DOI: 10.2352/issn.2168-3204.2016.1.0.24

Scalable Processing and Search in Package-based Repositories

Sven Schlarb, Rainer Schmidt, Mihai Bartha, Roman Karl; Austrian Institute of Technology; Vienna, Austria

Abstract

Subject of this paper is the architecture of the prototype implementation developed in the E-ARK project. It is specifically designed to support scalable and efficient data transformation, information extraction from archival information packages, and full-text search in the repository. As a continuation of previous work related to the use of Hadoop to process large data volumes, it presents a combined approach of using a distributed task queue for parallel processing together with Hadoop and HBase to allow computing intensive and long-running tasks being applied during ingest as well as the full-text indexing of very large document collections.

Introduction

This paper describes the architecture of the E-ARK prototype implementation of a scalable e-archiving system. It enables scalable data processing and search within package based repositories. Based on this, the E-ARK Integrated Prototype provides the reference implementation for a scalable archiving system. The purpose of the reference implementation is to allow memory organizations to evaluate and validate the architecture against their requirements. E-ARK is an ongoing 3-year multinational research project co-funded by the European Commission's ICT Policy Support Program (PSP) within the Competitiveness and Innovation Framework Program (CIP).

The main objective of the E-ARK project is the creation of a scalable open source digital archiving system based on best practices and standards. The core standard is the ISO Reference Model for an Open Archival Information System (OAIS) [3]. According to this reference model, data arrives at an archive as Submission Information Packages (SIPs) and are then stored in the archive as Archival Information Packages (AIPs). Upon client request the AIPs can be retrieved from the archive and repacked as Dissemination Information Packages (DIPs) for delivery. E-ARK is dealing with both, structured and unstructured data, including records from Electronic Records Management Systems (ERMS); Relational Database Systems (DBMS); Geographical Data, and Document Collections. OAIS provides immediate access to archived content and internal or public discovery services. This is accomplished by executing information extraction and transformation processes upon transfer of the SIP to the archive (SIP to AIP conversion/ingest), as well as during creation of the DIP (conversion from AIP to DIP).

In recent times the amount of data managed by archival institutions has been growing at an increasing rate. Many traditional infrastructures and data processing applications are unable to cope with the processing and transformation requirements of such massive data sets. This situation has promoted the development of new technologies, tools, and techniques which are usually associated with the term “big data”.

Based on open source big data technologies, the system architecture presented in this paper, addresses the high level scalability and extensibility requirements by offering a distributed storage and processing environment that can grow dynamically as data is added to the system. The system can handle both, very large individual files and a very large number of small files. Text-based content is automatically indexed during ingest thus enabling an archivist or end-user to perform full-text search in the entire repository and to immediately access the archived content.

In the following, we first present related work by introducing other repository systems which include mechanisms for handling large data volumes. We then briefly introduce the scenarios addressed within the scope of the E-ARK project. We continue with a detailed description of the system architecture, give an outlook to future work, and finally present our conclusions.

Related work

Use of Apache Hadoop in CERN'S archive system

Although not related regarding the type of content being archived, CERN's archive system with its over 70 Petabytes of data collected during the first run of the large hadron collider and with yearly growth rates of up to 40 to 50 Petabytes can – especially in terms of the amount of data being archived – be regarded as the epitome of big data environments [2]. While experimental data is mainly stored on tape, CERN also makes use of technologies based on Hadoop and HBase. By making use of MapReduce techniques they enable data mining, real-time data aggregation, and visualization [8].

RODA Open Source Repository

RODA is an open source digital repository which delivers functionality for all the main units of the OAIS reference model [6]. RODA provides distributed processing and the execution of digital preservation actions (e.g. migration) on a Hadoop cluster.

The European project SCAPE

The European project SCAPE (Scalable Preservation Environments) – co-funded by the European Commission under its 7th Framework Program and ended in September 2014 – developed an extensible infrastructure for the execution of digital preservation workflows. The application of digital preservation actions to massive input data sets, in the context of large digital repositories, was one of the main use cases addressed in the project [11]. Additionally the project investigated the integration of Hadoop with the Fedora Commons repository systems [7].

Distributed Forensic Cluster

In the context of digital forensics [9], preservation of storage device images, on a distributed storage system, is a completely different application domain in comparison to the scenarios described in this paper. However, several common characteristics can be identified. First, the repository employs Apache Hadoop for distributed storage and processing environment. Second, for archiving purposes and in order to enable distributed processing,

DOI: 10.2352/issn.2168-3204.2016.1.0.24

the images must be split into smaller parts (SIPs in OAIS terminology) putting the inherent integrity of the image at risk. This is similar to the E-ARK database archiving use case where large databases are converted to an archival format (e.g. SIARD) and split into multiple SIPs. The SIPs are then converted to AIPs and logically bound together by a parent AIP.

Scenarios

A variety of parameters need special attention in large-scale data processing scenarios. Factors such as the overall data volume, the number of individual files, and the distribution of file sizes can be used as an initial indicator to estimate the kind of infrastructure needed. Only by analyzing the requirements of a specific use case it is possible to compare alternative approaches for scalable data processing.

A standard scenario in E-ARK is the creation and processing of packages containing text-based documents. In addition, the following three special scenarios are being addressed:

- Archived databases (SIARD [20] format)
- Geographical data (GML format)
- Electronic records management system records

Each scenario is based on its own set of requirements and makes use of dedicated software for transforming data in various steps of an ingest process (e.g. creating an AIP, preparing a DIP for access purposes). In this context several relevant questions need to be answered: How computing intensive are the processes? Are there any relations that need to be evaluated between individual processing steps? Do any external information sources have to be taken into account?

In the following sections, we will briefly present the three special scenarios.

Archived databases

Current database archiving relies mostly on archiving the data according to the original data model in an open format like SIARD (Software Independent Archiving of Relational Databases) [20].

Databases archived in such a manner are often difficult to reuse because of complex data structures defined by the underlying entity relationship model.

E-ARK is developing a specialized tool to create a SIARD representation of a database [14]. This representation can be packaged as an E-ARK SIP and archived in the E-ARK repository.

To simplify access to archived databases, E-ARK is looking into methods to prepare de-normalized AIPs ready for dimensional analysis via OLAP.

Geographical data

Geographical data is a combination of the graphical representation of objects in space and related attributes. The basis of geodata is either vector or raster and can be stored as a set of files or as a database.

E-ARK gives recommendations how to prepare geographical data in an SIP to be submitted to the E-ARK repository. Furthermore, it develops scenarios for converting the AIP into a DIP allowing to render the geographical information in an adequate manner and to provide specialized tools to visualize and search in the geospatial dataset contained in the DIP.

Electronic records management systems

Another scenario in the E-ARK project is to standardize the way that content from Electronic Records Management Systems (ERMSs) can be archived.

The E-ARK approach reuses the MoReq specification [17] and gives guidance on how to reuse it for SIPs and AIPs. The goal of a MoReq2010 export to an archive is to assure that records metadata is preserved, and that the record's component contents are properly managed to assure long-term preservation.

Scalable processing and search infrastructure

Although systems for parallel and distributed computing have been studied since the early 1980's and parallel database systems were established already in the mid-1990's [1], a significant change in the last decade occurred with the advent of the MapReduce data processing paradigm [4] and the subsequent rise of open source technology for distributed storage and parallel data processing provided by Apache Hadoop.

In the following, we describe the E-ARK Integrated Prototype backend which is based on Apache Hadoop and related components that emerged in the Hadoop ecosystem during the last decade.

E-ARK Integrated Prototype Backend

The backend system of the reference implementation is built on top of the Hadoop framework and can be deployed on a computer cluster allowing the repository infrastructure to scale-out horizontally. This enables system administrators to increase the available system resources (i.e. for storage and processing) by adding new computer nodes. Using Hadoop, the number of nodes in a cluster is virtually unlimited and clusters may range from single node installations to clusters comprising thousands of computers.

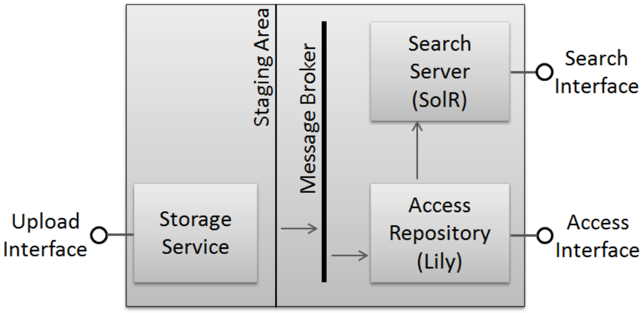

***Figure 2**. System Components deployed on top a Hadoop infrastructure which is employed by the E-ARK reference implementation. Hadoop can be deployed on a single node or on large clusters of commodity computers.*

Apache Hadoop based services

The E-ARK Integrated Prototype uses a Hadoop infrastructure with a backend system providing the following basic services:

- The Hadoop Distributed File System (HDFS) is the primary distributed storage used by Hadoop applications.
- Apache HBase is a distributed database (e.g. used by the Lily repository) that leverages the distributed data storage provided by HDFS.
- Hadoop MapReduce is a software framework for writing applications that process large amounts of data on Hadoop clusters.

DOI: 10.2352/issn.2168-3204.2016.1.0.24

The open source message broker software RabbitMQ [19] is used to facilitate asynchronous communication between the components and to trigger the ingestion of incoming data into the repository. The HDFS Storage service provides an HTTP interface allowing client applications to upload Information.

Lily Access Repository

The actual repository is realized based on the Lily framework [16] which provides a distributed data repository that can be deployed on top of Apache Hadoop clusters and establishes a link between the distributed database HBase and the search framework Solr. Lily provides a repository that is built on top of the HBase, a NoSQL database that has been built on top of Hadoop. Its main function is to serve as an access repository in addition to the package-based long-term repository. Lily defines a set of data types where most of them are based on existing Java data types. Lily records are defined using these data types as compared to using plain HBase tables, which makes them better suited for indexing due to a richer data model. The Lily Indexer is the component which sends the data to the Solr server and keeps the index synchronized with the Lily repository. Solr neither reads data from HDFS nor writes data to HDFS. The index is stored on the local file system and optionally distributed over multiple cluster nodes if index sharding or replication is used. One feature that is worth mentioning is that Lily can control the appropriate storage strategy for binary content fields (BLOB), which means that large files are stored in HDFS and the small files in HBase.

In an OAIS, the AIP as a whole is ingested into archival storage. In addition to this, in the E-ARK Integrated Platform the AIPs are extracted and the individual files are ingested into the access repository. Every item (or file) contained in an AIP is considered a record and the packages are then represented as a set of records sharing a common identifier.

SolR and Lucene search interface

The preferred option to use Solr with Lily is to use SolrCloud which takes care of the index sharding and also provides a mechanism for replication. The underlying Apache Lucene search library takes care of the full-text indexing and provides the search functionality. The search interface allows querying multiple fields which have been indexed during ingest. Additionally, it is possible to do sorting and ranking as well as faceted search.

Repository ingest workflow

The reference implementation utilizes the Lily Java API as part of a Hadoop MapReduce job in order to ingest large volumes of files in parallel. Basically, if an AIP creation is finished, the package is uploaded to the distributed storage (HDFS) using a dedicated service. Subsequently, each of the individual files contained in the package – depending on the mime-type of the file – will be indexed. The E-ARK integrated prototype provides a faceted query interface which allows performing search queries based on a previously generated full-text index created by using Apache Lucene.

The repository index is updated in periodic intervals. Depending on the data volume, the ingest of generated AIPs and their indexing can be time and resource intensive, therefore both processes have been implemented as parallel applications that can take advantage of a computer cluster.

Many of the underlying processes used during ingest or to create dissemination information packages are distributed on a cluster. Package transformations, such as data extraction, or indexing processes can run efficiently on very large data volumes in a distributed manner.

Usually relational database systems (RDBMS) are used to store the extracted information and as a backend for the search, discovery, and access services the repository offers. In addition, the RDBMS is used to manage the package transformation processes. It collects information about the status of a package, such as: acceptance of the delivery, validation and consistency verification, extraction from a physical container, digital preservation actions, archival package creation and validation, storage, etc.

As already presented in [12] and [10], by using scale-out architecture, based on Hadoop [5] and related NoSQL technologies, the ability to store and process very large data collections within the same environment can help open up a larger scope for action in face of the continued growth of data volume.

However, there are caveats that need to be considered with this approach. In the following, we will present a combined approach of using a distributed task queue for parallel processing using Hadoop and HBase [15] that allows for computing intensive and long-running tasks being applied during ingest as well as full-text indexing of very large document collections.

We have presented results of our work regarding the feasibility and advantages of using Hadoop to construct scalable data-flows with legacy components [12]. In the context of the E-ARK project we are extending this solution by system components allowing for efficient and fault tolerant information package processing.

Figure 1 gives a high level overview on the components of the integrated prototype that is being developed in the E-ARK project.

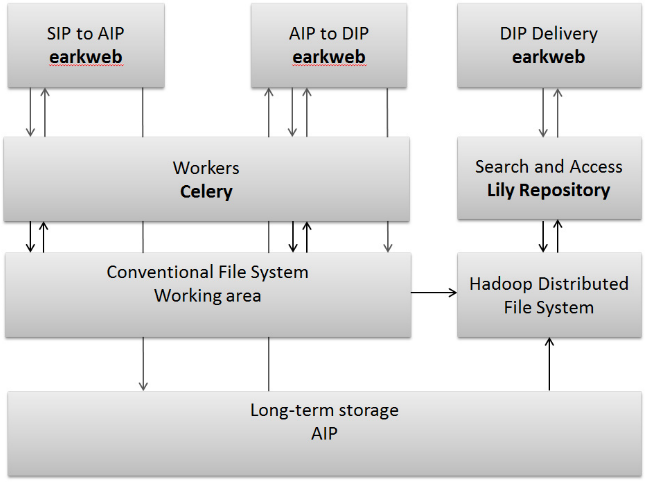

***Figure 1**. Component overview about the E-ARK integrated prototype.*

The 'earkweb' component coordinates package transformations between the package formats SIP, AIP, and DIP, and uses Celery [13], a distributed task queue, as its main backend. Tasks are designed to perform atomic operations on information packages and any dependency to a database is intentionally avoided to increase processing efficiency. The outcome and status of a task's process is persisted as part of the package. 'earkweb' also provides a web interface that allows to orchestrate and

DOI: 10.2352/issn.2168-3204.2016.1.0.24

monitor tasks, but is loosely coupled with the backend. The backend can also be controlled via remote command execution without using the web frontend. The outcomes of operations being performed by a task are stored immediately and the PREMIS format [18] is used to record digital provenance information. It is possible to introduce additional steps, for example, to perform a roll-back operation to get back to a previous processing state in case an error occurs.

Hadoop is an essential component in the integrated prototype that together with Lily – a document repository on top of HBase with Solr as its indexing component – provides a faceted query interface to do full-text on the archived data as well as search and access to individual files. If a SIP to AIP conversion is completed, a task can be launched to upload the AIP to HDFS and trigger the indexing of the AIP's content so that it becomes available to Lily's discovery service.

Future work

Future work in the E-ARK project will focus on the deployment of the system described in this paper at various archival institutions.

Furthermore, the E-ARK project is developing and integrating data mining related tasks to demonstrate the use of text mining or named entity extraction as part of the archival processes. It aims at showing how such components can be used effectively in a large scale repository environment.

Conclusion

We have presented a general approach that allows scalable processing of information packages and search in repositories together with a high level component overview of the integrated prototype that is currently being developed in the E-ARK project.

The particular benefit of the approach lies in the combined use of different scale-out strategies. Taking the concrete use cases into consideration, it can be decided if (1) an import of generated data into HDFS with subsequent Hadoop-based processing or (2) the use of celery workers is the appropriate data processing strategy.

Acknowledgement

Work presented in this paper is primarily supported by European Community's Seventh Framework Program through the E-ARK project under grant agreement number 620998.

References

[1] Borkar, V., Carey, M.J. and Li, C. 2012. Inside "Big Data Management": Ogres, Onions, or Parfaits? *Proceedings of the 15th International Conference on Extending Database Technology* (Berlin, Germany, 2012), 3–14.

[2] Cancio, G., Bahyl, V., Kruse, D.F., Leduc, J., Cano, E. and Murray, S. 2015. Experiences and challenges running CERN's high capacity tape archive. *Journal of Physics: Conference Series*. 664, 4 (2015), 042006.

[3] CCSDS 2012. *Reference Model for an Open Archival Information System (OAIS)*. CCSDS - Consultative Committee for Space Data Systems.

[4] Dean, J. and Ghemawat, S. 2004. MapReduce: Simplified Data Processing on Large Clusters. (2004).

[5] Dean, J. and Ghemawat, S. 2008. MapReduce: Simplified Data Processing on Large Clusters. *Commun. ACM*. 51, 1 (Jan. 2008), 107–113.

[6] Faria, L., Ferreira, M., Castro, R., Barbedo, F., Henriques, C., Corujo, L. and Ramalho, J.C. 2009. RODA - A Service-Oriented Repository to Preserve Authentic Digital Objects. *Proceedings of the 4th International Conference on Open Repositories* (2009).

[7] Jurik, B.A., Blekinge, A.A., Ferneke-Nielsen, R.B. and Møldrup-Dalum, P. 2015. Bridging the gap between real world repositories and scalable preservation environments. *International Journal on Digital Libraries*. 16, 3 (2015), 267–282.

[8] Presti, G.L., Curull, X.E., Cano, E., Fiorini, B., Ieri, A., Murray, S., Ponce, S. and Sindrilaru, E. 2014. Streamlining CASTOR to manage the LHC data torrent. *Journal of Physics: Conference Series*. 513, 4 (2014), 042031.

[9] Pringle, N. and Burgess, M. 2014. Information assurance in a distributed forensic cluster. *Digital Investigation*. 11, Supplement 1, (2014), S36 – S44.

[10] Schlarb, S. 2013. An open source infrastructure for quality assurance and preservation of a large digital book collection. *Archiving Conference, Archiving 2013 Final Program and Proceedings* (2013), 234–238.

[11] Schlarb, S., Cliff, P., May, P., Palmer, W., Hahn, M., Huber-Moerk, R., Schindler, A., Schmidt, R. and Knijff, J. van der 2013. Quality assured image file format migration in large digital object repositories. *iPRES 2013: 10th International Conference on Preservation of Digital Objects, 2-6 September 2013* (Lisbon, Portugal, Sep. 2013), 9–16.

[12] Schmidt, R., Rella, M. and Schlarb, S. 2015. Constructing Scalable Data-Flows on Hadoop with Legacy Components. *11th IEEE International Conference on e-Science, e-Science 2015, Munich, Germany, August 31 - September 4, 2015* (2015), 283.

[13] Celery (http://www.celeryproject.org).

[14] *Database Preservation Toolkit*.

[15] HBase (https://hbase.apache.org).

[16] Lily (http://www.ngdata.com/on-lily-hbase-hadoop-and-solr).

[17] MoReq2010 (http://moreq.info).

[18] PREMIS (http://www.loc.gov/standards/premis).

[19] RabbitMQ (http://www.rabbitmq.com/).

[20] 2013. *SIARD*. Swiss Federal Archives.

Author Biography

Sven Schlarb works as a Scientist at the Austrian Institute of Technology (AIT). Since 2008 he participated in various EU projects, such as PLANETS, IMPACT, SCAPE, and E-ARK. His research interest lies in digital libraries and archives with a focus on scalability and big data.

Rainer Schmidt works as a Scientist at the Austrian Institute of Technology (AIT). His research interest lies in Parallel and Distributed Computing, Large Scale Machine Learning and Big Data.

Mihai Bartha is an engineer in the Digital Safety and Security (DSS) department (with a specialization in Big Data and Digital Preservation) at the Austrian Institute of Technology (AIT). His research interests include: Digital Preservation, IT Security, Real-time 3D Visualization. More specifically, his work examines distributed big-data processing.

Roman Karl works as an engineer at the Austrian Institute of Technology (AIT). He studied computer science on the Vienna University of Technology with a focus on algorithmics and mathematical optimization. His research interests include big data technologies.