An Architectural Overview of the SCAPE Preservation Platform

Rainer Schmidt
AIT Austrian Institute of Technology
Donau-City-Strasse 1, Vienna, Austria
firstname.lastname@ait.ac.at

ABSTRACT

Cloud and data-intensive computing technologies have introduced novel methods to develop virtualized and scalable applications. The SCAPE Preservation Platform is an environment that leverages cloud computing in order to overcome scalability limitations in the context of digital preservation. In this paper, we provide an overview of the platform architecture and its system requirements. Furthermore, we present a flexible deployment model that can be used to dynamically reconfigure the system and provide initial insights on employing an open-source cloud platform for its realization.

1. INTRODUCTION

The SCAPE project is developing tools and services for the efficient planning and application of preservation strategies for large-scale, heterogeneous collections of complex digital objects [4]. The SCAPE Preservation Platform, developed in this context, provides the underlying hardware and software infrastructure that supports scalable preservation in terms of computation and storage. The system is designed to enhance the scalability of storage capacity and computational throughput of digital object management systems based on varying the number of computer nodes available in the system. It supports interaction with various information and data sources and sinks, the coordinated and parallel execution of preservation tools and workflows, and the reliable storage of voluminous data objects and records. At its core, the SCAPE Platform functions as a data center service that provides a scalable execution and storage backend which can be attached to different object management systems using standardized interfaces. The architecture aims at addressing scalability limitations regarding the number and size of the managed information objects and associated content.

The SCAPE preservation platform also supports a flexible software deployment model allowing users to reconfigure the system on demand. Packaging, virtualization, and automated deployment of tools and environments plays an important role in this context. A SCAPE preservation workflow may depend on dozens of underlying software libraries and tools, which must be made available on the computing infrastructure provided by the Platform. This in turn drives the need for a strategy that can resolve such context dependencies in a distributed and dynamically scaling environment on demand without requiring to perform expensive data staging operations over a network. SCAPE is employing a consistent packaging model in order to manage and sustain the preservation components developed within the project. Packaging and virtualization provide important concepts for the deployment and operation of the SCAPE Preservation Platform (and the tools and environments it depends on). In this paper, we describe a fully virtualized prototype instance of the SCAPE Execution Platform that has been recently set-up at AIT. Using a *private cloud* model, it allows us to deploy required preservation tools together with a parallel execution environment on demand and co-located with the data.

The rest of the paper is organized as follows: we provide an overview of the platform architecture and its key concepts in section 2. Section 3 presents design considerations for the packaging and deployment model. Section 4 discusses the application of data-intensive computing frameworks. A prototype setup of the preservation platform is presented in section 5. Section 6 reviews related work and section 7 concludes the paper.

2. ARCHITECTURAL OVERVIEW

The SCAPE Preservation Platform provides an digital object management and computing platform that (a) interacts with other SCAPE sub-systems like the Planning and Watch and the Result Evaluation Framework, and (b) supports the efficient execution and coordination of SCAPE *action components* like preservation tools and workflows.

2.1 Main System Entities

The main entities that make up the SCAPE Platform are the Execution Platform and the Digital Object Repository.

2.1.1 Execution Platform

The SCAPE Execution Platform provides a tightly coupled data storage and processing network (called a cluster) that forms the underlying infrastructure for performing data-intensive computations on the SCAPE Platform. The Execution Platform specifically supports the deployment, identification, and parallel execution of SCAPE tools

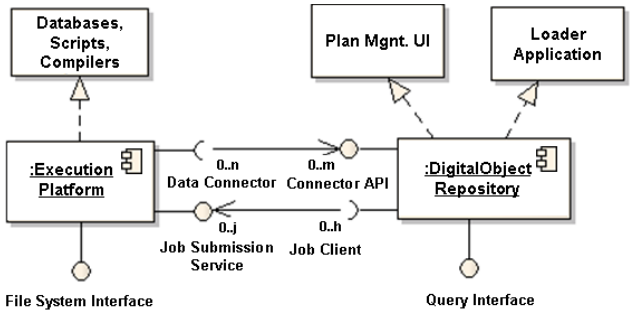

Figure 1: The SCAPE Platform comprises of two distinct system entities. The Execution Platform provides system-level support for storage and execution. The Digital Object Repository provides user-level support for data management and active preservation.

and workflows, and integrates with different data sources and data sinks. The system provides a set of command-line tools that support users in directly interacting with the system, for example to carry out data-management and preservation actions on the cluster. The Execution Platform does not provide graphical user interfaces per se but provides the below-described services to interact with client applications.

2.1.2 Digital Object Repository

A SCAPE Digital Object Repository (DOR) provides a data management system that interacts with the Execution Platform for carrying out preservation actions. The SCAPE DOR exposes also services to other entities developed in the context of SCAPE, for example for the Planning and Watch components. A SCAPE Digital Object Repository exchanges information with the Execution Platform via a defined API and may store or replicate its content directly to the Execution Platform's storage system. The DOR understands and manages Preservation Plans [2], triggers their execution, and may report to the Watch component. The repository may choose to preserve portions or the entire outcome of a workflow that has been executed against the content a DOR manages. It is therefore required that a DOR employs a corresponding data model as well as a scalable object store. Moreover, the object repository is responsible for aiding its user community in depositing, curating, and preserving digital content. A SCAPE repository reference implementation that is integrated with the platform's storage and execution environment is presently under development [1].

2.2 Interfaces and Services

Although the SCAPE Platform is designed to support software components and services provided by other SCAPE Sub-projects, its core entities may operate also independently from external services. A particular preservation scenario, for example, can be carried out autonomously once all required prerequisites (like data, tools, workflows) have been made available on a Platform instance. Figure 1 shows the interdependencies between the object repository and execution platform of the SCAPE Preservation Platform. The entities may interoperate with another using two defined services; (1) the data connector API, and (2) the job submission service. These services represent the two core functionalities of the SCAPE Platform: data management and computation. Although a typical Platform deployment might involve only a single repository and a single execution platform, the system is not limited to this configuration. Both, the Data Connector API and the Job Submission Service maintain an n:m relationship with their clients.

2.2.1 Data Connector API

The Data Connector API provided by the DOR is a service that allows clients to efficiently create, retrieve and update digital objects. The interface is specifically designed to support bulk data exchange allowing clients for example to access data directly through the storage system. The connector API is used by the Job Execution Service to efficiently obtain and update content and metadata from the repository that manages a particular information object. The execution platform resolves data based on references and may access data from different repositories or other data sources. Additionally, one repository can supply data to multiple (perhaps differently configured) clusters. A job execution can also be performed independently from a DOR and only rely on the Platform's internal data management component (e.g. a distributed file system and/or database).

2.2.2 Job Execution Service

This service provides an interface for performing and monitoring parallel data processing operations (jobs) on the platform infrastructure. The object repository acts as a client to this service in order to actively perform preservation operations (as for example defined by a preservation plans) against the data it manages. Depending on the repository implementation and use-case, the processed data may or may not reside on the Platform's storage network prior to the execution. A job execution service can be utilized by multiple clients and/or repositories. Also, a digital object repository may use the Platform's storage network without implementing a client to the job execution service. On the other hand, a SCAPE Digital Object Repository may maintain its own storage layer and use the execution platform only on demand. An example for a loosely integrated repository is the implementation of an active cache that can be used for performing scalable preservation activities like for example file identification without directly exposing the storage layer of the repository.

3. INFRASTRUCTURE DEPLOYMENT

The SCAPE Platform architecture does not prescribe a specific deployment or infrastructure provisioning model. The system may be set-up using a private or institutionally shared hardware infrastructure, or be hosted by an external data center. The architecture can also take advantage of virtualization and can be deployed on a private or public IaaS infrastructure. Depending on its level of integration, a single Platform instance may also be shared between multiple tenants. In SCAPE, a central deployment of the preservation platform, called the SCAPE Central Instance, provides a secure and project-wide shared hardware and software environment for evaluation and demonstration purpose. At the time writing this paper, a number of private instances of the SCAPE platform are being set-up at institutions participating the SCAPE project.

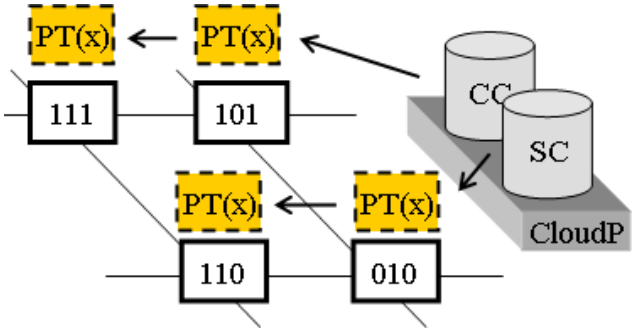

Figure 2: A setup for hosting the SCAPE Preservation Platform as a private cloud environment. Data is stored on persistent storage partitions directly on the nodes. The SCAPE software environment (PT) is deployed using transient virtual machine instances on demand on the nodes using a cloud platform.

3.1 Packaging Complex Environments

In addition to packaging software components, SCAPE is making use of virtual machine images in order to package complex software environments. This method allows us to provide the SCAPE Preservation Platform in the form of reusable images that can be published and launched on-demand using a private and/or public infrastructure like Amazon EC2[1]. Virtual machine images have been proven to be particularly useful for packaging complex systems which require a tedious installation and configuration procedure, if being installed manually. The SCAPE Platform is a complex distributed system that requires considerable effort and experience in order to be installed and configured. Provided as an pre-configured images, the Platform can be easily deployed on an arbitrary number of nodes, either manually or automatically based on cloud services (see section 5).

3.2 Employing a Cloud Hosting Model

The SCAPE Preservation Platform employs cloud technology on multiple levels includes hosting, computation, and storage. Infrastructure-as-a-Service (IaaS) provides a cloud hosting model that is well suited for an automated and scalable deployment of the platform environment. Figure 2 shows a simplified setup that utilizes a private cloud platform, as for example provided by the Eucalyptus [6] environment, to host an instance of the preservation platform. For simplicity, we assume that the cloud platform provides only two services, here called Cloud Controller (CC) and Storage Controller (SC). The cloud controller is capable of deploying virtual machine images on top of the cloud nodes (represented by the solid bordered boxes in the figure). The storage controller provides a service to store and retrieve the virtual machine images. Using this model, it is possible to bring up an instance of the preservation platform in a specific configuration. A platform instance comprising a (potentially large) number of platform nodes (PT(x)). Different nodes may have their own roles and behave differently within the platform instance. However, except nodes that provide external services, the platform internals are not visible to users/administrators.

[1]http://aws.amazon.com/ec2/

3.3 Storing Data in the Cloud

When operating a computer cluster, frequently occurring node failures are an expected rather than an exceptional case. Consequently, the number of nodes within a cluster may dynamically grow and shrink over time. File systems like HDFS[2] can deal with such behavior by replicating and dynamically recovering data between the nodes. A virtual machine instance that might have been deployed in a cloud environment operates on a transient file system by default. This is to say that data that is stored using the virtual machine's file system will be erased once the instance is shut down. For its storage network, the SCAPE preservation platform, however, demands a persistent storage media, mainly for two reasons. (a) We expect that most deployments will be hosted in (research departments of) individual institutions rather than in large data centers. Here, it cannot be guaranteed that the IT-infrastructure can be kept-up-and-running over very long periods without any interruptions (e.g. caused by maintenance work). Using transient storage, it would however be virtually impossible to shut down the entire cluster without loosing the data it holds. (b) A major design goal of the system is to support reconfiguration by deploying software environments, like specifically configured platform nodes, on demand. It is therefore required to separate the software environment's internal file system from the medium used to store content on the platform. The employment of a network attached storage system, however, would not satisfy the platform's scalability requirements, which demand to store data on a storage medium that is local to the processing unit of the node. As illustrated in figure 2, this can be solved by providing cloud nodes that provide a persistent storage layer upon with a software environment can be deployed dynamically (and operated locally to the data). While it is possible to establish such a configuration in a private cloud setting, this is usually not supported by commercial cloud offerings. Cloud storage is commonly provided as shared services that must be accessed via a network connection. The instantiation of environments on distinct physical computer nodes is in fact contradicting the public cloud model and can usually only be realized in a private setup.

4. APPLYING DATA-INTENSIVE TECHNOLOGIES

Preserving large volumes of loosely structured objects like data from scientific instruments, digitized objects, or multimedia content provides a resource demanding challenge. Both, scalable storage and processing capabilities are required to manage such data sets. In recent years, technologies like scalable file-systems, distributed databases, and frameworks for efficiently processing large quantities of data have emerged. We argue that the employment of scalable technologies can address and significantly enhance performance limitations of existing digital object management and preservation systems. These technologies were initially developed to capture and analyze vast amounts of data generated by Internet applications. Examples are data sets produced by social networks, search engines, or sensor networks. Such data sets have exceeded data volumes that can be organized using traditional database management tools. MapReduce [3] provide a prominent example of a distributed framework that

[2]http://hadoop.apache.org/hdfs/

is capable of processing huge amounts of data on top of a distributed file system. The MapReduce paradigm has also proven to be applicable to a range of domains. The SCAPE preservation platform is a software project that aims at employing scalable data management techniques for the purpose of digital preservation.

5. PRIVATE-CLOUD SETUP

AIT has started to deploy an initial version of the SCAPE Preservation Platform within a private cloud environment. The SCAPE infrastructure provides a Fully Automated Installation (FAI) server for configuring the cloud nodes. FAI is an automated installation framework that can be used to install Debian systems on a cluster. The service allows us to easily add new nodes to the system, which can be booted via a network card using PXE, a pre-boot execution environment most modern network cards support. The cloud infrastructure, presently consisting of 20 nodes, has been set up using the Eucalyptus cloud software stack. Eucalyptus is a private cloud-computing platform that provides REST and SOAP interfaces which are compliant with Amazon's EC2, S3, and EBS services. The infrastructure's front-end hosts the Eucalyptus Cloud Controller, the Cluster Controller, and the Walrus storage service. The worker nodes in the cloud run the XEN hypervisor and a Debian distribution that includes a Xen *Dom0 kernel*.

The initial preservation platform is based on an Apache Hadoop[3] cluster running MapReduce and HDFS, a set of preservation tools, and a number of MapReduce programs that have been developed to execute tools and/or specific workflows against data sets on the cluster. Using the cloud environment, a platform instance can be brought up dynamically by specifying a particular virtual machine image and the desired size of the cluster. The deployment of the platform instance supports the previously described requirements, namely dynamic deployment of environments and persistent storage. Each cloud node is configured with a physical data partition that can be hooked into the file system of a virtual machine instance. The platform nodes utilize this mechanism to establish a distributed file system that uses physical file system partitions underneath. Since data is already replicated by the Hadoop file system, it is not required to employ additional data redundancy mechanisms like RAID.

6. RELATED WORK

The employment of distributed and replicated storage and/or computation is a design decision that has been taken by a number of preservation systems. Prominent examples for systems that support geographically distributed and replicated data are LOCKSS [5] and iRods [7]. Preservation services like those developed in the context of the Planets project [8], provide a model that can be used to evaluate preservation tools in distributed environments. Many data management systems have been extended and/or configured to operate in cloud-based hosting environments. DuraCloud[4] and Fedorazon[5] are examples for repositories that leverage distributed cloud storage. The SCAPE platform is intended to support existing digital object repositories and preservation environments. It leverages data-intensive computing techniques to achieve scalability regarding storage, throughput, and computation allowing users to perform preservation actions and data analysis tasks at scale. Cloud and virtualization technologies are employed to support dynamic reconfiguration of the specific software environment required to interpret and manipulate a particular data item closely to its storage location.

[3]http://hadoop.apache.org/
[4]http://www.duracloud.org/
[5]http://www.ukoln.ac.uk/repositories/digirep/index/Fedorazon

7. CONCLUSION

The architecture of the SCAPE Preservation Platform aims at a versatile design that is intended to be applicable to digital content from many domains and to different preservation and information management systems. This paper discusses general design decisions and provides an overview of possible hosting models. We conclude that a private cloud environment, as described in this paper, can provides a very powerful, secure, and versatile solution for hosting the preservation platform in an institutional environment.

Acknowledgments

Work presented in this paper is primarily supported by European Community's Seventh Framework Programme through the project SCAPE under grant agreements No 270137.

8. REFERENCES

[1] Asseg, F., Razum, M., and Hahn, M. Apache Hadoop as a Storage Backend for Fedora Commons. In *7th International Conference on Open Repositories* (Edinburgh, UK, 2012).

[2] Becker, C., Kulovits, H., Guttenbrunner, M., Strodl, S., Rauber, A., and Hofman, H. Systematic planning for digital preservation: evaluating potential strategies and building preservation plans. *Int. J. on Digital Libraries 10*, 4 (2009), 133–157.

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

[4] King, R., Schmidt, R., Becker, C., and Schlarb, S. SCAPE: Big Data meets Digital Preservation. *ERCIM News 2012*, 89 (April 2012).

[5] Maniatis, P., Roussopoulos, M., Giuli, T. J., Rosenthal, D. S. H., and Baker, M. The LOCKSS Peer-to-Peer Digital Preservation System. *ACM Trans. Comput. Syst. 23*, 1 (Feb. 2005), 2–50.

[6] Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., and Zagorodnov, D. The Eucalyptus Open-Source Cloud-Computing System. In *9th IEEE/ACM International Symposium on Cluster Computing and the Grid* (2009), CCGRID '09, IEEE, pp. 124–131.

[7] Rajasekar, A., Wan, M., Moore, R., and Schroeder, W. A Prototype Rule-based Distributed Data Management System. In *in: HPDC workshop on "Next Generation Distributed Data Management, Paris, France.* (2006).

[8] Schmidt, R., King, R., Jackson, A., Wilson, C., Steeg, F., and Melms, P. A Framework for Distributed Preservation Workflows. In *Proceedings of The Sixth International Conference on Preservation of Digital Objects (iPRES)* (San Francisco, USA, 2009).