Characterisation technology, Release 2 + release report

Deliverable D9.2 of the SCAPE Project

Authors

Per Møldrup-Dalum (The State & University Library Aarhus), Lynn Marwood (The British Library), Sven Schlarb (The Austrian National Library), Alan Akbik (Technische Universität Berlin), Ivan Vujic (Microsoft Research Limited), Carl Wilson (Open Planets Foundation)

May 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 is the second iteration of the SCAPE D9 deliverable, D9.2 building upon the work presented in the first iteration. It describes the activities of the Characterisation Components workpackage during the second year of the SCAPE project. These activities are separated into four sections:

- The evaluation of the performance of 2 format identification tools, DROID and Tika, chosen after the first iteration, across the Govdocs1 test corpus in a Hadoop Map/Reduce, parallel environment, the SCAPE platform.
- The creation and publication of a data set showing how the results of format identification tools and their individual versions have varied over time, and the application of emerging Open Linked data standards to the curation and publication of SCAPE experimental data.
- The development of a tool that extracts domain specific, semantic information from harvested web pages, and its application to a real world use case.
- Microsoft Research's development of a set of Azure platform services to batch characterise, migrate, compare, and quality assure a wide variety of document formats.

Each activity group is covered in more detail. The section on Identification & Feature Extraction tools describes the experiments carried out to evaluate Apache Tika at The British Library, and DROID at The Austrian National Library, with regard to the feasibility of executing them in parallel environments. The details of the Hadoop cluster hardware and configuration are given for each experiment, which are both performed on the same test corpora used in the first iteration, the 1 million file Govdocs1 corpora. The evaluations demonstrate that real performance gains are realised executing the tools as Hadoop Map Reduce jobs. They also show consideration must be given to the configuration of Hadoop jobs, specifically the size of the individual Map tasks. Small Map tasks, lead to more Map tasks and degrading performance, but by ensuring Map tasks carry a significant workload, performance gains of an order of magnitude may be possible.

The Data Publication Platform (DPP) is described and its work divided into the Results Evaluation Framework (REF) technical activities, and a set of curation and publication guidelines that are been developed using the principles of linked open data. The report details the REF work to generate, process and publish format identification data for 5 different tools run across a subset of the test corpora. The curatorial activities of the Data Publication Platform are also described, using the Tika identification data covered in this report as an example. This section concludes that the coming years DPP work may be part of the sustainability work package.

Next the work of the Technical University of Berlin developing a tool that mines web archives for domain specific semantic information is presented. The crawl process and information extraction algorithms are described, followed by a case study with the National Library of Netherlands where the tool was used to gather publisher and journal metadata. The possible application of the tool to supply preservation watch metadata is discussed.

The Azure based document toolset developed by Microsoft Research is covered in the final activity section. This first describes the architecture implemented, and the supporting technologies. It then describes the different feature extraction and document comparison services developed, using screen shots of the Silverlight web application.

Finally a conclusion confirms the performance gains obtained by parallelising format identification tools, and tries to look at areas where investigation might yield improved performance. A roadmap for the workpackage activities for the coming year is outlined.

Table of Contents

1 Introduction

This report was created by SCAPE Work Package 9 (PC.WP.1), Preservations Components — Characterisation Components. The report documents work package activities from March 2012 until April 2013. The report builds upon previously published documents by this work package: "WP9 evaluation framework for characterisation tools" (van der Knijff, 2011), "Evaluation of characterisation tools" (Wilson, 2011) "Characterisation technology, Release 1 + release report" (Møldrup-Dalum M. R., 2012), and "PC.WP1 checkpoint CP070" (Møldrup-Dalum W. V., 2013).

1.1 Objectives

The Description of Work (SCAPE, 2010) describes this work package as having an overall coherency between the objectives assigned and the methods for attaining these objectives. During the last year it became apparent that this coherency only exists at a very high abstraction level. Therefore, in working towards these objectives, the work has been split into separate strands covering three subjects:

- Characterisation and identification of digital objects on the SCAPE platform.
- Characterisation and identification of office documents on the Microsoft Azure platform.
- Automated feature extraction of high volume text corpora.

In addition, the work of implementing a reference data publication platform has been moved from the Action Services Components work package to this one, adding another unique strand. These four strands cover very different areas of digital preservation and this report mirrors this division. The four strands are outlined in the following subsections.

1.1.1 Identification Tools on the SCAPE Platform

The characterisation of large collections of digital objects requires tools that perform efficient, high quality characterisation and that can be executed on the SCAPE platform. This report describes the evaluation of two identification tools Apache Tika and DROID with regard to:

- The feasibility of adapting them to the SCAPE platform.
- How they actually perform on the Hadoop platform.

The motivation behind the selection of these two tools is described in (Møldrup-Dalum M. R., 2012) and (Møldrup-Dalum W. V., 2013). These two documents also lay out intended objectives for the work described in this report with regard to the characterisation tools on the SCAPE platform. As can been seen in this report the actual work differs from the intended. This gap is a consequence of our decision to focus effort on evaluating the performance of the tools on the Hadoop platform. Such an evaluation is a necessary step prior to adapting a tool for the SCAPE platform. In this work package we distinguish between the SCAPE platform and the Hadoop platform, even though the SCAPE platform is based on the Hadoop platform. This distinction separates bespoke adaptation of a tool for execution as Hadoop Map Reduce task, and using the SCAPE adaption tools. This workpackage has concentrated on executing the tools on the Hadoop platform.

Apache Tika was evaluated on a Hadoop instance at the British Library while DROID was evaluated on a separate Hadoop instance at the Österreichische Nationalbibliotek. These two Hadoop deployments are described in detail in their respective sections. Both tools are evaluated using the Govdocs1 corpora which is described in section 2.1.2. This data set was also used for the initial

evaluation as reported in the D9.1 "Characterisation technology, Release 1 + release report" (Møldrup-Dalum M. R., 2012).

1.1.2 Data Publication Platform

The Data Publication Platform (DPP) is an adjunct to the work package, intended to address the publication, discoverability, and longevity of the experimental data created by the SCAPE project. The SCAPE project has examined the principles of linked data[1], and considered them well suited to the publication and preservation of experimental data sets. Consequently work has begun to apply them to the datasets used, and results set produced by the project. There are two parts to the DPP effort:

1 Software development, both putting together existing components to process and store data, and on visualisation of the results.

2 Curatorial and advocacy activities that document best practises for the online publication of datasets while assisting the tool developers and the testbeds with implementing the practices.

1.1.3 Information Extraction on Text and Web Corpora

We have designed and implemented a system that analyses medium to large collections of text documents to gather information relevant to preservation watch. The system consists of a focused crawler and an unsupervised information extraction system. We have performed a series of evaluations and have applied the system to a use case from the National Library of the Netherlands.

1.1.4 Characterization of office documents on MSR Azure platform

Microsoft Research is a technology partner in SCAPE. Thus our contribution is in research and technology development that meet the need and requirements of memory institutions rather than needs of Microsoft as a technology company. As part of this task, we reviewed the current state-of-the-art techniques and tools in document characterization, with the focus on documents produced by *office productivity tools* such as Adobe publishing tools, Microsoft Office suite, Open Office applications, and similar. Based on that, we selected a set of tools to include in the services we are designing and implementing.

1.2 Terminology

In order to avoid any confusion regarding terminology, this report uses the same terminology as the Evaluation of Characterisation Tools report (Wilson, 2011) that states the following:

In practice the term "characterisation" is often used to indicate quite different things. To avoid any confusion, this document follows the terminology used in the JHOVE2 project. Here, characterisation is loosely defined as the process of deriving information about a digital object that describes its character or significant nature2. This process is subdivided into four aspects:

- *identification—the process of determining the presumptive format of a digital object;*
- *feature extraction—the process of reporting the intrinsic properties of a digital object that are significant to preservation planning and action;*

[1] http://www.w3.org/DesignIssues/LinkedData

[2] JHOVE2 actually uses 2 separate definitions: "(1) Information about a digital object that describes its character or significant nature that can function as an surrogate for the object itself for purposes of much preservation analysis and decision making. (2) The process of deriving this information. This process has four important aspects: identification, feature extraction, validation, and assessment."

- *validation—the process of determining the level of conformance of a digital object to the normative syntactic and semantic rules defined by the authoritative specification of the object's format;*
- *assessment—the process of determining the level of acceptability of a digital object for a specific purpose based on locally defined policy rules.*

In particular we will adopt this terminology and the definitions for Characterisation, Identification, and Feature Extraction.

2 Identification and Feature Extraction Tools

2.1 Selected tools: Apache Tika and DROID

We have previously selected Apache Tika and DROID as the main candidates for performing identification and, in the case of Tika feature extraction. The selection is documented in the first iteration of this deliverable. This selection is substantiated first in (Wilson, 2011) and then further in (Møldrup-Dalum M. R., 2012). The latter report describes the two tools in detail, including how to obtain, deploy, and use them. The same report evaluates the two tools with regard to functional correctness, coverage, and briefly covers performance.

2.1.1 Planning the Evaluation

The intention of the experiments described in this section is to investigate how Apache Tika and DROID might perform on the SCAPE platform, rather than assessing format coverage or functional correctness. Those subjects were covered in the first iteration of this report (SCAPE D9.1), leading to the selection of the two tools. A re-evaluation of functional correctness might well be required if either, or both tools are capable of performing adequately on the Hadoop and SCAPE platforms. This would be wise both to test the latest version of the tools, and to ensure that the transfer to a parallel execution environment hasn't affected the function of the tools.

2.1.2 The Govdocs1 Corpus

The evaluations were carried out against the Govdocs1[3] test corpus previously used by this work package during the SCAPE project (Møldrup-Dalum M. R., 2012). This corpus is a freely available set of about 1 million files. Forensic Innovations, Inc. have provided the ground truth identification data for this corpus[4]. The Govdocs1 corpus is described in more detail in (Møldrup-Dalum M. R., 2012). The corpus was favoured as it was used in the initial assessment of the tools for the first iteration of the report giving some continuity. The use of a freely available corpus means that:

- All SCAPE partners are free to obtain the test corpus reproduce the evaluation.
- Evaluations performed on the corpora are directly comparable to other results.

2.2 DROID

The following describes a feasibility study regarding the integration and use of the DROID[5] file format identification tool with the SCAPE platform. After introducing the DROID tool and prerequisites to enable the tool execution on the SCAPE platform, an evaluation experiment provides more insights into the applicability of the tool in a large-scale execution environment.

[3] http://digitalcorpora.org/corpora/files
[4] This ground truth is accessible at: http://digitalcorpora.org/corp/files/govdocs1/groundtruth-fitools.zip
[5] DROID (Digital Record Object Identification), Version 6.1, http://digital-preservation.github.io/droid/

2.2.1 The DROID software tool

The DROID software tool is developed by The National Archives (UK) to perform automated batch identification of file formats. Identification is carried out by matching known byte sequences or signatures that are read from a signature file that is curated, regularly updated and published by The National Archives (UK). For the evaluations described here version 6.1 of DROID was used with version 67 of the signature file.[6]

2.2.2 DROID API and Implications for Hadoop Integration

2.2.2.1 Processing many small files using Hadoop

As a matter of fact, the sizes of file instances to which we can possibly apply file format identification vary significantly from several Kilobytes to many Terabytes and even more. In the context of this DROID experiment we focus on using DROID to perform format identification of many small files as can be found in an office or web content context. These files are usually smaller than 50 Megabytes. On a typical Hadoop system used for distributed processing, we face Hadoop's "Small Files Problem" (Cloudera, 2009).

Briefly, the files we want to process are too small for presentation as direct input for the Map function. In fact loading 1000 small files into Hadoop File System (HDFS) - which requires quite some time - and defining them as input to our Hadoop Job would cause the Hadoop JobTracker to create 1000 Map tasks. Given the task creation overhead this would result in a bad processing performance.

One approach to overcoming this problem is to put all the small files into one large SequenceFile[7], which is an input format that can be used to aggregate content using some identifier as key and the byte sequence of the content as value. Many Web Archives take this approach as they harvest small files into larger archive files as a business as usual activity.

Another approach, chosen in this experiment is to provide references to all files to be processed in a text file, and then use this text file as input for the Hadoop job. This requires that all worker nodes of the cluster can access the referenced file paths, e.g. by adding mount points to the cluster nodes so that a file path references the same file on each cluster node. The Hadoop framework does not generate one task per file, but the size of the task depends on the split size of the input text file, i.e. all file paths contained in a 64MB section (default configuration) of the text file.

2.2.2.2 DROID's support for processing input streams

Some scenarios require the processing of input streams, for example if data has to be read from archive formats, such as zip files, and the byte sequence is available for immediate processing. A stream can be made temporarily persistent if a file instance is required, but this entails additional I/O operations, which might negatively impact performance.

Although the DROID core[8] API offers methods that allow reading from stream objects[9], it is important to note that file system metadata is used to access the file instance, in case it is needed for the format identification process. One example is the use of a file extension, one indicator of format. Input streams don't require names, and the extension of the byte stream may not be known.

Ideally, the processing could then be performed on the byte sequence or input stream with no need to access a file instance on the file system. However, this is not possible with the current DROID

[6] http://www.nationalarchives.gov.uk/documents/DROID_SignatureFile_V67.xml
[7] http://wiki.apache.org/hadoop/SequenceFile
[8] https://github.com/digital-preservation/droid/tree/master/droid-core
[9] http://docs.oracle.com/javase/6/docs/api/java/io/InputStream.html

implementation, because it assumes that the object is available as a file instance and relies upon having to access the file system.

For this reason, the British Library's Web Archive has created the nanite software project[10], a DROID derivative, which allows identification to be performed on input streams.

2.2.2.3 DROID initialisation in a parallel environment

As already mentioned, DROID uses a signature file for file format identification. This file must be loaded and initialized by DROID before performing identification.

Initialising the signature file takes a significant time compared to the individual identification tasks. To give a rough idea, illustrated in Figure 1, initialising Version 67 of the DROID signature file (about 70 Kilobytes) took 6047 milliseconds while identification took 278 milliseconds. These measurements were taken on a development machine (Dual core 2.8 GHz, 2 Gigabyte RAM).

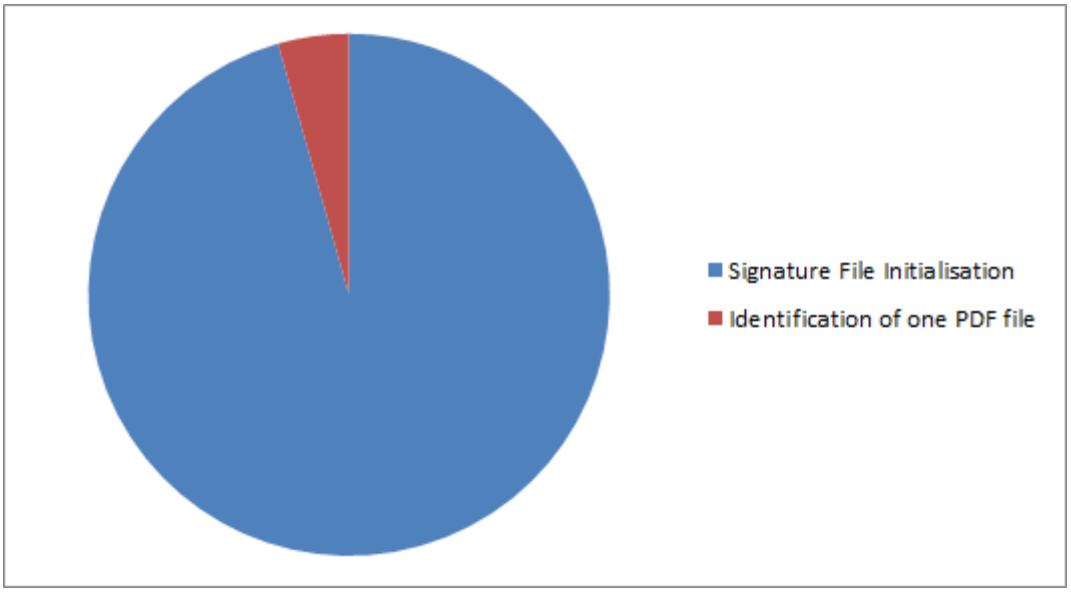

Figure 1 Time needed for initialising the signature file compared to the execution time of identifying one single PDF file.

The SCAPE Platform is built upon the Hadoop framework which means that each task runs in it's own Java Virtual Machine in order to isolate it from other running tasks. One consequence of this is that it is not possible to share a Java object, in this case an initialised signature file instance between the different tasks – even tasks running on the same machine, but on different cores.

The time needed to parse the signature file must be taken into account when configuring the number of files to be identified in a single task. Depending upon the input type, this can be done by parameterising the split size, allowing control over the number of tasks created for a Hadoop job.

For the example shown in Figure 1 this means that identifying 20 identical files would require more than half of the execution time to initialise the signature file. In consequence, a minimum of at least 200 file identifications should be processed per signature file initialisation, meaning less than 10% of the execution time is taken on initialisation as shown in Figure 2.

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

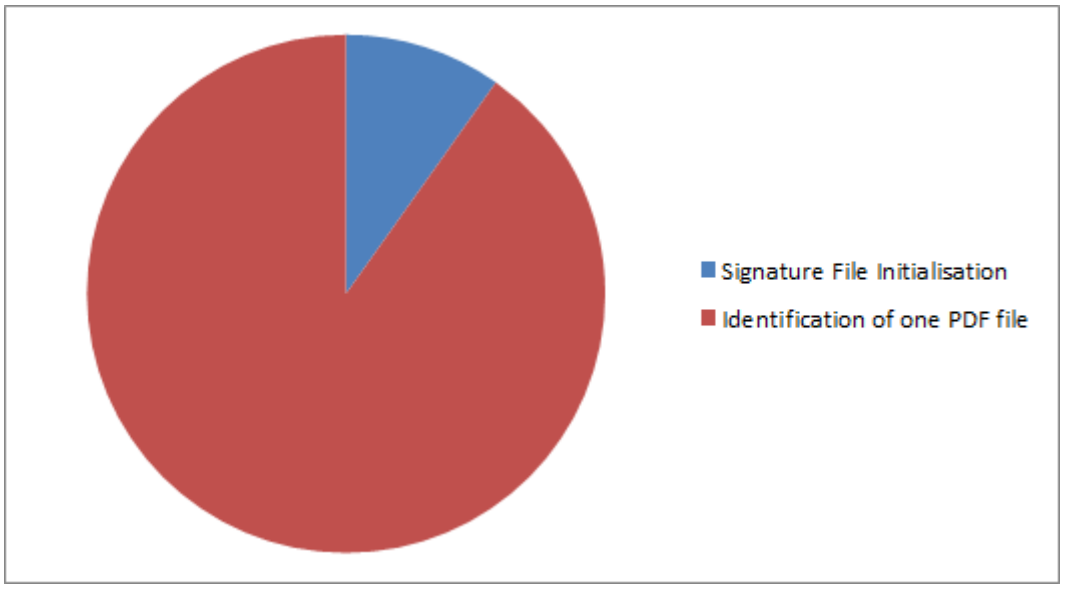

Figure 2 If running the identification of 200 PDF files in one task, initialising the signature file does not carry so much weight in the overall execution time.

2.2.3 Experiment using DROID on the SCAPE Platform

This section describes an experiment that tests file format identification against the Govdocs1 corpus (see section 2.1.2), using the SCAPE Platform instance at the Austrian National Library. The corpus was downloaded March 2013 and contained 986277 file instances.

The goal of the experiment was to evaluate the performance and reliability of the DROID software tool using the SCAPE Platform with Hadoop. Functional correctness of the DROID tool is not a focus of this evaluation.

First the hardware environment and Hadoop configuration is described. Next the process of preparing the data and making it available on the cluster is outlined. The next part describes the DROID identification. Finally, the result of the experiment and an outlook to further work is provided.

2.2.3.1 Experimental Hardware Environment and Hadoop Configuration

The experimental cluster available at the Austrian National Library consists of one controller machine (Master) and 5 worker machines (Slaves). The master node has two quadcore CPUs (8 physical/16 HyperThreading cores) with a clock rate of 2.40GHz and 24 Gigabyte RAM. The slave nodes have one quadcore CPU (4 physical/8 HyperThreading cores) with a clock rate of 2.53GHz and 16 Gigabyte RAM.

The cluster machines are connected by a gigabit Ethernet and the cluster network has a shared file server that stores global persistent data, accessed by the slave nodes as needed.

Regarding the basic Hadoop configuration, in the current set-up, five processor cores of each machine have been assigned to Map Tasks, two cores to Reduce tasks, and one core is reserved for the operating system. This is a total of 25 processing cores for Map tasks and 10 cores for Reduce tasks, which means that, in principle, depending on the type of data processing, 25 map tasks and 10 Reduce tasks can run in parallel.

2.2.3.2 Making data available for processing on the cluster

The Govdocs1 corpus was downloaded as 1000 tar-files with `gzip` compression (`.tar.gz`). They have been unpackaged on the file server in order to make the files accessible to all cluster nodes.

A 51 Megabyte large text file containing all paths to the individual files was created and loaded into HDFS to serve as input for the Hadoop jobs. The advantage of this approach is that it employs Hadoop's default TextInputFormat[11].

Please note that the size of the text file is smaller than Hadoop's default split size (64 Megabyte). This means that only a single task, running on one single core, would be created for processing the complete list with no benefit of running tasks in parallel at all.

The number of records processed in one task can be controlled by the Hadoop parameter:

```
mapred.line.input.format.linespermap
```

If setting this parameter does not have the desired effect, it is possible to take advantage of Hadoop's default behaviour to create at least one task per input file. Using the Unix command

```
split -a 4 -l NUMLINES govdocs1_absolute_file_paths.txt
```

the complete text file containing all paths can be split into different files with the desired number *NUMLINES* of file path lines per file which corresponds to the desired number of file identifications to be processed per task.

It's important to keep an eye on the number of records processed per task because this directly influences the task run time. As a rule of thumb it is recommended to ensure that "each task runs for at least 30-40 seconds" (Cloudera, 2009).

2.2.3.3 DROID Identification Hadoop Job[12]

As Table 1 shows, by loading 4932 text files into HDFS as input for the Hadoop job, 4932 map tasks were created each processing 200 file identifications in 1 hour, 1 minute, and 5 seconds. By loading 1233 text files, 1233 map tasks were created each processing 800 file identifications in 58 minutes and 6 seconds.

Table 1 Hadoop job runtimes for the Droid identification of the Govdocs1 corpus

Number of map tasks	Records per map task	Hadoop Job runtime (hh:mm:ss)	Average runtime per map task incl. DROID Initialisation
4932	200	01:01:05	18 seconds
1233	800	00:58:06	71 seconds

Another observation is that there is not much difference regarding the overall job execution time comparing the two job configurations. Even though the average task runtime is much higher in the example processing 800 file format identifications per task, this does not have an important impact on the overall execution time.

The Reducer of the Hadoop job is not described as it simply counts the file instances per format identification map task output in order to produce an aggregated result.

[11] http://hadoop.apache.org/docs/current/api/org/apache/hadoop/mapreduce/lib/input/TextInputFormat.html
[12] https://github.com/shsdev/droid-identify-hadoopjob

Regarding runtime stability, both jobs had task failures (9 and 10 failures respectively), which were rescheduled successfully, meaning the overall job execution was successful.

2.2.3.4 DROID Identification Results

Figure 3 gives an overview of the results of the DROID file format identification across the Govdocs1 corpus with format indicated by Pronom Unique Identifiers[13] (PUID) for more than 10000 file instances. See Table Droid identification result (Austrian National Library Cluster) in the Annex for the full identification results list.

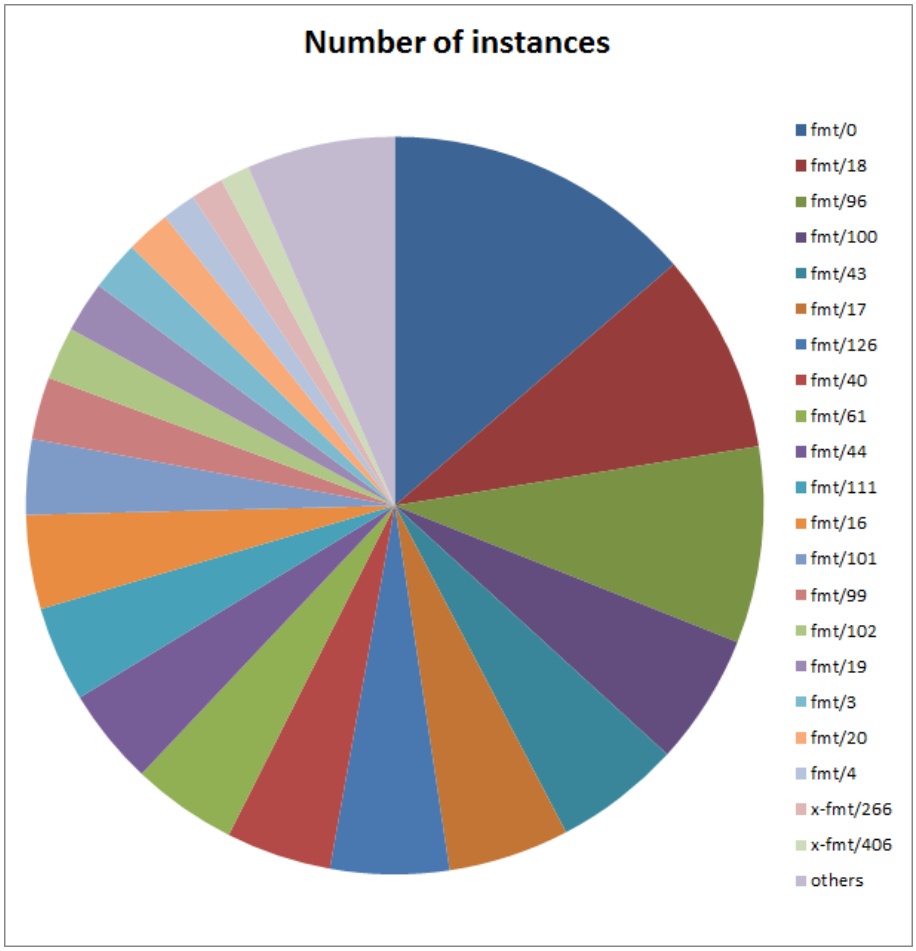

Figure 3 Diagram showing the DROID identification result for the Govdocs1 for the Pronom Unique Identifiers with more than 10000 instances.

A significant gap in the identification results (fmt/0 means that no format was assigned as the file could not be identified) can be observed. This is because no container signature file was used in this experiments the primary focus is not a functional evaluation, but rather to show the feasibility, and provide insights regarding performance and reliability for deploying DROID on the SCAPE Platform.

2.2.3.5 Comparing the DROID results to Apache Tika

A complete Tika evaluation considering different versions of the tool will follow in a dedicated section. But just to give an idea of how the SCAPE Platform execution of the two identification tools

[13] http://www.nationalarchives.gov.uk/aboutapps/pronom/puid.htm

compare to each other, the experiment was repeated with Apache Tika Version 1.0[14] in the same environment using the same data set leading to the results shown in Table 2.

Table 2 Hadoop job runtimes for the Tika 1.0 identification of the Govdocs1 corpus

Number of Input Files (map tasks)	File Path Lines Per File (Identifications per Task)	Hadoop Job runtime (hh:mm:ss)	Average runtime per map task
4932	200	00:19:35	5 sec
1233	800	00:16:56	20 sec

There is a significant difference regarding the overall performance, Apache Tika was approximately up to 4 times faster compared to DROID in this experiment.

See Table Tika identification result (Austrian National Library Cluster) in the Annex for the identification results list.

2.2.4 Conclusions

We have presented some points to consider when deploying the DROID file format identification tool on the SCAPE Platform. We have shown that for some scenarios, the Droid API is insufficient because file instances, rather than streams must be available for the identification process. We have also shown that Droid's initialisation of the signature file, required before performing format identification is not insignificant in relation to multiple identification tasks, and must be taken into consideration when setting up a Hadoop job.

We have described an experiment using the SCAPE Platform instance at the Austrian National Library. The Govdocs1 corpus was made available so that the cluster was able to perform parallelise format identification.

The identification was performed on the complete data set, with a low number of task failures that were successfully rescheduled by Hadoop.

The experiment shows the feasibility of DROID deployment on Hadoop and the SCAPE platform, primarily for parallel execution of the file format identification. As the data was made available on a file server accessible to the cluster the Hadoop File System was not used to hold or process the test corpus data.

[14] https://github.com/shsdev/tika-identify-hadoopjob

2.3 Apache Tika

This section describes the evaluation of Apache Tika[15] carried out at the end of the second year of the SCAPE project. The first part describes Apache Tika and the Tika Evaluation Tool developed to test it. The next section covers the platform on which the evaluation was executed and the data preparation process. The final sections describe the results of the evaluation and the conclusions reached.

The aim of the evaluation is to determine:

a) How well Tika format identification perform when executed on a Hadoop Map/Reduce cluster,

b) How well Tika identified a corpus of files (see section 2.1.2).

The Hadoop instance is described in the Experimental Hardware Environment and Hadoop Configuration section below.

At the end of year 1 a similar exercise was carried out using the Scape-Tool-Tester in a single threaded environment. Due to the need to perform year 2 evaluations on a Hadoop cluster, a new Tika evaluation tool was created, the Tika Evaluator Tool (described below).

2.3.1 Apache Tika and the Tika Evaluation Tool

Apache Tika is an open source toolkit to detect and extract metadata and structured text content from documents. For the purposes of this evaluation only the 'detect' module was used to identify the mime types of the files in the Govdocs1 corpus.

The Tika Evaluation Tool is a software tool, developed in Java by the British Library (BL). It consists of a number of modules providing functionality to carry out an end-to-end evaluation of Apache Tika. Specifically it:

- uses Tika to identify the mime type of every file in the Govdocs1 corpus and compares the result to the ground truth mime type;
- records the time taken to complete identification of the whole Govdocs1 corpus.

At the tools core is a map/reduce process that takes a list of file names as input. For each file listed it calls the Apache Tika `detect()` method to ascertain the mime type, and outputs the results to a CSV file.

The Tika Evaluator source code is published on the Open Planets Foundation Github organisation page.[16] https://github.com/openplanets/cc-benchmark-tests/tree/master/TikaEvaluatorTool

2.3.2 Tika Platform Evaluation Experiment

This section describes the experiment to perform file identification across the Govdocs1 corpus (see section 2.1.2) using the Apache Tika tool, running on a Hadoop cluster at the British Library.

It comprises four parts; the first describes the evaluation environment, followed by a description of how the data was prepared and loaded into HDFS. The next part describes the Hadoop map/reduce process and finally there is a brief description of how the evaluation software was executed.

[15] http://tika.apache.org
[16] https://github.com/openplanets/cc-benchmark-tests/tree/master/TikaEvaluatorTool

2.3.2.1 Experimental Hardware Environment and Hadoop Configuration

The Tika Evaluator Tool was run as a Cloudera Hadoop map/reduce process on a HP ProLiant DL 385p Gen8 host with 32 CPUs, 224 Gb of RAM and a clock rate of 2.295 GHz, using ESXi, a virtualisation tool, to run 32 virtual machines. The Hadoop configuration is a Cloudera (cdh4.2.0) Hadoop 30 node cluster consisting of a manager node, a master node (providing namenode and jobtracker services) and 28 slave nodes (providing data node and task tracking services), located at the British Library. Each node runs on its own virtual machine with 1 core, 500 GB of storage and 6 GB of RAM.

2.3.2.2 Data Preparation and HDFS

Two HDFS directories were set up, `TikaEvaluatorInput` to hold the input files used by the 'map' process, and `TikaEvaluatorOutput` to hold the output from the 'reduce' process.

The input files consist of:

- A file containing the names of all the files in the Govdocs1 corpus (one filename per line) called `FileList.txt`.
- A subdirectory, `FilesToIdentify`, into which was copied all of the test files in the Govdocs1 corpus[17].

The output files, generated by the 'reduce' process, are `part-r-00000`, `part-r-00001`, to `part-r-`*n* where '*n*' is the number of reduce processes, as specified by the number of reducers, in this case 14.

2.3.2.3 Map/Reduce Process

The 'map' process, shown in Figure 4, calls the Apache Tika 'detect' method that identifies the mime type of the specified file. The 'detect' method accepts a number of objects as parameters; the variant used by the evaluation accepts a file input stream (i.e. a stream of the file's contents) and the filename as input parameters[18]. The detect method outputs the file's detected mime type.

[17] It should also be noted that each of the files in the GovDocs1 corpus was copied into HDFS as an individual file, rather than being collated into sequence files.

[18] It is also possible to pass just a file input stream object, without the filename, to the Tika detect method but initial tests showed that the lack of a filename significantly reduced the number of files that Tika was able to identify correctly. The detect() method

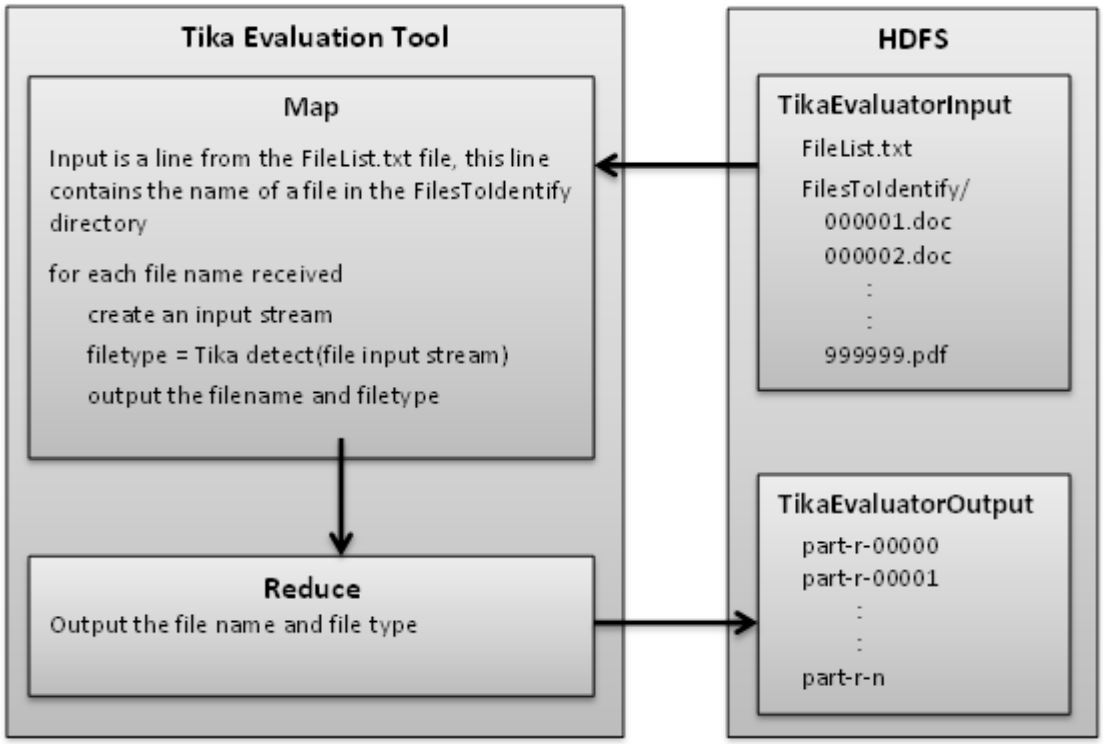

Figure 4 Diagram showing Tika Evaluator Map/reduce Process

Note that the map/reduce process shown in Figure 4 is concerned solely with identifying the format (MIME type) of each file in the Govdocs1 corpus. The Tika Evaluation Tool provides additional modules to carry out auxiliary tasks such as moving the test corpus into HDFS, merging the output from the Hadoop reduce process, comparing the Tika identified mime type with the ground truth mime type and generating a report showing the results of this comparison. The output from the map process is a list of comma separated values consisting of the filename, the file mime type as identified by Tika, and the time taken in milliseconds to call the Tika detect() method.

The reduce process collects the output from the map process and outputs it unchanged to the HDFS `TikaEvaluatorOutput` directory.

2.3.2.4 Tika Identification Hadoop Job

The Hadoop job was initiated using the command:

```
hadoop jar TikaEvaluatorHadoopDist.jar
```

The result of each of the twelve evaluations is shown in the Results section below.

2.3.3 Results

This section shows the results obtained from the evaluations in terms of the number of files identified correctly or otherwise, and the time taken to carry out each evaluation.

The raw result files generated during the evaluation can be found at https://github.com/openplanets/cc-benchmark-tests.

The conclusions drawn from these results are presented in the subsequent section.

2.3.3.1 File Types Identified

2.3.3.1.1 Overall results

Table 3 shows the number of files processed by Apache Tika; the number correctly identified; the number that were identified incorrectly; and the percentage of files identified correctly.

Table 3 Files identified correctly and incorrectly by Tika

Test	Tika Version	Files Processed	Files Identified Correctly	Files Identified Incorrectly	Files Correctly Identified (%)
1	1.0	973693	945555	28138	97.110
2	1.1	973693	945516	28177	97.106
3	1.2	973693	938138	35555	96.348
4	1.3	973693	938148	35545	96.349

2.3.3.2 Speed of Identification

Table 4 below shows the time taken for each evaluation. The 'Number of Input Files' is the number of files processed by each map process. This is controlled by the value of the mapred.line.input.format.linespermap configuration parameter that specifies how many lines from `FileList.txt` each map task will process. During the evaluation the `mapred.line.input.format.linespermap` configuration parameter was set to 3 different values, 20, 200 and 800.

The 'Number of Input Files (map tasks)' is the number of map tasks Hadoop creates to process the test files. This is determined by dividing the total number of test files by the number of files each map task can process.

The 'Overall Hadoop Job Runtime' is the total time taken by the Hadoop map/reduce process to complete the evaluation.

The 'Average Runtime per Map Task' is the average number of seconds each Map Task takes to complete, it is calculated by multiplying the number of Hadoop nodes by the total completion time in seconds and dividing by the number of map tasks.

The blue shaded horizontal bars grouping rows indicate similar Hadoop job configurations, each group having an identical number of files paths per Map task. This governs the number of Map tasks created, and the total runtime of the evaluation, more of which in the conclusion.

Table 4 Time taken to carry out file identification

Test	Tika Version	Number of Input Files (map tasks)	File Path Lines Per File (Identifications Per Task)	Overall Hadoop Job Runtime (hh:mm:ss)	Average Runtime Per Map Task (seconds)
1	1.0	49316	20	05:55:27	12.1
2	1.1	49316	20	05:53:37	12.0
3	1.2	49316	20	05:47:21	11.8
4	1.3	49316	20	05:08:29	10.5
5	1.0	4932	200	01:10:15	23.9
6	1.1	4932	200	01:10:49	24.1
7	1.2	4932	200	01:12:31	24.7
8	1.3	4932	200	01:11:30	24.4
9	1.0	1233	800	00:49:22	67.3
10	1.1	1233	800	00:49:10	67.0
11	1.2	1233	800	00:49:54	68.0
12	1.3	1233	800	00:49:32	67.5

Note that the times shown in Table 4 do not include the time taken to move the test corpus into HDFS or to compare the Tika mime types with the ground truth mime types.

2.3.4 Conclusions

2.3.4.1 File Identification

Table 3 shows that the functional performance of Tika, in terms of how well it identifies file mime types, degrades slightly between versions 1.0 and 1.3 in comparison to the ground truth. In order to assess why this might be an investigation was carried out to determine which files Tika was able to identify correctly in version 1.0, but failed to identify correctly in version 1.3.

Table 5 Type and number of files identified correctly in version 1.0 but incorrectly in version 1.3

Tika v1.3 Mime Type	Number of Files
application/msword	1
application/octet-stream	61
application/rss+xml	4
application/x-tika-msworks-spreadsheet	2
application/zip	2
message/x-emlx	10
text/plain	6
text/x-log	8107
text/x-matlab	2
text/x-perl	2
text/x-python	4
text/x-sql	295

Table 5 shows the mime types and numbers of the files that were identified correctly in Tika version 1.0 but identified incorrectly in version 1.3.

Further investigation, carried out into files identified by Tika 1.3 as '`text/x-log`', shows that these are text files with a file extension of '`.log`'. These files were identified by Tika versions 1.0 and 1.1 as having a mime type of '`text/plain`', which matches the ground truth mime type. Similarly, Tika versions 1.2 and 1.3, when used with just an input stream, also identified these files as '`text/plain`', again matching the ground truth.

However, when Tika versions 1.2 and 1.3 were provided with the filename, they identified files with a '`.log`' extension as having a mime type of '`text/x-log`'. This suggests that the later versions of Tika are using the file extension to differentiate between different types of plain text file, i.e. mime type identification is becoming finer-grained. This could be considered an improvement if the extra granularity is useful, but might equally be considered to be a 'muddying of the waters' depending upon the application.

2.3.4.2 Speed of Identification

The results of the timings shown in Table 4 show that the configuration of Hadoop, specifically the number of files processed by each map task, has a significant impact on the time taken to complete the evaluation.

When the '`mapred.line.input.format.linespermap`' parameter is set to 20 (tests 1 to 4) the number of map tasks is very high at 49316, and the total job runtime is between 5 and 6 hours. When the '`mapred.line.input.format.linespermap`' parameter is increased to 200 (tests 5 to 8) the number of map tasks created reduces to 4932 (fewer Map processes) but the total job runtime is just over an hour, 20% of the time taken in tests 1 - 4. This difference is almost certainly caused by the overhead of creating a large number of map tasks i.e. in the first set of evaluations (tests 1-4) Hadoop spent much of its time creating and initialising the many map tasks, each of which ran for only a short period of time processing a small number of input files.

When the 'mapred.line.input.format.linespermap' parameter is set to 800 (tests 9 to 12) the evaluation completes in under an hour, more quickly than when the parameter is set to 200 but the improvement is less marked.

Table 5 below shows the average runtime per input file for each test, calculated by dividing the average runtime per map by the number of files processed by each map.

Table 6 Average Runtime per input File

Test	Tika Version	File Path Lines Per File (Identifications Per Task)	Average Runtime Per Map Task (seconds)	Average Runtime per Input File (seconds)
1	1.0	20	12.1	0.6050
2	1.1	20	12.0	0.6000
3	1.2	20	11.8	0.5900
4	1.3	20	10.5	0.5250
5	1.0	200	23.9	0.1195
6	1.1	200	24.1	0.1205
7	1.2	200	24.7	0.1235
8	1.3	200	24.4	0.1220
9	1.0	800	67.3	0.0841
10	1.1	800	67.0	0.0838

Test	Tika Version	File Path Lines Per File (Identifications Per Task)	Average Runtime Per Map Task (seconds)	Average Runtime per Input File (seconds)
11	1.2	800	68.0	0.0850
12	1.3	800	67.5	0.0844

The results in Table 4 also show that the performance of the Tika tool appears to improve between versions 1.0 and 1.3 for tests 1 to 4, in particular there appears to be a significant improvement in speed between versions 1.2 and 1.3. However subsequent tests (tests 5 to 12) do little to back up these findings, even suggesting a slight degradation in performance between versions 1.0 and 1.3. It is quite possible that these variations are caused by differences in the Hadoop executions, as they are fairly small and show little correlation with version.

3 The Data Publication Platform

During the first year of SCAPE it became apparent that the project had the potential to create a variety of datasets as a result of testing tools and workflows at scale. One of the first large sets was created when testing format identification tools across the Govdocs1 corpus, as covered in D9.1. This led to some thought as to how these large datasets should be documented and published, and what might happen to the datasets at the end of the project. As a project researching the preservation of digital content including scientific datasets, it is logical that we do all possible to ensure our own data is accessible and understandable in the long term.

The standards that underpin the efforts to assemble the Semantic Web, using Linked Data to provide a machine interpretable web of data, are designed to solve this sort of problem. At the start of the second year of the project work began applying these principles to the format identification data generated by testing tools on the Govdocs1 corpus. This work became known as the Results Evaluation Framework (REF) and the results, and their publication are described later.

Because the initial work focussed on data produced for deliverable D9.2, the effort for the REF work started in this work package. The REF work has now led on to addressing the more general question of how the SCAPE project should go about publishing its experimental outputs, both for ease of sharing and re-use in the project, and for longevity after the projects end. This more general work has been given the title of ”The Data Publication Platform”.

So the Data Publication Platform is not a characterisation tool but the initial effort came from the Preservation Components work package. The best practices and methods learned publishing the identification data will be applied to other data produced in the other sub projects and work packages.

The Data Publication Platform is an effort to take some of the principles of Linked/Open Data and apply them to the data published by the SCAPE project. The activities can be divided into two general classes:

- A set of curatorial / tutorial activities that augment the SCAPE datasets with machine interpretable metadata to aid future understanding and reuse of the data.
- A set of technical activities, performed to address specific problems as required. Where possible existing, open source components / public services are been used, but some development "glue" is required to use them together.

These classes of activity aren't exclusively technical and nontechnical, in particular development effort can be required as part of the curatorial activity to process, and re-purpose some of the data. These activities will combine to provide published SCAPE data a permanent URI, which identifies a location where the raw data and machine/human interpretable representations are provided.

3.1 A Two Stream Approach

There are two general types of Data Publication Platform activity:

1 **Technical / Development**
This stream concentrates leveraging and integrating existing tools and platforms into an infrastructure where the data can be published and preserved. It also provides development effort to produce automated solutions for parts of the curatorial process, including some bespoke work to process and visualise particular data sets.

2 **Curatorial/Advocacy** These efforts involve researching and documenting best practices around Linked Open Data, and then providing wider project with assistance in implementing these best practices. Direct assistance will be provided to project staff to help them to:

- make their raw datasets available on the web as machine readable data;
- enhance datasets with machine readable metadata, licensing, and open availability as soon as possible;
- post process data for reuse e.g. discoverability, querying, and producing visualisations of results;
- looking for opportunities to link to other datasets where possible.

These two streams of activity are documented separately for logical structure, but are occurring in parallel and involve the same staff. While the technical activities, in particular the Results Evaluation Framework work, have their roots in this work package the non-technical activities have more to do with the sustainability of SCAPE outputs, and are a better match for the sustainability work package. Because the majority of the effort this year was spent on a specific technical implementation, publishing format identification test data, the curatorial work, which is just beginning in earnest, is also described in this document, along with a plan to move some of the activities to the sustainability work package.

3.2 The Technical Approach

3.2.1 Results Evaluation Framework

3.2.1.1 Background

The first iteration of this deliverable detailed the comparison of format identification tools against the Govdocs1 Corpus (Møldrup-Dalum M. R., 2012). The approach was to run the latest version of the selected tools: DROID, FIDO, and Tika, across the entire million file corpus, and then compare the identification results and time taken to run. The comparisons were used to select tools for further development, and adaption to run on the SCAPE platform.

Another interesting question is how consistent/accurate different versions of these tools have been over their history. This information is important to institutions that have used older versions of these tools to identify their digital content. Given that the tools and the signature files that they rely on have improved over time, it's quite possible that older versions of the tools gave inaccurate results. These inaccuracies may have been corrected in later versions.

One good example is the identification of the Microsoft Word. docx format introduced in Office 2007. This format uses a zip file as a container for the multiple files that make up a single document. When the format was introduced format identification tools recognised the zip file signature and identified these documents as zip files, the container format, rather than Word documents. Any repository containing metadata generated using the older tools potentially holds misleading format information.

Another possibility is that older formats get re-classified because of changes in signature precedence, or changes to the algorithms in subsequent versions.

In order to highlight these potential issues it was decided to run as many versions of the format identification tools and signature files as possible over the Govdocs1 corpora and publish the results, as the Results Evaluation Framework (REF).

3.2.1.2 Gathering Tools

In order to make the results as comprehensive as possible some tools not chosen for the original evaluation were chosen for the REF. The full list of tools is given in Table 7

Table 7 List of tools chosen for original REF evaluation

Tool	Homepage
DROID	http://www.nationalarchives.gov.uk/information-management/our-services/dc-file-profiling-tool.htm
FIDO	http://www.openplanetsfoundation.org/software/fido
FITS	https://code.google.com/p/fits/
Apache Tika	http://tika.apache.org/
*nix File Utility	http://www.darwinsys.com/file/

For DROID there was the additional task of tracking down as many versions of the signature file as possible. Once all combinations of tool version and signature file were counted there were 65 different versions of tool and signature in total.

3.2.1.3 Cutting Govdocs1 Down to Size

Rather than try to run every tool over the entire million-file corpus the decision was made to try to reduce the size of the test corpus. Running 67 tools over 1 million files would be time consuming, not just in terms of execution, but also when processing and consolidating the results. The distribution of file formats in the corpus is not even, with PDFs alone making up nearly 10% of the content.

The ground truth file that accompanies the corpus was used to select duplicate files, with respect to format, that could be removed. This was done by first removing the fields in the ground truth file that weren't directly related to format id:

- Last Saved
- Last Printed
- Title
- Number of Pages
- SHA-1
- Image Size
- File Name (for now)
- File Size (for now)
- Other characteristics...

Leaving these properties in the ground truth file:

- Extension
- Description
- Version (& related information)
- Valid File Extensions
- Accuracy of Identification
- Content
- Creating Program (or library)
- Description Index (serial code)
- Extension Valid (Y/N)

These fields were then sorted to give a list of unique identifications in the file; there were 4653 of these in total, for 87 different file extensions. The discrepancy between the number of extensions and the number of identifications is caused by the variety of ways that files of a single format can be created. As an example there were 3,337 different variations of PDF, taking into account version, creating program, etc.

Up to 10 files representing each identification format were chosen, less when there weren't 10 instances available in the corpus. This gave a reduced size corpus containing about 21,000 files; a blog post giving more details is available on the OPF web site.[19]

3.2.1.4 Running Tests and Processing Data

The task of running all versions of the tools over a corpus is performed by the REF code, a PHP Github project[20]. The project contains all of the versions of the tools tested, along with the archive of DROID signature files[21]. A brief description of the projects general function is given below, a fuller description is given in a post on the Open Planets Foundation website[22].

The REF project requires some configuration to:

- Set up a database to record the results of identification tests.
- Register the tools to be tested.

[19] http://openplanetsfoundation.org/blogs/2012-07-26-1-million-21000-reducing-govdocs-significantly
[20] https://github.com/openplanets/ref
[21] https://github.com/openplanets/ref/tree/master/tools/Droid/signature_files
[22] http://openplanetsfoundation.org/results-evaluation-framework-first-release

- Add test data to the data directory.

The full project documentation is provided in the Github README file[23].

Once configured, the registered tools can be tested on the data, and the raw results are recorded in a relational database. These are then converted into RDF triples, while aligning results that are identical.

At this point the data requires further processing to align results that are, in fact, different representations of the same result. This takes into account issues such as variations in tool output over versions of the tool, e.g.

<droid:puid>"fmt/18"</droid:puid> == <droid:hasPronomPUID>"fmt/18"</droid:hasPronomPUID>

It also maps identical results across different tools, e.g.

<droid:puid>"fmt/18"</droid:puid> == <fido:puid>"fmt/18"</fido:puid>

REF calls this component the Semantic Matching Engine. Its task is essentially to create new owl:sameAs predicates that identify equivalent results.

3.2.1.5 Displaying Format Identification Data Online

The final challenge was to provide a web interface to the format identification dataset. The interface presents the results by file format. When the user selects a particular format they are presented with a graph displaying the identification results of all versions of all tools, for the set of files (at most 10 files) that were selected for the particular format. The interface is hosted online at this URL, http://data.openplanetsfoundation.org/ref/.

The figure below shows the format selection panel presented to the user:

Figure 5 Format choice scree from Results Evaluation Framework interface

Selecting the Portable Document Format (PDF) icon allows the user to choose a particular version of the format, as shown in the next figure

[23] https://github.com/openplanets/ref#readme

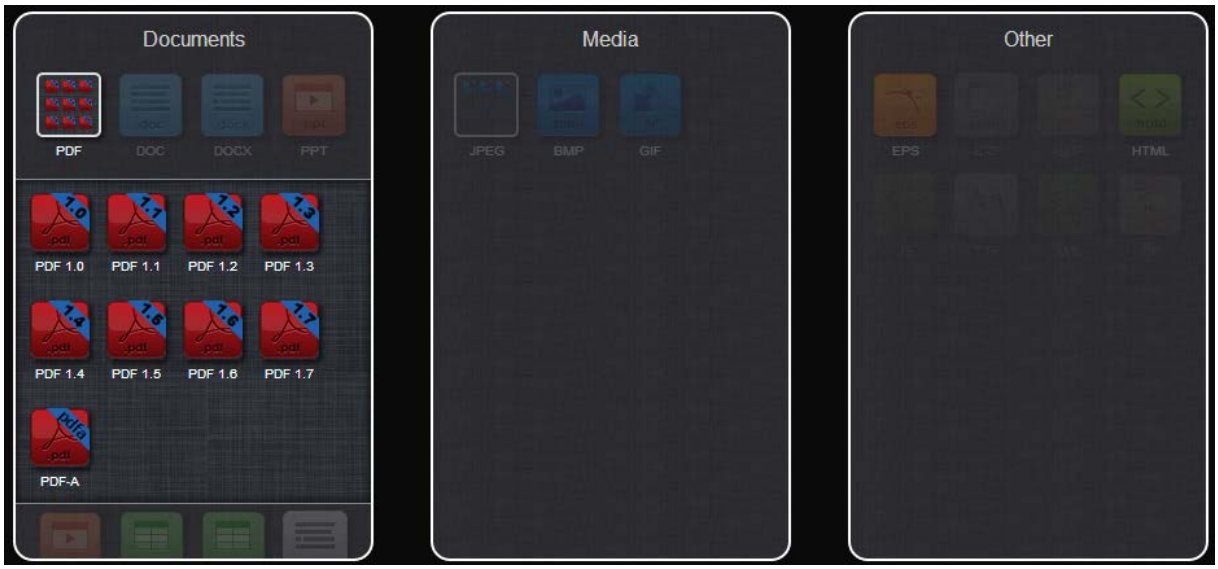

Figure 6 PDF version selection from Results Evaluation Framework interface

Selecting PDF 1.6 produces the following graph for the format:

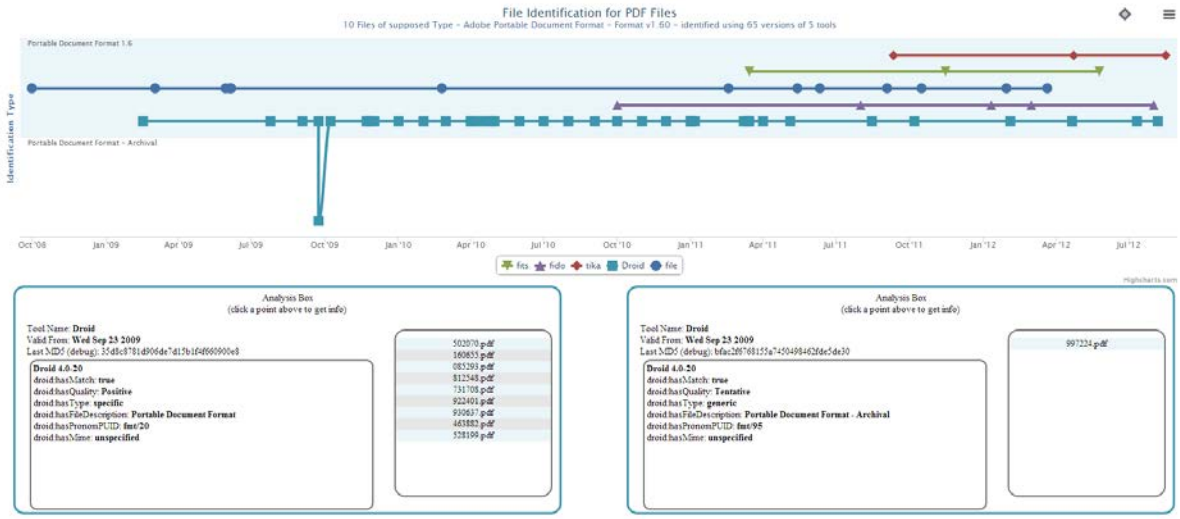

Figure 7 PDF version 1.6 format identification results

Although the graph isn't entirely clear in the document the original can be viewed online[24]. A quick guide to the graph is given below:

- Each coloured line represents the identification results of a particular tool.
  - FITS == green triangles
  - FIDO == purple triangles
  - Tika == red diamonds
  - DROID == pale blue squares

[24] http://data.openplanetsfoundation.org/ref/pdf/pdf_1.6/

- File == dark blue circles

- Each shaped point on a line indicates the results of a different version or signature file for the particular tool. The line for DROID has many more points due to the regular publication of new signature files.

One point of interest is the bottom most point on the graph, which appears as a downward spike in the bottom, pale blue line. This represents the misidentification of a single PDF 1.6 file as a PDF/Archival (PDF/A) format file by a single version of the DROID signature file, version 20. In general the straight lines in the upper bar indicate consistent identification of the format by the different versions of the tool.

The graph can also be downloaded as an image file, a PNG example showing the identification data for Microsoft Office docx files is displayed below.

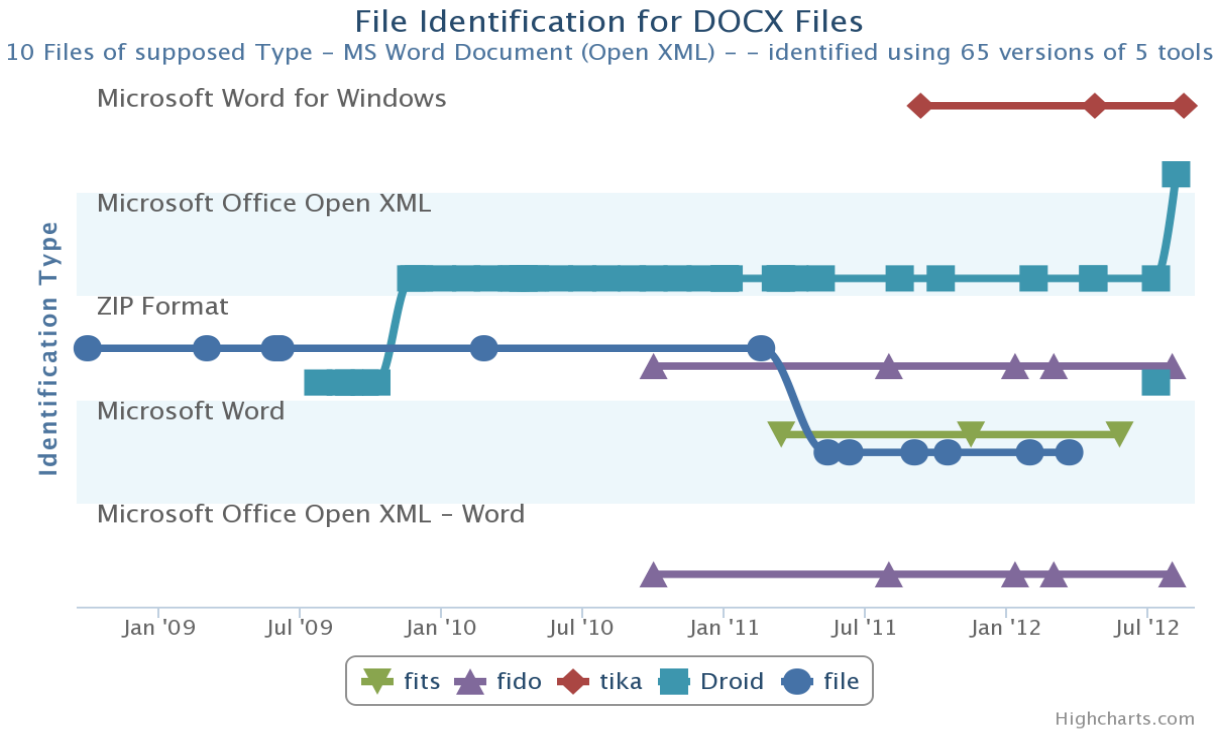

Figure 8 Microsoft Office docx format identification results

Looking at the central horizontal bar on the graph shows that older versions of DROID and file have misidentified the docx files as zip archives, while recent versions of DROID and Fido have at different times misidentified doc x files.

3.2.1.6 Other forms of REF Data

The web interface also allows the user to download the source RDF data that is used to plot the graphs. This machine interpretable data can be used to inform preservation systems of potential inaccuracies in their past format identification activities. It is planned to produce an adaptor to import the format identification data into SCOUT, SCAPE's automated preservation watch system. SCOUT evaluates an organisations collection metadata and against trigger conditions to suggest when preservation planning, and potentially action, is necessary due to a detected potential issue. Possible misidentification of format might be such a trigger.

3.3 The Curatorial Approach

The curatorial activities can be broadly divided into research tasks, and advocacy/training tasks. Initially the aim is to provide project staff with a process that allows them to put their datasets online quickly, and begin adding metadata. The process needs to be as simple as possible, to prevent it becoming a "barrier to entry", that is the process must be nearly as easy doing nothing. Once the data is online, open licensed, and has a minimal but useful set of metadata the problems of availability and longevity become easier to solve.

3.3.1 A Quick Introduction to Linked Data

The Data Publication Platform is the SCAPE projects efforts to apply the principles of the Semantic Web to its own data outputs. In order to aid the reader in understanding the theory underlying the Data Publication Platform a quick introduction to Linked Data is provided.

Linked Data describes a method for publishing data so that it can be linked with other data, meaning that when you find data of interest it is easier to find other, related data. In 2006 Tim Berners-Lee put forward four principles of Linked Data[25]:

1 Use URIs as names for things.

2 Use HTTP URIs so that people can look up those names.

3 When someone looks up a URI, provide useful information, using the standards (RDF[26], SPARQL[27]).

4 Include links to other URIs, so that they can discover more things.

These principles are deceptively simple, as most data is still not linked, beyond providing HTTP links for human readers using browsers.

In 2010 a 5 star rating system for Linked Open Data was added[25]. Linked Open Data is data that abides by the principles stated above, Linked *Open* Data is Linked Data released under an open license, that doesn't impede its free reuse. The Creative Commons CC-BY license[28] is an example open licence, as is the Creative Commons CC0 Public Domain license[29]. The five stars of Linked Open Data are described below:

★	Available on the web (whatever format) but with an open licence, to be Open Data
★★	Available as machine-readable structured data (e.g. excel instead of image scan of a table)
★★★	Available as machine-readable structured data plus non-proprietary format (e.g. CSV instead of excel)
★★★★	All the above plus, Use open standards from W3C (RDF and SPARQL) to identify things, so that people can point at your stuff
★★★★★	All the above, plus: Link your data to other people's data to provide context

[25] http://www.w3.org/DesignIssues/LinkedData
[26] http://www.w3.org/RDF/
[27] http://www.w3.org/TR/rdf-sparql-query/
[28] http://creativecommons.org/licenses/by/3.0/
[29] http://creativecommons.org/about/cc0

The Data Publication Platform is in effect an effort to apply these principles to SCAPE datasets in order to ensure their widespread and long-term availability and usefulness.

In terms of the 5 stars of linked data the aim is that SCAPE should initially publish 3 star data that is it is online, with an open license and in a non-proprietary format. Once this is close to a business as usual opportunities to link to other relevant datasets can be explored.

3.3.2 The Open Data Certificate

The Open Data Institute[30] (ODI) is a non-profit organisation, based in the UK that aims to promote the creation and use of Open Data through advocacy and training. One current ODI initiative is the Open Data Certificate[31], which specifies what good open data looks like in four main categories:

- **Legal Information** about the licence, ownership, and privacy implications of the data.
- **Practical Information** about the timeliness, quality, and sustainability of the data.
- **Technical Information** about where the data can be found and its format.
- **Social Information** about how to engage with the data owner and others about the data.

A certificate can be awarded at one of our levels:

- A **basic certificate** is for data that meets the fundamental requirements for open data.
- A **pilot certificate** is for initial forays into publishing open data, which provides enough to prompt experimentation with the data.
- A **standard certificate** is for routinely published open data that encourages reuse.
- An **established certificate** is for open data that provides reliable key information infrastructure.

In practise the certificate alpha is an online form[32] that a user fills in with information about aspects of their dataset. At the end of the process the user is presented with a certificate marking the status of their data, marked from **basic** through to **established** as explained above.

A realistic first aim for SCAPE data is the **pilot certificate** which, in combination with the third star of Linked Open Data would mean that all SCAPE data:

1 Is available online at a URI.

2 Is marked as having an open licence.

3 Is in a non-proprietary, machine-readable form.

4 Provides sufficient support to prompt some reuse of the data.

While meeting these criteria appears fairly straightforward, most data published on the Internet doesn't meet them.

[30] http://www.theodi.org
[31] http://theodi.github.io/open-data-certificate/
[32] http://theodi.github.io/open-data-certificate/certificate.html

At this stage the certification process is a pilot, this may mean that it is never finalised if it doesn't satisfy its aims. An upside is that the SCAPE project can feed back their experiences of using the process, and help to improve it.

3.3.3 Applying the Certificate Criteria

The SCAPE project is just starting to test the process of applying the certification criteria to the datasets it creates. Initially it is envisaged that raw data will be published on the web using a Github[33] repository. Github is an online revision control service, used for source code, but the versioning and attribution functionality, i.e. all changes are versioned, and change information is audited, make it ideal for experimenting with versioning of data sets. Every Github project can make use of a README markdown file, used to provide project documentation. These files are human readable but can be structured to be machine interpretable, and they seemed a good candidate as a place to add metadata to data sets.

The Tika benchmarking data produce for this report was used as a test case for the certification process. The project and readme file can be found on Github[34]. The following headings mirror those found in the README, and provide a little background information.

3.3.3.1 Overview

This section provides some basic information about the dataset that is covered at the top level of the certification form, specifically:

Title: A descriptive title for the data

Creator Details: The name and URL for the SCAPE project

Dataset URLs: URLs for representations of the data (CSV and JSON in this case)

3.3.3.2 Data Release Type

Indicates the basis of release, whether a one off, or part of an established publication process, this a one off set of related data at the moment, though this can change.

3.3.3.3 Dataset Type

Gives some idea of the type of data in the set, statistical data in this case.

3.3.3.4 Description

Details how the raw data was produced, and documents its structure, in this case describes the CSV fields.

3.3.3.5 Copyright and License / Open Data

Clearly states the license terms for the data, and emphasises that the dataset can be freely used. Also indicates the certification level of the data.

3.3.3.6 Personally Identifiable Data

Provides a clear indication as to whether any of the data contains personal information and is therefore liable to be subject to Privacy Law.

3.3.3.7 Using the Data

Provides a little explanation about the data formats used with some references.

[33] https://github.com/

[34] https://github.com/openplanets/cc-benchmark-tests

3.3.3.8 Discoverability

These questions are taken directly from the certificate and force the user to think about how interested parties might find the dataset. Ideally both of these would be true.

3.3.3.9 Applicability

Meant to indicate the "timeliness" of the data, indicating whether it has a shelf life, or is time sensitive.

3.3.3.10 Quality

Gives an indication as to how people can ask questions about, or raise issues about the data, in this case it's the Github Issues[35] Tracking page associated with the project. Ideally this section should state the URL of the SCAPE projects data publication / data quality policy, but the policy is still been written.

3.3.3.11 Guarantees

Indicates how long the data might be available for, as it may not be appropriate to keep this data available beyond the project's life the date of July 2014 is given for now, this may be changed.

3.3.3.12 References

These questions are used to indicate whether the dataset uses terms from other vocabularies, or any codes, etc. In this case MIME types are used to identify formats, so an explanation and a link is provided. There is also a link to the original data that the identification was performed against, i.e. the Govdocs1 corpus. The information in this section will be used to look for opportunities to link to other datasets at a later date.

3.3.3.13 Support

A contact for the dataset and a link to the issues page again.

3.4 Next Steps

3.4.1 Getting Our Data Online

The curatorial activities have so far concentrated on devising a simple methodology for putting data online that follows current best practise for Open Data. The next activities will be to liaise with the Preservation Components and Testbed subprojects to help staff to implement the process for their datasets. Feedback gained from providing one-on-one assistance will be used to improve the process and possibly be fed back to the Open Data Certification process itself.

[35] https://github.com/blog/831-issues-2-0-the-next-generation

3.4.2 Using Our Online Data

Once experimental datasets are available as Open Data the DPP Team will help publishers to make use of their data, including the use of visualisation techniques. A first experiment with the Tika dataset is available here: http://data.openplanetsfoundation.org/cc-benchmark-tests/Visualisations/d3-parsets/. This uses a parallel sets[36] visualization technique to compare the Tika identification result to the ground truth, and indicate how long the identification took.

3.4.3 Referencing other Datasets

The DPP Team and other SCAPE staff will re-examine their datasets looking for opportunities to link to other open datasets. One obvious candidate would be the Linked Data PRONOM[37] prototype, which provides a Linked Data endpoint for the contents of the PRONOM[38] registry. Success here depends as much on the availability of good quality data external to the project, so it is difficult to set targets.

A more achievable aim is to ensure that the project takes the opportunities to provide links between its own datasets so that users can navigate between related datasets.

3.4.4 Selection Activities

As more raw datasets are made available online in this manner, the curation process will have to address selection of datasets for long-term publication. Much of the raw data created is useful during the project, but may not be required in the long term / externally to the project, e.g. benchmarking times for prototype workflows or the results of characterisation tools that are under development. Other datasets may have genuine long-term utility, for example the comparison of results of format identification tools over time.

During the last phase of SCAPE (2014) the curation activities will start to concentrate on the selecting datasets that it will guarantee to make available beyond the end of the project. Criteria for preserving sets are not finalised but will include:

- Usefulness - whether the dataset is likely to be of long term interest.
- Referenced - whether the dataset is referenced in an external publication.
- Ease of Reproduction - whether a dataset is difficult / time consuming to recreate.
- Service Provision - whether the dataset underpins a publically available service

3.5 Conclusion

The DPP work has expanded over the past year from a discrete effort to publish a specific set of experimental data, the Results Evaluation Framework, to an advocacy task promoting recent initiatives and best practices for publishing open, linked data on the web. The most realistic applications of this work are:

- The possibilities it presents for sharing and re-using results amongst SCAPE partners.
- The publication of results sets that can be cited from SCAPE publications.
- The development of adaptors that can import the data for use in other SCAPE components, particularly the planning watch components.

[36] http://eagereyes.org/parallel-sets
[37] http://test.linkeddatapronom.nationalarchives.gov.uk/doc/file-formats
[38] http://www.nationalarchives.gov.uk/PRONOM/Default.aspx

Given that these are really project wide activities dealing with the longevity of SCAPE outputs, the effort and activities that will make up the Data Publication Platform in the coming year are best carried out as part of the sustainability work package. The sustainability work package should consider publishing a data policy document online, as a formal outcome of this work.

4 Information Extraction on Text and Web Corpora

4.1 Overview

This section describes the work in the fields of text and Web (HTML) page characterization carried out in year 2 of SCAPE. The goal is to characterize large collections of HTML pages, discovering and extracting information relevant to the domain of digital content preservation. We make use of Information Extraction technologies to extract semantic relations from natural language text on the Web, and make this information machine processable for automatic preservation watch. In the following, we give an outline of the system developed in year 2 of SCAPE, an overview of evaluation results and describe an application of the system to a use case from the National Library of the Netherlands.

4.2 System Outline

We initially developed a prototypical open-domain Information Extraction system specifically designed to capture complex, N-ary relations. We evaluated this system, named KrakeN, on a small dataset with regards to information completeness. A scientific paper on the prototype and the evaluation was submitted and accepted at the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (Löser, 2012).

Building on the prototype, we designed and implemented an expanded system consisting of two components: A focused crawler for retrieving text from the Web and a system for the normalization relational information by leveraging distributional evidence.

The first component, the focused crawler, makes use of lists of keywords and search engine APIs, such as the Bing API[39]. Each keyword on the list is individually queried against the Web and all Web pages found by the search engine are crawled and stored. The crawling process can thus be directed by providing domain specific keywords; for example, by providing a list of keywords that relate to the domain of digital content preservation, we can gather Web pages relevant to the domain.

The second component makes use of distributional evidence to normalize relations. A corpus of text data, which in our system is provided by the focused crawler, is processed sentence by sentence. For each sentence, linguistic analysis consisting of part-of-speech tagging, Named Entity detection and dependency parsing is performed. For each pair of Named Entities that co-occur in at least one sentence, all patterns are extracted with a novel algorithm developed by us. Entity pairs that are observed in similar patterns are then grouped with a clustering approach. Each resulting cluster is interpreted as one distinct semantic relation, and all entity pairs in the cluster as instances of this relation. The relations that are identified depend on the text collection that is processed; if a text collection from a specific domain such as digital preservation is passed to the system, it identifies key relations in this domain.

[39] http://www.bing.com

4.3 Evaluation

We evaluated the system against Yago, a well-known knowledge base based on Wikipedia. We set the focused crawler to find text that contains entities that are also in the Yago knowledge base and used the unsupervised extraction system on this text. We then compared the extracted information against the structured information in Yago and measured precision, recall and f-measure values. We find that our pattern generation method (indicated as PROP in Figure 9 below) outperforms previous approaches based solely on shallow syntactic or lexical features. A detailed discussion of the pattern generation method and comparison to other approaches is given in (A. Akbik, 2012).

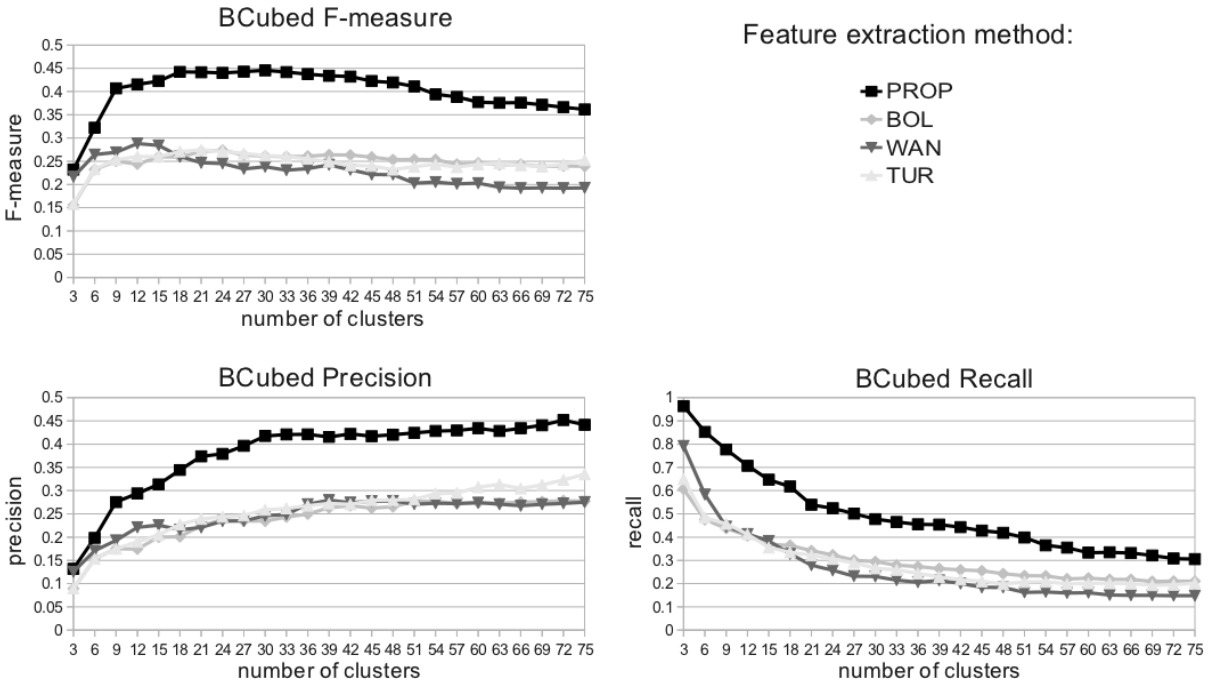

Figure 9 Comparison of different feature extraction methods against our proposed system (PROP). Our system outperforms related works by up to 60% in overall f-measure.

4.4 Application

The designed system has reached a level of maturity to allow for the definition and execution of workflows to gather relational data specific to the domain of digital content preservation. We have expanded collaboration with two SCAPE partners, namely KEEP SOLUTIONS and the National Library of the Netherlands, to address a prototypical real world scenario with our system. We address a real case scenario from the National Library of the Netherlands, where information on publishers and journals is needed to recognize when the preservation of digital content is becoming endangered and for maintaining the long-term, continuous and authentic access to digital assets. To address this use case, we have conducted a focused crawl of the Web, from which we have extracted relational data for automatic ingestion into the Scout preservation-monitoring tool. We have evaluated the results against two manually curated repositories that contain similar data, namely e-Depot[40] and Keepers Registry[41]. Comparing the information with the Keepers registry we find that more that 50% of the automatically fetched data is not in the registry and should be added, proving that this method is

[40] http://www.kb.nl/en/expertise/e-depot-and-digital-preservation
[41] http://thekeepers.org/thekeepers/keepers.asp

effective and can provide a much needed contribution for the automatic watch of the publisher community. We have submitted a joint contribution to the 10th International Conference on Preservation of Digital Objects (iPres 2013), in which we demonstrate how we use information extraction technologies to find and extract machine-readable information on publishers and journals for ingestion into automatic digital preservation watch tools. We show that the results of automatic semantic extraction are a good complement to existing knowledge bases on publishers, finding newer and more complete data.

4.5 Dissemination Activities

In year 2 of SCAPE, we have also expanded the range of dissemination activities of our work. We have submitted scientific publications to the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX 2012)[1], the International Conference on Computational Linguistics (COLING 2013) (A. Akbik, 2012) and the International Conference on Preservation of Digital Objects (iPres 2013). In addition, group member Alan Akbik has given several invited talks on our work, including a guest talk at the University of Toronto and a talk at the Big Data Analytics Day at the Technical University of Berlin.

4.6 Current and Future Work

We are currently focusing on large-scale experiments with the intent of processing very large collections of Web pages. One longstanding and challenging aim is to process the 3-trillion page ClueWeb corpus, the main reference corpus for work on large Web collections. To this end we have begun porting large parts of our system to the Apache PIG data analytics platform, which offers a high level language for expressing data processing programs that compile down to MapReduce code. Effectively, this enables us to parallelize large parts of our system in order to handle very large data sets.

The technologies and methods used in the above described application use case are not specific to the publishing domain and can be applied to other monitoring needs, opening new possibilities for institutions to automate their watch processes. Using information extraction with automated preservation watch systems allows monitoring of non-technical domains, such as social, economic or organizational, where formally specified data is scarce. For example, monitoring economical or organizational changes in companies that support file formats or tools, like company bankruptcy or takeover, may allow the discovery of significant preservation risks. Also, this method allows monitoring of institutional specific domains, like the producer or target community, from which pre-existing formally specified data is rare and mostly manually created by institution itself. Further research on how to use these technologies and methods to monitor digital preservation related domains will be done in the next year of the SCAPE project.

5 Characterization of office documents on MSR Azure platform

Over the first 18 months Microsoft Research Cambridge has designed and implemented an Azure based architecture with functions that support a four step workflow for batch-mode document conversion: ingest and characterization of document collections, conversion, comparison, and reporting.

Initially, MSR had focused on the conversion of common proprietary formats into XML based formats. However, the set was expanded with the set of converters to support representations of documents in multiple formats in order to increase the value that the user can derive from digital content. The value is increased through the use in multiple scenarios and on multiple platforms.

5.1.1 Quick MSR Azure platform overview

This section provides an overview of the MSR SCAPE architecture implementation on the Windows Azure platform with focus on characterization. The described solution makes use of a number of key Microsoft technologies including Silverlight, RIA Services, WCF, SQL Azure and Windows Azure.

5.1.1.1 Architecture components

The following MSR SCAPE Azure 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 a metadata layer, which describes Data, Reports, Logs, and Workflows. It maps stored data and metadata in SQL Azure.
- **Tools and Resources Layer** represents the Action services and tools we use for Characterization, Conversion, Comparison, and Reporting.
- **Data store** is a virtually unlimited storage in BLOB.

On Figure 10, note the highlighted *Content Representation Layout* part of the schema that indicates architecture components responsible for storing and processing metadata about documents.

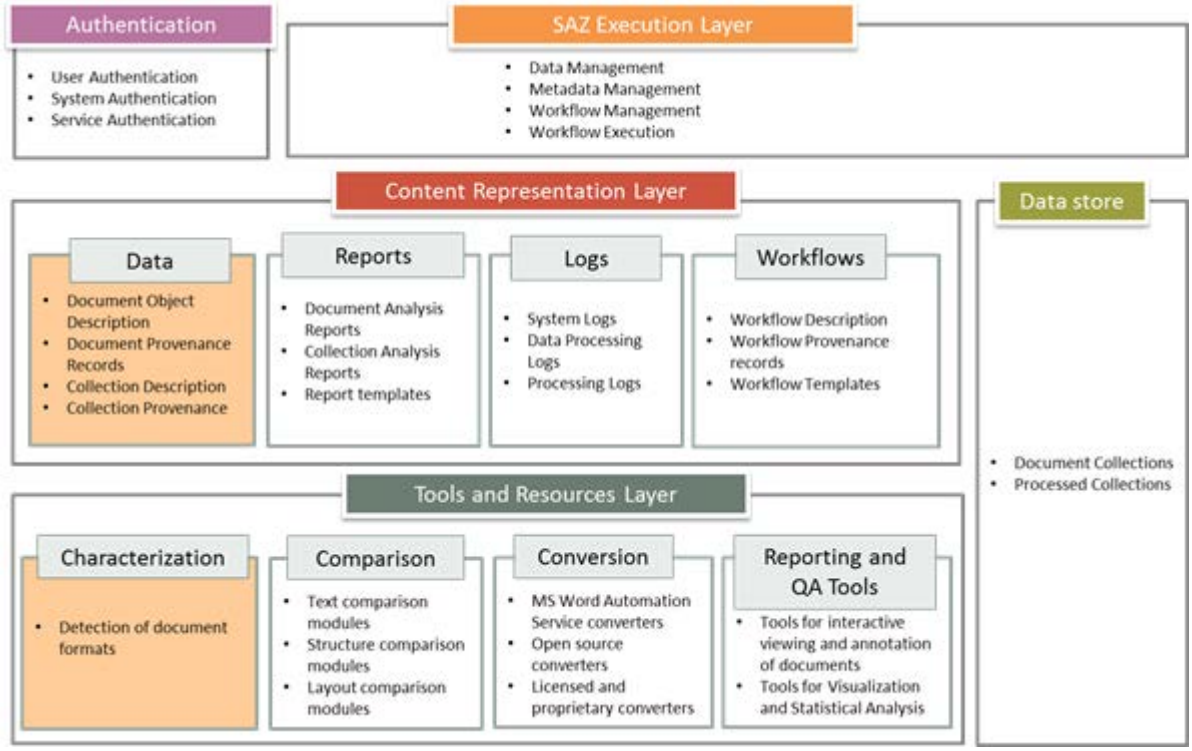

Figure 10 Architecture components of MSR SCAPE Azure

5.1.1.2 SCAPE Azure Architecture

Figure 11 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.

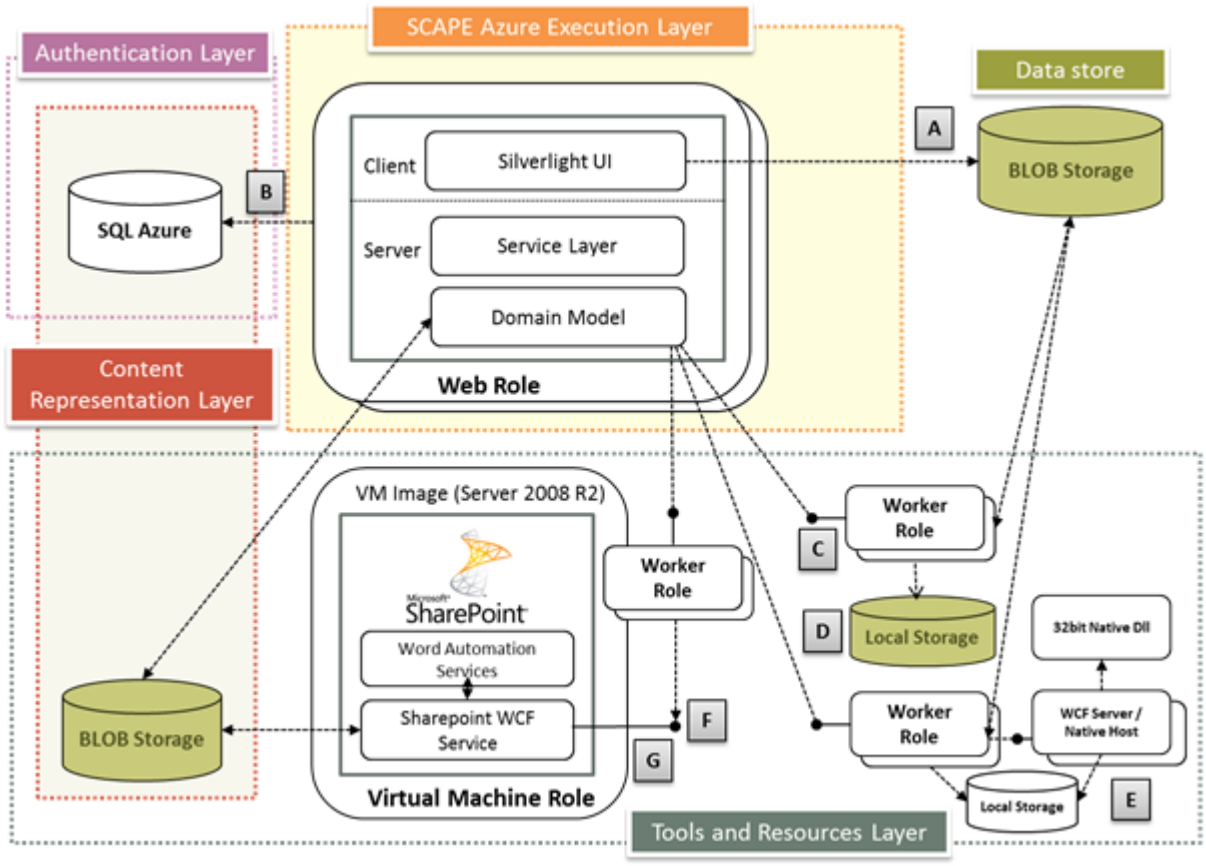

Figure 11 System architecture of SCAPE Azure v1.0

Version v1.0 includes a method for characterizing original and converted documents. It first creates a common representation of each document, transforming them into XPS format. XPS versions of documents are passed through OmniPage OCR software that provides analysis of the layout and content of individual documents. The OCR output is then processed to compare the two documents.

Legend for Figure 11 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 document characterisation metadata as well as user profiles and profile management information. Alternative database solutions can be employed either locally or within the cloud, e.g. MySQL)

[C] A worker role (*processing node*) encapsulates a number of discrete actions such as data processing functions, diagnostics, analysis, QA methods, etc.

a The actions of a worker role can be exposed externally and internally
b Replicating Worker Roles attains scalability. 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. This 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 a 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:

a SharePoint Word Automation Services (WAS) have been enabled within the SCAPE portal
b 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)
c 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 this figure) to perform OCR before analysis is performed and results transferred to BLOB storage

Glossary for Figure 11:

- **SQL Azure** – cloud-based, scale-out version of MS SQL Server
- **Web Role** –used for frontend (Silverlight client) and overall logic of the system
- **Worker Role** –used 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

5.1.2 Exposing metadata about documents and conversions on SCAPE Portal UI

Metadata is currently extracted from the file properties or generated through OCR comparison techniques. The SCAPE Azure Portal provides two main views for displaying document metadata:

1 Collection Pivot view

2 Document comparison view

5.1.2.1 Collection Pivot Viewer

Collection Pivot Viewer is an application that enables browsing, sorting, and filtering of a large collection of objects represented as images within a zoom-able canvas.

In the *Collection pivot* view, following metadata is shown:

- Size
- Author
- Date created
- Page, paragraph, line and character counts
- Title
- Company
- Subject
- Page, paragraph, line and word count differences
- Good, bad partial and no match qualities (%)
- Average mismatched rectangles per page

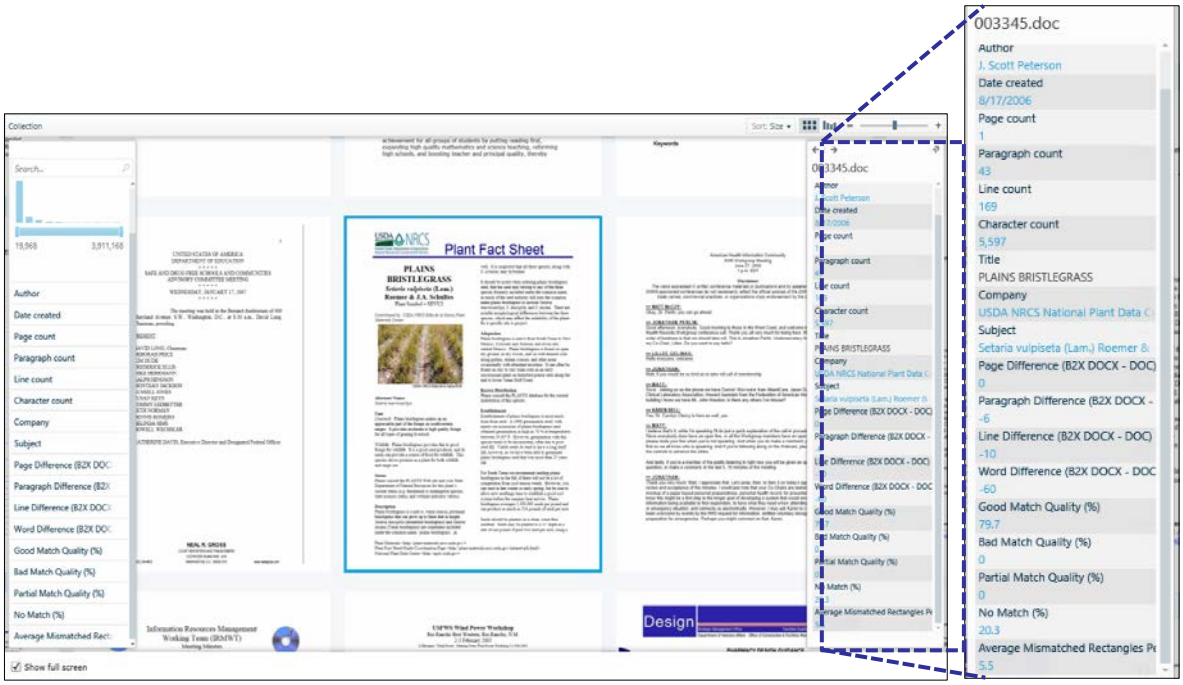

Figure 12 Collection Pivot View showing the metadata associated with the selected document

5.1.2.2 Document comparison view metadata

By selecting highlighting options, using check boxes from the *Show highlights* pane (*Match, Missing* and *Error*) one can see the match quality as a text overlay in the *Document Comparison View*.

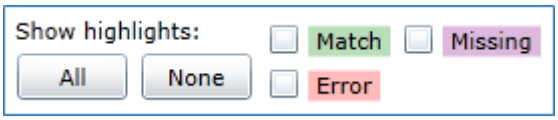

Figure 13 Highlight options in the document Comparison View.

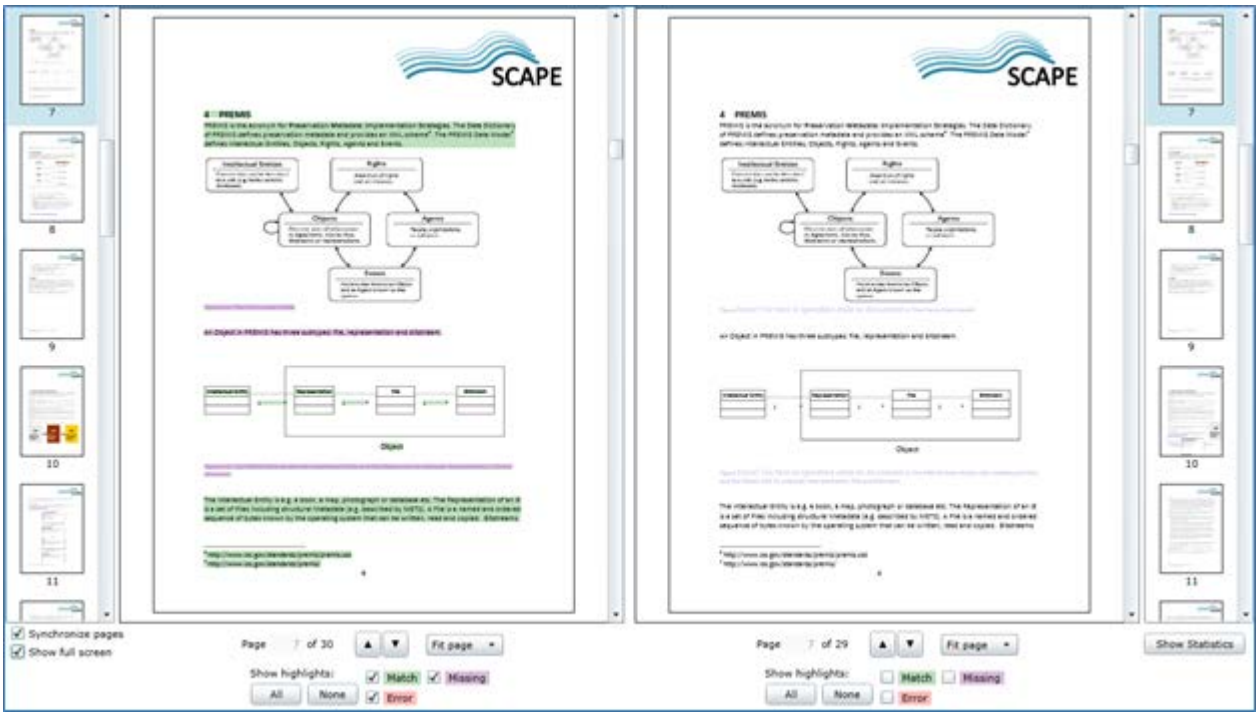

Figure 14 Document Comparison view

By clicking the *Show Statistics* button additional metadata is shown in the *Comparison Statistics* pop up:

Matches (%)
Errors (%)
Missing (%)
Pages, paragraphs, lines and words in uploaded file
Pages, paragraphs, lines and words in B2X file

Figure 15 Comparison statistics displayed in the pop up window

5.1.3 Experiment using Droid and Apache Tika on the MSR Azure Platform

Experiments with Droid and Tika on MSR Azure platform are postponed due to depreciation of current VM Role model in Microsoft Azure service.

MSR used the Windows Azure VM Role to run applications in Azure. The VM Role feature provided better control of the operating system and applications running within a virtual machine. However, the Windows Azure VM Role is not supported by Azure any more. Instead, the applications now need to run using Windows Azure Virtual Machine. This requires changes to the existing implementation.

Please find more information in the Moving from VM Role to Windows Azure Virtual Machines MSDN article (MSDN). For that reason, the performance testing and benchmarking work is postponed after the migration to Windows Azure Virtual Machines.

6 Conclusion

Here we present some overall conclusions drawn from the document, specifically comparing the two Hadoop based format identification experiments and looking at the implications for executing the tools on the SCAPE platform.

We first compare the two Hadoop evaluations against the single threaded evaluations run last year and presented in the last iteration of this report. We then look to see if comparison against a common benchmark gives any insight into the relative performance of the two Hadoop clusters used.

We finally compare the job configuration and timing results of the different Hadoop job configurations and present some conclusions and ideas about optimal configuration of Map Reduce processes in general.

The section won't cover the conclusions regarding the other work package activities as the outputs of these aren't directly comparable, and their conclusions are self-contained.

6.1 Comparing Performance of Parallel and Single Threaded Format Identification

Both of the format identification tools evaluated on the Hadoop clusters were also evaluated last year for the first iteration of this deliverable. This evaluation was carried out single threaded on a dual Xeon server; the data was hosted on a network file system accessed over Gigabit Ethernet. The machine had access to 70GB of RAM, detailed specification details, such as CPU speed, aren't compared as the comparisons of timings between parallel and single threaded environments can only be rough and ready.

The comparisons below are made using the fastest parallel job configurations for the two clusters. Differences in software versions are ignored for simplicity, the idea is to look for obvious trends. The table below shows the average time taken per Govdocs1 file for the different evaluations.

Table 8 Comparison of identification performance per object

	Single threaded D9.1 evaluation.	BL Hadoop Cluster	ONB Hadoop Cluster
Apache Tika	0.016 sec/object	0.084 sec/object	0.025 sec/object
Droid	0.02 sec/object		0.089 sec/object

Assuming that the Govdocs1 corpus has a million files we can project total single threaded run times to identify all files in the corpus:

Table 9 Projected single threaded times for identification evaluations

	Single threaded D9.1 evaluation.	BL Hadoop Cluster	ONB Hadoop Cluster
Apache Tika	4.6 hours	23.3 hours (50 mins)	6.9 hours (17 mins)
Droid	5.6 hours		24.7 hours (58 min)

The figures in brackets give the actual time taken by the parallelised Map Reduce jobs run on the cluster.

The comparison between the performance of the ONB Hadoop cluster and the single threaded performance from the previous year is interesting: while the time taken to carry out Tika identification is fairly similar on the different platforms, DROID performs significantly slower on the Hadoop cluster in comparison. The reason for this is unknown, but might be worth investigating to see if the Map Reduce performance of DROID can be improved. Comparisons between the two clusters are difficult to draw because of the differences in hardware, both virtual, and the physical storage. Again investigating why two broadly similar clusters give such a marked variation in performance may give some insights into the performance of the Map Reduce tasks.

One firm conclusion that can be drawn is that parallelising identification yields performance improvements. The ONB cluster reduced DROID identification time by 80%, while the BL cluster did the same for Tika identification. The ONB cluster brought the Tika time down by more than an order of magnitude, which is the scale of performance improvement sought after by the project. There is good reason to think that these figures don't represent the full potential of parallelised preservation environments. There is more work that can be done to improve I/O performance of DROID in particular, implementing stream-based identification as was mentioned in the evaluation. This is only been considered at the moment, as the work package can't follow up every possibility.

During the editorial process of this report ONB performed an interesting experiment where they ran the DROID tool as a single-threaded process for baseline comparison. This experiment is described in the blog post "Droid file format identification using Hadoop"[42] The results depicted in the single-threaded experiments relates directly to the results from section DROID Identification Results.

6.2 Hadoop Job Configuration

The parallel format identification evaluations clearly demonstrate the importance of considering the division of tasks amongst Hadoop cluster Map tasks. The Tika evaluations performed at the British Library where only 20 identification jobs were assigned to each map task actually ran slower than the single threaded test carried out last year. This configuration meant that the workload of managing the Map tasks outweighed any performance benefits offered by parallelisation. Like DROID, Tika does have to parse an XML file of signatures, although this is contained within the Java jar archive, rather than as a separate file. Processing so few files per Tika initialisation may also have a negative impact on performance.

Looking at the configurations shows that the best comparative performance came when the Map tasks were at their largest. That is to say that there were fewer Map tasks created, and they each performed more work, and ran for longer. In both cases the execution time for an individual Map task was about a minute. This may prove to be a good guideline when considering parallelising preservation tasks and workflows, i.e. a Map task should take more than 30 seconds. This sort of

[42] http://www.openplanetsfoundation.org/blogs/2013-05-24-droid-file-format-identification-using-hadoop

performance tuning is probably best done in the Testbed and Platform workpackages. It is only because the work done in this workpackage so clearly illustrates the issue that it is covered here.

7 List of references

A. Akbik, L. V. (2012). Unsupervised Discovery of Relations and Discriminative Extraction PAtterns. *International Conference on Computational Linguistics.*

Cloudera. (2009). *The small files problem.* Hentet fra Cloudera Blog: http://blog.cloudera.com/blog/2009/02/the-small-files-problem/

Cloudera. (2009). *Tips for improving mapreduce performance*. From Cloudera Blog: http://blog.cloudera.com/2009/12/7-tips-for-improving-mapreduce-performance

Löser, A. A. (2012). KrakeN: N-ary Facts in Open Information Extraction. *Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction.*

Møldrup-Dalum, M. R. (2012). *Characterisation technology, Release 1 + release report.* Deliverable, SCAPE.

Møldrup-Dalum, W. V. (2013). *PC.WP1 checkpoint CP070.* Checkpoint, SCAPE.

MSDN. (u.d.). *Moving from VM Role to Windows Azure Virtual Machines.* From MSDN: http://msdn.microsoft.com/en-us/library/windowsazure/dn133483.aspx

SCAPE. (2010). *SCAPE Description of Work (Annex I of the Grant Agreement with the European Commission).* Description of Work, SCAPE.

van der Knijff, A. B. (2011). *WP 9: evaluation framework for characterisation tools.* SCAPE.

Wilson, v. d. (2011). *Evaluation of characterisation tools.* Checkpoint, SCAPE.

8 Annex

8.1 Tables

8.1.1 Tika identification result (Austrian National Library Cluster)

Mime type	Number of instances
application/pdf	231236
text/html	178593
image/jpeg	109282
text/plain	94398
application/msword	78055

application/vnd.ms-excel	65167
application/vnd.ms-powerpoint	51464
image/gif	36302
application/xhtml+xml	34907
application/xml	34360
application/postscript	26581
text/csv	18339
application/x-gzip	14021
image/png	4125
application/x-shockwave-flash	3473
application/rtf	1125
application/vnd.google-earth.kml+xml	987
message/rfc822	967
application/zip	950
model/vnd.dwf	299
text/x-java-source	289
application/x-sh	239
application/vnd.openxmlformats-officedocument.presentationml.presentation	218
application/rss+xml	183

application/vnd.openxmlformats-officedocument.wordprocessingml.document	163
application/rdf+xml	106
image/x-ms-bmp	72
text/x-diff	63
application/x-tex	59
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	39
image/x-xbitmap	36
application/octet-stream	31
image/tiff	31
application/javascript	22
image/x-portable-bitmap	15
text/troff	14
application/x-123	12
application/x-tika-msoffice	10
application/x-bibtex-text-file	9
video/x-ms-wmv	6
application/x-compress	5
application/winhlp	4
application/xslt+xml	3

application/vnd.oasis.opendocument.text	2
application/x-elc	2
image/vnd.adobe.photoshop	2
image/vnd.dwg	2
application/vnd.ms-powerpoint.presentation.macroenabled.12	1
application/vnd.ms-word.document.macroenabled.12	1
application/vnd.openxmlformats-officedocument.presentationml.slideshow	1
application/x-bzip	1
application/x-stuffit	1
audio/x-ms-wma	1
image/x-icon	1
image/x-portable-pixmap	1
message/news	1

8.1.2 Droid identification result (Austrian National Library Cluster)

PUID	**Number of instances**
fmt/0	134644
fmt/18	86531
fmt/96	84950

fmt/100	56605
fmt/43	54618
fmt/17	52350
fmt/126	51251
fmt/40	45475
fmt/61	45414
fmt/44	41984
fmt/111	41282
fmt/16	40672
fmt/101	32502
fmt/99	26987
fmt/102	22753
fmt/19	22031
fmt/3	21590
fmt/20	19140
fmt/4	14647
x-fmt/266	14021
x-fmt/406	12721
fmt/42	7665

x-fmt/408	6738
fmt/15	6518
fmt/98	4948
fmt/11	3975
fmt/123	2860
x-fmt/9	2662
fmt/57	2579
fmt/124	2572
x-fmt/390	2448
fmt/39	2330
fmt/59	2048
x-fmt/391	1678
fmt/276	1265
fmt/109	1138
x-fmt/383	1040
fmt/14	949
x-fmt/263	948
fmt/97	916
fmt/244	626

fmt/41	510
fmt/110	477
fmt/50	475
fmt/108	465
fmt/189	422
fmt/125	413
fmt/122	379
x-fmt/398	374
fmt/505	370
fmt/107	365
fmt/106	340
x-fmt/49	299
fmt/45	295
x-fmt/394	284
fmt/53	216
fmt/354	201
fmt/56	175
fmt/38	161
fmt/103	149

fmt/281	149
fmt/12	146
x-fmt/88	141
x-fmt/65	137
fmt/52	125
fmt/506	121
fmt/157	106
x-fmt/226	83
fmt/116	72
x-fmt/44	63
fmt/288	62
fmt/471	51
fmt/158	47
fmt/280	46
x-fmt/280	37
fmt/185	35
fmt/37	35
fmt/156	30
fmt/390	28

x-fmt/274	28
fmt/332	24
fmt/501	20
x-fmt/275	18
fmt/63	17
fmt/104	16
fmt/55	16
fmt/146	15
fmt/95	15
fmt/94	12
fmt/75	11
fmt/503	10
x-fmt/393	9
x-fmt/114	7
x-fmt/207	7
x-fmt/407	7
x-fmt/91	7
fmt/133	6
fmt/328	5

fmt/355	4
fmt/388	4
x-fmt/416	4
fmt/77	3
fmt/78	3
x-fmt/235	3
x-fmt/420	3
fmt/13	2
fmt/130	2
fmt/473	2
fmt/488	2
fmt/76	2
x-fmt/115	2
x-fmt/117	2
x-fmt/428	2
x-fmt/429	2
x-fmt/92	2
fmt/132	1
fmt/136	1

fmt/144	1
fmt/284	1
fmt/290	1
fmt/32	1
fmt/329	1
fmt/34	1
fmt/341	1
fmt/353	1
fmt/372	1
fmt/374	1
fmt/396	1
fmt/468	1
fmt/58	1
fmt/74	1
fmt/91	1
fmt/93	1
x-fmt/108	1
x-fmt/119	1
x-fmt/219	1

x-fmt/268	1
x-fmt/317	1
x-fmt/367	1
x-fmt/430	1
x-fmt/453	1
x-fmt/64	1
x-fmt/8	1

Table: Droid identification result