Enhancing Grid Infrastructures with
Virtualization and Cloud Technologies

First Year Infrastructure Operations Report

Deliverable D5.3 (V1.1)
16 June 2011

Abstract

This document summarizes the achievements of the infrastructure operations activity during the first year of the project. It describes the technical specifications of the infrastructure that has been put into operation, the process that was followed to establish it, the problems encountered and the various solutions that were applied. It also provides statistics about the usage of cloud resources and an assessment of their utilization.

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

Contributors

Name	Partner	Sections
Evangelos Floros	GRNET	All
Charles Loomis	LAL/CNRS	Pre-production services. Benchmarks.
Stuart Kenny	TCD	Marketplace
Ioannis Konstantinou	GRNET	Alternative File Systems

Document History

Version	Date	Comment
0.1	10 May 2011	Initial draft with TOC.
0.2	15 June 2011	First version for review (without executive summary)
1.0	15 June 2011	Final version.
1.1	16 June 2011	Minor changes.

Contents

List of Figures

List of Tables

1 Executive Summary

This document presents the activities carried within WP5 during the first year of the StratusLab project. The document summarizes the work performed and the tangible results that have been delivered, providing a retrospective view of the success stories. In parallel it outlines the various issues identified, and concludes with a glimpse of the work in progress due to be completed during the second half of the project. In particular the documents covers the following aspects of WP5 activities: cloud service operations, grid service operations, appliance creation and maintenance, support services, and testing and benchmarking.

Cloud services comprise the foundations for many of the activities that take place within WP5 and are important for the work of the rest of the project's WPs. For this reason they consume the majority of effort allocated for WP5 and naturally correspond to a large fraction of the activities that took place in the first year. These activities mainly are related to the provision of the project's reference cloud service. This service can be considered the production-level service that the project provides to external parties. A significant amount of work and resources are also dedicated to beta-testing and deployment of pre-production level services. Most of the software for the these services are integrated from WP4 and WP6. Nevertheless, some development and integration effort is also provided by WP5 especially for what concerns new capabilities on the infrastructure level such as networking, file systems etc.

The provision of Grid sites on top of IaaS cloud services has been a core motivation for the project since inception. As such, this activity is considered fundamental for the project and an important criterion of the overall project success. The work done in this area includes the installation of pre-production and production grid sites that have been deployed on top of StratusLab's reference cloud service. These activities have been important to gain experience from the exercise and identify gaps and missing functionality. This know-how will help with the work that takes place in WP6 regarding advanced cloud services for grid sites, including grid site elasticity and dynamic management of grid resources.

The success of any cloud IaaS solution largely depends on the availability of a rich set of stable and secure Virtual Machine appliances. These appliances are an important asset for the cloud infrastructure since their availability becomes a major incentive for new users to join the service and exploit the offered cloud capabilities. StratusLab has focused on all aspects of VM appliance lifecycle: the

definition of recipes for appliance creation, the production of a large number of appliances, the development of the appliance curation and distribution functionality, and the provision of the respective service for the public. The last service, called the StratusLab Appliance Repository, has been a fundamental tool during the first year of the project. Coupled with the reference cloud service, it has given the opportunity to internal and external users to take advantage of the cloud technologies developed by the project.

The infrastructure services offered to internal and external users (i.e. the reference cloud service, the Appliance Repository, the pre-production testbed etc.) have to be coupled with proper support in order to ensure that the users are taking full advantage of their capabilities. For this reason much of the day-by-day work within WP5 has to do with the provision of user support, where by the term users we refer to system administrators that are using the StratusLab distribution and reference cloud users that exploit the IaaS cloud service. Also since WP5 is the infrastructure provider of the project it is natural that it will contribute to the internal processes of the project that require access to physical resources. Such processes are for instance the software build and integration services, the hosting of software repositories, etc.

Testing is also part of the WP5 activities. WP5 is the first "user" of the software developed or integrated by the project. As such it has been assigned with the responsibility to be the first to try and verify the maturity of the software produced. Typically the pre-release versions of the software exhibit various problems that have to be reported back to the developers. Therefore a cycle of iterative improvements is usually foreseen before the software is accepted for public release. Testing activities are coupled with benchmarking processes that aim to quantify the benefits of the developed cloud solutions.

The above success stories have raised the expectations from WP5 and from the project as a whole for the second year of the project. For the cloud services our goals are to further evolve and expand the provided infrastructure offering a complete range of IaaS cloud capabilities, including support for persistent storage, support for private IPs, and capability to deploy and manage virtual clusters. An important dimension that will be pursued is the provision of federated cloud resources, i.e. seamlessly utilizing cloud resources offered by two or more interoperable cloud resource providers. For what concerns the work on grid service integration, the focus in the second year will be to enhance their capabilities by exploiting the dynamic and elastic nature of the underlying cloud layer. This work will proceed in close collaboration with WP6 that is currently developing the relevant functionality exploiting the capabilities of the Claudia service manager.

2 Introduction

This document presents the activities carried in the context of WP5 during the first year of the StratusLab project. WP5 is the sole provider of physical infrastructure in the project and responsible for most of the activities that directly utilize or affect the physical layer of a cloud computing infrastructure. As such it, is responsible for the provision of production level cloud and grid services, for beta testing the software integrated from other work packages and for providing day-by-day operational support for the usage of infrastructure facilities and software services offered within the project or to external third parties.

In previous deliverables, and in particular D5.1 [2] and D5.2 [3], we have specified the range of activities planned by WP5 and the tools and procedures that facilitate their execution respectively. In this document we summarize the work achieved and the tangible results that have been delivered during the first year of the project, providing a retrospective view of the success stories, as well as the various issues identified, and a glimpse of the work in progress planned for the second half of the project.

The document is split in five different sections reflecting the range of activities that WP5 has undertaken. Starting with Section 3 we present the IaaS cloud service operated by the project, the physical infrastructure allocated for this purpose and the various research activities that focus on the improvement of quality of service offered from this infrastructure. The section also provides a set of metrics and statistics collected during the first year, that help us make various interesting conclusions for the way StratusLab's services have been used and guide us for finding ways to improve them during the second year. Section 4 focuses on the important activity of grid and cloud service integration mainly presenting our work for the provision of grid sites on top of virtualized cloud resources. The work on VM appliance creation, management and distribution which is complementary to the cloud services described in the previous sections is presented in Section 5. The rest of the document presents the various support services that WP5 provides (Section 6) and the Testing and Benchmarking activities that are currently ongoing within the work package (Section 7).

3 Cloud Services

3.1 Overview

This section focuses on the activities related to testing and provision of Cloud services based on the StratusLab distribution. Cloud services comprise the foundations for many other activities that take place within WP5 but also in the remaining WPs of the project, for this reason they consume the majority of effort allocated for WP5. In the following paragraphs we present the activities that took place in the first year regarding the provision of the reference cloud services, including an analysis of the physical resources allocated for this purpose and various interesting usage metrics and statistics. The reference cloud service can be considered the production-level service that the project provides to external parties. A significant amount of work and resources are also dedicated for beta-testing and deployment of pre-production level services. Most of the software for these services are integrated from WP4 and WP6. Nevertheless, some research and integration effort is also provided by WP5 especially for what concerns new capabilities on the infrastructure level such as networking, file systems etc.

3.2 Reference Cloud Service

The reference Cloud Service operated by WP5 is one of the activity's most significant contributions to the project. It stands out as a valuable asset of the project and a crucial tool for various other activities like testing, dissemination, training and end-user application development. In this section we take a closer look at the inner details of the service provisioning during the first year of the project.

3.2.1 Overview of Offered Cloud Services

The purpose of the Reference Cloud Service is to provide a production-level installation of the StratusLab cloud distribution. The cloud service follows the evolution of the cloud software developed in the context of WP4 and WP6 offering at each certain time the whole range of components that have been released so far by these activities.

The cloud service provides an Infrastructure as a Service (IaaS) cloud giving the ability to users to instantiate VMs and manage their lifecycle. This management is performed remotely by using a set of end-user command line tools. By the end

of the first year of the project users could execute the following commands on a VM in the reference cloud service:

stratus-run-instance Instantiates one or more VM images with a user defined hardware profile.

stratus-describe-instance Provides information about the running status of a user's VM instances

stratus-kill-instance Kills one or more running VM instances. The command is similar to unplugging a machine from the power socket.

The hardware profile of the VMs can be selected from a list of pre-defined profiles hardcoded in the user-side client tools. The default profiles shipped within these tools can be retrieved by running the stratus-run-instance command with "-l" command line switch:

```
$ stratus-run-instance -l
Type        CPU         RAM         SWAP
m1.xlarge   2 CPU       1024 MB     1024 MB
m1.small    1 CPU       128 MB      1024 MB
c1.medium   1 CPU       256 MB      1024 MB
c1.xlarge   4 CPU       2048 MB     2048 MB
m1.large    2 CPU       512 MB      1024 MB
```

The above values can be changed in the client-side python scripts although this has not been extensively documented. Thus although the end users have the ability to customize their VM hardware capabilities most of them so far have utilized only the above configurations.

Exploitation of the IaaS service actually assumes access to one additional core service offered by StratusLab, namely the Appliance Repository. The repository is the container of VM images and appliances. During instantiation prologue phase, the images are transferred over http to the cloud service and then are passed to the hypervisor that starts running it. More information about the Appliance Repository is provided with greater detail in Chapter 5.

Access to the service is open to anyone interested either inside or outside of the project. Requests for accessing the service are send by email to support@stratuslab.eu. External users need to provide their full contact details and the purpose of their usage. So far we have not rejected any request; on the contrary at any opportunity StratusLab members encourage colleagues from other projects to get access and try out the project's cloud services.

By the end of first year the service offers two ways for user authorization.

1. Username/password pair generated internally from StratusLab operations
2. Digital certificate issued by an IGTF accredited Certification Authority. In this case authentication can be done directly using this certificate or by generating a VOMS proxy certificate signed from the above.

Table 3.1: *Physical resources allocated for the reference cloud service*

Resource	Per node	Aggregate
CPU	Intel Xeon E5520	
Speed	2.67 GHz	
CPU×Cores	2×4 (HT enabled)	256 logical
Disk(s)	2×146 GB	2.3 TB
SAN storage		3 TB (NFS shared)
RAM	48 GB	768 GB
Ethernet	2×10 Gb/s	

For the first case, a username/password pair is generated by the cloud administrator and is communicated to the user by email. The second options requires that the user holds a valid digital certificate issued by an IGTF accredited CA. In addition if a VOMS proxy certificate is the preferred method of authentication, the user has to be enrolled in at least one EGI grid VO. So far, username/password has been the most popular method of authentication due to its simplicity and fast learning curve, despite being less secure. The certificate-based solutions although more secure are still underutilized due to their complexity of usage but also because not everyone is accustomed to PKI technologies.

For what concerns storage services, since by the end of the first year the cloud service was running v0.3, of the StratusLab distribution, which lacked any persistent cloud storage solution, these were restricted to offering the capability to users to allocate an extra volume during VM instantiation. This volume has to be mounted by hand inside the VM's file system after the VM is up and running. This is still a pretty basic capability since the storage is not persistent and the volume disappears when the VM is destroyed, thus losing all stored data.

3.2.2 Physical Infrastructure

The reference cloud service is running on top of a dedicated set of physical resources allocated by GRNET. The physical infrastructure comprises of 17 nodes located in the same rack. The nodes are dual quad-core Fujitsu Primergy RX200S5, configured with Intel Xeon E5520 running at 2.26GHz with 48Gbyte of main memory (DDR3-1066 ECC). Hyper-threading has been enabled in the CPUs thus a total of 16 logical cores per node are reported by the OS.

For what concerns storage, each node is equipped with 2x146GB SAS hard disks configured in RAID 1 (mirroring) thus the total storage space available for the system and applications is 146GB in each node.

Apart from the local storage available from each node, additional storage is provided to the cloud service from a central storage server installed in the datacenter. The server is an EMC Celerra NS-480 providing a total of 200TB over NFSv3

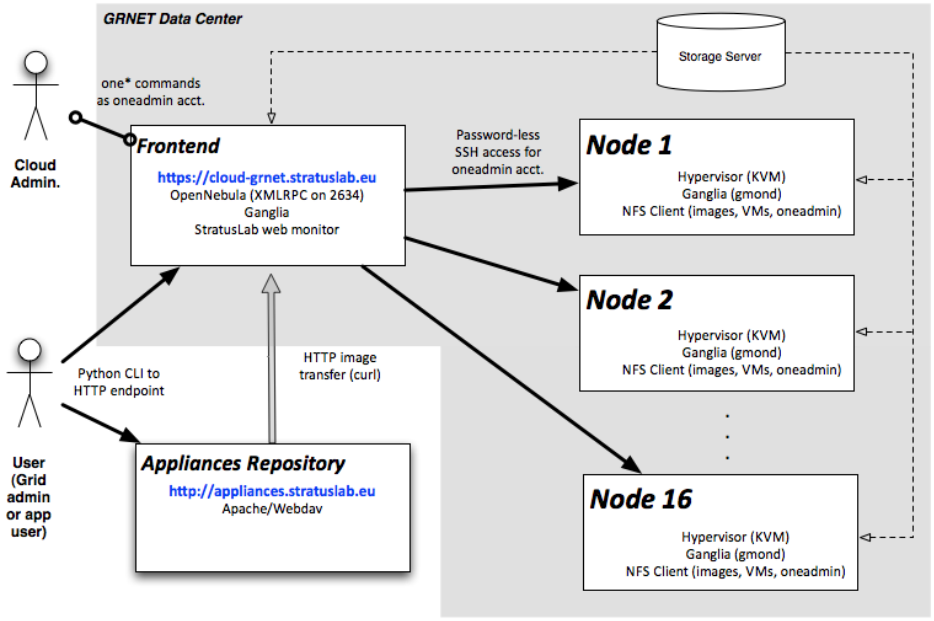

Figure 3.1: *Architecture of StratusLab production cloud service*

and iSCSI interfaces. For the cloud service a total of 3TB have been allocated and are shared among all nodes using the NFS interface. Additional volumes have been reserved from the storage server for various other services of StratusLab (appliance mirror, grid services etc.).

Figure 3.1 depicts the topology of the cloud service installation. One node acts as a frontend whereas the remaining 16 nodes are hosting the VM instances. The frontend is hosting the proxy/gateway service that allows external users to authenticate and to interact with the cloud service. This is also where the OpenNebula virtual machine manager (VMM) is installed. The hosting nodes are running the KVM hypervisor and through the orchestration of OpenNebula in the frontend, they permit the life-cycle management of VM instances.

Until StratusLab version 0.3, CentOS 5.5 was the base OS used for all nodes. By the end of the first year and with the upgrade to 0.4, it was decided to re-install all nodes with Fedora 14. The motivation for this OS migration was to take advantage of the newest Linux kernel and system libraries that are shipped with the latter.

Table 3.1 summarizes the number of resources offered for the service.

3.2.3 Service Utilization and Metrics

In this section we present various statistics and metrics we have collected during the first year of the project from the operation of the reference cloud service. These statistics actually cover the period from January 2011 when StratusLab v0.2 was deployed in the production infrastructure, till the end of May 2011 when the in-

frastructure was upgraded from v0.3 to v0.4.

Overall the total number of VMs instantiated were 2433. As of the time this document is authored, 2345 of them have reached a Done state (they have been shutdown and do not consume resources anymore), 83 are running, 2 are suspended and 2 have Failed. The actual number of failures is actually larger but it is typical that the users will run a stratus-kill-instance command for the failed instances in order to release the resource allocated for them thus these instances appear to be in *Done* state.

Among the instances, 2213 of them have been instantiated from raw images and 219 from qcow images. The usage of qcow images was introduced midway during the operation of the service and was actually suggested as the preferable disk image format to use for VM instances. The main reason is that this way we alleviate to a certain degree the performance problems of NFS since qcow images are inherently compressed that they take less time to download and require fewer resources during the prolog phase.

The average memory requested per instance was 761.62 MB and the average cores per VM was 1.63 cores. This shows that in general users instantiated small VMs mainly for testing purposes and for getting a first experience of the service usage. This is probably natural for the first months of service usage since people are trying to solve basic problems and become acquainted with the introduced technologies.

Another interesting statistic is the number of instances per user. Here we separated the internal users (people from the StratusLab project consortium) from external ones (people outside of the project evaluating the software). As the table shows on average external users instantiated 11.13 VMs whereas internal users started a much larger 158.33 number of instances. Obviously in the context of the project, people are starting and killing instances very often as part of the development, testing and validation procedures. On the other hand the number of instances for external users is encouraging since it shows that they are trying different things and some of them are already using the service for real applications. Certainly this is a number that we will try to improve for the second year of the project.

For what concerns the average lifetime of VMs the statistics show that this is 29.62 hours, with the maximum running time for a VM recorded at an impressive number of 119.68 days. Notice, that this statistic does not cover the instances that were still in Running state when this document was written so the latter number is actually larger. Indeed during this period we followed a policy from the operations team to avoid decommissioning VM instances despite that many of them were running without doing actual work. This way we wanted to verify the stability of the system and the ability to sustain various external events (power failures, data center maintenance) without affecting the the availability of the service.

Table 3.3 provides an overall picture concerning the distribution of VMs instantiated from the available images from the appliance repository. The most popular images were those based on CentOS distribution. This OS was used for a different number of test images and was utilized a lot during test and validation processes.

Table 3.2: *Aggregated reference cloud service metrics*

Metric	Value
Number of VMs instantiated	2433.
Number of VMs that reached Done state	2345.
Number of VMs active by the end of Y1	83.
Number of VMS suspended	2.
Number of VMs failed	2.
Total raw images instantiated	2213.
Total qcow images instantiated	219.
Average cores per VM instance	1.63
Average memory per VM instance (MB)	761.62
Average Bytes received by instances	54484822.85
Average Bytes transmitted by instances	14640433.58
Maximum Bytes transmitted from a single VM	2147441949.00
Maximum Bytes received by a single VM	2147321522.00
Average VMs per external user	11.13
Average VMs per internal user	158.33
Average running time of VMs (hours)	29.62
Maximum running time for a VM (days)	119.68
Average startup time for VM instances (time in Prolog state) (minutes)	6.069

Table 3.3: *Number of machine instances by OS*

CentOS	1435	59%
Ubuntu	231	9%
ttylinux	533	22%
SL5	190	8%
Fedora	5	$< 1\%$
Other	39	2%
TOTAL	2433	100%

Table 3.4: *Number of machine instances by requested CPU cores*

Cores	Instances
1	1460
2	586
3	5
4	329

The second most popular distribution was ttylinux which is a very minimal linux distribution aimed for a restricted number of use cases. This is explained from the fact that ttylinux has been the main image we used during demonstrations and tutorials since it is fast to instantiate and provides a complete linux environment to validate the cloud service capabilities.

Tables 3.4 and 3.5 present statistics regarding the most popular hardware configurations used when starting new instances. For what concerns the number of cores the most popular choice was to start instances with 1 core, with 2 cores and 4 cores being the other popular choices. For what concerns memory, more than half of the instances utilized only 128 MB. Again this shows that during this period most instances were used for very basic validation work that did not require demanding resources. As the project and the infrastructure evolves we should expect to see more powerful configurations being used in order to run real-life applications.

3.3 Testing and pre-production services

3.3.1 LAL Pre-production Cloud Service

LAL provides a cloud infrastructure that will eventually become a production public cloud service. This infrastructure currently consists of 1 front-end node (Class LAL2, see Table 3.6) and 7 hosts (Class LAL3) for running virtual machines. The

Table 3.5: *Number of machine instances by requested memory*

Total memory (MB)	Number of instances
128	1300
256	84
512	153
1024	216
1740	264
2048	373
4097	33
7168	1
Other	9

Table 3.6: *Physical Machine Configurations at LAL*

	LAL1	LAL2	LAL3
CPU	Opteron (64 bit)	Xeon (64 bit)	Xeon (64 bit)
Speed	2.2 GHz	2.33 GHz	2.67 GHz
CPU×Cores	2×2	2×4	4×6
Disk(s)	1×80 GB	1×160 GB	2×250 GB
RAM	8 GB	16 GB	36 GB
Ethernet	2×10 Gb/s	2×10 Gb/s	2×10 Gb/s

system currently uses the standard configuration with an NFS shared file system mounted from the front-end node. The exercise of deploying this infrastructure has provided valuable feedback to the project concerning policies and required functionality.

When first trying to deploy the hardware and make the cloud infrastructure accessible from outside of the laboratory, the CNRS/IN2P3 security officers were very reluctant to open all of the ports in the firewall for these machines. This is a dramatic change to the current policies and they viewed this as a serious security hole. After lengthly discussions, a large number of common ports were opened for virtual machines running on the infrastructure with access to the physical machines restricted to machines at LAL. This highlighted the need to inform security personnel concerning cloud technologies so that they understand the need for breaking with previous firewall policies. This also highlighted the need within the StratusLab distribution for better monitoring of network activity and the development of dynamic firewalls for virtual machines. Work on this additional functionality has started in other work packages.

Two security incidents involving hijacked virtual machines highlighted the need for better logging and improved tools for forensic analysis. Because of the first incident (at LAL), the logs for OpenNebula were improved to provide better tracking of allocated IP addresses and identities of users. A quarantine mechanism was also introduced that saves machine images for a configurable time period (default is 48 hours) after they have completed. The second machine highjacking (at GRNET) occurred after the quarantine mechanism was in place. The forensic analysis was greatly simplified as all of the changed files could be found and inspected with a simple 'diff' between the initial image and hijacked one. This quarantine mechanism works only for a cloud deployment using a shared file system; it must be extended to include 'SSH' deployments.

A condition from the CNRS-IN2P3 security officers for opening this service to the public was the registration of users of the service. The registration must collect simple identification information (name, email address, etc.) but must also require that the user agree to the existing EGI Acceptable Use and Security Policies. To satisfy this requirement, a registration service was developed to handle this registration. The information is kept in an LDAP server which is compatible with the StratusLab authentication services, allowing access via username/password pairs and via grid certificates. This has been integrated as part of the standard StratusLab release.

The remaining issue with this infrastructure is the lack of significant storage resources. Already local users have had problems with launching large numbers of virtual machines because of a lack of disk space within the system. Foreseen funding has unfortunately not appeared, so storage hardware will likely be purchased with StratusLab funds instead. This will mean a reduction in the overall funded human effort, but believe that this will better serve the aims of the project and allow faster evolution of the StratusLab distribution.

3.3.2 GRNET Pre-production services

In addition to the reference cloud service, GRNET has allocated a few more physical nodes in order to host and provide support and pre-production services. In particular three nodes are dedicated for deploying and beta testing snapshot versions of the StratusLab distribution in order to test drive them, validate them and give the green light for public release. These three nodes are sufficient for a very basic deployment in which one node acts as a frontend and the two other nodes act as VM hosts.

The pre-production infrastructure has played multiple roles during the first year of the project. It has been used extensively to test and to validate the StratusLab distribution before being officially released. It has been used as an experimentation testbed for trying out new ideas and for prototyping new services (see next section). As such it has been formatted and re-configured many times remaining nevertheless a stable and consistent project service.

In addition to the above resources, two nodes have been provided to WP6 in order to be used for the testing and integration activities of the Service Manager (Claudia) and three nodes have been allocated to WP4 as a testbed for the developments of this activity. Both installations are considered highly unstable and are used only for the initial development and integration efforts of both activities. Once the code produced is stable enough it is then deployed in the pre-production infrastructure for final testing.

3.4 Service prototyping

3.4.1 Persistant Storage

A prototype "persistent disk service" has been developed within the project to provide a service similar to Amazon's Elastic Block Store (EBS). This service allows users to create new disk images of a specified size and then to mount those disk images on running virtual machines. A single disk image can be mounted on only one active virtual machine at a time. The disk image is persistent and can be used to store data or service state independent of the lifecycle of a particular machine image. The system uses a standard, open-source iSCSI target (iSCSI Enterprise Target–IET) to make the disks available to running virtual machines. This system could also be used as an image store to both cache images and make them available to the hypervisors on the physical machines. This approach shows promise, but has not yet been tested throughly.

3.4.2 Alternative file systems

Experience with the current production, pre-production, and test infrastructures has highlighted the performance and scalability issues with using NFS as a shared machine image store for cloud infrastructures. For this reason we decided to experiment with various other distributed and shared file-system technologies in order to identify one that would offer us the best performance and would fit our specific

technological setup. During the first year we investigated two different solutions, namely Ceph and GlusterFS.

Ceph

Ceph[1] is a distributed file system designed for reliability, scalability, and performance. The storage system consists of some (potentially large) number of storage servers (bricks), a smaller set of metadata server daemons, and a few monitor daemons for managing cluster membership and state. Although still in alpha version, Ceph has gained significant publicity and the client side has been merged with Linux kernel since version 2.6.34. For this reason it was our first choice for experimentation.

The system was installed in our pre-production infrastructure comprising of 3 nodes which at that point were installed with CentOS 5.5. The installation process was not straightforward since unfortunately the default CentOS kernels do not come with Ceph support and there are not pre-compiled RPM packages in the distribution's repositories. Thus we had to compile the system from the sources and also re-compile the kernel in order to add support for the software.

After we finished with the installation, we replaced the shared directory where OpenNebula keeps its virtual machine images (i.e. /var/lib/one) with the Ceph directory and we started our tests. During the tests, we did not notice an increase in the performance compared to the NFS case. What is more, the Ceph filesystem was constantly crashing or was not responding (the Ceph monitoring daemon *cmon* had 100% CPU utilization and the file system could not be accessed). Therefore, we concluded that the Ceph filesystem is not mature enough to be adopted by a production service. Nevertheless, since it is a very promising system, we will continue to examine its applicability in the StratusLab production environment.

GlusterFS

We continued our examination of NFS alternatives with a popular and more production-ready distributed file system, the GlusterFS. GlusterFS[2] is a clustered file-system capable of scaling to several peta-bytes. It aggregates various storage bricks over Infiniband RDMA or TCP/IP interconnect into one large parallel network file system. Storage bricks can be made of any commodity hardware such as x86-64 server with SATA-II RAID and Infiniband or GigE interconnect.

The GlusterFS installation was a more straightforward procedure, since both the server and client packages were available for CentOS. The GlusterFS has several operation modes, and for our tests we selected the one with the best performance (ignoring, for instance fault tolerance). According to the GlusterFS manual[3] the best read and write performance is achieved with the use of striping, i.e., by distributing chunks of a file across many storage servers. In our setup, we utilized stripes of 1MB size across 3 storage servers. We conducted the same experiments with the Ceph case, and we were serving the directory that contained

[1] http://ceph.newdream.net/

[2] http://www.gluster.org/

[3] http://gluster.com/community/documentation/index.php/Gluster_3.2:_Configuring_Distributed_Striped_Volumes

the virtual machine images through a GlusterFS shared directory. Nevertheless, we did not notice significant improvement compared to the NFS case, when we tried to increase the number of concurrent served virtual machines. In this case, the performance degradation was similar to the NFS case.

3.4.3 Networking

VMs that are instantiated in the reference cloud service are accessible over the network from a public IP address. A total of 120 public IP addresses in the range 62.217.122.130-240 have been allocated for the cloud. A few of them have been reserved for the CE and SE nodes of the production grid site running on the cloud. The rest are randomly assigned whenever a new VM is instantiated. Public IP addresses are a critical resource for the cloud service and in reality it is the resource that is putting the limit on the number of VMs that the reference cloud service can sustain. To overcome this problem there are three solutions:

Allocate a large range of public IP addresses This solution is not sustainable since in the end there will always be a shortage of DNS addresses. However many IP addresses we reserve users will have the tendency to acquire them all.

Use private IP addresses In principle StratusLab supports private and local IP addresses although this capability has not been enabled yet in the cloud service. This solution can satisfy use cases like virtual clusters in which a user may instantiate a large number of interrelated VMs but only a few of them (typically one) require access to the public network. The solution is also limited by the number of VLANs that can be configured in the hosting infrastructure. Although there is no hard limit imposed on this, in order for the service to scale the definition and management of VLANs should be done in an automated and scalable manner otherwise there will be lots of administrative effort for maintaing the service and making sure the private IP addresses are always available.

Use IPv6 The large IP space available from IPv6 would be the ideal solution for the public IP provision from cloud services in general. The requirement of private IP addresses could still be relevant in order to explicitly enforce sub-cluster isolation. The problem with this solution is the lack of support for IPv6 from the underlying cloud middleware and in many cases from the network hardware itself.

StratusLab does not provide an explicit way for defining firewall rules and blocking access to TCP/UDP ports in the VMs. For the reference cloud service we have decided to keep all ports open for the VMs in order not to enforce access restrictions to end-users and their applications. It is assumed that the users are responsible for the security of their VM instances and that if specific firewall restrictions are needed this will be either pre-configured in the VM appliances or will be applied upon VM instantiation through contextualization.

4 Grid Services

4.1 Overview

The provision of Grid sites on top of IaaS cloud services has been a core motivation for the project since its inception. As such, this activity is considered fundamental for the project and an important criterion of the project's success. In this section we analyze the work done regarding the installation of pre-production and production grid sites that have been deployed on top of StratusLab's reference cloud service. These activities have been important to gain experience from the procedure and identify gaps and missing functionality. This know-how will help with the work that takes place in WP6 regarding advanced cloud services for grid sites including grid site elasticity and dynamic management of grid resources.

4.2 Deployment tests for grid sites

Tests with deployment of grid sites on a virtualized environment started as soon as we had an initial cloud service available in the project, even before the first release (v0.1) of the StratusLab distribution. This early experience helped us develop the required knowledge for what concerns technical issues, like the creation of VM appliance with grid software, and operational issues, like the acquisition of digital certificates and integration of the test grid sites with centralized monitoring and testing services (Nagios, SAM tests etc.).

Once we gained enough experience from the above tests the whole process was formalized by the deployment and certification of a production grid site. In parallel we prepared a number of VM appliances for the basic node roles of a gLite-based grid site: the Computing Element, the Storage Element, the Worker Node, the User Interface and the APEL service. All of these images are currently available from the appliance repository.

The certified grid site is named HG-07-StratusLab. The site was certified within the GRNET NGI (the Greek National Grid Initiative) and has joined the national grid infrastructure (HellasGrid). The site offers one CE and 8 dual-core WNs thus providing a total capacity of 16 cores for job execution. The site also supports MPICH-2 and OpenMPI jobs. Communication among the nodes for MPI execution is supported through ssh host-based authentication. Each WN is configured with 2GB of main memory. The site also comprises a SE that offers a total

storage space of 2TB. It should be noted that this storage is configured directly as an NFS mount-point from the local storage server and is not yet virtualized (e.g. it is not managed as a persistent storage service from the StratusLab command line tools).

The site currently supports the StratusLab VO (vo.stratuslab.eu) as well as the required EGI-wide and local ops (operations) VOs. Obviously the job processing capacity of the site at this phase is rather limited but the priority has been to validate the procedure and identify various issues than to provide a powerful production infrastructure for end-user applications. In the future depending on the workload and potential requests to support additional VOs, it would be trivial to expand the workload execution capacity of the site (i.e. add more cores and WNs). The GStat page[1] contains all of the details for the site.

4.3 Production grid site operation

HG-07-StratusLab has been running as a production grid site for the last quarter of the first year of the project with no significant issues or problems to report. As it was more or less expected the fact that the nodes were deployed on Virtual Machines did not have any real impact on the stability and availability of the service. Performance-wise, the initial tests we run on the cloud infrastructure showed no performance penalty for end-user applications. Nevertheless, with the current setup of the cloud service that is using NFS for volume sharing it has been observed an overall performance degradation which although has not been verified with grid application and it is safe to assume that such a problem does exist.

The underlying middleware has followed the evolution of gLite and the nodes have been kept up-to-date with the major releases of the software or by applying any specific requirements communicated by the EGI operations groups. The respective VM appliances that were used to setup the site are also kept up-to-date with the grid middleware releases and are regularly updated on the Appliance repository.

One of the most useful outcomes of the whole deployment/operation activity of the grid site was the various issues and ideas recorded during this process that have been reported in the “Installing and operating a production grid site in the StratusLab cloud” [5] Technical Note. This document could potentially be a starting point for concrete collaboration with relevant DCI projects, i.e. EMI and EGI-InSPIRE, in order to provide optimized grid services on cloud environments or inversely cloud-services for optimal grid site operation.

4.4 Future work

At first sight, the provision of grid services on top of cloud technologies could be considered just another application for cloud services. Indeed, one can operate a grid site using virtualization technologies without any particular integration be-

[1] http://gstat-prod.cern.ch/gstat/site/HG-07-StratusLab/

tween the two layers. Nevertheless, in order for grid services to fully exploit the potential of cloud computing there should be a bridging of these two worlds on a technical and operational level.

Resource elasticity is one of the most well known advantages introduced by cloud computing. Grid sites should be able to capitalize on this by being able to dynamically adjust their dimensions based on fluctuating demands. Typical dimensions of a grid site are:

Processing capacity: being able to modify the size the cluster by adding or removing WNs on demand.

Processing capability: being able to modify the profile of the WNs by adding new ones with more CPU cores, local storage and memory.

Storage capacity: being able to modify the available storage provided by the SE node.

In order to implement such elasticity functionality the grid and cloud layer should work closely together. The control and definition of elasticity rules and triggers have to be defined in the cloud layer, since this is the place where virtual resources are conceptualized. The grid layer should provide all the necessary information about the current state of grid resource utilization (e.g. number of jobs in the CE) since this is the layer where all information about grid resources are kept.

Such functionality is currently being developed in StratusLab with the close collaboration of WP6 and WP5 activities. The idea is to be able to capitalize on the abilities of the Claudia Service Manager for elastic management of cloud resources. Elasticity is applied by monitoring the utilization of the grid site LRMS queues and then triggering the expansion or reduction of the resources when the workload exceeds or falls beyond a specific threshold respectively. For this purpose we are currently developing the probe and monitoring services that will run on a CE and communicate all necessary information back to the Service Manager. More details on this work can be found in D6.3 [4].

This dynamic behavior of grid sites on the other hand may cause inconsistency on the global level if the information about the sites new configuration are not promptly propagated to the top level information systems (e.g. top-level BDII), causing job management services like the WMS (Workload Management System) to make inappropriate job scheduling decisions. Thus also the top-level grid services should be able to cope with dynamic grid sites whose dimensions and capabilities are continuously changing.

5 Appliance Creation, Management and Maintenance

5.1 Overview

The success of any cloud IaaS solution largely depends on the availability of a rich set of stable and secure Virtual Machine appliances. These appliances are an important asset for the cloud infrastructure since their availability becomes a major incentive for new users to join the service and exploit the rest of the provided cloud capabilities. StratusLab has focused on all aspects of VM appliance lifecycle; from the definition of appliance preparation recipes, the preparation of a large number of appliances, the development of the appliance curation and distribution functionality as well as the provision of the respective service to the public. The latter service referenced as the *StratusLab Appliance Repository* has been a fundamental tool during the first year of the project, which coupled with the reference cloud service has given the opportunity to internal and external users to take advantage of the cloud technologies developed by the project. In the following sections we analyze all aspects of appliance provision as these are implemented in the context of WP5.

5.2 Image Creation

Almost all operating systems provide a mechanism for automated installation via PXE. Reusing this mechanism to automate the production of virtual machine images offers the possibility to document the contents of a particular image (through kickstart or other description files) and to regenerate them easily to incorporate regular security updates of the operating system. LAL has adapted the mechanisms used in Quattor for the automated installation of machines via PXE to provide a similar mechanism for cloud users. In this case, cloud users provide the network boot file and the machine description to create the virtual machine. A prototype of this works, but needs to be integrated into the main StratusLab distribution and command line interface.

5.3 Appliance Repository

For the first year an initial prototype of the virtual appliance repository was deployed as a WebDAV-enabled Apache webserver located at TCD. A backup server[1] has been provided by GRNET using a simple mirroring scheme. The contents of the primary repository are currently mirrored once every day using a regular Unix cronjob.

At time of writing the repository contains the following images:

- Base images:
  - Ubuntu 10.04
  - CentOS 5.5
  - ttylinux 9.7 (a minimal Linux operating system useful for testing)
- Grid Appliances:
  - Grid CE (SL5.5)
  - Grid SE (SL5.5)
  - Grid WN (SL5.5)
  - Grid UI (SL5.5)
  - Grid Apel (SL5.5)
- Bio-informations Appliances:
  - Bio Data (CentOS 5.5)
  - Bio Compute (Centos 5.5)

The goal of the prototype repository was to quickly and simply provide centralised storage that could be used by the project to share images. For the second year of the project the repository will evolve to become an appliance *metadata* marketplace. Storage of appliances will be handled by the appliance provider, and only the metadata associated with the appliance will be stored on the central registry.

The images provided through the appliance repository are intended to be freely available. Users can access the images using the StratusLab command-line tools, or by using standard tools such as *cURL* [1]. The Appliance Repository front webpage[2] shows the images currently available and also displays information about the images.

Write access to the repository is currently restricted to StratusLab project members. The web server is accessed using WebDAV (Web-based Distributed Authoring and Versioning), with authentication via the StatusLab LDAP server.

[1] http://appmirror-grnet.stratuslab.eu
[2] http://appliances.stratuslab.eu

5.4 Marketplace

As stated above, for the second year of the project the Appliance Repository is evolving to become an *Appliance Marketplace*. The Marketplace is it 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 will be 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 can be 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 marketpace including specification about the metadata format are available from the "StratusLab Marketplace Technical Note" [6]

A prototype Marketplace[3] has been made available for testing. The prototype already provides many of the expected features of the Marketplace including viewing and searching of metadata entries, direct querying of the metadata database, and browser-based upload of metadata files.

There are two primary use cases for images in the cloud:

1. creating new images and making them available and
2. instantiating a referenced image

5.4.1 Registering a New Image

New machine or disk images can either be created from scratch following the StratusLab guidelines or be modified versions of existing images. The workflow for creating an image and making it available involves the following steps:

Image Creation A new machine or disk image is created from scratch or from an existing image. The *stratus-create-image* command can facilitate this process.

Transfer to Storage The new image is transferred to network accessible storage. This may be storage provided by a StratusLab cloud or storage provided by the user or users' institute. The physical image must be available via one or more URLs.

Create Metadata The metadata entry for the image must be created. This is an XML file containing information about the image itself including the location URLs for the physical image. The *stratus-build-metadata* command may be helpful here.

Sign Metadata All of the metadata entries must be signed with a valid grid certificate. The *stratus-sign-metadata* will sign the metadata file with a given certificate, inserting the certificate information automatically into the metadata file.

[3] http://appliances.stratuslab.eu:8081/

Table 5.1: *Ways to Reference a Marketplace Image*

image identifier	MMZu9WvwKIro-rtBQfDk4PsKO7_)
Marketplace URL	http://mp.example.org/metadata/ ?id=MMZu9WvwKIro-rtBQfDk4PsKO7_
Direct or indirect URL to a valid metadata entry	http://tinyurl.com/my-favorite-image
URL for the physical location of the image	http://example.org/myimage

Upload Metadata The signed metadata entry can then be uploaded to the StratusLab Marketplace. This is done via an HTTP POST to a the Marketplace's http://mp.example.org/metadata/ URL, or by using the *stratus-upload-image* or *stratus-upload-metadata* commands.

The Marketplace will then validate the entry and verify the email address given in the image metadata. Once all of the checks have been done, the metadata will be visible in the Marketplace and other users can search for the entry.

5.4.2 Using a Registered Image

Referencing an existing image and instantiating it involves the following steps:

Request New Instance StratusLab users will create new machine instances through the *stratus-run-instance* command. Disk images can be used in the machine description to attach existing, public disks to a new instance. In both cases, the user can supply an image reference in a number of formats (see Table 5.1).

Resolve Entry The URL given by the user must be resolved into one or more metadata entries associated with the image. For any of the URLs that provide the metadata entry (or entries) directly, this is a simple HTTP GET on the URL. For location based URLs, this will involve a search in the Marketplace either on the location URL or on a checksum calculated on the fly.

Apply Site Policy The command *stratus-policy-image* will apply a configurable site policy to the image metadata. If none of the metadata entries for a particular image pass the site policy, then the image will not be instantiated.

Download Assuming that the image passed the site policy, then the *stratus-download-image* can be used to download a copy of the image based on the location URLs in the metadata entry.

Once the image is available, it will be instantiated and run. All of the work on the cloud side is expected to happen in the clone hook of OpenNebula.

6 Support Services

6.1 Overview

The infrastructure services offered by WP5 for internal and external users (i.e. the reference cloud service, the appliance repository, the pre-production testbed etc) have to be coupled with proper support in order to ensure that the users are taking full advantage of their capabilities. For this reason much of the day-by-day work within WP5 has to do with the provision of user support where the term ”user” refers to either system administrators that are using the StratusLab distribution or reference cloud users that exploit the IaaS cloud service. Also since WP5 is the resource provider of the project it is natural that it will contribute to the internal processes of the project that require access to physical resources. Such processes are for instance the software build and integration services, the hosting of software repositories, etc.

6.2 Infrastructure support services

6.2.1 Project Services

The project's web server, nexus repository (maven-generated artifacts), and yum repository (software packages in RPM format) are hosted on a machine at LAL (Class LAL1, see Table 3.6). This machine is regularly backed-up to ensure that the configuration and data for these critical services are not lost. Recently, this machine has also been used to host a test Markeplace instance for the HEPiX community, which is evaluating how the Marketplace could be used for sharing trusted machine images between grid resource centers.

The project's LDAP server, JIRA instance, and git repository are running on instances in the Amazon cloud. These are backed up regularly via the disk snapshotting mechanism provided by Amazon. As the StratusLab distribution becomes more feature-rich (specifically with persistent disk management and IP reservations), these services will be hosted directly within the StratusLab reference infrastructure.

6.2.2 Build and Test Services

LAL provides two machines (Class LAL2, see Table 3.6) that are used via the Hudson continuous integration server [1] to build and to test the StratusLab software. These machines are controlled by Quattor ensuring a stable, known base operating system configuration; moreover, they are reinstalled nightly to ensure that side effects from the hudson tests do not affect future tests.

LAL also provides two machines (Class LAL2) for regularly installing and testing the Quattor configuration of the StratusLab distribution. As for the build system, a clean installation is done nightly to ensure that clean installations function correctly.

6.3 Quattor Configuration Support

Quattor is a site management tool that allows for the automated installation, configuration, and maintenance of a resource center's resources. It is widely used by sites within the European Grid Infrastructure (EGI) for deploying grid services. LAL has created and maintains a Quattor configuration that allows the installation of a complete StratusLab cloud, similar to that created with the scripted installation with the *stratus-install* command. This configuration has evolved with the StratusLab distribution. The most difficult issue encountered has been the switch from CentOS 5.5 to Fedora 14 as the base operating system for the project. Fedora 14 requires many changes in the configuration modules for standard services and for the generated kickstart files. These are generic changes needed within Quattor to support more modern RedHat-based distributions, but the StratusLab effort is pushing this evolution forward rapidly. As for previous releases, the StratusLab 1.0 release should have a complete, working Quattor configuration. In addition to the maintenance of the Quattor configuration, the people involved also support the system administrators who are using Quattor to install the StratusLab software. The load so far has been light but is expected to increase markedly after the 1.0 release.

6.4 Support to early adopters of StratusLab distribution

The StratusLab distribution attracted during the first year of the project significant interest from external users and institutes who decided to download the software and try to deploy their own private cloud service. The distribution itself is publicly available from the project's software repository. Access is provided either from the source files using the git repository[2] or pre-packaged from the yum repositories[3]. Support for the installation process is provided either from the relevant sections in the project wiki or directly using the support mailing list support@stratuslab.eu.

[1] http://hudson.stratuslab.eu:8080
[2] http://stratuslab.eu/doku.php/git
[3] http://yum.stratuslab.eu/

Table 6.1: *Reference Cloud Service User Statistics*

Number of internal users	12
Number of external users	26
Total number of VMs instantiated by internal users	1900
Total number of VMs instantiated by external users	267
Number of external projects supported	8
Number of countries represented	10

Messages to this mailing list are directed to a selected number of StratusLab's members that have agreed to contribute with their expertise regarding cloud service installation, operation and usage.

6.5 Reference cloud service support

As mentioned, StratusLab offers a reference cloud service which is open for access for anyone interested to get a first hand experience of the cloud solutions integrated by the project. In order to request access anyone interested has to send an email to support@stratuslab.eu providing full contact details and a short description of the purpose of usage. Upon first contact the user is given an option to select between username/password or certificate based authentication. According to the method of choice the respective account in created in the service frontend and the connection details are send back to the user. Additionally the user is notified about the acceptable usage policy of the service which actually is the same with the one from EGI[4].

Users may contact the StratusLab support team for additional information, problem reporting or feature request. The list of users is retained in an off-line database and is used by StratusLab operations for announcement of significant events like service unavailability, service upgrade or security incidents.

So far the service has proved to be very popular among third parties attracting a total of 26 external users during the first six months of provision. Table 6.1 summarizes the service usage statistics.

[4] https://documents.egi.eu/public/ShowDocument?docid=74

7 Testing and Benchmarking

7.1 Overview

As already mentioned in previous sections, testing is an important part of the WP5 activities. WP5 is the first "user" of the software developed or otherwise integrated within WP4 and WP6. As such it has been assigned with the responsibility to be the first to try and verify the maturity of the software produced. Typically the pre-release versions of the software will exhibit various problems that have to be reported back to the developers thus a long circle of interaction is usually foreseen before the software is accepted for public release. Testing activities are coupled with benchmarking processes that aim to quantify the benefits of the developed cloud solutions.

7.2 Benchmarking

The project has developed a set of benchmarks to measure the performance of various cloud configurations and to ensure that the system can handle standard scientific use cases. At the beginning of the project, these were based on direct use of OpenNebula and are now being re-engineered to use the StratusLab client. Now that complete test clouds are deployed as part of the continuous integration tests in Hudson, these benchmarks can be run systematically with each build. The re-engineering and creation of hudson jobs for the benchmarks are expected to be completed for the StratusLab 1.0 release.

The benchmarks have been chosen to be representative of the range of various scientific applications. They fall into four categories:

- CPU Stress Tests: Tasks with high-CPU requirements but little or no input and output.
- IO Stress Tests: These test the IO performance of the cloud. These include jobs with large data output, large data input, and both large input and output.
- Parallel Applications: Both shared memory (OpenMP) and multi-task (MPI) parallel jobs.
- Workflow: An application with orchestrated flow of individual tasks.

These tests use computing resources (CPU, IO bandwidth, memory) in ways that are representative of real scientific analyses, but are not real applications. This ensures that the results are representative without having a large number of dependencies on application-specific libraries.

The more complicated benchmarks blur the lines between IaaS and PaaS infrastructures, accordingly some cooperation with VENUS-C is expected in the evolution and generalization of these benchmarks in the latter part of the project.

7.3 Testing of StratusLab distribution

The infrastructure operations activity is the first "adopter" of the work done in the context of WP4 and WP6. As such it is natural to act as a beta tester of the software integrated or developed by the above activities. Also this task is formally included in the description of work for the activity and the process has been documented in D5.2 [3] "Infrastructure Operations Policy and Tools".

In the first year of the project the contribution of the activity in the testing and certification of the StratusLab distribution has been crucial in order to deliver a quality and robust cloud software. An important tool for this process is the pre-production infrastructure allocated for this purpose (described in detail in Section 3). The three pre-production nodes have been re-installed from scratch many times in order to emulate realistic cases of software deployment and identify potential problems and incompatibilities.

8 Conclusions

Overall this has been an exciting first year for the whole project and WP5 in particular. The establishment of a computing infrastructure from ground up is always a challenging task. The ability to put this infrastructure to usage fast and to be able to start providing high level services has been one of the main concerns of WP5 during the first months of the project. WP5 was able to cope with these demands and succeeded in progressing with and finally achieving most of the goals set for the first year.

Definitely, the highlight of this effort has been the establishment and provision of the reference cloud service, a production level IaaS cloud that has been used for a large range of activities and from users inside and outside of the project. The reference cloud is currently the tip of the iceberg for StratusLab, providing visible access to the overall results of the project activities. Coupled with the appliance repository, it has allowed people to have access to a production quality IaaS cloud, has been used for developing real applications and for demonstrating StratusLab in various occasions.

The other significant success story was the certification of a grid site running on top of this cloud, as part of the EGI pan-European grid infrastructure, satisfying all qualitative requirements set forth by the latter. This activity has been important in order to prove that the deployment of grid services is indeed possible within cloud environments without having any technical or operational implications.

The above success stories have raised the expectations from WP5 and generally from the project for the second year. For the cloud services our goals are to further evolve and expand the provided infrastructure offering a complete range of IaaS cloud capabilities, including support for persistent storage, support for private IPs, and capability to deploy and manage virtual clusters. An important dimension that will be pursued is the provision of federated cloud resources, i.e. seamlessly utilizing cloud resources offered by two or more interoperable cloud resource providers. This approach appears to gain increasing attention from the european grid infrastructure community, posing as a very likely evolution of the current model for computing resource provision currently achieved from the usage of grid services. An extension of this capability is also the ability to perform cloud bursting over commercial cloud providers. The latter introduces also issues of cloud economics and raises interesting questions regarding optimal resource utilization since the usage of computing resources have in this case direct financial costs.

Finally, for what concerns the work on grid service integration, the focus in the second year will be to enhance their capabilities by exploiting the dynamic and elastic nature of the underlying cloud layer. This work will proceed with close collaboration with WP6 that is currently developing the relevant functionality exploiting the abilities of the Claudia service manager. Towards this we expect to dedicate considerable effort integrating this new functionality potentially deploying dynamic grid sites that can operate seamlessly within the EGI grid infrastructure.

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
CE	Computing Element
DCI	Distributed Computing Infrastructure
EGEE	Enabling Grids for E-sciencE
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
Front-End	OpenNebula server machine, which hosts the VM manager
IGTF	International Grid Trust Federation
Instance	see Virtual Machine / VM
LAN	Local Area Network
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
Private Cloud	Cloud infrastructure accessible only to the provider's users
Public Cloud	Cloud infrastructure accessible to people outside of the provider's organization
SAM	Service Availability Monitoring
SE	Storage Element
SSH	Secure SHell
TB	Terabyte(s)
Virtual Machine / VM	Running and virtualized operating system
VLAN	Virtual Local Area Network
VM	Virtual Machine
VO	Virtual Organization
Web Monitor	Web application providing basic monitoring of a single StratusLab installation

WebDAV	Web-based Distributed Authoring and Versioning
Worker Node	Grid node on which jobs are executed

References

[1] Haxx. cURL. http://curl.haxx.se/.

[2] The StratusLab consortium. Deliverable D5.1 - Infrastructure Operations. http://www.stratuslab.org/lib/exe/fetch.php?media=documents:stratuslab-d5.1-v1.0.pdf, 2010.

[3] The StratusLab consortium. Deliverable D5.2 - Infrastructure Tools and Policy Specifications. http://www.stratuslab.org/lib/exe/fetch.php?media=documents:stratuslab-d5.2-v1.0.pdf, 2011.

[4] The StratusLab consortium. Deliverable D6.3 -First Year Cloud-like Management of Grid Sites Research Report. To be published, 2011.

[5] The StratusLab consortium. Installing and operating a production grid site in the StratusLab cloud: Experience and issues. http://stratuslab.eu/lib/exe/fetch.php/documents:gridovercloud-v1.0.pdf, 2011.

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