves a problem with duplicate detection of scanned images that was non-existing before SCAPE it's already a great result.

Organisational fit was evaluated in three different scenarios. All evaluators gave positive response to this goal meaning that solutions evaluated met institutional requirements for solving the preservation issue as well as matching the institutional competences and technical capabilities meaning that the institutions was actually able to use the proposed solutions in real life settings.

The evaluation of the goal "**reliability – stability indicators**" is about evaluating whether solutions and technical developments can be proven to be stable seen from a development and maintainability view. The SCAPE functional review process as well as the defined development guidelines ensures that this will be the case. The latest version of the development guidelines has been developed as an output from the latest functional review carried out in year-2. It has thus not been possible to evaluate solutions strictly according to these guidelines but such evaluation will be carried out as part of the next functional review of SCAPE in month 24 and the output of this review will be included in the next evaluation report.

A **planning case study** has been carried out in cooperation between the Technical University of Vienna (TUW) and the State and University Library (SB). In terms of automation within planning processes significant improvements are already starting to show. Case studies have been carried out using version 3 of the planning tool Plato to help defining the baseline of planning efficiency when it comes to the actual time it takes to develop a preservation plan. Based on these base line results, future automated planning processes with Plato version 4 as developed in SCAPE will no doubt improve planning efficiency.

From a **commercial point of** view SCAPE developments have been followed by Ex Libris – a commercial software solution partner taking part of the SCAPE project. Ex Libris has been looking at SCAPE components and solutions and will in year-3 implement a proof-of-concept form selected components in cooperation with one or more SCAPE partners. It has not been possible to do this integration yet but it will be a natural part of the SCAPE work in the coming year so again future evaluations will take this aspect into consideration as well.

A small number of goals originally selected for evaluations were not found suitable for formal evaluation at this point of the project but these will all be included in future evaluations.

All in all the project is quite happy with the current state of developments based on the evaluations carried out. We feel that evaluations yield promising results as well as room for improvements which perfectly match the expectations we have for a project that is about half through the planned work.

9 Appendix A – metrics catalogue

Metric	Data type	Description	Example	Comments
NumberOfObjectsPerHour	integer	Number of objects that can be processed per hour	250	Could be used both for component evaluations on a single machine and on entire platform setups
IdentificationCorrectnessInPercent	integer	Defining a statistical measure for binary evaluations - see detailed specification below	85%	Between 0 and 100
MaxObjectSizeHandledInGbytes	integer	The max file size a workflow/component has handled	80	Specify in Gbytes
PlanEfficiencyInHours	integer	Number of hours it takes to build one preservation plan with Plato	20	Specify in hours
ThroughputGbytesPerMinute	integer	The throughput of data measured in Gbytes per minute	5	Specify in Gbytes per minute
ThroughputGbytesPerHour	integer	The throughput of data measured in Gbytes per hour	25	Specify in Gbytes per minute
ReliableAndStableAssessment	Boolean	Manual assessment on if the experiment performed reliable and stable	true	
NumberOfFailedFiles	integer	Number of files that failed in the workflow	0	
QAFalseDifferentPercent	integer	Number of content comparisons resulting in *original and migrated different*, even though human spot checking says *original and migrated similar*.	5%	Between 0 and 100
AverageRuntimePerItemInHours	float	The average processing time in hours per item	15	Positive floating point number

An attribute/measure catalogue is also developed in PW - this evaluation metrics catalogue will be merged with the PW catalogue in year-3.

9.1 Binary evaluation method (FMeasure)

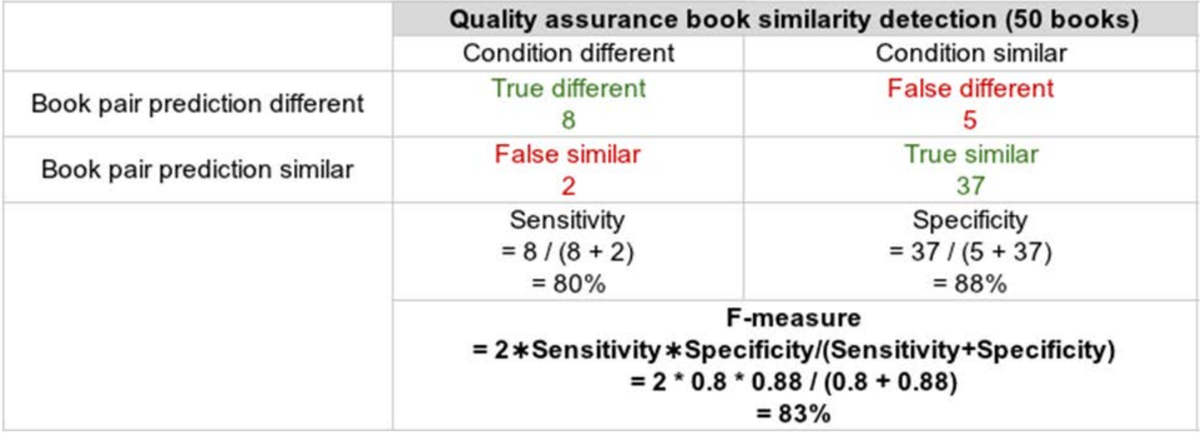
We use *sensitivity* and *specificity* as statistical measures of the performance of the binary classification test where

$$\textit{Sensitivity} = \Sigma \text{ true different} / (\Sigma \text{ true different} + \Sigma \text{ false similar})$$

and

$$\textit{Specificity} = \Sigma \text{ true similar} / (\Sigma \text{ true similar} + \Sigma \text{ false different})$$

and the F-measure is calculated on this basis as shown in the table below:



	Quality assurance book similarity detection (50 books)	
	Condition different	Condition similar
Book pair prediction different	True different 8	False different 5
Book pair prediction similar	False similar 2	True similar 37
	Sensitivity = 8 / (8 + 2) = 80%	Specificity = 37 / (5 + 37) = 88%
	F-measure **= 2∗Sensitivity∗Specificity/(Sensitivity+Specificity)** **= 2 * 0.8 * 0.88 / (0.8 + 0.88)** **= 83%**	

This is one suggested way which is nicely applicable if we test for binary correctness of calculations, i.e. it is applicable for characterisation and QA.

10 Appendix B – Evaluation Templates

10.1 Template for platform and system level evaluations

Field	Data type	Value	Description
Evaluation seq. num.	int	1	Use only of sub-sequent evaluations of the same evaluation is done in another setup than a previous one. In that case copy the Evaluation specs table and fill out a new one with a new sequence number. For the first evaluation leave this field at "1"
Evaluator-ID	email		Unique ID of the evaluator that carried out this specific evaluator.
Evaluation description	text		Textual description of the evaluation and the overall goals
Evaluation-Date	DD/MM/YY		Date of evaluation
Platform-ID	string		Unique ID of the platform involved in the particular evaluation - see Platform page included below
Dataset(s)	string		Link to dataset page(s) on WIKI For each dataset that is a part of an evaluation make sure that the dataset is described here: Datasets
Workflow method	string		Taverna / Command line / Direct hadoop etc...
Workflow(s) involved	URL(s)		Link(s) to MyExperiment **if applicable**
Tool(s) involved	URL(s)		Link(s) to distinct versions of specific components/tools in the component registry **if applicable**
Link(s) to Scenario(s)	URL(s)		Link(s) to scenario(s) **if applicable**

Evaluation points

Metrics must come from / be registered in the Metrics Catalogue

Metric	Baseline definition	Baseline value	Goal	Evaluation 1 (date)	Evaluation 2 (date)	Evaluation 3 (date)

10.2 Template for component level evaluations

Field	Data type	Value	Description
Evaluation seq. num.	int	1	Use only of sub-sequent evaluations of the same evaluation is done in another setup than a previous one. In that case copy the Evaluation specs table and fill out a new one with a new sequence number. For the first evaluation leave this field at "1"
Evaluator-ID	email		Unique ID of the evaluator that carried out this specific evaluator.
Evaluation description	text		Textual description of the evaluation and the overall goals
Evaluation-Date	DD/MM/YY		Date of evaluation
Dataset(s)	string		Link to dataset page(s) on WIKI For each dataset that is a part of an evaluation make sure that the dataset is described here: Datasets
Workflow method	string		Taverna / Command line / Direct hadoop etc...
Workflow(s) involved	URL(s)		Link(s) to MyExperiment **if applicable**
Tool(s) involved	URL(s)		Link(s) to distinct versions of specific components/tools in the component registry **if applicable**
Link(s) to Scenario(s)	URL(s)		Link(s) to scenario(s) **if applicable**

Technical setup

Field	Data type	Value	Description
Description	String		Human readable description of the "platform" - e.g. Bjarne' s Linux PC
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 Gbytes	integer		Total amount of RAM on all nodes
Operating System	String		Linux (specific distribution), Windows (specific distribution), other?
Storage system/layer	String		NFS, HDFS, local files, ?

Evaluation points

Metrics must come from / be registered in Metric Catalogue

Metric	Baseline definition	Baseline value	Goal	Evaluation 1 (date)	Evaluation 2 (date)	Evaluation 3 (date)

10.3 Template for Plato Evaluation

Step	Description	Status
1. Consult TUWIEN	write to becker@ifs.tuwien.ac.at	
2. Follow evaluation workflow in Plato (Plato manual)	1. Define requirements 2. Evaluate alternatives 3. Analyse results 4. (Build Preservation Plan)	
3. Generate report in Plato	see Plato manual or consult TUWIEN	
4. Upload and link to report on WIKI	Link Plato report here	

10.4 Template for Platforms

[name]

Field	Data type	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 Gbytes	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 Gbytes 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, ?
Network layer between nodes	String		Speed of network interfaces, general network speed

11 Appendix C – Evaluations

11.1 Large Scale Digital Repository Testbed (LSDR)

11.1.1 LSDR-2

Evaluation specs platform/system level

Field	Data type	Value
Evaluation seq. num.	int	1
Evaluator-ID	email	william.palmer@bl.uk
Evaluation description	text	The migration of TIFF files to JP2, followed by validation of the new JP2 files using Jpylyzer. The evaluation is to test the processing speed, reliability and correctness of such a migration and the tools used.
Evaluation-Date	DD/MM/YYYY	06/11/2012
Platform-ID	string	Platform BL-0
Dataset(s)	string	30 master TIFF files from JISC1 19th Century Digitised Newspapers (465MB total)
Workflow method	string	Hadoop calling command line tools and Java code, one workflow per file. The code consists of two parts - a Java wrapper for Hadoop and a "workflow" style Java class that is executed once per map/file. A text file containing locations of input files is given as input to the wrapper. The wrapper code performs the following, once per input file/map: * Copies file to local temporary storage for processing (from HDFS) * Calls the "workflow" class * Stores outputs from the workflow class in HDFS * Queries the workflow class for success/failure of workflow and reports this in the final overall output from the wrapper (a CSV file: original name, success boolean, output filename) The "workflow" class performs the following: * Checksums the input file (Java code) * Extracts metadata from the input file (Exiftool) * Migrates the input file (OpenJPEG) * Extracts metadata from the output file (Exiftool) * Extracts Jpylyzer info from the output file (Jpylyzer) * Checks the Jpylyzer output against the Jpeg 2000 profile used to encode the file (Java code) * Generates a short report containing Jpylyzer’ s isValidJP2 and whether the Jpeg 2000 profiles match (Java code) * Checksums all files (Java code) * Zips all files with a BagIt style structure (Java code) * Output includes a log of all commands lines run, with stdout/stderr from

		each tool
Workflow(s) involved	URL(s)	
Tool(s) involved	URL(s)	Debian "testing" fairly up to date at time of test OpenJPEG - nb. That the 1.3 version in the Debian "testing" repositories does not work with TIFF input files. You need to build the 1.5.1 binaries from source. Hadoop 1.0.4 (Apache compiled .deb) Jpylyzer 1.6.3 (from GitHub, compiled using pyinstaller 2.0) Exiftool (from Debian testing)OpenJDK 6 (from Debian testing)
Link(s) to Scenario(s)	URL(s)	LSDRT2+Validating+files+migrated+from+TIFF+to+JPEG2000

Platform BL 0

Field	Data type	Value
Platform-ID	String	Platform BL 0
Platform description	String	This is a pseudo-distributed single-node Hadoop instance running on a virtual machine on our work laptops and is used for our development. Initial evaluation will be performed on this platform with the long term goal being to run against both experimental DPT platform and using the BL cluster.
Number of nodes	integer	1
Total number of physical CPUs	integer	1
CPU specs	string	1 Intel Core i5-2540M CPU @ 2.6GHz
Total number of CPU-cores	integer	1
Total amount of RAM in Gbytes	integer	2GB
average CPU-cores for nodes	integer	1
average RAM in Gbytes for nodes	integer	2GB
Operating System on nodes	String	Debian "testing", fairly current as of test date
Storage system/layer	String	HDFS on virtual disk.
Network layer between nodes	String	n/a

Evaluation points

Metric	Baseline definition	Baseline value	Goal	Evaluation 1 (06-11-

				2012)
NumberOfObjectsPerHour	**Processing speed** with shell script	50	1600**	87.4
ThroughputGbytesPerHour	**Processing speed** with shell script	0.766	25**	1.355
ReliableAndStableAssessment	**Reliability and correctness** The workflow completed successfully and no failures were encountered at runtime. However, there is an incompatibility with OpenJPEG and the BL j2k profile: when coder bypass is enabled the outputs of the files show compression artefacts. Also, one converted file failed to open and was corrupt, despite Jpylyzer assessing its headers as valid. This shows that Jpylyzer validation should not be used alone for checking the success or otherwise of the migration.		true	false
OrganisationalFit			true	
NumberOfFailedFiles	**Reliability** No files failed during the workflow. However, when visually reviewing files, one file was found that would not open in various programs, despite Jpylyzer assessing its headers as valid.		0	0*

Previous tests were run on the same platform, but with different data, to compare the relative times taken for the following methods of executing a single command line migration from TIFF to JP2 using OpenJPEG:

1. Batch file
2. Hadoop - Java class calling migration command line
3. Hadoop - Java class executing the migration command line in a Taverna workflow via Taverna command line tool
4. Hadoop - Java class executing the migration command line in a Taverna workflow via Taverna Server instance in Tomcat

When looking at average runtime per file this gave an indication of the average overhead per file for each method:

1. N/A (baseline)
2. 0.69s
3. 10.17s
4. 25.84s

** The goal values assume that we want to complete the migration of the JISC Newspapers collection (2.2 million images) over two months (60 days) and that the sample data we have used here are representative of the collection as a whole. These values are subject to change.

11.1.2 LSDR-3

Evaluation specs component level

Field	Data type	Value
Evaluation seq. num.	int	1
Evaluator-ID	email	william.palmer@bl.uk
Evaluation description	text	The migration of TIFF files to JP2 followed by validation of the new JP2 files using Jpylyzer and Matchbox. The evaluation is to test the processing speed, reliability and correctness of such a migration and the tools used.
Evaluation-Date	DD/MM/YY	06/11/2012
Platform-ID	string	Platform BL-0
Dataset(s)	string	30 master TIFF files from JISC1 19th Century Digitised Newspapers (465MB total)
Workflow method	string	Hadoop calling command line tools and Java code, one workflow per file. The code consists of two parts - a Java wrapper for Hadoop and a "workflow" style Java class that is executed once per map/file. A text file containing locations of input files is given as input to the wrapper. The wrapper code performs the following, once per input file/map: * Copies file to local temporary storage for processing (from HDFS) * Calls the "workflow" class * Stores outputs from the workflow class in HDFS * Queries the workflow class for success/failure of workflow and reports this in the final overall output from the wrapper (a CSV file: original name, success boolean, output filename) The "workflow" class performs the following: * Checksums the input file (Java code) * Extracts metadata from the input file (Exiftool) * Migrates the input file (OpenJPEG) * Extracts metadata from the output file (Exiftool) * Extracts Jpylyzer info from the output file (Jpylyzer) * Checks the Jpylyzer output against the Jpeg 2000 profile used to encode the file (Java code) * Extract features from input file (Matchbox) * Extract features from output file (Matchbox) * Compare SIFT data (Matchbox) * Compare Profile data (Matchbox) * Generates a short report containing Jpylyzer' s isValidJP2, whether the Jpeg 2000 profiles match and whether the Matchbox SIFT comparison resulted in a value >0.9 (Java code)

		* Checksums all files (Java code) * Zips all files with a BagIt style structure (Java code) * Output includes a log of all commands lines run, with stdout/stderr from each tool
Workflow(s) involved	URL(s)	NA
Tool(s) involved	URL(s)	Debian "testing" fairly up to date at time of test OpenJPEG - nb. That the 1.3 version in the Debian "testing" repositories does not work with TIFF input files. You need to build the 1.5.1 binaries from source. Hadoop 1.0.4 (Apache compiled .deb) Jpylyzer 1.6.3 (from GitHub, compiled using pyinstaller 2.0) Exiftool (from Debian testing)OpenJDK 6 (from Debian testing) OpenCV 2.4.2 (compiled from source) Matchbox (from GitHub, compiled from source)
Link(s) to Scenario(s)	URL(s)	LSDRT3 Validating Migrated Images 'Visually'

Platform BL 0

Field	Data type	Value
Platform-ID	String	Platform BL 0
Platform description	String	This is a pseudo-distributed single-node Hadoop instance running on a virtual machine on our work laptops and is used for our development. Initial evaluation will be performed on this platform with the long term goal being to run against both experimental DPT platform and using the BL cluster.
Number of nodes	integer	1
Total number of physical CPUs	integer	1
CPU specs	string	1 Intel Core i5-2540M CPU @ 2.6GHz
Total number of CPU-cores	integer	1
Total amount of RAM in Gbytes	integer	2GB
average CPU-cores for nodes	integer	1
average RAM in Gbytes for nodes	integer	2GB
Operating System on nodes	String	Debian "testing", fairly current as of test date
Storage system/layer	String	HDFS on virtual disk.
Network layer between nodes	String	n/a

Evaluation points

Metric	Baseline definition	Baseline value	Goal	Evaluation 1 (06-11-2012)
NumberOfObjectsPerHour	**Processing speed** with shell script	50	1600**	45
ThroughputGbytesPerHour	**Processing speed** with shell script	0.766	25GB**	0.697GB
ReliableAndStableAssessment	**Reliability and correctness** The migration completed successfully, success/failure of each of individual migration workflow was noted in the overall output. One file did not migrate to JP2 successfully and this outcome was identified in the output from the workflow and in the overall report. The same issues about OpenJPEG/BL profile were present in the output files as in LSDR2-1. The failed migration did not affect the rest of the migration, which completed successfully.		true	true
OrganisationalFit			true	N/A – not able to evaluate yet
NumberOfFailedFiles	**Reliability** One file failed during the migration. However, this did not stop the rest of the migration from completing and the failure was clearly identified in the outputs.		0	1*

** The goal values assume that we want to complete the migration of the JISC Newspapers collection (2.2 million images) over two months (60 days) and that the sample data we have used here are representative of the collection as a whole. These values are subject to change.

11.1.3 LSDR-6

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 loosing 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**	10	1000	18

	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.			
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 spacedue 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	161 in 3190 ~= 5% (test 2nd-16th October 2012)	.1%	0.412 % (5th-9th November 2012)

	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.			

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.

11.1.4 LSDR-11

Evaluation specs component level

Field	Data type	Value
Evaluation seq. num.	int	1
Evaluator-ID	email	sven.schlarb@onb.ac.at
Evaluation description	text	Matchbox evaluation applied to a data set of 40 books (~330 page images per book) from the Austrian Books online collection of the Austrian National Library. The performance of duplicate page detection is determined by the average runtime of Matchbox per book. The Taverna Workflow Workbench is used in batch processing mode without hadoop. Matchbox is executed on a server with 4 physical cores, and Taverna is configured to process 4 books in parallel. Additionally, an evaluation of the correctness of the Matchbox duplicate detection has been performed using a small sample of 7 books with ground indicating which pages should be identified as duplicate pages.
Evaluation-Date	DD/MM/YY	28/11/12
Dataset(s)	string	40 books from the Austrian Books online collection of the Austrian National

		Library
Workflow method	string	Taverna Workflow Workbench, batch processing using "tool" service components, 4 processes in parallel server with 4 physical cores (without hadoop).
Workflow(s) involved	URL(s)	http://www.myexperiment.org/workflows/3318.html
Tool(s) involved	URL(s)	Matchbox
Link(s) to Scenario(s)	URL(s)	LSDRT11 Duplicate image detection within one book

Technical setup

Field	Data type	Value
Description	String	FUE-L Rack Server at ONB
Total number of physical CPUs	integer	1
CPU specs	string	Intel(R) Xeon(R) CPU, E5540 @ 2.53GHz
Total number of CPU-cores	integer	4
Total amount of RAM in Gbytes	integer	12GB
Operating System	String	Ubuntu Linux Server 12.04.1 LTS
Storage system/layer	String	NFS

Evaluation points

Metric	Baseline definition	Baseline value	Goal	Evaluation 1 (28/11/12)
NumberOfObjectsPerHour	Number of books (~330 page images per book) processed per hour.	-	1	0.18
AverageRuntimePerItemInHours	Average runtime of processing one book in hours.	-	1	5.4
NumberOfFailedFiles	Number of book processings that failed in the workflow.	0	0	0
IdentificationCorrectnessInPercent	Average F-Measure (Precision and Recall combined)	-	98	96

11.2 Web Content Testbed (WCT)

11.2.1 WCT-1

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	DD/MM/YY	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

11.2.2 WCT-3

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

11.2.3 WCT-4

Evaluation specs platform/system level

Field	Data type	Value
Evaluation seq. num.	int	1
Evaluator-ID	email	markus.raditsch@onb.ac.at
Evaluation description	text	The workflow has been implemented as a native JAVA map/reduce application. It uses the Apache Tika™ 1.0 API (detector call) to detect the MIME type of the inputStream for each file inside the ARC.GZ container files. 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. Each test ARC.GZ file has a size of approximately 500MB and is the container for around 30000 files. The 100GB test sample (200 x 500MB) is a subset of the original data set produced by a web crawler at the ONB. Output of the map/reduce program is a MIME type distribution list of the

		analysed input, containing all identified MIME types plus the occurrence count for each identified MIME type. **Goal / Sub-goal:** Performance efficiency / Throughput) The result has been measured as GB/min/platform Reliability / Stability Indicators) The processing application has been implemented as a JAVA JAR map / reduce application) All needed components (program logic, Hadoop method implementations, dependencies, Apache Tika™ 1.0 JAR) are integrated) The result has been measured "manually" and reflected as a boolean value (true = met the requirements) Reliability / Runtime stability) Use Hadoop admin interface to identify failed tasks.) Use Hadoop output to identify dropped records / any reported errors.) The result has been measured as an integer value reflecting the number of identified run time failures.
Evaluation-Date	DD/MM/YY	28/08/12
Platform-ID	string	Platform ONB 1
Dataset(s)	string	100GB sub set of Austrian National Library - Web Archive
Workflow method	string	Hadoop map / reduce application implemented in JAVA (jar).
Workflow(s) involved	URL(s)	
Tool(s) involved	URL(s)	Hadoop cluster, tb-wc-hd-archd, Apache Tika™ 1.0 API
Link(s) to Scenario(s)	URL(s)	http://wiki.opf-labs.org/display/SP/WCT4+Web+Archive+Mime-Type+detection+at+Austrian+National+Library

Platform ONB 1

Field	Data type	Value
Platform-ID	String	ONB 1
Platform description	String	Experimental cluster (setup 06.2012). Cloudera CDH3u5. 8 (HT) cores per node. Using max. 7 cores for map / reduce slots (one for the OS). Map / reduce slots ratio 6 / 1.
Number of nodes	integer	5
Total number of physical CPUs	integer	5
CPU specs	string	Xeon X3440@2.53GHz Quad core CPU

Total number of CPU-cores	integer	40 Cores (5 * 8 Cores)
Total amount of RAM in Gbytes	integer	80GB (5 * 16GB)
average CPU-cores for nodes	integer	8 Cores
average RAM in Gbytes for nodes	integer	16 GB
Operating System on nodes	String	Ubuntu 10.04.04 LTS (64bit)
Storage system/layer	String	HDFS
Disk subsystem	String	2 x 1TB DISKs; configured as RAID0 => 2TB effective disk space
HDFS replication factor	integer	2
Network layer between nodes	String	The CONTROLLER and the NODEs are connected to a GBit high performance network switch (guarantees the full GBit performance for each port).
Controller: CPU specs	String	2 x Xeon E5620@2.40GHz Quad core CPU
Controller: RAM	integer	24 GB
Controller: Disk subsystem	String	3 x 1TB DISKs; configured as RAID5 => 2TB effective disk space

Evaluation points

Metric	Baseline definition	Baseline value	Goal	Evaluation 1 (28/08/12)
ThroughputGbytesPerMinute	Virtual machine, Ubuntu Linux, 2GB RAM, Core i5 2,5GHz (single Processor VM configuration), Taverna Workbench workflow, TIKA 0.7 in API mode.	0,08	5	16,17
ReliableAndStableAssessment	The workflow incorporates different technologies (script, jar, beanshell, Taverna, Unix tools) which makes it hard(er) to implement a reliable error handling (compared to a Java map/reduce implementation).	false	true	true
NumberOfFailedFiles	n/a (much smaller data set)	n/a	0	0

11.2.4 WCT-8

Evaluation specs platform/system level

Field	Data type	Value
Evaluation seq. num.	int	1
Evaluator-ID	email	markus.raditsch@onb.ac.at
Evaluation description	text	The web archiving team at the Austrian National Library produces information about the content of a web archive during the harvesting process. The result

		is stored as huge text files. Each line of the log holds the meta data of one object. The application reads the file content line by line, extract the mime type "item 10 (Subtype)" and count all occurrences. **Goal / Sub-goal:** Performance efficiency / Throughput) The processing of these text files is very time consuming and needs parallelized processing) The workflow uses text files produced by the web crawler) The hadoop split size has been 64MB (the default value)) The result has been measured as GB/min/platform Reliability / Stability Indicators) No external tools are used) The processing application has been implemented as a JAVA JAR map / reduce application) All needed components (program logic, Hadoop method implementations, dependencies) are integrated) The result has been measured "manually" and reflected as a boolean value (true = met the requirements) Reliability / Runtime stability) Use Hadoop admin interface to identify failed tasks.) Use Hadoop output to identify dropped records / any reported errors.) The result has been measured as an integer value reflecting the number of identified run time failures.
Evaluation-Date	DD/MM/YY	20/08/12
Platform-ID	string	Platform ONB 1
Dataset(s)	string	Austrian National Library - Web Archive
Workflow method	string	Hadoop map / reduce application implemented in JAVA (jar).
Workflow(s) involved	URL(s)	
Tool(s) involved	URL(s)	
Link(s) to Scenario(s)	URL(s)	http://wiki.opf-labs.org/display/SP/WCT8+Huge+text+file+analysis+using+hadoop

Platform ONB 1

Field	Data type	Value
Platform-ID	String	ONB 1
Platform description	String	Experimental cluster (setup 06.2012). Cloudera CDH3u5.

		8 (HT) cores per node. Using max. 7 cores for map / reduce slots (one for the OS). Map / reduce slots ratio 6 / 1.
Number of nodes	integer	5
Total number of physical CPUs	integer	5
CPU specs	string	Xeon X3440@2.53GHz Quad core CPU
Total number of CPU-cores	integer	40 Cores (5 * 8 Cores)
Total amount of RAM in Gbytes	integer	80GB (5 * 16GB)
average CPU-cores for nodes	integer	8 Cores
average RAM in Gbytes for nodes	integer	16 GB
Operating System on nodes	String	Ubuntu 10.04.04 LTS (64bit)
Storage system/layer	String	HDFS
Disk subsystem	String	2 x 1TB DISKs; configured as RAID0 => 2TB effective disk space
HDFS replication factor	integer	2
Network layer between nodes	String	The CONTROLLER and the NODEs are connected to a GBit high performance network switch (guarantees the full GBit performance for each port).
Controller: CPU specs	String	2 x Xeon E5620@2.40GHz Quad core CPU
Controller: RAM	integer	24 GB
Controller: Disk subsystem	String	3 x 1TB DISKs; configured as RAID5 => 2TB effective disk space

Evaluation points

Metric	Baseline definition	Baseline value	Goal	Evaluation 1 (20/8/2012)
ThroughputGbytesPerMinute	Serial processing using bash scripts, unix tools and self written java helper tools. Quad core processor 2,66GHz.	0,35	5	11,93
ReliableAndStableAssessment	The baseline workflow incorporates different technologies (script, jar, Unix tools) which makes it hard(er) to implement a reliable error handling (compared to a Java map/reduce implementation).	false	true	true
NumberOfFailedFiles	Failing on the single input file can be monitored.	0	0	0

11.3 Research Data Set Testbed (RDST)

11.3.1 RDST-1

Evaluation specs component level

Field	Data type	Value
Evaluation seq. num.	int	1
Evaluator-ID	email	holly.zhen@stfc.ac.uk
Evaluation description	text	Characterisation tool for NeXus files - NeXus file format validation - Metadata extraction
Evaluation-Date	DD/MM/YY	09/11/2012
Dataset(s)	string	OPF STFC scientific datasets
Workflow method	string	- script - Java code calling a command line tool
Workflow(s) involved	URL(s)	n/a
Tool(s) involved	URL(s)	- NeXus Data Format Windows Distribution Kits - To extract metadata from NeXus files, the command line tool requires a XML mapping file which is written by ISIS for each instrument. They are still working on producing the mapping files for all of their instruments.
Link(s) to Scenario(s)	URL(s)	General Scientific Data Handling Scenarios

Local setup

Field	Data type	Value
Description	String	Windows
Total number of physical CPUs	integer	1
CPU specs	string	2nd generation Intel® Core™ i5-2557M processor with Intel® Turbo Boost Technology 2.0
Total number of CPU-cores	integer	1
Total amount of RAM in Gbytes	integer	6
Operating System	String	Windows 7 Professional 64
Storage system/layer	String	local files

Evaluation points

Metric	Baseline definition	Baseline value	Goal	Evaluation 1 (date) 09/11/2012

ThroughputGbytesPerMinute	Instrumental data from older instruments is stored in raw format rather than nexus. The raw format contains less useful metadata. Nexus files from the EMU instrument were used for this evaluation. The revised evaluation figure was obtained using the whole set of testing data, instead of a filtered subset originally used. Further improvements will be considered. As of Nov 2012, ISIS has approximately 3.5T of nexus files.	n/a	70G	65.7G

11.3.2 RDST-2

Evaluation specs component level

Field	Data type	Value
Evaluation seq. num.	int	1
Evaluator-ID	email	erica.yang@stfc.ac.uk
Evaluation description	text	Raw to Nexus format migration - *STFC ISIS facility* The typical format of these files are RAW or NeXus. NeXus is an international standard for neutron and synchrotron communities. RAW is facility specific: many historic data files are in this format. Increasingly, NeXus format is being adopted as the standard format for instrument data.
Evaluation-Date	DD/MM/YY	14/11/2012
Dataset(s)	string	OPF STFC scientific datasets
Workflow method	string	- command line - Java
Workflow(s) involved	URL(s)	n/a
Tool(s) involved	URL(s)	-NeXus Data Format Windows Distribution Kits -raw2nexus as part of the Mantid software framework (http://www.mantidproject.org/Main_Pag)
Link(s) to Scenario(s)	URL(s)	General Scientific Data Handling Scenarios

Technical setup

Field	Data type	Value
Description	String	Windows
Total number of physical CPUs	integer	1
CPU specs	string	2nd generation Intel® Core™ i5-2557M processor with Intel® Turbo Boost Technology 2.0
Total number of CPU-cores	integer	1
Total amount of RAM in Gbytes	integer	6
Operating System	String	Windows 7 Professional 64
Storage system/layer	String	local file system

Evaluation points

Metric	Baseline definition	Baseline value	Goal	Evaluation 1 (date) 14/11/2012
ThroughputGbytesPerMinute	The evaluation is completed on a single machine. A nexus file is created from a RAW file which contains the data, and a collection of log (text) files containing sample environment data. As of Nov 2012, ISIS has roughly *16.5Tb* of RAW and log files. With the evaluated value of 1.73Gb/min, it would take about *7 days* to process *16.5Tb* of data. Our projected goal is to achieve it in a day,	n/a	12Gb	1.73Gb
NumberOfObjectsPerHour	With an evaluated throughput of 1.73Gb/min, *we* expected the NumberOfObjectsPerHour to be much higher. This could be due to the varied size of files. Log files are typically very small, some of them can be as small as *1kb* but the *RAW* files can be well *over 10Gb.* On average 6 log files are needed with one RAW file to create on nexus file. The number of log files required differs from	n/a	15,300	1152

	instrument to instrument. As of Nov 2012, ISIS has roughly *11,000,000* files, of which *9,500,000* log(txt) files and *1,500,000* RAW files. With the evaluated value of *1152/hr*, it would take about *13 months* to process the whole set of files. Taking into consideration that we have a lot of very small files, and using Hadoop for parallelisation, we have projected a conservative goal of achieving it in a month.			