Final Data Center Deployments

Authors
Daniel Pop (West University of Timisoara), Tomasz Parkola (Poznań Supercomputing and Networking Center)

May 2014

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

Executive Summary

In addition to shared deployment of the infrastructure for large-scale testing called SCAPE's 'Central Instance', the deployments at UVT and PSNC, hosted on two different national and academic data and computing centers, are installed on hardware facilities and virtualization services (cloud middleware stack) available at partners' location. These installations are available to all SCAPE users, as well as additional scientific users that require data center hosting to cope with domain specific data volumes and use-cases

This report addresses the installation of the SCAPE Platform as a hosted preservation environment within data center infrastructure. The installation integrates the SCAPE Platform instance with large collections of video and medical datasets and makes them available to the user via remote access. In cooperation with TB.WP6, preservation tools and associated workflows are under deployment on the Platform instance to enable scalable processing, specifically for medical data services and semantic data analysis and interlinking in case of PSNC and UVT, respectively. The development and deployment of preservation workflows is, at the time of the delivery of this report, an ongoing task and therefore it will be addressed in detail in the documentation of milestone MS100 – Executable large-scale workflows, due in M42 (July 2014).

In case of PSNC, SCAPE platform is installed on (i) a production Hadoop cluster comprising 48 cores and 30TB storage space on HDFS and (ii) a development cluster comprising 4 virtual machines provisioned over an OpenStack private cluster. The deployments are also integrated with a cloud archiving environment, namely PLATON U4[1], developed in the frame of the PLATON project coordinated by PSNC. The clusters have the same configuration in terms of tools and SCAPE components: Cloudera Hadoop distribution (HDFS, MapReduce environment and HBase), HDFS PACS application for receiving DICOM files sent from WCPT hospital and a HDFS HL7 server for receiving the HL7 documents. The access to the infrastructure is provided by dedicated APIs: DICOM download service to access the DICOM files and Medical Data Centre search and browse interface used as a basic tool to access anonymised data for educational purposes.

UVT provides (i) a dedicated Hadoop cluster comprising 32 cores and 400GB storage space on HDFS and (ii) access to shared, mixed CPU-GPU cluster (Infragrid) with more than 300 cores available, plus seven NVidia Tesla M2070Q (with 6GB GDDR5) compute nodes. Additionally, (iii) an Amazon Web Services (AWS)-compatible private cloud powered by Eucalyptus software enables dynamic scaling out of compute and storage resources according to changing needs. Access to IBM BlueGene/P supercomputer (1024 CPUs with 4GB RAM per node) is possible as well. Different tools and SCAPE components are made available on different clusters, with Cloudera Distribution for Hadoop CDH4 fully installed on the dedicated Hadoop cluster. Specific tools for semantic data analysis and interlinking are made available on the Infragrid cluster. Remote access to the infrastructure is possible using SSH clients or HDFS FTP server for file transfer developed by UVT. To ease the deployment of SCAPE tools and components on heterogeneous clusters, UVT is using Puppet recipes for most common packages of SCAPE execution platform, including Taverna server, CDH4, Tomcat server and various components developed by PC subproject, such as jpylyzer, pagelyzer or matchbox.

[1] http://storage.pionier.net.pl/

Table of Contents

1. Introduction 1
2. Installation at Poznan Supercomputing and Networking Center 2
2.1. Infrastructure 2
2.2. SCAPE Components Deployed on the Infrastructure 3
2.3. Services for Remote Access 4
2.3.1. DICOM Download Service 4
2.3.2. Medial Data Centre Search and Browse 6
2.4. Executable Preservation Workflows for Medical Data Management 9
3. Installation at West University of Timișoara 9
3.1. Infrastructure 10
3.2. SCAPE Components Deployed on the Infrastructure 13
3.3. Services for Remote Access 15
3.3.1. Remote Access to Infrastructure 15
3.3.2. Remote Access to Storage Layer 16
3.3.3. Remote Access to Execution Layer 16
3.4. Executable Preservation Workflows for Semantic Data Analysis and Interlinking 17
4. Conclusions 20

1. Introduction

The primary goal of the SCAPE Platform is to enhance the scalability of preservation activities. In order to support this vision, SCAPE is applying data-intensive computing technologies in the digital preservation domain. This involves the employment of a scalable architecture and technologies for parallel processing, automated data decomposition, and error recovery. The preservation environment delivered by the SCAPE platform builds on top of a set of frameworks like Apache Hadoop, Taverna or Fedora 4 repository, as previously discussed in [*1*].

The SCAPE Platform described in [1, 2] exhibits a complex system composed of multiple software components and technologies combined to deliver a flexible and scalable environment for digital preservation. The Final Platform Release [3] delivers a pre-configured virtual machine providing a basic environment that allows users to experiment with different technologies. Contrasting the easy installation of the pre-configured virtual machine on a desktop computer, the deployment of SCAPE platform in a data center raises specific challenges due to sharing, accessing and availability of resources, such as managing the nodes of the infrastructure, service provisioning and maintenance of the integrity of software stack. We will present the infrastructure available in national (PSNC) and academic (UVT) data centers, what SCAPE components were installed in each deployment and how these can be accessed by users in order to execute preservation workflows.

Although migrating library applications to Cloud environment is not an easy task, many libraries are interested in using Cloud infrastructure services broadly across their businesses, whether it is a Public, Private or Hybrid Cloud. In this report we will also investigate the installation of SCAPE platform on two Infrastructure-as-a-Service (IaaS) environments, at PSNC and UVT.

The document is structured as follows: Chapter 2 presents the SCAPE installation at PSNC Data and Computing Centre, while Chapter 3 details the infrastructure and how to access it at UVT High Performance Centre. Each chapter starts by presenting the infrastructure (sections 2.1 and 3.1, respectively), afterwards detailing the SCAPE components installed at each partner. Sections 2.3 and 3.3 address the technical details of remote access to infrastructure and services on the two deployments. Both chapters describe in their last section the preservation workflows to be implemented. In the last chapter of the document we are discussing the lessons learnt during the deployment of the SCAPE platform in the two computing centres.

2. Installation at Poznan Supercomputing and Networking Center

Instytut Chemii Bioorganicznej Pan – represented by the Poznań Supercomputing and Networking Center (PSNC), affiliated to the Institute of Bioorganic Chemistry of the Polish Academy of Sciences, was founded in 1993 to build and develop computer infrastructure for science and education in Poznań and in Poland. This infrastructure includes metropolitan network POZMAN, High Performance Computing (HPC) Center, as well as the national broadband network PIONIER, providing the Internet and network services on international, domestic and local levels. With the development of the computer infrastructure, PSNC has been managing research and development within the field of new generation computer networks, high performance - parallel and distributed - computations and archive systems, cloud computing and grid technologies. PSNC is working also on the themes of green ICT, future Internet technologies & ideas, network safety, innovative applications, web portals, as well as creating, storing and managing digital content.

PSNC provides a broad range of data center services, including grid processing (e.g. QosCosGrid, UNICORE, gLite), SMP & cluster batch processing (e.g. SLURM, Torque), storage services (e.g. Lustre, NFS, CIFS, iSCSI) as well as cloud computing access interfaces based on the OpenStack middleware (e.g. virtual machines, object and block storage).

2.1. Infrastructure

PSNC infrastructure for the SCAPE project scenario related to medical data preservation is composed of two main clusters. The first one is a development platform and the second one is a production mode platform. Development platform is used for:

- Prototyping activities, such as validation of the tools and resourced used in the medical data scenario
- Testing activities, such as medical data transmission from the hospital to PSNC or demo applications for the access interface to anonymised medical data.
- Application of the cloud-computing environment (OpenStack) to run medical data preservation activities.

The second one is dedicated to production mode environment. It contains the same toolset as the development platform and additionally provides:

- Production mode environment with high reliability and stability level.
- More processing power for better performance and faster data processing.
- Larger storage space, so that it can handle all of the data envisioned to be stored at the PSNC's data center facilities.

Table 1 provides an overview of the PSNC's production cluster technical aspects, while Table 2 details technical information about PSNC's developer platform.

Attribute	Value
ID	**PSNC Hadoop Platform**
Description	6 physical servers *Fujitsu®* RX300 S4
Number of nodes	6

Total number of physical CPUs	12
CPU specs	Intel® Xeon® Processor 2.83 GHz
Total number of CPU cores	48
RAM	80 GB
Avg. CPU-cores per node	8
Avg. RAM per node	13 GB
Storage system	Fujitsu® FibreCAT SX40 Storage Subsystem (~30 TB for HDFS)
Network layer (inter-nodes)	1 GB Ethernet
Operating system	Ubuntu 12.04 LTS

Table 1 Technical details of PSNC's production mode cluster

Attribute	**Value**
ID	**PSNC Developer Platform**
Description	4 virtual machines launched in cloud powered by OpenStack
Number of nodes	4
Total number of physical CPUs	14
CPU specs	Intel® Xeon® Processor 2.33 GHz
Total number of CPU cores	32
RAM	16 GB
Avg. CPU-cores per node	8
Avg. RAM per node	4 GB
Storage system	Sun® NAS Storage connected via 4 Gbps Fibre Channel (~120 GB for HDFS)
Network layer (inter-nodes)	Virtual
Operating system	Ubuntu 12.04 LTS

Table 2 Technical details of PSNC's development cluster

The infrastructure covers also cloud archiving environment called PLATON U4[2]. This storage infrastructure has been developed in scope of the PLATON project and was coordinated by PSNC. PLATON U4 archiving services provide geographically dispersed, heterogeneous and replicated storage space for the purposes of archiving and long-term preservation. The idea in the context of medical data is that the data that are copied into PSNC facilities are located not only in the HDFS (which is then used in the data processing), but also in the PLATON U4 archiving services. Therefore the data are more secure (located on two different storage systems) and also ready for processing (because they are located in HDFS) as well as stored in reliable cloud storage infrastructure (because they are copied to PLATON U4 as well).

2.2. SCAPE Components Deployed on the Infrastructure

Each of the clusters available at PSNC has the same configuration in the context of the components identified or developed in the framework of SCAPE project. There are two groups of tools, which are deployed on the SCAPE clusters. One of them is responsible for data transferring and data storage, and the second one is responsible for data processing. The first group includes:

[2] http://storage.pionier.net.pl/

- HDFS PACS, which is a server application for receiving DICOM files, sent from the WCPT hospital. It is a customized version of the Picture Archiving and Communication System based on dcm4che toolset[3]. The tool was modified in a way that it can store the data (DICOM files) on the HDFS cluster and also archive them in the cloud storage. In order to send data the user needs to use the client application, which is part of the dcm4che toolset. The modified version of the server application is available at: https://git.man.poznan.pl/stash/projects/SCAP/repos/dicom/browse

- HDFS HL7, which is a server application for receiving HL7 documents, sent from the hospital. It receives textual data encoded in XML and stores them on HDFS as well as on the cloud storage. In order to send data the user needs to use a client application developed in the framework of the SCAPE project together with the HDFS HL7 server application. The whole toolset is available at: https://git.man.poznan.pl/stash/projects/SCAP/repos/hl7/browse

The group of tools dedicated to help with data processing is composed of the components included in the Cloudera toolset. The components used in the medical data preservation and processing setup are as follows:

- Hadoop execution environment, which is intended to run analysis jobs on the XML files provided by the WCPT hospital. The files include information about the patient's medical history, including laboratory results, diagnosis results and general information about the disease.
- HDFS component, which is used to store the medical data for processing purposes. It also acts as a copy of the data, which are also stored in the cloud storage environment.
- HBase component used for keeping information about medical data. The information is needed to provide fast access to the information needed in the educational and full data access scenarios identified by the WCPT hospital.

2.3. Services for Remote Access

The medical data stored at PSNC can be accessed using a dedicated API for DICOM data retrieval or with the use of a dedicated interface for searching and browsing anonymised data. There are two main components, which provide access to the data:

- DICOM download service which provides access to specific DICOM files required either by the WCPT hospital (when providing its users with access to full patient's history) or by the Medical Data Center search and browse interface when presenting CT or RTG examination results in an online DICOM viewer.

- Medical Data Centre search and browse interface is used as a basic tool to access anonymised data for educational purposes. This functionality is still under development as it is to be completed in M42 as stated in Description of Work (MS100 Executable large-scale workflows). Nevertheless the access interface is already designed and initially tested. The toolset has been also already deployed in the PSNC's data center environment as a prototype.

2.3.1. DICOM Download Service

[3] http://www.dcm4che.org

DICOM download service provides access to the DICOM files via the two subsequent access services. The first one is able to provide list of URLs to DICOM files based on the accession number related to particular referral and therefore examination. The second one is for downloading particular DICOM files with the use of the URLs provided by the first service. In the following part of this section, the API for the two services is described, namely DICOM List Service API and DICOM File Service API.

DICOM List Service API returns a list of URLs to all DICOM files for a given accession number. The URLs in the list can be directly used to access DICOM files via the DICOM File Service API. In order to obtain a list of URLs to DICOM files for a given accession number the following HTTP request needs to be submitted:

```
https://<mdc.scape.psnc.pl>:<8080>/dicom/<hospital_name>?AccessionNu
mber=<accession_number>
```

where:

- <server> is the domain name or IP of the server on which the DICOM List Service API is located,
- <port> is the port number under which the DICOM List Service API is available,
- <organization_name> is the name of organization which owns the files to be downloaded,
- <accesstion_number> is the referral identifier for the examination results (in form of DICOM files).

Based on above description, the following request is an example of requesting examination results:

```
https://mdc.scape.psnc.pl:8080/dicom/wcpt?AccessionNumber=3206/KT/20
11
```

For such a request the following response will be generated:

```
https://mdc.scape.psnc.pl:8080/dicom/wcpt/1.2.840.113619.2.278.3.168
428016.836.1323671144.715.7/1.2.840.113619.2.278.3.168428016.836.132
3671144.893.139
https://mdc.scape.psnc.pl:8080/dicom/wcpt/1.2.840.113619.2.278.3.168
428016.836.1323671144.715.7/1.2.840.113619.2.278.3.168428016.836.132
3671144.893.21
https://mdc.scape.psnc.pl:8080/dicom/wcpt/1.2.840.113619.2.278.3.168
428016.836.1323671144.715.7/1.2.840.113619.2.278.3.168428016.836.132
3671144.893.23
https://mdc.scape.psnc.pl:8080/dicom/wcpt/1.2.840.113619.2.278.3.168
428016.836.1323671144.715.7/1.2.840.113619.2.278.3.168428016.836.132
3671144.893.237
https://mdc.scape.psnc.pl:8080/dicom/wcpt/1.2.840.113619.2.278.3.168
428016.836.1323671144.715.7/1.2.840.113619.2.278.3.168428016.836.132
3671144.893.244
https://mdc.scape.psnc.pl:8080/dicom/wcpt/1.2.840.113619.2.278.3.168
428016.836.1323671144.715.7/1.2.840.113619.2.278.3.168428016.836.132
3671144.893.320
https://mdc.scape.psnc.pl:8080/dicom/wcpt/1.2.840.113619.2.278.3.168
428016.836.1323671144.715.7/1.2.840.113619.2.278.3.168428016.836.132
3671144.893.397
```

```
https://mdc.scape.psnc.pl:8080/dicom/wcpt/1.2.840.113619.2.278.3.168
428016.836.1323671144.715.7/1.2.840.113619.2.278.3.168428016.836.132
3671144.893.449
https://mdc.scape.psnc.pl:8080/dicom/wcpt/1.2.840.113619.2.278.3.168
428016.836.1323671144.715.7/1.2.840.113619.2.278.3.168428016.836.132
3671144.893.450
https://mdc.scape.psnc.pl:8080/dicom/wcpt/1.2.840.113619.2.278.3.168
428016.836.1323671144.715.7/1.2.840.113619.2.278.3.168428016.836.132
3671144.893.468
```

In the response, each line identifies a certain URL to the DICOM file in the examination related to a given accession number (which is provided in the request).

DICOM File Service API provides access to the DICOM files identified by the unique identifier. Normally, the URL to the DICOM files should be obtained from the DICOM List Service API by providing accession number of a certain examination. The DICOM File Service returns DICOM files for given `SeriesInstanceUID` (DICOM tag) and `AffectedSOPIntanceUID` (DICOM tag). The `SeriesInscanceUID` is unique in the context of whole organization and identifies specific examination, while the `AffectedSOPIntanceUID` is unique identifier of a DICOM file within a certain examination. In order to obtain a DICOM file from the DICOM File Service the following request needs to be submitted:

```
https://mdc.scape.psnc.pl:8080/dicom/<hospital_name>/<series_instanc
e_uid>/<affected_sop_instance_uid>
```

where:

- <server> is the domain name or IP of the server on which the DICOM File Service is running,
- <port> is the port exposing the DICOM File Service,
- <organization_name> is the name of the organization which owns the file to be downloaded,
- <series_instance_uid> is the unique identifier of the examination in which the file to be downloaded exists,
- <affected_sop_instance_uid> is the unique identifier of the DICOM file within the examination.

Based on above description, the following request is an example of requesting a DICOM file:

```
https://mdc.scape.psnc.pl:8080/dicom/wcpt/1.2.840.113619.2.278.3.168
428016.836.1323671144.715.7/1.2.840.113619.2.278.3.168428016.836.132
3671144.893.468
```

In the response, requested DICOM file is returned as a data stream.

2.3.2. Medial Data Centre Search and Browse

The Medical Data Centre search and browse interface is for accessing anonymised data stored at the data center. The idea is to provide an easy and simple interface to be used during various courses on medical universities for educational purposes. The search and browse functionality needs to take into consideration different criteria, such as type of disease or type of examination. The results can be then provided to students or practitioners to analyse and discuss specific aspects of the medical case. The crucial aspect here is a professional description of particular examination results such as laboratory results or diagnosis. Therefore, it is important to have real-life cases coming from the hospital that conducts professional patient's treatment. The functionality for searching and browsing has been developed as a web portal that allows users to view anonymised medical cases coming from the WCPT hospital. In order to search medical cases stored in the PSNC's data center, the user can apply the following criteria:

- description and code of disease type according to ICD-10 (searches in underlying disease and concurrent disease),
- description and code of procedures according to the ICD9,
- age of the patient (range can be provided),
- place of residence (city of residence),
- date of admission to the hospital,
- date of discharge from the hospital,
- sex of patient,
- conducted diagnosis (multiple choice from: CT, EKG, RTG, USG, Bronchoscopy).

The user interface, which allows specifying the criteria, is depicted on *Figure 1*.

Figure 1 Search box in the Medical Data Centre search and browse user interface

When the search is executed then the list of results is presented to the user. The results include medical cases that match specified criteria. The user can then browse the list of results and present details of a specific case. The details include links to online viewer that displays DICOM files using the DICOM List Service API and DICOM File Service API. An example of results returned by the service is depicted on the Figure 2, while the online DICOM viewer is presented on Figure 3.

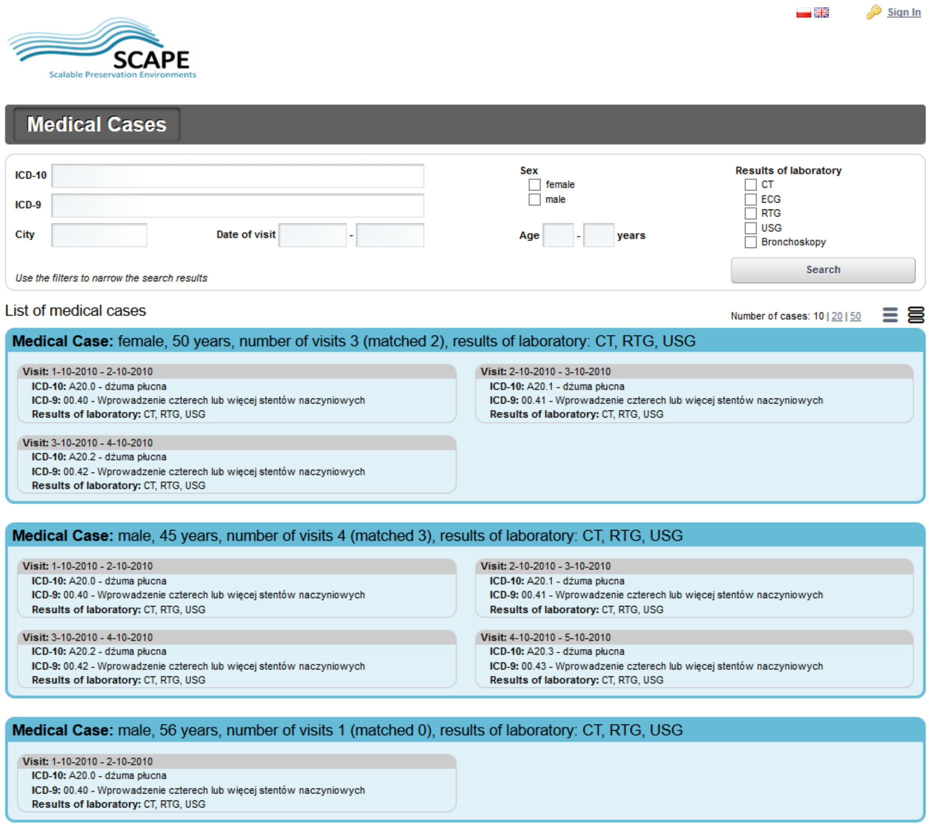


Figure 2 Search results: list of medical cases presented in a detailed view

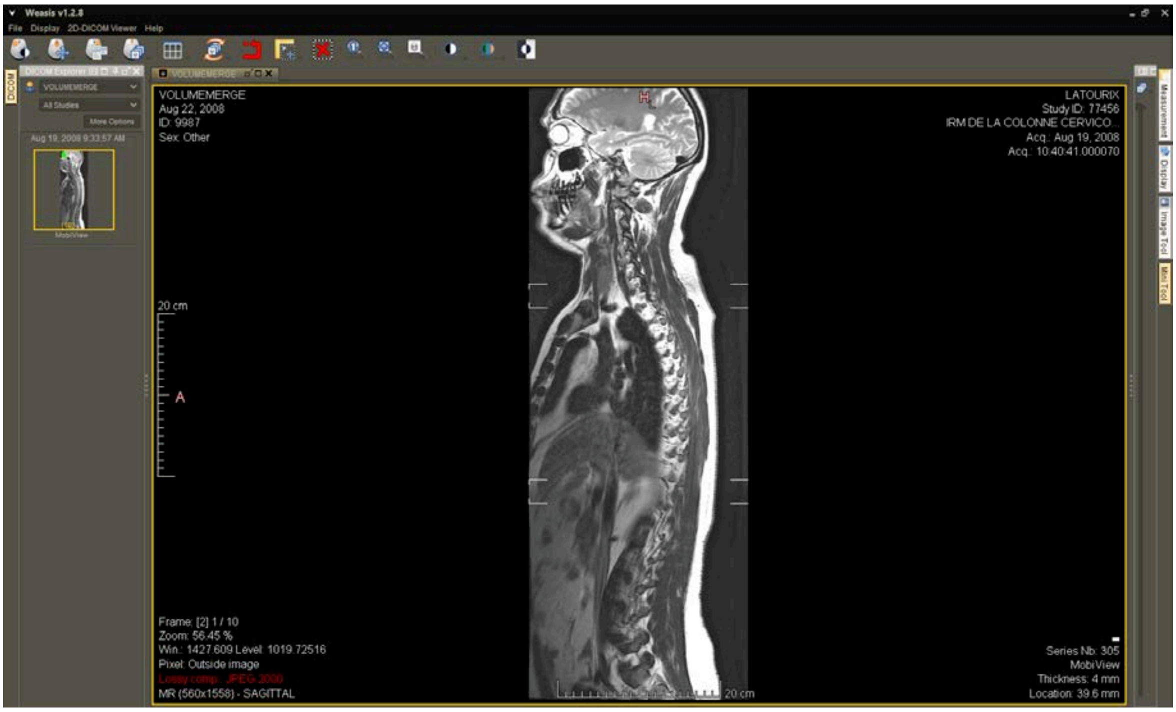
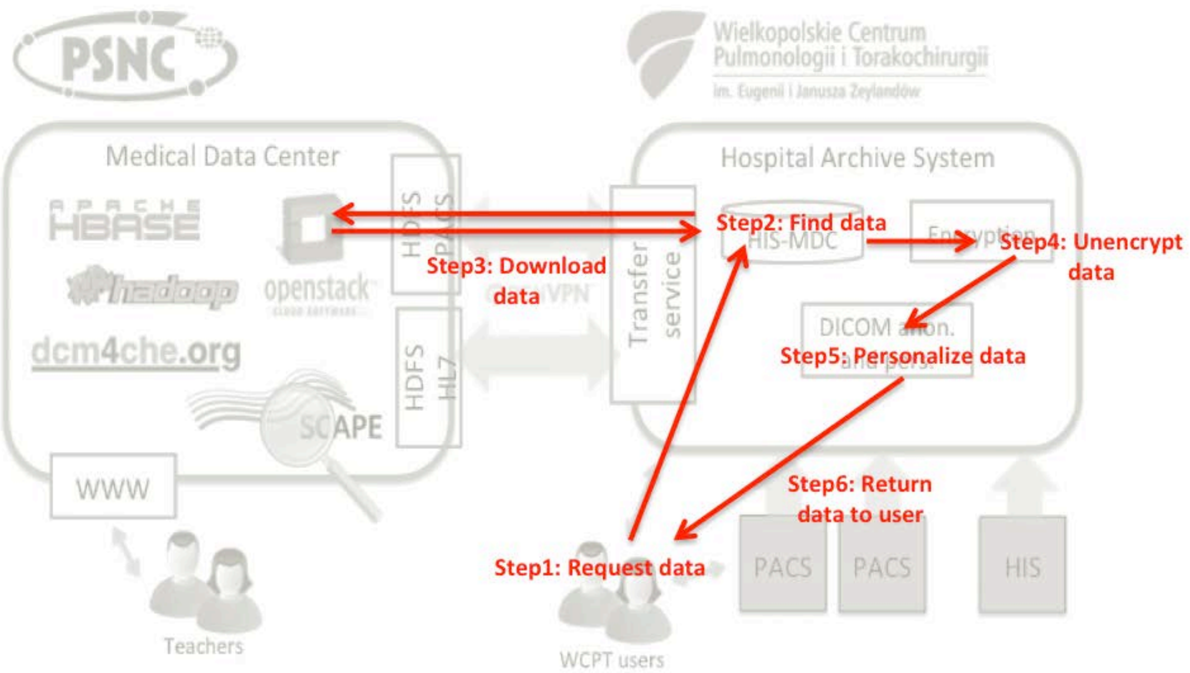


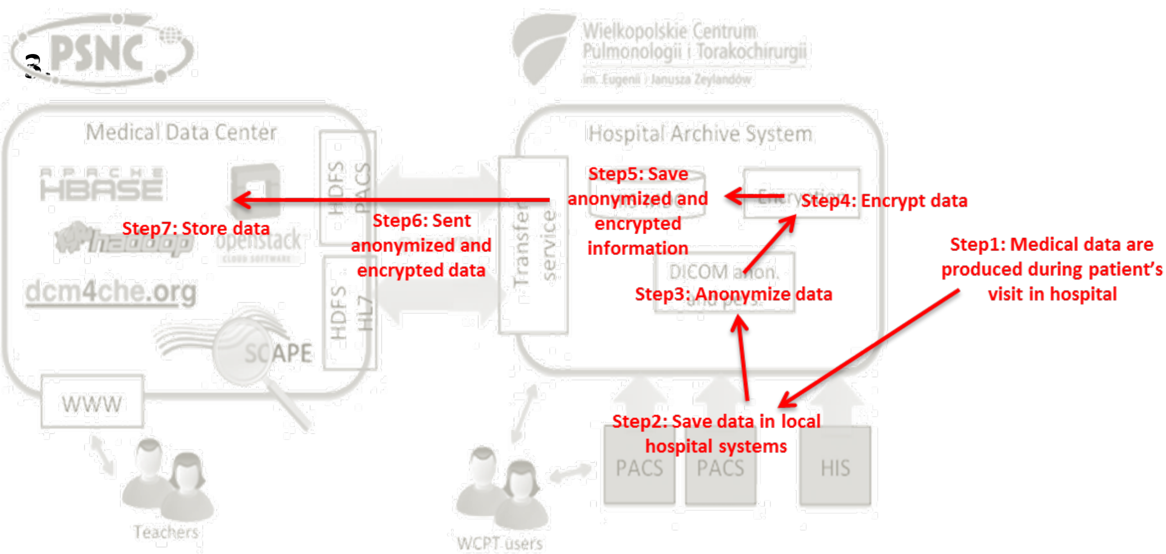
Figure 3 Online DICOM files viewer

2.4. Executable Preservation Workflows for Medical Data Management

The figures below illustrate the access and ingest scenarios defined by PSNC together with WCPT. More details about the implementation of these scenarios and obtained results will be provided in *MS100 Executable large-scale workflows.*



3.



Installation at West University of Timișoara

West University of Timișoara (UVT) was founded on 1944 as a classic university meant to be representative for the west part of Romania, now with more than 15,000 students. The Faculty of Mathematics and Computer Science has two departments: the Department of Mathematics from 1948 and the Department of Computer Science from 1971. The computational power of the High Performance Computing (HPC) center (http://hpc.uvt.ro) managed by Department of Computer Science is sustained by an IBM cluster (InfraGrid) with 400+ cores (since 2009), a supercomputer IBM BlueGene P with 4000+ cores (since 2011) and a GPU cluster with 3000+ CUDA cores (since 2012). UVT presently supports access to its facilities based on different mechanisms. Parallel computations on clusters or BlueGene/P are typically performed via SSH connections by the users.

3.1. Infrastructure

Within SCAPE project, UVT is offering both a dedicated infrastructure and remote access to shared infrastructure.

UVT installed one **dedicated** Hadoop Cluster based on HP servers (see Table 3 Technical details of dedicated UVT Hadoop Platform). The remote access to this cluster is described in section 3.3.

Attribute	**Value**
ID	**UVT Hadoop Platform**
Description	Platform setup for preservation experiments
Number of nodes	8 HP ProLiant DL-385 servers
Total number of physical CPUs	16
CPU specs	CPU AMD Opteron 2.4 GHz, dual core, 1 MB L2 cache per core
Total number of CPU cores	32
RAM	32 GB
Avg. CPU-cores per node	4
Avg. RAM per node	4 GB
Storage system	• NFS Staging: max 512GB • HDFS Storage: ~400GB
Network layer (inter-nodes)	Each node is equipped with 2 NICs 1 Gb/s: • 1 NIC for NAS access • 1 NIC for Hadoop Operation
Operating system	CentOS 6
Hadoop ecosystem	Cloudera CDH 4 distribution: • HDFS and MapReduce • Apache HBase • Apache Hue • Apache Oozie • Apache ZooKeeper

Table 3 Technical details of dedicated UVT Hadoop Platform

The following tables describe the **shared** infrastructure available to SCAPE partners at UVT. Detailed and up to date description of UVT's infrastructure is available here[4]. The remote access to the shared infrastructure is provided through a bundle of services, which are in detail described in section 3.3.

Attribute	Value
ID	**InfraGRID CPU Cluster**
Description	Grid-services oriented infrastructure
Number of nodes	40
Total number of physical CPUs	80
CPU specs	CPU Intel Quad-Core 2.00Ghz
Total number of CPU cores	320
RAM	800 GB
Avg. CPU-cores per node	8
Avg. RAM per node	20 GB
Storage system	145GB local SAS
Network layer (inter-nodes)	• 40 Gbps 4xQDR Infiniban • 4 Gbps FiberChannel Fabrics with remote storage
Scheduling system	IBM® LoadLeveler

Table 4 Technical details of InfraGRID cluster (CPU)

Attribute	Value
ID	**InfraGRID GPU Cluster**
Description	GP GPU cluster
Number of nodes	• 7 Compute Nodes Intel XEON 3.46Ghz + NVidia Tesla M2070Q (448 cores, 6GB GDDR5) • 2 Head Nodes Intel XEON 2.66Ghz
Total number of physical CPUs & GPUs	CPU: 14 (7x2) GPU: 7
CPU & GPU specs	CPU: 6-core Intel XEON 3.46Ghz GPU: NVidia Tesla M2070Q (448 cores, 6GB GDDR5)
Total number of CPU & GPU cores	Total CPU cores: 42 Total CUDA cores: 3,136
RAM	288 GB
Avg. CPU/GPU-cores per node	6 / 448 cores per compute node
Avg. RAM per node	32 GB
Storage system	18 TB (60x300 GB SAS disks) raw
Network layer (inter-nodes)	2x40 Gpbs Infiniband
Scheduling system	IBM® LoadLeveler

Table 5 Technical details of InfraGRID cluster (GPU)

Attribute	Value
ID	**IBM® BlueGene/P**
Description	Fully loaded single BlueGene/P rack
Number of nodes	1024
Total number of physical CPUs	1024
CPU specs	Quad-Core PowerPC-450 850Mhz

[4] http://hpc.uvt.ro

Total number of CPU cores	4096
RAM	4096 GB
Avg. CPU-cores per node	4
Avg. RAM per node	4 GB
Storage system	48 x 320 GB SAS
Network layer (inter-nodes)	3-D Torus network
Scheduling system	IBM® LoadLeveler

Table 6 Technical details of BlueGene/P

On top of the bare metal infrastructure described above we have set up a **virtualized infrastructure**, a Private Cloud, Amazon Web Services (AWS)-compatible powered by Eucalyptus[5] software enabling pooling compute, storage, and network resources that can be dynamically scaled up or down as application workloads change. The hardware used for this deployment has the following configuration (for each blade server): CPU: 2 x 2.0Ghz Quad-Core Intel Xeon CPU; 10GB RAM, 148 SAS 15K RPM Storage, 2 x 1Gbps, 1x 40Gpbs Infiniband network interfaces and as storage connection 2 x 4Gbps Fibre Channel. Currently one blade server is used to host the collocated Eucalyptus services and 7 others for hosting VMs.

Attribute	**Value**
ID	**Eucalyptus Cloud**
CPU	1 to 8 vCPU servers available
Memory	0.25GB 0.5GB, 1GB, 2GB, 4GB, 6GB, 8GB available.
Network	Public communication network – 1Gbps / 10Gbps uplink Private communication network – 1Gbps Storage network – 4Gbps FC
Storage	The storage is offered using a FC network from a shared NAS where block devices are applicable. Also NFS is used to share object store service (S3).
Images	Pre-created instance-store (without persistence); block device images (with persistence).
Supported OS	Any Linux distribution supporting kernel v2.6+ enabled for KVM (32 & 64bit architecture)
Management & Configuration Access	Eucalyptus two types of web console access: Administrative console: account level management (define users, groups, policies for institution account access) User web console: user level management (define security groups, ssh keys, start/stop/terminate instances, create volumes, create snapshots, allocate IP addresses) User CLI console: same as user web console but using the command line interface tools.
How are IP	The entire project has 15 public IP addresses shared. VMs are not

[5] www.eucalyptus.com

addresses managed?	automatically assigned with public IP addresses; each user must manually assign one at boot time;
Backup	Backup can be achieved through the instant snapshot function.
Firewalling	Each account can define its own firewall policy using security groups. The default security group drops any communication with the instances;

Table 7 Technical details of Eucalyptus cloud

3.2. SCAPE Components Deployed on the Infrastructure

The architecture of the SCAPE Preservation Platform, as described in D4.1 SCAPE Architecture Design [1], defines four layers, briefly described below:

- **Storage Layer** is the main storage environment, accessible by the Execution platform; preferred SCAPE storage solution is **Hadoop Distributed File System (HDFS)**;
- **Execution Layer** provides robust and scalable computational services; the **MapReduce** paradigm that provides a parallel programming and a distributed runtime environment is commonly utilized within SCAPE for processing wide range of content type;
- **Data Management Layer** hosts the Digital Object Repository (DOR) that maintains structured information about the objects; three out of four SCAPE repositories use **Fedora Commons**[6] as underlying core repository component;
- **User Application Layer** delivers user applications that are only loosely coupled with the data management, execution, and storage layers; the majority of application layer has been implemented using **Taverna**[7] **software stack**, including the Taverna Workbench and the Taverna Server.

Atop of widely used components of the four layers of SCAPE architecture, we looked to identify the most used and mature tools in Test Beds (TB) and Preservation Components (PC) subprojects (D18.1 Final Evaluation Report [4]) and inspired by D4.2 Final Platform Release [3] as well, the following tools were included in the deployment list: MapReduce tool executer (**ToMaR**), **Pagelyzer**, **Jpylyzer**, **Matchbox** and **xcorrSound**.

Storage Layer

The Data Storage Services provide the project partners with storage space on UVT's infrastructure. The storage services include both GPFS storage (available as FTP/SFTP) and Hadoop HDFS storage. The HDFS storage service is accessible both directly from Hadoop jobs and also by means of the HDFS FTP Server developed by UVT in the frame of the project. See section 3.3.2 for details on Remote Access to Storage Layer.

Execution Layer

The execution services provided by UVT cover several facilities, including (but not restricted to): Batch Scheduling, MapReduce services, and QosCosGrid Compute API.

6 www.fedora-commons.org
7 www.taverna.org.uk

The Batch Scheduling service is based on the IBM® LoadLeveler scheduling system. The service allows the SCAPE partners to use UVT's shared facilities. This service also allows access to specialised resources, for instance the GPU computing nodes, of particular interest for semantic data analysis and interlinking workflow (section 3.4). The Batch Scheduling service is complemented by the QosCosGrid API developed by Application Department from PSNC. The QCG API allows integration with partners' applications. Reader is referred to section 3.3.3 for technical details on Remote Access to Execution Layer.

Besides the two batch oriented services described above, UVT also provides access to Hadoop resources both on a SCAPE dedicated cluster, UVT Hadoop Platform (see section 3.1), and on demand Hadoop clusters based on the SCAPE Cloud Deployment Toolkit[8]. See section 3.3.3 for details on executing MapReduce jobs on dedicated Hadoop cluster.

On top of providing the aforementioned execution services, UVT also provides support for customising existing services for the requirements of SCAPE Users. Examples include specific runtime configurations, both software and hardware, such as off-screen CUDA rendering support provided specifically to BUT, allowing the execution of OpenGL applications on top of headless GPU systems.

Data Management Layer

UVT also provides IaaS hosting services for SCAPE Consortium members. These hosting services include the ability of deploying Virtual Machines on top of the Eucalyptus Cloud. For instance, FIZ team members use the IaaS hosting facility leveraging on four VMs (`scape-fiz-1.info.uvt.ro` to `scape-fiz-4.info.uvt.ro`) to host Fedora Directory development.

User Application Layer

Different tools and utilities developed in SCAPE project are managed at this layer through Puppet recipes and can be installed on a variety of infrastructures. For now, these tools are installed on UVT Hadoop Platform and made available for registered users. The Open Planets GitHub repository[9] contains Puppet recipes for most common components and tools of the SCAPE User Application Layer: Taverna server, Tomcat server, jpylyzer, pagelyzer and xcorrSound. Below is an example of Puppet module for Taverna installation:

```
class scape_taverna::install{

  class{'scape_tomcat':}

  user { "add-taverna-user":
    name   => "taverna",
    ensure => present,
  }

  exec { "download-taverna-server":
```

[8] http://wiki.opf-labs.org/display/SP/Portability+of+SCAPE+Platform+over+Multiple+Cloud+Environments
[9] https://github.com/openplanets/scape-puppet-modules

```
    command => "wget -O /mnt/taverna_server.war
http://web.info.uvt.ro/~caius.bogdanescu/taverna_server.war",
    path    => '/usr/bin',
    onlyif  => "test ! -e /mnt/taverna_server.war"
  }

  scape_tomcat::deploywebapp {'taverna':
    appname     => taverna,
    source      => "/mnt/taverna_server.war",
    destination => "${scape_taverna::tomcatWebApps}/taverna.war",
    require     => File['apply-tomcat-server-template'],
  }
}
```

3.3. Services for Remote Access

3.3.1. Remote Access to Infrastructure

In order to receive access to the UVT resources available, one must download the resources request form[10], fill it in, sign it and send it by email [PDF format] at bgsupport@hpc.uvt.ro. After receiving an account for accessing the infrastructure, one can proceed with one of the following modes of access.

"Low level" access to **dedicated** (UVT Hadoop Platform) and **shared** (InfraGRID CPU/GPU Cluster and BlueGene/P) infrastructure is possible in using Secure Shell (`ssh`) protocol either

- using **command line** mode:
- From Windows: using PuTTy[11] or OpenSSH [12]clients
- From a Linux-based distribution: using installed ssh client

- or using **Graphical User Interface (GUI)** mode:
- From Windows / Linux: using NoMachine NX[13] client

providing server name `head-infragrid.info.uvt.ro`, port `22` and protocol SSH.

Once logged into UVT's head node, to get access to UVT Hadoop Platform use `ssh` command:
`ssh ieat-hdp-1.priv.hpc.uvt.ro`

If connected in GUI mode, then one can display and manage the Hadoop ecosystem using Cloudera Manager (Free Edition) available at:
`http://ieat-hdp-1.priv.hpc.uvt.ro:7180/cmf/services/status`

10 http://hpc.uvt.ro/wiki/BlueGene?action=AttachFile&do=view&target=BGP-request_form-v0.4.pdf
11 http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
12 http://www.openssh.org
13 https://www.nomachine.com

3.3.2. Remote Access to Storage Layer

Besides direct access to HDFS from Hadoop CLI, the HDFS FTP server, running on one of the Hadoop cluster nodes, allows remote manipulation of the HDFS file system from legacy applications or commodity file transfer utilities. Check project page on Open Planets GitHub[14] for source code and installation instructions. In order to access the service users need to use a FTPS client for connecting to ftp.hdfs1.scape.info.uvt.ro. The operations allowed are typical to any FTP service, except some permission commands that could not be mapped to HDFS commands. Alternatively, the HDFS can be accessed using Hadoop tools, for example `hadoop fs -ls` – to list HDFS content once logged onto UVT Hadoop Platform. The GPFS storage is mounted on InfraGRID cluster as `/u` and can be operated using FTP/SFTP.

3.3.3. Remote Access to Execution Layer

In order to execute MPI applications on InfraGRID cluster, do follow the next steps:

1. Create a Job description file
2. Request the resources specified in the JOB execution file using the `llrun` command
3. Run your application using `mpirun`

A sample Job description file looks like:

```
Example1.job
# @ job_type = MPICH
# @ environment = COPY_ALL;
# @ node = 2
# @ tasks_per_node = 2
# @ class = small
# @ queue
```

To execute the job on the cluster, issue the following command:

```
llrun -f Example1.job
```

Using `class` parameter one can control the target where job is executed and the number of nodes and time allocated to the job. Use `llclass` to learn more about available classes, worth mentioning at this point being `gpu` class that will submit the job on the GPU cluster, all other classes submitting the job to the CPU-based cluster.

The same method for executing jobs on the cluster can be used for other types of jobs, not only MPI, and can be triggered/executed from outside the cluster by means of SSH connections. This allows Taverna Server to execute remote SSH jobs on InfraGRID cluster.

Besides direct access to the LoadLeveler scheduling system, the data center services are accessible by QosCosGrid/QCG GRID middleware that provides both an API usable by various applications, and a series of clients allowing simple access from within scripts, for example:

- `qcg-sub`: submit a job

[14] https://github.com/openplanets/maroodi

- `qcg-list`: list active jobs
- `qcg-info JOB_ID`: get job information
- `qcg-cancel JOB_ID`: cancel a job

QCG uses a Job Description Language similar to the LoadLeveler one:

```
Example2.job
#QCG queue=scape
#QCG host=infragrid
#QCG nodes=4:5
#QCG walltime=PT8H
#QCG memory=1024
#QCG stage-in-file=input.txt -> input.txt
#QCG stage-out-file=output.txt -> output.txt
#QCG application=scape-tool1
```

Executing MapReduce applications on UVT Hadoop Cluster is accomplished by using common Hadoop tools and services once logged into the cluster.

3.3.4. Remote Access to Eucalyptus Cloud

The Private Cloud, Amazon Web Services (AWS)-compatible powered by Eucalyptus software is remotely accessible as well:

- https://cloud.info.uvt.ro:8888/ - Web interface for resources management (create, start, stop VMs, create/delete security groups, keypairs, volumes etc)
- https://cloud.info.uvt.ro:8443/ - Web interface for administrative actions: change password, manage account users/permissions, create/delete certificates, keys etc.

Note that accounts are manually created. For this purpose please send email to support@info.uvt.ro with organisation details (organisation name, plus name, surname and email address of each user).

Additional tutorials are available online for Environment setup[15] and VM control[16].

3.4. Executable Preservation Workflows for Semantic Data Analysis and Interlinking

The figure below illustrate the semantic data analysis and interlinking workflow defined by BUT and UVT. Workflow is available on myExperiment platform: http://www.myexperiment.org/workflows/4268.html

[15] http://solsys.ro/kb/cloud/eucalyptus3/tutorials/usage/Introduction/Environment
[16] http://solsys.ro/kb/cloud/eucalyptus3/tutorials/usage/Introduction/VMControl

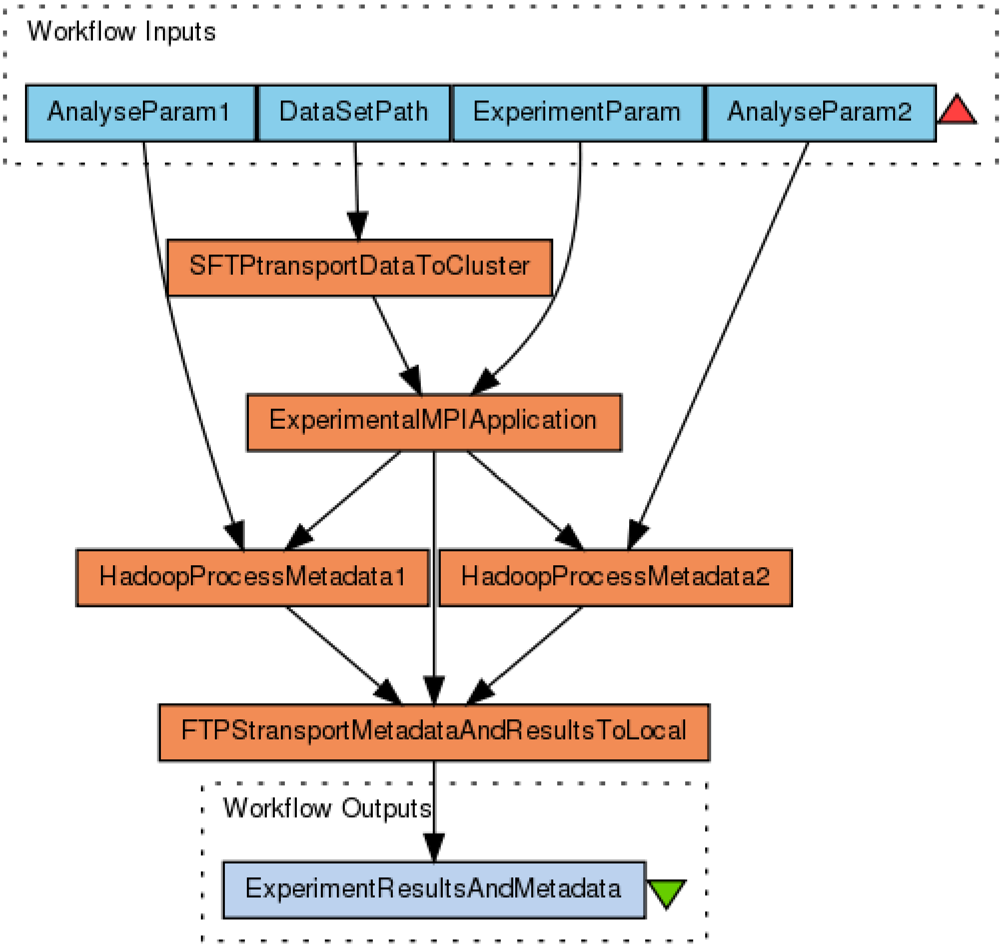


Figure 4 Semantic data analysis and interlinking workflow

Inputs of the WF:

DataSetPath Path to local data set directory on BUT local machine.

ExperimentParam Input parameters for experimental MPI application for video content analysis.

AnalyseParam1 1st set of parameters for Hadoop tool for analysing metadata generated by the experimental MPI application.

AnalyseParam2 2nd set of parameters for Hadoop tools for analysing metadata generated by the experimental MPI application.

Outputs of the WF:

ExperimentResultsAndMetadata Metadata and results of experiment and results of Hadoop analysis.

Experimental MPI Application running on cluster, providing video semantic analysis, while saving details about cluster environment (e.g. memory usage, available drivers, names of currently used nodes, processing time etc.) to XML metadata. The experimental applications will be applications for

- reconstruction of 3D model from set of images, there will be cca. 270,000 images from stereo camera, we will divide them to smaller groups and reconstruct smaller model from each group of images, and
- Annotations of mountain scenery images - we try to align real images to synthetically rendered panoramas (from high resolution DEM) while varying some parameters (sampling, resolution etc.).

The dataset will contain text and jpeg images and some text files (XML format). The metadata generated by the experiment will be large scale as the application on the same dataset will be run many times with varying parameters' values.

HadoopProcessMetadata1(2) is a Hadoop application analysing generated metadata, extracting some particular information about environment w.r.t. AnalyseParameter1(2), for example list of results which were being processed for longer than 1 minute or average memory usage of cluster when processing inputs from 1 to 100.

SFTPtransportDataToCluster and **FTPStransportMetadataAndResultsToLocal** are jobs responsible for datasets and metadata transfer (inputs or results) to/from HDFS.

Details on the implementation of this scenario and obtained results on UVT infrastructure will be provided as part of *MS100 Executable large-scale workflows.*

4. Conclusions

In this report we presented the installation of SCAPE Platform at two data and computing centers, a national one (PSNC) and an academic one (UVT). Each partner is working closely with preservation scenario providers WCPT and BUT, respectively, with whom we identified the tools and SCAPE components required to execute the preservation workflows.

The installation at PSNC is focused on storing, accessing and processing of medical records composed of DICOM files and HL7 documents encoded as XML files. Specific tools for handling medical records, such as Picture Archiving and Communication System (PACS), were customised for the integration within SCAPE Platform (HDFS storage) and metadata about medical data is stored on HBase. This setup will allow further distributed processing to be applied on medical data. The system is setup on two clusters, one for production – deployed on a bare-metal infrastructure – and another one for development – deployed on IaaS powered by OpenStack – that will ensure compliance with quality assurance policies and procedures implemented at PSNC. The environment setup so far will allow the implementation and execution of ingest and access scenarios defined together with WCPT.

The final deployment of SCAPE at UVT's data and computing center comprises of a dedicated cluster for SCAPE Execution Platform (Cloudera Hadoop Distribution, Taverna server and frequently used preservation components), access to a distributed execution infrastructure (InfraGRID) composed of mixed CPU and GPU machines and to a parallel processing environment (IBM BluGene/P) and provisioning of virtual machines on a IaaS cloud computing environment powered by Eucalyptus (AWS compatible). In order to orchestrate the installation of SCAPE Platform on a variety of hardware and software-enabled infrastructures, UVT team has been developing the SCAPE Cloud Deployment Toolkit that orchestrates the installation of SCAPE components on cloud environments.

The installations on the two data centers demonstrate the versatility of the SCAPE Preservation Platform, applying preservation ideas and methods in new contexts: medical data processing and semantic data analysis and interlinking. Specific tools have been developed or customised to support WCPT and BUT scenarios, such as HDFS PACS or off-screen CUDA rendering allowing execution of OpenGL applications on headless GPU systems.

Contrasting to Cloud deployment presented in D4.1 [1] by AIT, who is using pre-installed VM images to configure VM instances in cloud environment, UVT is relying on Puppet configuration management system to orchestrate the installation of software packages and to ensure integrity between the multiple components of the SCAPE Platform.

References

[1] SCAPE Deliverable D4.1 Architecture Design. Available at: http://www.scape-project.eu/deliverable/d4-1-architecture-design-first-version

[2] SCAPE Deliverable D2.3 Technical Architecture Report

[3] SCAPE Deliverable D4.2 Final Platform Release

[4] SCAPE Deliverable D18.1 Final Evaluation Report