StratusLab

Enhancing Grid Infrastructures with
Virtualization and Cloud Technologies

Infrastructure Operations Final Report

Deliverable D5.5 (V1.1)
5 June 2012

Abstract

This document serves as a final report of the activities and achievements of WP5 throughout the whole duration of the project. The document covers the areas of infrastructure operation, service provisioning, support, testing and benchmarking. In addition, the document provides a record of the practical knowledge accumulated during the provision of various public cloud services over a period of almost two years.

CAPACITIES

StratusLab is co-funded by the
European Community's Seventh
Framework Programme (Capacities)
Grant Agreement INFSO-RI-261552.

The information contained in this document represents the views of the copyright holders as of the date such views are published.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED BY THE COPYRIGHT HOLDERS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE MEMBERS OF THE STRATUSLAB COLLABORATION, INCLUDING THE COPYRIGHT HOLDERS, OR THE EUROPEAN COMMISSION BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THE INFORMATION CONTAINED IN THIS DOCUMENT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Contributors

Name	Partner	Sections
Evangelos Floros	GRNET	All
Stuart Kenny	TCD	Physical Infrastructure [3] and Service provisioning [4] sections
Mohammed Airaj	LAL	Benchmarking [5]
Guillaume Philippon	LAL	Physical Infrastructure [3], LAL resources
Gabriel Tézier	LAL	GPFS experiences [8]
Ruben Montero	UCM	Internal Review
Charles Loomis	LAL	Proofreading

Document History

Version	Date	Comment
0.1	30 April 2012	Initial draft ToC.
0.2	9 May 2012	Updated ToC after first review. Further breakdown of chapters into sections
0.3	17 May 2012	First complete draft. Missing introductory and conclusion chapters, figures and tables
0.9	22 May 2012	Release candidate version ready for review
1.0	29 May 2012	Applied review comments
1.1	5 June 2012	Final version

Contents

List of Figures 6

List of Tables 7

1 Executive Summary 8

2 Introduction 10

3 Physical Infrastructure 11

3.1 Overview 11

3.2 GRNET site 11

3.3 CNRS/LAL site 13

3.4 TCD site 13

4 Service Provisioning 14

4.1 Overview 14

4.2 Cloud services 14

4.3 Appliance Marketplace 15

4.4 Virtualized Grid services 15

5 Testing and Benchmarking 19

5.1 Overview 19

5.2 Testing 19

5.2.1 Testing and certification infrastructure 19

5.2.2 Certification procedure 20

5.3 Benchmarking 20

5.3.1 Benchmarking suite and tools 21

5.3.2 Sample Benchmarks 22

6 Support Activities 25
6.1 Overview . . . 25
6.2 Support for Public Cloud Service Users . . . 25
6.2.1 Enrollment procedure . . . 25
6.2.2 Profile of cloud users . . . 27
6.2.3 Common problems and issues . . . 27
6.3 Internal support to project activities . . . 28

7 Economic Impact 29
7.1 Overview . . . 29
7.2 Total Cost of Ownership . . . 29
7.3 Comparison with Amazon EC2 . . . 31
7.4 Feedback and Analysis . . . 31

8 Lessons Learned and Good Practices 33
8.1 Overview . . . 33
8.2 Physical infrastructure design . . . 33
8.2.1 Compute nodes . . . 33
8.2.2 Storage infrastructure . . . 34
8.2.3 File systems . . . 35
8.3 Cloud service operations . . . 40
8.4 Software integration . . . 41
8.5 Security . . . 42

9 Conclusions 44

References 47

List of Figures

3.1 GRNET datacenter hosting StratusLab's infrastructure 12

4.1 Number of jobs per month 17

4.2 Site normalized CPU utilization per month 17

4.3 Availability/Reliablity metrics per month for HG-07-StratusLab . . 18

6.1 Cloud services user registration form 26

7.1 StratusLab cloud service TCO cost breakdown 30

8.1 Single VM/storage management frontend 36

8.2 Separated VM and storage server frontends 36

8.3 Centralized large capacity storage server 37

8.4 Shared storage accros all infrastructure nodes 37

List of Tables

3.1 Node allocation in GRNET 12

7.1 TCO breakdown in Euros 30

1 Executive Summary

Infrastructure operations have been an integral part of StratusLab. Starting from the very first month of the project, the operations activity (WP5) followed the evolution of StratusLab distribution, provided the necessary resources for software development, testing and certification, and acted as cloud service provider to both external and internal users and applications. Through this process, StratusLab operations contributed significantly to the overall success of the project and, despite the fact that the project was not aiming to provide infrastructure services, it managed to gather valuable experience and relevant know-how for what concerns the provision of large-scale public IaaS cloud services.

From the point of view of physical infrastructure, two large sites were involved with resource provisioning offering compute, storage and network resources required by the various activities of the project. The sites are operated in GRNET and LAL. One additional site in TCD is contributing with smaller resources in order to provide the Appliance Marketplace public instance and a project-wide appliance repository for image storage and distribution.

Overall, the deployment and maintenance of public cloud services has been one of the primary tasks of StratusLab's operations activity. These cloud services attracted a considerable amount of users around the globe and was exploited by most major European DCI projects like EGI, EMI, IGE and EDGI. In parallel WP5 provided end-user support both of internal and external users.

Another important aspect has been the deployment and hosting of a production virtualized grid computing cluster on top of the public cloud. The project operated an EGI certified grid site for more than a year, validating the ability to offer production level end-user service over cloud infrastructures.

WP5 played a significant role in the software testing and integration activities. This was necessary in order to confirm that the software released from WP4 was stable and also to evaluate its performance characteristics. The work package contributed also with the necessary resources required for the various testing, certification, development and pre-production activities.

One of the major premises of Cloud computing is the reduction of expenses stemming from the consolidation of computing resources and the benefits of economy of scale. In order to investigate whether this benefits of cloud computing are valid WP5 performed an economic analysis of its internal cloud operations which was published as project Deliverable D5.4. This study verified that indeed economy of scale and infrastructure consolidation may deliver economic benefits to cloud adopters even for small-to-medium scale cloud deployments over a relatively short period of time.

WP5 is completing its activities leaving as a legacy two public cloud services, running in GRNET and LAL respectively, an Appliance Marketplace instance, operated by TCD, and a record of its experiences and good practices from almost two years of infrastructure operations. These experiences cover the areas of cloud infrastructure design and deployment, cloud service operations, software integration and security.

According to the sustainability plans defined by the consortium, these services will outlive the lifetime of the project. Withing this initiative our aim is to expand the existing infrastructure management knowledge base and keep it up to date with the technology evolution.

2 Introduction

Infrastructure operations have been an integral part of StratusLab. Starting from the very first month of the project, the operations activity (WP5) followed the evolution of StratusLab distribution, provided the necessary resources for software development, testing and certification, and acted as cloud service provider to both external and internal users and applications. Through this process, StratusLab operations contributed significantly to the overall success of the project and, despite the fact that the project was not aiming to provide infrastructure services, it managed to gather valuable experience and relevant know-how for what concerns the provision of large-scale public IaaS cloud services.

This document serves as a final report of the activities and achievements of WP5 throughout the whole duration of the project. More importantly the document provides a record of the practical knowledge accumulated during the provision of various public cloud services over a period of almost two years. This knowledge is probably one of the most important non-tangible assets that WP5 delivers as a result with the conclusion of StratusLab, believing that it will provide useful guidelines to other relevant initiatives and to potential cloud computing providers.

The document is structured as follows:

Section 3 presents the status of physical infrastructure allocated by the various providing partners by the end of the project.

Section 4 focuses on the cloud service provisioning as well as the results from running a production grid site over the project's public cloud service.

Section 5 reports on the testing and benchmarking activities that took place within WP5.

Section 6 briefly describes the various support services provided by the operations team.

Section 7 offers a condensed version of the Economic Impact study that was presented in detail in D5.4 [8] offering an updated outlook of the various implications of this particularly interesting subject.

Section 8 provides a detailed analysis of lessons learned and suggested good practices gathered from the operation of cloud services and the relevant activities of WP5.

3 Physical Infrastructure

3.1 Overview

This section provides an overview of the status of total physical resources committed by various partners at the end of the project. In general, two large sites were involved with resource provisioning offering compute, storage and network resources required by the various activities of the project. The largest percentage of these resources have been dedicated for hosting the cloud services provisioned by the project. A significant number of resources were also required for the various testing, certification, development and pre-production activities.

3.2 GRNET site

GRNET has been the primary hardware resource provider since the beginning of the project and more or less remained as such during the 24 months of the project. GRNET has allocated a total of 32 nodes, all in a single rack, located in the company's datacenter hosted in the Ministry of Education in Athens (YPEPTH).

The nodes are dual quad-core Fujitsu Primergy RX200S5, configured with Intel Xeon E5520 running at 2.26 GHz and 48 GB of main memory. The CPUs support Hyper-Threading capabilities (HT) allowing two threads running parallel per core. This feature has been enabled thus a total of 16 logical cores per node are available to the operating system. Each node is equipped with 2×146 GB SAS hard disks configured in RAID 1 (mirroring) thus the total storage space available for the system and applications is 146 GB. Apart from the local storage additional storage is provided from a central storage server installed in the datacenter. The server is an EMC Celerra NS-480 offering a total of 200 TB over NFSv3 and iSCSI interfaces. For StratusLab we have allocated a total of 20 TB shared into multiple volumes using the NFS interface.

The allocation of physical resources to specific tasks and services varied significantly during the course of the project. Similarly the physical configuration was adjusted in some cases as part of our efforts to optimize the hosted services performance. For example when the first version of StratusLab Persistent Disk Service (pDisk) was introduced that utilized iSCSI and LVM for volume management, two additional hard disks (1 TB each) where installed in the cloud front-end node in order to suffice the demand for fast access to the iSCSI service (tgtd) overcoming

# of nodes	Usage
16	Public cloud service running StratusLab distribution
2	Pre-production infrastructure
2	Certification infrastructure
4	WP4 integration (used by hudson continues integration system)
2	WP5 development/experimentation nodes
2	WP6 development
1	DHCP and other infrastructure-wide services.
3	Auxiliary cloud service used during upgrades e.g. for temporary hosting of critical VM instances.

Table 3.1: *Node allocation in GRNET*

the network overhead of NFS.

Table 3.1 shows the allocation of nodes and their usage in the GRNET infrastructure. Figure 3.1 depicts GRNET's datacenter where these nodes are located.

Figure 3.1: *GRNET datacenter hosting StratusLab's infrastructure*

The nodes are interconnected with 4×1-Gbit ethernet adaptors each. One additional ethernet network is used for monitoring and management. The storage server provides 2×4 Gbit Fiber Channel (FC) interfaces used for connection with 4 data movers. Each data mover is used to serve a single rack in the datacenter. The datacenter provides 2 redundant 10 Gbit links to the Geánt pan-european academic network.

3.3 CNRS/LAL site

LAL is the second cloud infrastructure available for StratusLab project. LAL has allocated a total of 9 nodes for computing, all in a single rack, located in LAL's datacenter hosted in the Paris 11 University.

The node are dual hexa-core Dell C6100, configured with Intel Xeon X5650 running at 2.67 GHz and 36 GB of main memory. The CPUs support Hyper-Threading capabilities (HT) allowing two threads running parallel per core. This feature has been enabled thus a total of 24 logical core per node are available to the operating system. Each node is equipped with 1×300 GB SAS hard disk. All nodes are connected by 1 Gb per second link to a switch connected by 10 Gb per second link to our core network.

Apart from the local storage additional storage is provided from a central storage server installed in the datacenter. The server is a dedicated blade on a HP c7000 blade center connected to a HP MDS600 disk array. A pool of 10 disks configured in RAID6 are allocated for StratusLab for a total of 20 TB, 5 TB are used for NFS shared filesystem, the rest of storage is allocated to StratusLab pDisk service using LVM and iSCSI to provide storage for Virtual Machine. The storage server is connected by 10 Gb per second link directly on our core network.

3.4 TCD site

Two additional central services were provided by TCD, the appliance repository, and the appliance Marketplace. For the first year of the project an initial prototype of a virtual appliance repository was deployed as a WebDAV-enabled Apache webserver, with a mirrored backup server at GRNET. The goal of the prototype repository was to quickly and simply provide centralised storage that could be used by the project to share images. Write-access was available to project members only, although the images were publicly accessible. The appliance repository was maintained as a service for the duration of the project, with 200 GB of storage allocated.

For the second year of the project the repository evolved to become an appliance *metadata* marketplace. The Marketplace provides a central image metadata registry. This was accessible to both project members and StratusLab users.

Both services are hosted within VMs and occupy minimum computing resources.

TCD also provides a StratusLab cloud for local users. The deployment consists of a front-end machine with 5 VM hosts. Each VM host has a dual quad-core CPU with 16 GB of RAM. Storage is managed by the StratusLab Persistent Disk service. The Port Address Translation solution developed within the project by IBCP is used to provide access to VM's started in the private IP address range. The resources are in use by local TCD users and are also part of the EGI Federated Clouds Task Force testbed.

4 Service Provisioning

4.1 Overview

The deployment and maintenance of public cloud services has been one of the primary tasks of StratusLab's operations activity. Another important aspect has been the deployment and hosting of a production virtualized Grid cluster on top of the public cloud. This section provides details regarding the provisioning of the above services, their impact within the project and their exploitation by third parties.

4.2 Cloud services

A large part of WP5 activities has been dedicated to the deployment, maintenance and provision of public cloud IaaS services. These IaaS clouds served primarily as a demonstration and validation testbed, and provided a reference cloud instance based on StratusLab distribution, that gave the opportunity to external users to evaluate the features of StratusLab. Internally the services were used both for production and testing purposes. In particular, the cloud services were utilized successfully for deploying and providing the production grid site of the project. It also served as a production platform for porting various Bioinformatics applications from IBCP.

The primary cloud service for most of the project lifetime, has been the reference cloud provided by GRNET. The service is currently hosted in 16 nodes of GRNET's infrastructure providing a total of 256 cores and 768 GB of memory to VMs.

During the second year of the project a secondary site was deployed in LAL initially to serve as local production cloud. During the last months the site was opened for public access. By the end of the project the two sites use the same centralized LDAP server for user authentication. This allows both internal and external users to choose freely which to sites they would like to use for their VMs.

Both sites offer similar set of cloud services, namely VM management and persistent storage management. The VM images instantiated in the sites are typically registered in the project's Marketplace and physically located in external network-accessible repositories.

The total number of accounts created for the sites is 70. Ten accounts respond to internal users and the remaining 60 have been created for external users. The cloud services have been very popular among DCI projects and has been used for proof of concepts and testing in projects like EGI, EMI, IGE and EDGI. Furthermore, the service attracted people from different domains and continents that got to hear about StratusLab and got interested in the project.

4.3 Appliance Marketplace

The Marketplace is at the center of the image handling mechanisms in the StratusLab cloud distribution. It contains metadata about images and serves as a registry for shared images. The actual image contents are kept in cloud, grid, or web storage, external to the Marketplace.

The metadata stored in the Marketplace contains a pointer to the image contents. The Marketplace is interacted with through a RESTful interface. The metadata format is based on a standard format, and can be extended by users as required. Detailed information about the Marketplace including specification about the metadata format are available from the "StratusLab Marketplace Technical Note" [7].

At time of writing the metrics related to the Marketplace were as below:

No. of Marketplace metadata entries	140
No. of Marketplace endorsers	34

The Marketplace has received interest from external communities. EGI has deployed its own instance of the service[1]. This is currently being evaluated by the EGI Federated Clouds Task Force [3].

4.4 Virtualized Grid services

The applicability of cloud infrastructures to host production level grid services has been one of the primary motivations of StratusLab. For this reason once a stable cloud service became available, deploying a grid site on top of it became one of the first priorities. Over more than a year WP5 has operated a production grid site (HG-07-StratusLab) certified by the Greek NGI (National Grid Initiative).

The site is deployed on a total of 15 VMs with the following distribution: 1 CE, 1 SE, 12 WNs and 1 APEL node. This is a basic setup and reflects the typical services that a grid site has to provide in order to be certified. All the VMs are configured with 2 CPU cores, and 4 GB of memory. Thus the total cores available to Grid jobs is 32. The total storage provided by the SE is 3 GB.

During this period it was validated that higher-level services (like grid services) can be hosted with no real implications in complete virtualized environments, taking advantage of the flexibility offered by the latter. However, the operations team faced two really important problems during the operation of the Grid site.

Failures of the LDAP service running inside the CE This caused erroneous reporting of the site's state to the centralized monitoring and accounting ser-

[1] http://marketplace.egi.eu

vice of EGI, and in return impacted for a period of a few weeks the site's availability and reliability metrics. The issue appeared to be solved by itself after the service was upgraded (following the normal upgrade cycle of grid sites) and although the actual causes were not confirmed it was most probably either a bug in the software itself or a misconfiguration of the system that was fixed during the upgrades.

Degraded performance with iSCSI/LVM based pDisk service With the introduction of the first version of StratusLab's persistent disk service (pDisk), which was depending exclusively on iSCSI and LVM in order to create and share volumes among VMs, the reference service could not take full advantage of its capabilities due to limited ability to exploit iSCSI in its current setup. An attempt to provide the service using storage from the centralized server with NFS, proved to be very problematic due to the network overhead and the delays introduced by this configuration. This impacted seriously the VMs running the various grid services, which would freeze for a few minutes due to the heavy load in the underlying cloud infrastructure. To overcome this problem an auxiliary cloud was setup using the traditional NFS-based storage management approach until pDisk could evolve in order to support other back-end storages also. It should be noted that this problem had limited impact to the availability of the grid site, since it was trivial to move the VMs to the back up infrastructure within a few hours and bring the site back online.

The grid site supported numerous Virtual Organizations (VOs) from various scientific domains. The site was heavily used during this period of operations with thousands of jobs submitted for execution. Figures 4.1, 4.2 and 4.3 depict the site utilization and the availability/reliability metrics from the installation date of the site (February 2011) till the end of the project. The total number of grid users that submitted at least one job to the site is 114.

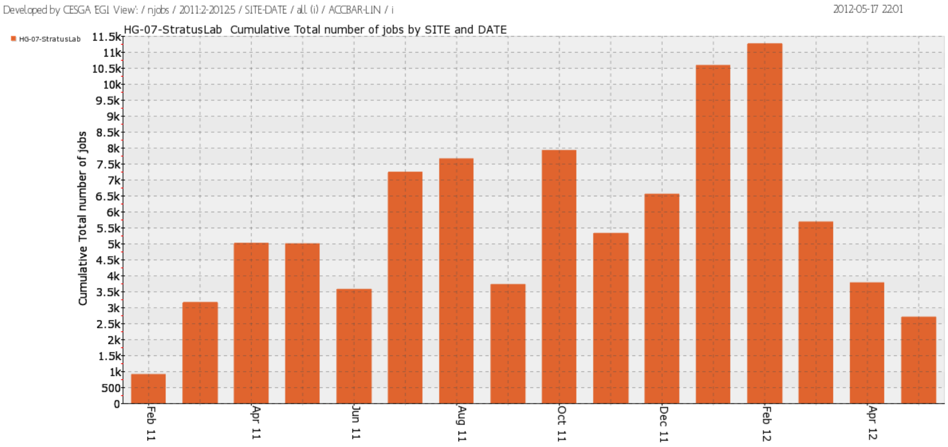


Figure 4.1: *Number of jobs per month*

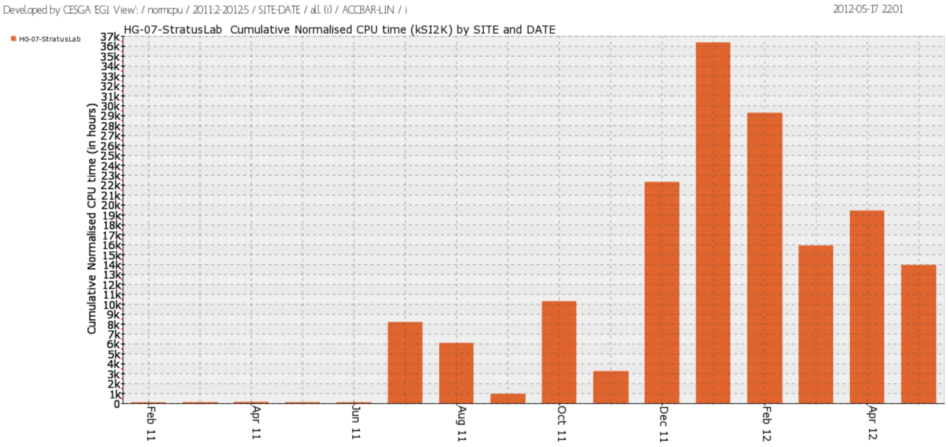


Figure 4.2: *Site normalized CPU utilization per month*

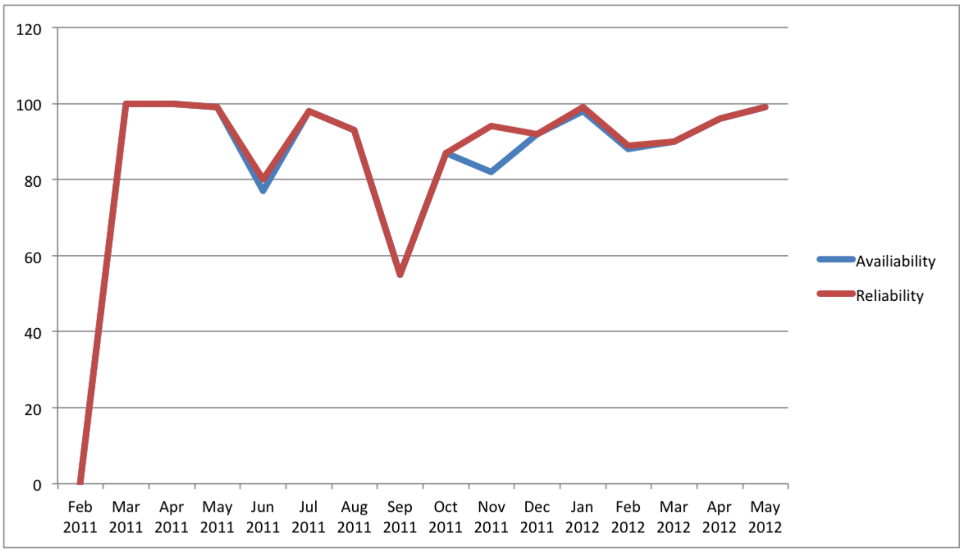


Figure 4.3: *Availability/Reliablity metrics per month for HG-07-StratusLab*

5 Testing and Benchmarking

5.1 Overview

WP5 played a significant role in the software testing and integration activities. This was necessary in order to confirm that the software released from WP4 was stable and also to evaluate its performance characteristics. In this section we focus on the two interrelated activities of Testing and Benchmarking.

5.2 Testing

Testing is an important part of software pre-release activities. It validates the stability of the software, it verifies the software installation procedure and validates the end-user documentation. Overall, it provides important early feedback to software developers and at the end facilitates the delivery of quality software to end users. WP5 collaborated closely with WP4 in order to establish an automated comprehensive testing and certification procedure that enabled the unattended verification of StratusLab distribution releases. This process was supported by a dedicated testing infrastructure, a set of integration tools and a clearly defined certification procedure.

5.2.1 Testing and certification infrastructure

Software testing requires a small number of dedicated physical machines that play different roles in the various stages of testing. Cloud software is inherently complicated. In order to support the testing procedures physical machines should be completely dedicated for this purpose since in many cases the OS has to be re-installed and configured from scratch before deploying the cloud software stack to be tested.

In StratusLab we allocated a number of dedicated nodes from GRNET's infrastructure for the following purposes: one node dedicated for running the Hudson integration server; three nodes for automated hudson job deployments and test; 2 nodes for certification through hudson and two nodes for additional tests and experimentation with different configurations and new services.

5.2.2 Certification procedure

The software certification procedure was introduced during the second year of the project in order to streamline the interaction between the development and operations team and to accelerate the “time-to-market” for the software developed by the former. The main tool for implementing the certification procedure is the Hudson integration system and an intermediate certification code repository (hosted in git). Once a release is ready to pass from certification, the development team commits it in the certification repository. Afterwards, two hudson jobs are run; the first one attempts to make a clean installation, verify that all the installation steps are completed properly and that the installed services are working as they should by instantiating a number of Virtual Machines. The second job performs a similar task but on an existing system thus testing the ability to seamlessly upgrade the previous version of the software to the latest one.

Before deploying the new software release to the production sites the operations team typically performs one more manual test installation of the packages from the certification repository on the pre-production nodes. This last step will usually reveal a few minor issues since it gives the opportunity to perform a few ad-hoc tests or to stress test the installation. When both certification jobs pass successfully and the pre-production installation reveals no problems, the generation of the final release packages are triggered manually and the release package repository is populated.

5.3 Benchmarking

StratusLab developed application-oriented benchmarks for evaluating the performance of the cloud implementation. These Benchmarks have been created to test the efficiency of cloud services, allowing quantitative measurements on the performance of different types of realistic applications. They could be run on different cloud configurations.

The StratusLab Benchmarks cover a range of common scientific application patterns: CPU-Intensive, Simulation, Analysis, Filtering, Shared Memory, Parallel, and Workflow.

- CPU-intensive: High-CPU requirements but little or no input and output data.

- Simulation: Small input, but significant output data with high-CPU requirements.

- Analysis: Large input data but relatively small output data.

- Filtering: Large input and output data.

- Shared Memory: OpenMP-based programs

- Parallel: MPI-like programs

- Workflow: Multiple interdependent tasks.

All these benchmarks are configured and installed on a specific appliance, a Centos 5.5 appliance referenced in the StratusLab Marketplace[1]. In order to test StratusLab Benchmarks, stratus-run-benchmark client command was used to deploy virtual machines referencing this appliance, with a requested amount of CPU, RAM, and SWAP, and run automatically the specified test benchmarks on these virtual machines. All the tests presented below were run in the StratusLab Reference Cloud at GRNET. For each test/algorithm we measure the Elapsed time and CPU time.

StratusLab benchmarks are also all tested via the continuous integration hudson-server, and could be deployed on LAL StratusLab Cloud, other production StratusLab Cloud.

5.3.1 Benchmarking suite and tools

StratusLab project developed the stratus-run-benchmark client command that tests a complete range of common scientific application patterns: CPU-Intensive, Simulation, Analysis, Filtering, Shared Memory, Parallel, and Workflow. It covers a large set of applications: sequential, multi-threaded, OpenMP, MPI, i/o parallel, kepler/workflow, master/worker, cpu stress. The application benchmarks are, by default, parameterized to allow a wide range of input sizes, output sizes, and running times to be evaluated. The stratus-run-benchmark command allows the selection of one specific type of application to test and also permits a complete test of all the above types of applications. The performance of these tests are measured and stored in XML format.

For each type of application we are using a specific option:

–openmp OpenMP benchmark

–io IO/Parallel, Large input data but relatively small output data, Large input data but relatively small output data and mall input, but significant output data with high-CPU requirements benchmark.

–mpi MPI benchmark

–cpu-intensive High-CPU requirements but little or no input and output data

–workflows Multiple interdependent tasks, Kepler Workflow benchmark

–all Run all benchmarks

–output-folder=OUTPUT_FOLDER folder for xml output files

The stratus-run-benchmark command is an extension of stratus-run-instance command, thus the standard options of the later can also be used.

The algorithms implemented for each type of application:

[1] Appliance identifier: Gr8NgmZ5N6sLza_xwDv9CkHlLYa

1. OpenMP Benchmarks
    - Conjugate Gradient Method: Solves linear system $Ax = b$
    - Jacobian type iteration: Solves linear sysrem $Ax = b$
    - Matrix multiplication $C = AB$
2. MPI Benchmarks
    - MPI asynchronous non blocking communication
    - MPI synchronous blocking communication
    - MPI persistent communication
    - MPI standard communication
3. I/O Benchmarks
    - Implements the important features of Parallel I/O
    - Writing to a file: MPI_FILE_write_at
    - Reading from file: MPI_FILE_read_at
    - Both: Reading from a file and writing to another
4. CPU Benchmarks
    - Fork Stress
    - CPU Stress
5. Workflow Benchmarks We are using Kepler Platform for workflows design and management
    - The workflow is built using the customizable components available in Kepler namely: directors, actors, parameters, relations and the ports
    - It executes an OpenMP Application

For testing purposes, we are fixing the parameters (matrix sizes, tolerance, ...) for each algorithm.

These algorithms are installed by default in the benchmark appliance referenced above.

5.3.2 Sample Benchmarks

1. Shared Memory benchmarks

    ```
    $ stratus-run-benchmark --openmp \
    --output-folder=/home/user/benchmark \
    -t c1.xlarge Gr8NgmZ5N6sLza_xwDv9CkHlLYa
    ```

This command will deploy the specific benchmark appliance, run all the Shared Memory StratusLab benchmarks, and retrieve output results in the /home/user/benchmark folder. Parameter c1.xlarge implies that we requested for (CPU=4, RAM=2048MB, SWAP=2048MB).

```
$ ls /home/user/benchmark
openmp-cg.xml  openmp-jacobi.xml  openmp-matrix.xml
```

For each test we run the application benchmark, in sequential and multithreaded mode. CPU time and elapsed time measure the performance of each mode.

The example below, illustrate openmp/jacobi application evaluation in sequential and multithreaded modes. The number of iterations is defined by default to be 104. The first part with represent sequential mode, second part with nb_threads defined, represent multithreaded mode. The time unit is expressed in seconds.

```
$cat openmp-jacobi.xml
 <benchmark name='openmp_jacobi'>
     <parameters>
        <nb_iterations>104</nb_iterations>
     </parameters>
     <results>
       <elapsed_time unit='sec'>2.692E+00</elapsed_time>
       <cpu_time unit='sec'>2.662E+00</cpu_time>
     </results>
 </benchmark>
 <benchmark name='openmp_jacobi'>
     <parameters>
        <nb_threads>4</nb_threads>
        <nb_iterations>104</nb_iterations>
     </parameters>
     <results>
       <elapsed_time unit='sec'>1.798E+00</elapsed_time>
       <cpu_time unit='sec'>7.181E+00</cpu_time>
     </results>
 </benchmark>
```

2. I/O Parallel and CPU intensive benchmarks

```
$ stratus-run-benchmark --io  --cpu-intensive  \
--output-folder=/home/user/benchmark \
-t c1.xlarge Gr8NgmZ5N6sLza_xwDv9CkHlLYa
```

This command will deploy the specific benchmark appliance, run all the I/O Parallel, CPU intensive StratusLab benchmarks and retrieve output results in the /home/user/benchmark folder.

```
$ ls /home/user/benchmark
io_mpi_io.xml  io_mpi_i.xml  io_mpi_o.xml
```

The example below, illustrates a MPI-I/O Parallel application: nb_thread parameter equal to CPU number of the image. The application reads in parallel a file of 10 MB and writes a file of the same size. The time unit is expressed in seconds.

```
 <benchmark name=’mpi_io’>
     <parameters>
        <nb_threads>4</nb_threads>
     </parameters>
     <inputfile>
        <name>10</name>
        <size unit=Mbytes>10</size>
     </inputfile>
     <outputfile>
        <name>10</name>
        <size unit=Mbytes>10</size>
     </outputfile>
     <results>
       <elapsed_time unit=’sec’>0.025</elapsed_time>
       <cpu_time unit=’sec’>0.025</cpu_time>
     </results>
 </benchmark>
```

CPU intensive benchmarks, run a fork stress test, maximally loading all the four virtual machine’s CPUs. The duration of this test, by default is 600 seconds.

3. Running all the benchmarks

```
$ stratus-run-benchmark --all \
--output-folder=/home/user/benchmark \
-t c1.xlarge Gr8NgmZ5N6sLza_xwDv9CkHlLYa
```

This command will run all the StratusLab benchmarks, and retrieve their xml outputs in the /home/user/benchmark folder. Kepler workflow models an openmp-matrix application with Kepler SDF Director and input/output actors.

6 Support Activities

6.1 Overview

Part of the operations activity has been the delivery of support to users regarding the infrastructure services both internally within the project and externally. In this section we briefly report on the types of support offered by WP5 thought the duration of the project.

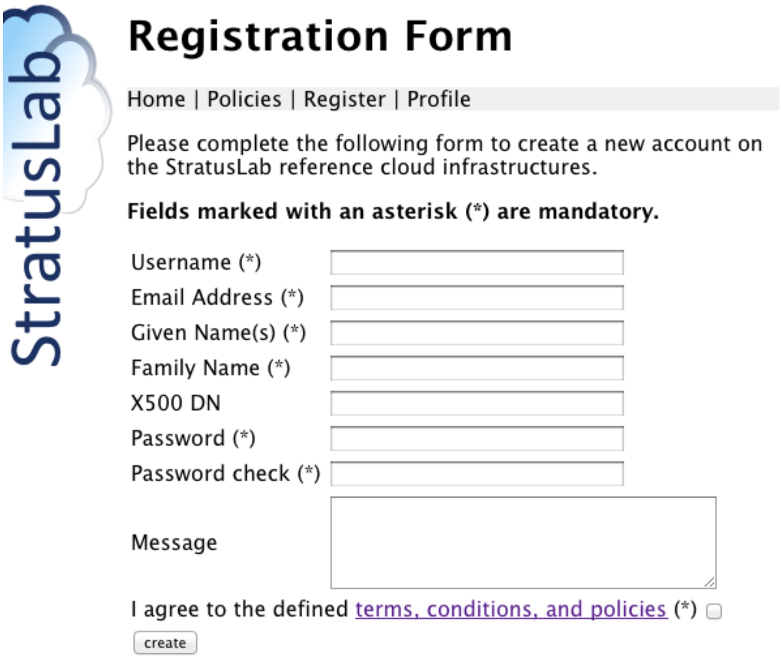
6.2 Support for Public Cloud Service Users

Despite the fact that StratusLab is not an infrastructure-provider project it was set from the beginning as goal to offer public cloud services with the highest possible quality of service and level of support, still providing these services on "best effort" terms.

6.2.1 Enrollment procedure

During the first months we followed an off-line enrollment procedure according to which the interested user would send an email to StratusLab mailing list, requesting access to the reference infrastructure, stating shortly the intented purpose of use and any relevant projects involved. The username/password tuples were generated by WP5 operations and send by email to the user unless certificate-bases access was requested in which case only the addition of the users DN in the cloud frontend ACL was sufficient.

During the last quarter of the project and with the deployment of the second reference cloud service in LAL the procedure was automated with the introduction of a centralized on-line registration form (Figure 6.1). Once the form is filled a notification mail is send to a member of WP5 in order to evaluate and accept the request. By using the registration page a user gets access to both reference cloud services in GRNET and in LAL using the same credentials. This is achieved by using the same LDAP server for user authentication in both sites. The LDAP server is hosted in LAL's site and the GRNET frontend has been configured to use it for user authentication. Both sites have retained their flexibility to add users locally in order to provide them access only to local resources.



Registration Form

Home | Policies | Register | Profile

Please complete the following form to create a new account on the StratusLab reference cloud infrastructures.

Fields marked with an asterisk (*) are mandatory.

Username (*)	
Email Address (*)	
Given Name(s) (*)	
Family Name (*)	
X500 DN	
Password (*)	
Password check (*)	
Message	

I agree to the defined terms, conditions, and policies (*) □

Figure 6.1: *Cloud services user registration form*

6.2.2 Profile of cloud users

By the end of PM23 a total of 71 accounts have been created in the reference cloud service and are shared between the two sites in LAL and GRNET. Among them 12 accounts belong to internal users and the rest 59 to users not directly related with the project. These external users originate from different countries and continents and represent a diverse set of communities and scientific disciplines. A large fraction of users are located in France and are active in the Life Sciences domain. This is of course due to the fact that the project had strong presence in the country with the Bioinformatics partner from IBCP promoting strongly the project locally. Other countries appearing in the list are Switzerland, Greece, Germany, UK, Ireland, Poland, Romania, Italy, Norway and Spain. Outside Europe the service attracted users from Australia, Vietnam, South Africa, United States, Tunisia, India and Singapore.

For what concerns the scientific domain, apart from Life Sciences there was also strong interest from users from High Energy Physics, Earth Sciences, Computer Science and Communications. Notably, it was not only the academia that got interested into StratusLab, but also a significant fraction of users (20%) represented the industry or public bodies (e.g. NRENs, National Observatories etc.).

As expected, StratusLab software and cloud services attracted the interest of many DCI projects currently running. Thus many user accounts where allocated for people working in EMI, EGI, IGE, PRACE and EDGI. Most of this people used the service for general evaluation but also for running a few very focused use case scenarios in order to assess the usage of the software for production purposes. As a result these users created many production level appliances that later re-used in other environment or have brought their existing appliances and used them within StratusLab. Also the Appliance Marketplace instance provided by TCD was a good example of the applicability of the marketplace concept and motivated other projects to adopt the technology for their own purposes.

6.2.3 Common problems and issues

User support was provided with two means of communication: the project web page and the respective support mailing list (support@stratuslab.eu). The latter in particular was the main channel of communication for problems, issue reporting and feature requests. The most popular requests we received were:

- Problems
    - Problem with the infrastructure: VMs failing to start
    - Infrastructure not available at all
    - Custom appliances contextualization problems, typically leading to connection failures
    - Degraded network bandwidth in VMs (resolved with the support for virtio_net).

- General requests
  - Request for new cloud account
  - Request access to appliance repository in order to store own appliances
  - Instructions for preparing own appliances
  - Support for virtio_net and virtio_blk drivers
- Feature requests
  - Request to support private IPs and custom firewall rules
  - Provision of a web frontend
  - VM Snapshotting
  - Improve the description of error messages
  - Ability to remove entries from Marketplace
  - Request to support additional base OSs for StratusLab distribution
  - VM automatic shutdown after deadline

6.3 Internal support to project activities

WP5 has provided and supported the infrastructure used for various purposes inside the project. In particular WP4 and WP6 were offered with the necessary hardware for their development and testing purposes. Roughly 25% of GRNET's physical infrastructure was dedicated for the above purposes. WP5 collaborated closely with these WP's in order to setup and maintain the basic OS's and in some cases deployed a complete StratusLab site to serve as a development platform for high-level service, in particular those developed by WP6.

Additionally, WP5 contributed with physical resources to the various dissemination activities, like demonstrations and tutorials. For the latter a number of short lived accounts were usually created in the reference cloud service whenever a new event had to be delivered. These tutorial/demo accounts had restricted access policies and limited resource utilization capabilities, and typically where deactivated after the delivery of the event.

7 Economic Impact

7.1 Overview

One of the major premises of Cloud computing is the reduction of expenses stemming from the consolidation of computing resources and the benefits of economy of scale. In order to investigate whether this benefits of cloud computing are valid WP5 performed an economic analysis of its internal cloud operations which was published as project deliverable D5.4. In this chapter we provide a quick overview of the results report in this document and some updated assessments based on the interactions we had from the presentation of this results in various conferences and scientific events.

7.2 Total Cost of Ownership

A first question we tried to answer was how much it actually costs to operate the reference cloud services. In order to calculate this we took into account the cost for the hardware, the cost of the datacenter, the power consumption, the manpower for datacenter operations, the manpower for the cloud site maintenance and the costs for network connectivity. Although not all of the datacenter resources were dedicated for StratusLab we isolated a reasonable fraction of resources that we identified being used for the provision of StratusLab services. In the total infrastructure cost apart from the hardware required for running the service we took also into account the cost for hosting the necessary support sites like the pre-production and testing sites that we consider as a necessary part for the provisioning of production level services. The detailed cost breakdown is listed in Table 7.1 and depicted in Figure 7.1.

The only cost we did not include in our calculations is for the software itself the rationale being that the software in any case is open source following a free-license scheme, thus no loyalties need to be paid for its usage. Obviously, in this case the software user will have to invest some effort in order to adapt the software exactly to the specific environment, which in some cases may need some moderate development activities. In our case we make the assumption that such expenses are already calculated in the administration manpower but also in the hardware costs for the additional services (e.g. pre-production sites).

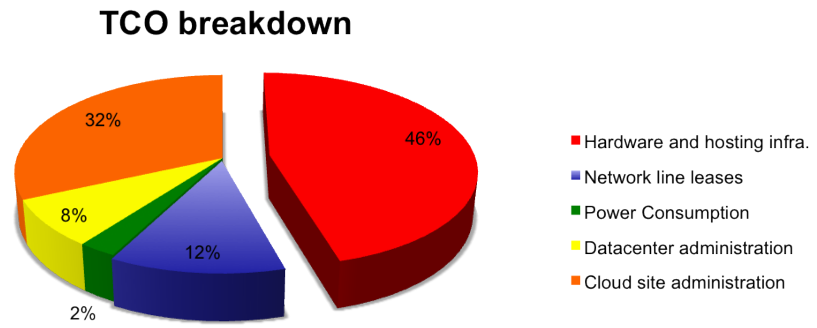


Figure 7.1: *StratusLab cloud service TCO cost breakdown*

Type of cost	Euros
Hardware and hosting infrastructure	115,200
Network line leases	30,000
Power Consumption	6,600
Data Center administration	20,000
Cloud Site administration	80,000
TCO	251,800

Table 7.1: *TCO breakdown in Euros*

7.3 Comparison with Amazon EC2

We compared the cost for provisioning our cloud service with the most popular IaaS cloud offering, namely Amazon EC2. Based on the average profile of VM instantiated in our cloud, we identified the matching Amazon EC2 instance and compared the respective cost. Based on the accounting records collected during the first year of the cloud site operations the average resources consumed by VMs were 1 core and 600 MB of main memory per instance. This matches roughly with either t1.micro and t1.small EC2 instances. During the period of operation the prices of these instances where 0.019 euro/hr and 0.07 euro/hr respectively. The average price per VM calculated for StratusLab is 0.0712 euro/hr which is competitive to t1.small. Obviously t1.micro is much cheaper but offers very limited I/O capabilities and is not suitable for all use cases that StratusLab VMs can cover.

Another comparative analysis we did was to calculate the cost for hosting the HG-07-StratusLab grid production site in the reference cloud and compare it with the theoretical cost for running it on Amazon EC2. Using again the t1.small instance prices the total cost calculated (21,888 euro) was pretty much the same with hosting the site in StratusLab (21,896 euro). Actually this comparison is very modest since in StratusLab we utilized much bigger VMs for hosting the grid cluster (4 cores and 4 GB main memory per VM) delivering very good performance with limited loses due to the virtualization layer. Thus we argue that in this case our private cloud provided better services for similar costs with Amazon and therefore the overall value more money is favoring the private cloud approach.

7.4 Feedback and Analysis

The final results of the report were very positive for what concerns the cost effectiveness of private clouds. Our figures more or less suggested that in a period of two years a medium scale investment of roughly quarter million euro would deliver economical benefits by breaking even the costs for the provision of computing resources using an open source cloud distribution. Our results were confirmed from other resources also and especially from the report of the Magellan project [9]. The latter did an economic analysis of a scientific HPC cloud and it reached to the same conclusion that economy of scale delivers results very fast for private installations. Magellan though was a much bigger project, which invested in larger HPC infrastructures comparing to StratusLab, so in their case the benefits were much more profound and evident.

The report was presented in various venues, most notably in the 2nd International Workshop of Cloud Computing platforms (CloudCP 2012) [5] and the EGI Community Forum 2012, receiving positive comments and a number of interesting questions. One issue that was noted is that in our calculations we do not take into account flexible charging policies available from Amazon like spot instances that may potentially reduce the expenses for running high-capability instances in EC2. Indeed this introduces a more complicated aspect that private clouds should take into account. Another interesting question is the benefits stemming from resource elasticity which at some point was tested also with Grid sites within StratusLab. In this scenario the total cost over a period of time may reduce significantly if there are only short peeks of workload in which a large number of VMs have to be instantiated to cover the temporary demand. During the rest of the period the site would operate with only a minimum number of VMs significantly reducing the overall cost of the service provisioning.

Since the deliverable was released Amazon has announced new reduced prices for EC2 instances which of course renders the companies offering much more attractive price-wise comparing to the numbers we used for comparison. We believe that this is the natural process of commercial services that will always try to improve their offerings first in order to challenge the competition of other commercial products but also to make their solutions more competitive comparing to open source alternatives. This will most probably be a common trend in the cloud computing market in the years to come. Moreover, this implies that economic analysis and the private versus commercial cloud question has to be revisited every time an organization wishes to make the transition to cloud infrastructures or upgrade/redesign their existing solutions.

Still we believe that private clouds based on open-source solutions like StratusLab bring with them many benefits and will remain an attractive cost-effective solution for medium to large-scale installations.

8 Lessons Learned and Good Practices

8.1 Overview

This section summarizes the experiences and good practices gathered from two years of cloud operations within WP5. This accumulated knowledged covers the areas of hardware infrastructure, service provisioning, software integration and security and provide useful guidelines for any interested organization wishing to deploy private cloud services whether these are based on StratusLab distribution or any other cloud software stack.

8.2 Physical infrastructure design

Setting up the necessary physical computing infrastructure is the first step when establishing a large scale public service like IaaS clouds. Maintaining this infrastructure is one of the major tasks that an operations team undertakes during the duration of service provisioning. Proper infrastructure architecting and installation impacts significantly on the quality of service delivered, affects the incurring administration overheads, defines the ability to expand the infrastructure in the future and may lead to cost reductions. From the point of view of functional requirements we consider two as the most important factors when designing a cloud service infrastructure: performance and scalability. The range of services should be able to expand both in terms of size and volume without performance losses, maintaining in parallel the same levels of availability, reliability and simplicity of administration.

8.2.1 Compute nodes

The total number and type of compute nodes define to a large extent the capacity and performance capabilities of the delivered services. In StratusLab's case, for the reference cloud service in GRNET, we relied on fat nodes with 16 cores, 48 GB main memory each, which proved to be very adequate for the purpose. During the course of the project it was evident that having a few more GB of local disk space, than the 140 GB available, would be useful for various deployment scenarios (like the shared file-system setup described below). The RAID-1 configuration (2 disks mirrored for high availability) proved invaluable at least in two occasions that a hard disk failure was experienced in our nodes. Generally speaking, hardware failures are inevitable and proper infrastructure design should cater for high-availability solutions and redundancy of resources, in order to achieve limited downtimes and protect from potentially disastrous data losses.

From the performance point of view, typically by choosing the latest CPU technologies it is more or less ensured that you will get the best performance and also the best support for virtualization technologies. Also scalability is not a real issue when it comes to compute resources: the more nodes you add in your infrastructure the more capacity you will get and also higher aggregated performance numbers. This is not to say that a cloud service like the one installed in GRNET or in LAL is adequate for high-performance applications, but for typical use cases and regular scientific and commercial scenarios, it will take time for the applications to reach their capacity limits.

8.2.2 Storage infrastructure

The storage infrastructure needed to back the cloud service appeared to be the most challenging aspect of infrastructure design. Storage capabilities define to a large extent the performance and scalability potentials of a given physical setup. Indeed many of the infrastructure limitations we hit during the cloud site operations were accounted for limited or inadequate storage capabilities. During the lifetime of the project we identified and experimented with four different architectural configurations for combining storage with computing capabilities and delivering them as part of a cloud service based on hardware virtualization and abstraction layer. These configurations are:

Configuration 1 Single frontend providing both VM and storage management capabilities (Figure 8.1). This is the typical setup a newcomer will deploy during the first tests with cloud computing sites for basic evaluation and proof of concept purposes. A single node acts as a VM management (running OpenNebula) and storage management node (running for example pDisk or sharing VM image files over ssh or NFS). This setup suffers from scalability and performance restrictions. The front-end is a bottleneck receiving all the traffic related to VM management actions and more importantly with the data movement requests from clients nodes. This solution cannot scale to more than a handful of hosting nodes and 20 to 30 VM instances running in parallel.

Configuration 2 Separated VM control and storage server (Figure 8.2). This setup improves slightly the performance behavior of the system but still suffers from the network congestion created on the storage server from all the traffic created by the running VMs.

Configuration 3 Centralized storage server providing storage directly to all the nodes over a fast network (typically Fibre Channel) (Figure 8.3). This setup is adequate for high-performance an high-availability configurations. On the other hand this solution is significantly more expensive and requires capital investment for the installation of the storage infrastructure and possibly for the development of necessary interfacing code to adapt the cloud software with the back-end hardware API. Also this solution does not offer unlimited scalability. In case the cloud service expands significantly attracting hundreds or thousand users (which is reasonable even for mid-scale corporate private clouds) there is a strong chance that the infrastructure will reach its limits both in terms of interconnection bandwidth but also in total available storage. In this case expanding the infrastructure will require again significant investment in terms of money and manpower.

Configuration 4 In this setup a shared/parallel file-system is used (Figure 8.4). Each node is participating with a fraction of storage to the total amount of storage available to all the infrastructure. The single VM management front-end setup remains the same but the workload for storage distribution and file management is shared now between all the participating nodes (potentially across literally ALL the nodes) of the infrastructure. This setup holds the potential to be the most cost effective, robust, scalable and optimal in terms of performance. Nevertheless, it requires a high-performance file system capable of sustaining the imposed service requirements. Current open-source solutions seem to perform either suboptimally (e.g. GlusterFS) or still in development state (e.g. Ceph). Commercial solutions like GPFS probably are able to satisfy this requirements nevertheless impose high licensing and maintenance costs. Also, in many cases these file systems are optimized to support the distribution of large number of small-to-medium sized files. In the case of cloud computing that VM images are quite large (10+ GB) this file systems may prove to be inadequate. These questions are worth investigating in the context of future initiatives and funded projects. In the context of WP5 we carried a number of medium scale tests the results of which are described in the following section.

8.2.3 File systems

As mentioned above, storage management proved to be one of the most critical parts of cloud operations. In search for optimal solutions the operations team experimented with various different distributed file system setups. Below we briefly report our findings from these experiments.

8.2.3.1 NFS

NFS [6] was the first file system of choice when moving from a centralized to shared network file system solution. It is also the default file system supported by GRNET's EMC Celerra storage server. Unfortunately the specific hardware supports only NFSv3 which is considered rather obsolete nowadays. The performance of the setup was overall average although it did manage to serve a moderate workload (up to 120 VM instances running at the same time). Among other issues, NFS underlined the problem of delayed VM instantiation times. In some cases for large

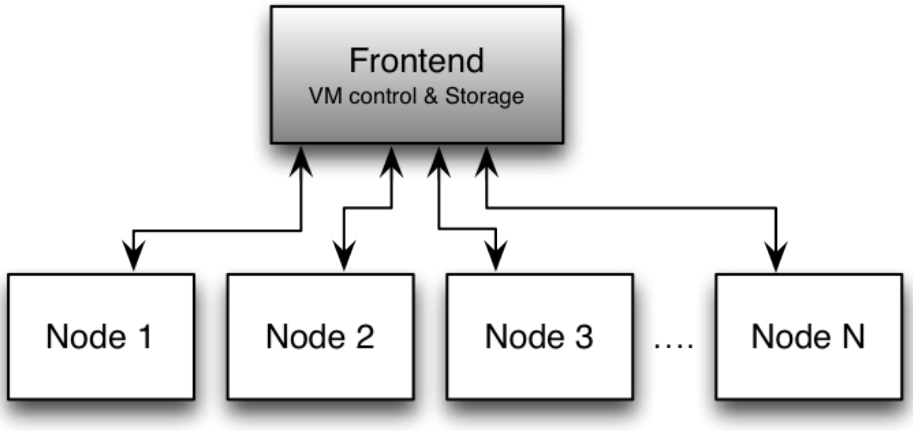


Figure 8.1: *Single VM/storage management frontend*

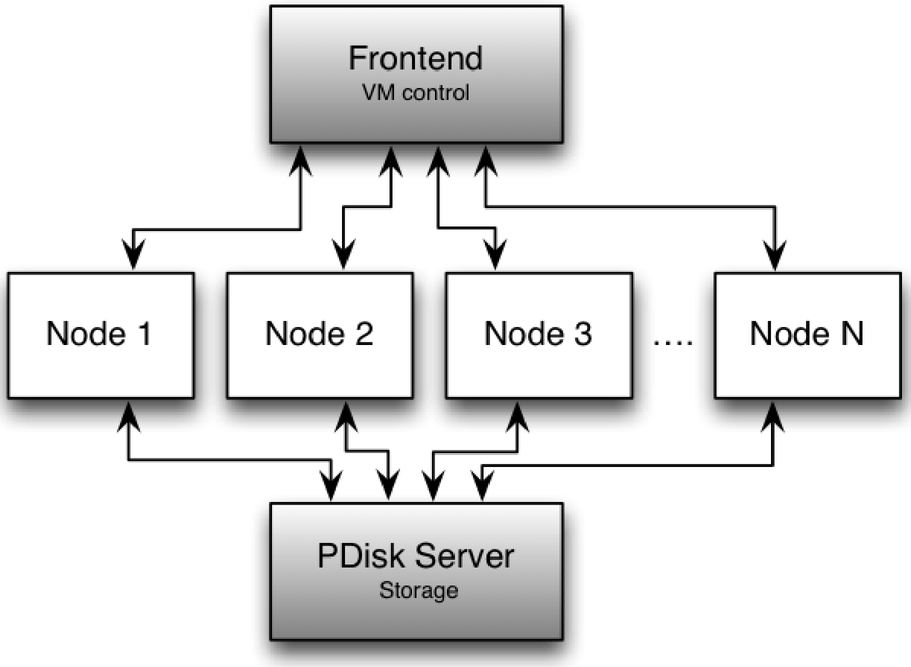


Figure 8.2: *Separated VM and storage server frontends*

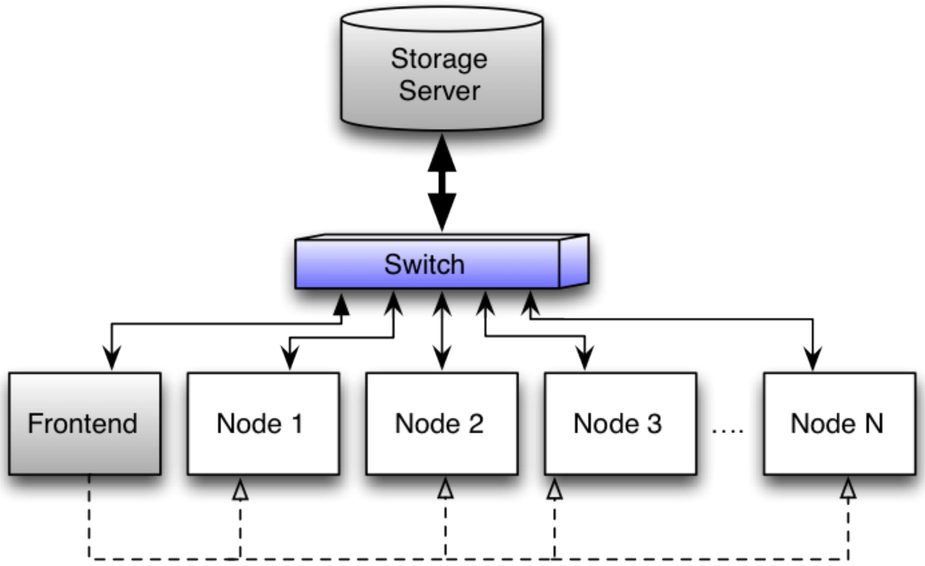


Figure 8.3: *Centralized large capacity storage server*

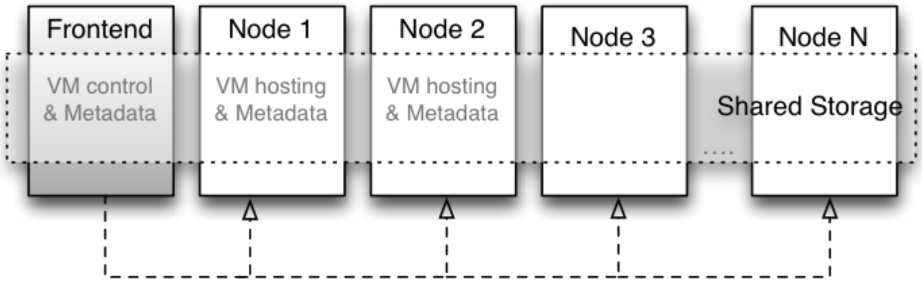


Figure 8.4: *Shared storage accros all infrastructure nodes*

VMs it would take up to 15 minutes to move from PENDING to RUNNING state. In order to improve this it became evident that some short of caching mechanism should be put into place in order to accelerate the VM startup times.

NFS on the other hand provided good support for OpenNebula's live migration capabilities which is very useful from the administration point of view when a node has to be taken offline for maintenance and all the VMs have to be moved to a different host. In general a file-based approach for VM image management proved to be very flexible for cloud site administrators allowing to perform easily tasks like VM cloning, image backup and transfer of images between different sites.

With the introduction of the first version of pDisk, the NFS based storage approach proved to be extremely slow and suboptimal. The first version of pDisk supported creation and sharing of disk volumes for VMs using LVM and iSCSI exclusively. An attempt to use LVM/iSCSI over the existing NFSv3 file system exhibited extremely low performance characteristics. In this setup the cloud front-end fall back to acting as a storage server with the EMC merely providing a large storage space to the frontend node only. This created a network and I/O bottleneck in the front-end node which could not sustain the increased workload. As a result a few images, especially those stored in qcow format, where spontaneously freezing and were becoming completely unresponsive for a few minutes. The above behavior impacted the site operations for a few months. The storage configuration had to fall back to a pure centralized setup by adding a large disk locally in the front-end which in turn to be inadequate for handling large volumes of VM instances running in parallel. These experiences led to the development of the file-based pDisk service which during the writing of this report is under certification.

8.2.3.2 Ceph

Ceph [10] is an open source distributed file system. It is one of the latest developments in the high-performance file system arena and carries many promises for delivering an efficient, fault-tolerant, robust general purpose file system. During the course of the project we made two trial installations of Ceph in our pre-production machines. The first time it was during the end of the first year of the project, while looking for alternatives to NFSv3. Unfortunately the Ceph version available at that point was in a very alpha state and unstable, inadequate overall to support a production environment. We revisited the Ceph option after a few months, this time in order to pair it with the LVM/iSCSI version of pDisk. This time the software proved to be much more stable and we managed to deploy it successfully in the pre-production testbed. After running a few tests the performance recorded which pretty much similar to that of NFS thus we did not consider that it was worth adopting it in the production service since the overall storage space available from NFS exceeded significantly the aggregated distributed storage of the 16 participating nodes of the reference cloud service.

Despite these results the operations team still believes that Ceph in the future will be one of the strongest contestants as a storage management back-end of cloud computing infrastructures. The reason for this is that Ceph's block storage subsystem (RADOS Block Device or RBD) provides direct support to QEMU/KVM virtualization systems, enabling them to directly create and manage distributed block devices for VM storage. This is expected to offer virtualization block storage capabilities with increased management and performance characteristics. For example the RBD-based version of QEMU (qemu-rbd) appears to be a credible alternative to the LVM/iSCSI solution provided by pDisk and although it is currently out of scope of the project, porting pDisk on Ceph and RBD still remains within the roadmap of an open-source StratusLab initiative.

8.2.3.3 GlusterFS

GlusterFS [1] is a popular open source clustered file system. It was one of the first alternative (together with Ceph) that we tried in the preproduction infrastructure in order to evaluate it as a replacement to NFS. Contrary to Ceph at that point, GlusterFS was much more stable and mature software and we were able to have a basic installation ready with StratusLab distribution installed on top. Nevertheless some basic performance tests (read/write I/O performance) showed that GlusterFS had slightly worse behavior than the existing NFS system thus we decided also to exclude it as a file system alternative.

8.2.3.4 GPFS

GPFS (General Parallel File System) [2] is a high-performance shared-disk clustered file system developed by IBM. It is used by many of the world's largest commercial companies as well as some of the faster supercomputers in the world. Contrary to the other two similar alternatives that we evaluated (Ceph and GlusterFS), GPFS is neither open-source nor cost free. On the contrary it is an expensive system and for large installations the acquiring of GPFS licenses is one of the major capital investments that and organization must do. Nevertheless, it is considered one of the best solutions for distributed file systems exhibiting excellent performance characteristics compared to its competition. The closed code nature of the system and the fact that it is supports only enterprise-grade Linux distributions, was one of the reason that its evaluation within StratusLab took a long time to complete and actually was only done near the end of the project.

The first results from the tests with GPFS showed that the system stands true it its reputation delivering excellent I/O bandwidth results. A few simple tests of file creation showed a peak bandwidth throughput of 235 MB/s which is almost 90% of the performance using the local disk directly. Thus GPFS appears to be a credible candidate for cloud-storage file system. Still it remains an expensive solution and its inflexible policy of supporting only commercial Linux distributions will discourage a large number of organizations that wish to invest in open source solutions. From the point of view of StratusLab we would still like to have the opportunity to perform a few more tests and actually deploy a complete large scale cloud on top of a GPFS file system.

8.2.3.5 iSCSI

iSCSI [4] is a popular IP-based storage networking standard that allows the sharing of storage volumes between hosts over the internet. It is not a file-system per se since it covers only the networking and sharing part of volume management. Coupled with LVM though, iSCSI provides a straightforward solution for volume creation, management and sharing within a cloud computing infrastructure. For this reason the combination of LVM/iSCSI was selected as the underlying technology for the first version of StratusLab's pDisk service. Performance-wise, iSCSI can deliver very good results within a local infrastructure and is supported by most major storage server technologies (NetApp, EMC etc).

Nevertheless the adoption of iSCSI within StratusLab was not trouble-free. First of all we experienced problems related with the iSCSI target demon implementation shipped with Linux (tgtd) which in some cases appeared incapable of managing large volumes of data requests. Additionally, as mentioned already, our effort to provide LVM/iSCSI over NFS proved to be more than inadequate. The conclusion is that in order to offer LVM/iSCSI storage management you need a large storage server that natively supports iSCSI and can deliver large volumes without relying on any other underlying network file system. This is the case for example of the NetApp port of pDisk that was developed in LAL and currently serves the public cloud operated on this site.

8.2.3.6 Overall

We believe that the storage management part of cloud infrastructure remains an open challenge and one of the main configuration complexities for prospective IaaS cloud providers. The solutions described above although representative do not by any means cover all potential implementation scenarios. Storage infrastructure needs careful planning from the early stages and taking strategic decisions that define the performance and scalability of the service in the future. Potential cloud providers should take into account the volume of users they want to serve, their specific application requirements regarding I/O, and of course the total capital expenditures that an organization is able or wishing to invest for the required hardware and software infrastructure. Thoughtful planning is also required in order to avoid vendor lock-ins which for the storage technologies appears to be very possible.

8.3 Cloud service operations

From the point of view of operations, running a cloud site is very similar to other public internet services. Two groups of people are involved in the provisioning:

Infrastructure administrators These are the people responsible to manage the hardware and the overall infrastructure in the datacenter. Their daily workload is moderate since their intervention is required during hardware failures (e.g. a crashed disk), a server upgrade (new hard disk added in a machine) and in some cases in an infrastructure reconfiguration (e.g. network recabling, relocation of servers etc.).

Cloud service administrators These are the people responsible actually to install and manage the service on the software level. They are in close interaction with the cloud software developers, the infrastructure administrators and in the case of StratusLab, with the external cloud users for support purposes.

One of the major concerns of cloud operations is the uninterrupted provisioning of cloud services. IaaS clouds by design offer flexibility and high availability. Problems are usually isolated (e.g. a node failure) and do not impact the end users since VMs can be easily reallocated without disrupting their operation.

Global downtimes are still possible to happen of course and indeed cause serious disruptions since all VMs and hosted services have to be brought off-line. VM snapshotting is useful in this case since it enables to bring the service back to the previous state typically with minimum interaction from the users. In the case of StratusLab there were a few occasions of global downtimes. At some point during the first year of the project, the datacenter had to be recabled in order to correct some network installation errors. Also an air-conditioning failure during the second year forced us to shutdown the systems in order to reduce power consumption and avoid a potential datacenter overheat.

Apart from hardware failures a typical cause for scheduled global downtime is a cloud service upgrade. For example when a new version of StratusLab distribution is released the service has to be brought off-line through the whole duration of the upgrade. In some cases this upgrade can be quite major, for example it may require the installation of a new hosting Operating System or the new cloud software installed may be incompatible with the previous one thus imposing a more complicated migration path (e.g. some system databases have to be imported to a new schema). In the case of critical hosted services (for example the production grid site) the impact of a global downtime can be limited by using a smaller scale secondary cloud site where you can temporarily migrate any “critical” VM until the upgrade is completed.

8.4 Software integration

Global downtimes due to cloud software upgrades was already identified as a potential problem for cloud operations. This issue generally falls within the software integration part of operations. Typically, operations teams interact with the software development team which provides the service software, implements requests and applies bug fixes in problems reported by the former. Within StratusLab it soon became evident that a close interaction between development and operations is important in order to ensure seamless software integration. This close interaction was pursued in the two ways:

- The participation of operations team in the agile processes established by WP4 (Scrum)

- The establishment of a clearly defined automated certification process for new software releases

The need for close interaction between Development and Operations has already been as an important aspect of service provisioning and has led to the development of DevOps agile development movement and the relevant methodologies and tools. Our experience from StratusLab confirmed the importance of DevOps and the benefits it brings for achieving an unobstructive, high-quality service provisioning.

8.5 Security

Security is of course of paramount importance for the provisioning of any internet based service and even more for IaaS clouds where the provider is hosting numerous third-party applications some of which potentially critical. Strong security measures have to be applied on the hardware and network layer, including:

- Hardware redundancy to avoid data losses due to infrastructure failures.
- Strong authentication mechanisms for physical hosts user access control. For example in StratusLab nodes we opted for public-key based authentication wherever possible in order to prevent password-guessing attacks.
- Monitoring mechanisms that can identify and notify unusual and potentially rogue activities.
- Carefully planned firewall rules that protect the hosting systems but still allow for maximum flexibility and exploitation of the infrastructure from the end-user applications.
- The definition of a clear incident management plan that can be easily followed in order to quickly mitigate any security incident.

During the operations of StratusLab cloud service there was only one verified security incident; a well-known username/password combination was used by intruders to gain access to a ttylinux VM instance which then was used to launch attacks to other systems outside StratusLab. The incident was identified and reported by the Network Operations Center in GRNET. The immediate actions from our side was to shutdown the VM and remove the respective VM image from the Marketplace prohibiting other users to instantiate it and probably cause the same problem. An important part of security incident handling is the forensic analysis that identifies the root of the problem and defines further action plans to permanently mitigate the threat. The ability to save complete VMs in files and then re-instantiate them under controlled environment is one more benefit that virtualization (and thus cloud computing) brings to operations. In StratusLab this ability was further enhanced by the quarantine functionality that was implement in the

VM management layer which automatically saves an retains the images of shutdown instances for a short period of time in case such forensic analysis has to be performed.

This incident also showed that VM security is one of the major concerns for IaaS clouds since it is a asset that cannot be fully controlled by the cloud provider. In this case mechanisms like the VM image signing and endorsement implemented in the Appliance Marketplace can improve the assurance of the service since it implicitly enforces accountability. Nevertheless, it seems that it would make sense to establish some form of official VM image certification procedures that will perform a number of sanity tests to a new candidate image to ensure some basic security requirements before allowing the image to be registered in a public Appliance Marketplace.

9 Conclusions

WP5 is completing its activities leaving as a legacy two public cloud services, running in GRNET and LAL respectively, an Appliance Marketplace instance, operated by TCD, and a record of its experiences and good practices from almost two years of infrastructure operations. During this period we have run into a good share of problems and obstacles, which from one hand hindered the smooth progress of the work package, on the other hand motivated us to dig deeper, identify gaps, investigate alternatives and overall led to the development of important know-how which is one of the most important outcomes of the activity.

In this document we have presented the final outcome of infrastructure operations, including cloud service provisioning, grid site provisioning, software integration testing and benchmarking. As a final outcome of the activity the document provided a set of good practices and lessons learned for cloud operations covering the areas of infrastructure architecture, service management, integration management and security, along with a brief reference to the economic analysis described in more detail in D5.4.

According to the sustainability plans defined by the consortium, the abovementioned results will outlive the lifetime of the project. The two IaaS clouds along with the Appliance Marketplace will continue to be provided and supported on best-effort basis, offering the necessary infrastructure for the envisioned open-source StratusLab initiative. Within this context, we finally plan to expand the existing infrastructure management knowledge base and keep it up to date with the technology evolution.

Glossary

Appliance	Virtual machine containing preconfigured software or services
Appliance Repository	Repository of existing appliances
API	Application Programming Interface
BDII	Berkeley Database Information Index
CA	Certification Authority
CapEx	Capital Expenditure
CE	Computing Element
DCI	Distributed Computing Infrastructure
EC2	Elastic Computing Cloud
EDGI	European Desktop Grid Initiative
EGI	European Grid Infrastructure
EGI-InSPIRE	EGI Integrated Sustainable Pan-European Infrastructure for Researchers in Europe. The European funded project aimed to establish the sustainable EGI
FC	Fiber Channel
Front-End	OpenNebula server machine, which hosts the VM manager
IaaS	Infrastructure as a Service
IGE	Initiative for Globus in Europe
Instance	see Virtual Machine / VM
LAN	Local Area Network
LVM	Logical Volume Manager
Machine Image	Virtual machine file and metadata providing the source for Virtual Images or Instances
NFS	Network File System
NGI	National Grid Initiative
Node	Physical host on which VMs are instantiated
OS	Operating System
pDisk	Persistent Disk
Private Cloud	Cloud infrastructure accessible only to the provider's users
Public Cloud	Cloud infrastructure accessible to people outside of the provider's organization
RAID	Redundant Array of Interdependent Disks
SAS	Serial Attached SCSI
SCSI	Small Computer System Interface
SSH	Secure SHell

TB	Terabyte(s)
TCO	Total Cost of Ownership
Virtual Machine / VM	Running and virtualized operating system
VM	Virtual Machine
VO	Virtual Organization
WebDAV	Web-based Distributed Authoring and Versioning
Worker Node	Grid node on which jobs are executed
WN	Abbreviation of Worker Node

References

[1] GlusterFS. http://www.gluster.org.

[2] GPFS: A Shared-Disk File System for Large Computing Clusters. In *Proceedings of the FAST'02 Conference on File and Storage Technologies. Monterey, California, USA*, pages 231–244, January 2002. ISBN 1-880446-03-0.

[3] EGI. Federated Clouds Task Force. https://wiki.egi.eu/wiki/Fedcloud-tf:FederatedCloudsTaskForce.

[4] IETF - Network Working Group. RFC3720 - Internet Small Computer Systems Interface (iSCSI). http://tools.ietf.org/html/rfc3720, April 2004.

[5] I. Konstantinou, E. Floros, and N. Koziris. Public vs Private Cloud Usage Costs: The StratusLab Case. 2nd International Workshop on Cloud Computing Platforms (CloudCP), 2012.

[6] R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh, and B. Lyon. Design and implementation or the sun network filesystem, 1985.

[7] The StratusLab consortium. StratusLab Marketplace - Technical Note. http://stratuslab.eu/lib/exe/fetch.php/documents:marketplace-v3.0.pdf, 2011.

[8] The StratusLab consortium. Deliverable D5.4 - Economic Analysis of Infrastructure Operations. http://www.stratuslab.org/lib/exe/fetch.php?media=documents:stratuslab-d5.4-v1.2.pdf, 2012.

[9] U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research. The Magellan Report on Cloud Computing for Science. http://science.energy.gov/~/media/ascr/pdf/program-documents/docs/Magellan_Final_Report.pdf, December 2011.

[10] S. A. Weil. Ceph: Reliable, scalable, and high-performance distributed storage. Ph.D. thesis, University of California, Santa Cruz, December 2007.