Enhancing Grid Infrastructures with
Virtualization and Cloud Technologies

Report on the Detailed Evaluation of StratusLab Products

Deliverable D2.2 (V1.0)
15 June 2011

Abstract

This document provides a detailed, internal evaluation of the StratusLab v1.0 cloud distribution, providing feedback and informing the roadmap for the second year of the project. This document has evaluated the distribution in three areas: 1) use cases defined in the continuous integration system, 2) requirements and recommendations identified from user and system administrator surveys conducted at the beginning of the project, and 3) scenarios and requirements from the EGI User VIrtualization Workshop. The document identifies areas in which to concentrate efforts in the future. Notably, it reinforces the focus of the work plan for the coming year on issues related to federation of cloud infrastructures.

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.

Contributors

Name	Partner	Sections
Charles Loomis	CNRS-LAL	All

Document History

Version	Date	Comment
0.1	23 May 2011	Initial skeleton.
0.2	11 June 2011	Complete draft for internal review.
1.0	15 June 2011	Final draft.

Contents

List of Figures

1 Executive Summary

This document provides a detailed, internal evaluation of the StratusLab v1.0 cloud distribution. It is intended to provide feedback to the people within the project concerned with development and integration and to give them an clear view of where the StratusLab distribution stands with respect to identified requirements and priorities.

This document has evaluated the current StratusLab v1.0 cloud distribution in three areas: 1) use cases defined in the continuous integration system, 2) requirements and recommendations identified from user and system administrator surveys conducted at the beginning of the project, and 3) scenarios and requirements from the EGI User Virtualization Workshop.

A significant number of use cases are tested automatically and systematically via the project's continuous integration server–Hudson. These include the full user tutorial, which encompasses the virtual machine lifecycle and handling images through the Marketplace, as well as tests for the supported authentication methods, image creation, registration server, quota management, image policies, and application benchmarks. These need to be expanded to include:

- Performance benchmarks
- Tests of the persistent storage
- Tests of the service manager, Claudia
- Validation of all StratusLab-provided base, grid, and bioinformatics images

All of these will be added incrementally as we learn how to better parameterize and automate these tests.

The current distribution substantially meets the requirements and recommendations identified early in the project through user and system administrator surveys and enumerated in the D2.1 deliverable [7]. The areas which need further attention in the second year of the project include:

- Performance and scalability: As the project has done with testing use cases, it also needs to develop an infrastructure to run performance and scalability metrics and to track these as the distribution evolves.
- Storage: Services that provide storage must evolve into production-level services and expand to include file-based access as well.

- Network Services: These need to expand to provide better sandboxing of virtual machines, particularly via dynamic VLANs and dynamic firewalls.
- Commercial users: More effort needs to be made to contact commercial users to ensure that their needs are met and to see if they are interested in adopting the StratusLab distribution.

Making improvements in these areas will make future versions of the StratusLab cloud distribution even more appealing to users and system administrators.

Given the importance of the EGI community for the widespread use of the StratusLab distribution and ultimately its sustainability, it is worthwhile to evaluate the current StratusLab distribution in terms of EGI's scenarios for integrating virtualization technologies and requirements emerging from the breakout sessions of the User Virtualization Workshop (12-13 May 2011) for monitoring, accounting, virtual machine management, and information services.

Generally, the User Virtualization Workshop broadly validated the existing work plan of the project. The StratusLab distribution already satisfies a significant number of stated requirements and handles the "standalone" scenarios involving the execution of simple virtual machines.

The scenarios and requirements that StratusLab does not yet satisfy revolve around federating resources–the major topic for the second year of the project. This workshop has provided a concrete set of tasks in the areas of accounting, monitoring, and publishing of information, which will allow the StratusLab to support federated cloud deployments.

Overall, the distribution responds well to the identified use cases and requirements. Areas which need improvement had already been identified and tentatively added to the project's second year work plan. This evaluation will help solidify that work plan leading into the second year of the project.

2 Introduction

This document provides a detailed, internal evaluation of the StratusLab v1.0 cloud distribution. It is intended to provide feedback to the people within the project concerned with development and integration and to give them an clear view of where the StratusLab distribution stands with respect to identified requirements and priorities. This document does *not* provide an external evaluation of the distribution by current users (researchers and system administrators), which will be done with the survey in the next WP2 deliverable due in PM14.

After a brief description of the services and other products (e.g. prepared appliances) provided with the StratusLab v1.0 cloud distribution, the document enumerates the use cases and tests that are routinely run through the project's continuous integration system–Hudson. These provide continuous evaluation of the distribution as it evolves and ensure that it continues to satisfy the core use cases. By design, these test features that have been implemented in the distribution. After this, the distribution is evaluated against two sets of criteria: requirements identified in the initial user and system administrator surveys [7] and scenarios identified by the European Grid Infrastructure as necessary for integrating cloud and virtualization technologies on that e-infrastructure.

The document concludes with a summary of the evaluation, identifies gaps compared to the stated requirements, and suggests priorities for further development.

3 Overview of StratusLab Distribution

The StratusLab v1.0 cloud distribution provides compute, storage, and networking services necessary for an "Infrastructure as a Service" (IaaS) cloud. It also provides services that allow sharing between users and sites. For example, the Marketplace allows sharing of virtual machine and disk images and the authentication system allows the use of federated identities. The release also includes a set of "base" images (virtual machine images with minimal operating systems for ttylinux, CentOS, and Ubuntu), grid service images, and some customized bioinformatics images.

The description here is intended only to provide a cursory overview of the StratusLab v1.0 cloud distribution and associated products. A full description can be found in the "StratusLab Toolkit 1.0" deliverable D4.2 [8] and on the project's website. An overview of the StratusLab services can be found in Figure 3.1.

3.1 Compute Services

Virtual Machine Manager OpenNebula [5] serves as the virtual machine manager for the StratusLab distribution. The version packaged with the distribution is based on the OpenNebula 2.2 version with some StratusLab-specific patches and enhancements. An authentication proxy permits a wide range of authentication methods to be supported.

Service Manager Claudia [9] provides the ability to manage services–ensembles of machines acting together–as a single unit. It also allows rules to be defined based on monitoring metrics that allow the system to perform autoscaling. A wide range of authentication methods is also supported for Claudia via a similar authentication proxy.

3.2 Storage Services

Persistent Storage Service A prototype service is included in the distribution that allows users to create and use persistent disks. These have a life cycle independent of that of a virtual machine and allow the persistent storage of service state or user data.

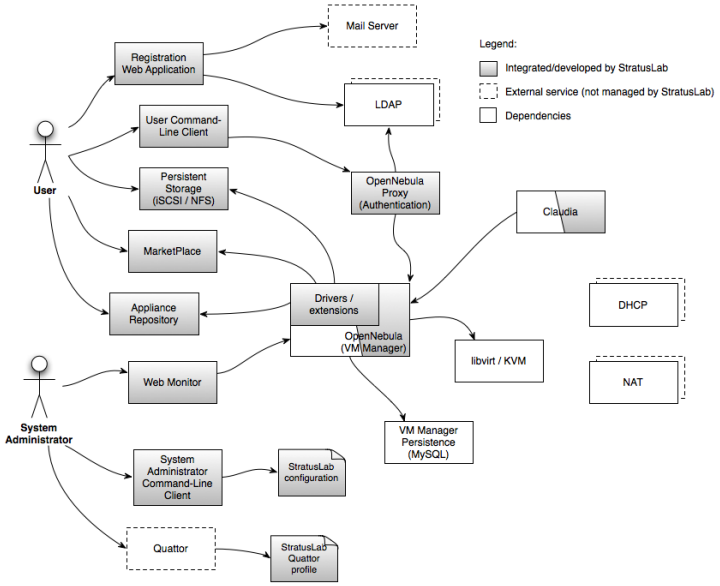

Figure 3.1: *Overview of StratusLab v1.0 Services*

3.3 Networking Services

The networking services are not directly visible to end users. Instead users launching virtual machines can specify an IP address from one of three classes:

- Public: An address visible from the internet. Appropriate for user-level services.
- Local: An address visible only within the cloud. Outgoing external internet access is done via Network Address Translation (NAT). Appropriate for parallel jobs like those using MPI.
- Private: An address visible only within the physical machine. As for local addresses, all outgoing external internet access is through NAT. Appropriate for workers in master/worker frameworks in which the worker contacts the master to obtain tasks and to return computed results.

Additionally, users may specify that a specific address be allocated to a given virtual machine instance. This is necessary to support services which are secured via a host certificate keyed by the DNS name or IP address.

3.4 Supporting Services and Tools

User Command Line Client The distribution includes a Command Line Interface (CLI) written in python. This allows users to access the cloud services provided by the StratusLab distribution. The client is very portable and easily installs and runs on Windows, Mac OSX, and Linux.

System Administrator Command Line Client The distribution also includes a CLI for system administrators and allow a scripted installation of the StratusLab front-end and hosts as well as tools for testing the installation.

Web Monitor A simple web interface that allows a system administrator to view the current state of the StratusLab cloud.

Registration Service A web service that allows users to register for using a StratusLab cloud. User information is kept in an LDAP server that can then be used by the StratusLab authentication systems to identify users.

3.5 Appliances

Marketplace This service provides a registry for metadata information about virtual machine and disk images. It allows such images to be shared between users. The metadata also provides information that allows system administrators to judge the trustworthiness of requested images.

Appliance Repository This provides storage for virtual machine images. This will eventually be phased out in preference to cloud storage services.

Base Images The project currently provides a set of virtual machine images containing minimal installations of common operating systems: ttylinux, CentOS (a RedHat derivative), and Ubuntu.

Grid Appliances A set of machine images are provided that allow a complete gLite grid site to be installed on a StratusLab cloud. The images require that valid host certificates be provided through the contextualization for secured services. The grid services must be configured as with services running on a physical host.

Bioinformatics Images Two customized images have been created for the bioinformatics community: one containing common analysis software and another providing an interface to common databases.

4 Continuous Evaluation

A core tool that supports the agile development process adopted by the project is the Hudson [6] continuous integration server. The allows the code base to be validated after each change is committed. Continuous integration jobs are often limited to unit and simple functional tests. StratusLab, however, has also defined jobs which install the StratusLab distribution from scratch and run complete functional tests against the installed system, including a large number of identified use cases. All of the activities have contributed to these tests.

As this document deals with the evaluation of the distribution, the jobs which test use cases are described. As these are part of the software process, the release by definition satisfies these use cases. Future plans regarding jobs to test use cases are also described.

4.1 Existing Tests

Jobs have already been defined in the Hudson server to test a number of common use cases. The jobs themselves can be found in the Fedora 14 view [1] of the Hudson server. In the descriptions of the jobs below, only the name of the job is given.

cloud_Test_tutorial_Fedora14 This job tests the standard virtual machine lifecycle and interactions with the Marketplace. For the lifecycle it tests the stratus-run-instance, stratus-describe-instance, and stratus-kill-instance commands by running the base ttylinux image. Before killing the machine, the test ensures that it can be successfully pinged. The commands tested for the Marketplace are stratus-build-metadata, stratus-sign-metadata, stratus-validate-metadata, and stratus-upload-metadata. The Marketplace is then queried to ensure that the uploaded metadata description exists.

cloud_Test_smoke_Fedora14 This job runs a subset of the tests defined via the stratus-test command. There are three tests to check that virtual machines have the right type of requested IP address (public, local, or private), a test to ensure that the CPU quota enforcement works, a test to ensure the registration of a new node with OpenNebula works, and a test that uploads to the Appliance Repository works.

cloud_Test_user_Fedora14 During this job the StratusLab client is installed and each of the commands is run with the "–help" option to ensure that there are no

[1] http://hudson.stratuslab.eu:8080/view/Fedora14/

problems with referenced modules and that the help option exists.

cloud_Test_smoke_createImage_Fedora14 This job tests the stratus-create-image command that takes a reference base image, adds requested packages, runs a defined configuration script, creates a new machine images, and registers this image.

cloud_Test_registration_Fedora14 This test ensures that the basic functionality of the Registration Service works. All of the static pages must exist. It tests the registration of a new user with both a username/password pair and a certificate Distinguished Name (DN). It then tests that changes to the user record are possible and that the password resetting workflow is correct.

cloud_Test_Policy_Fedora14 System administrators can define a policy that determines whether a given machine image is trustworthy enough to run on their clouds. The system uses the metadata associated with the entry in the Marketplace. This test creates metadata entries and evaluates them against a range of policies to ensure that the system accepts or rejects the associated images correctly.

cloud_Test_marketplace_Fedora14 This job tests the commands stratus-build-metadata, stratus-sign-metadata, stratus-validate-metadata, and stratus-upload-metadata, like the tutorial job described above. It additionally also restarts the Marketplace and retrieves the uploaded metadata entry to ensure that the information is persistent.

cloud_Test_ldapAuthentication_Fedora14 This test uses the user entry created in the registration test described above. The test ensures that the stratus-describe-instance *fails* with an authentication error because the user has not been added to the cloud-access group. It then adds the user to the cloud-access group and confirms that the stratus-describe-instance command *succeeds* using both authentication methods.

cloud_Test_Benchmarks_Fedora14 This test ensures that the defined application benchmarks run. These test a range of common scientific application patterns.

4.2 Future Plans

The defined tests systematically verify a large subset of use cases for the StratusLab cloud distribution. However, these tests must be expanded to include even more of the functionality provided. Missing use cases include:

- Performance benchmarks
- Tests of the persistent storage
- Tests of the service manager, Claudia
- Validation of all StratusLab-provided base, grid, and bioinformatics images

All of these will be added incrementally as we learn how to better parameterize and automate these tests.

From the above descriptions of the tests, some duplication of tests and a mixture of server-side and client-side tests exist. The existing tests need to be refactored to separate clearly the server-side and client side tests and to remove duplication between the tests. The client side tests must also be run multiple times to test the various supported authentication methods as well as the various installation methods (i.e. via a tarball and via an RPM package).

5 Previously Identified Requirements

Near the beginning of the project, the WP2 work package surveyed both researchers and system administrators to determine their experience with cloud technologies, their requirements for a cloud distribution, and the relative priorities of those requirements. The survey, raw results, and analysis can be found in the D2.1 deliverable [7].

That document concluded with a list of 25 requirements and recommendations based on the analysis of the surveys. Here we revisit them and evaluate StratusLab v1.0 cloud distribution and project work plan with respect to those requirements and recommendations.

The following section lists them (in italics) and provides an analysis for each one. Important gaps are identified and listed in the concluding section of this chapter.

5.1 Requirements and Recommendations

- *Although the research community is the primary target of the project, the project needs to make a stronger effort in contacting commercial enterprises.*

  This is a continuing concern for all work packages within the project. Contact with other national and European projects as well as with the academic community using EGI remains strong. However, contacts with industrial and commercial entities outside of those in the project are weak or nonexistent. A concerted effort is needed in this area in the second year of the project.

- *The project should support installation of the cloud distribution on RedHat and Debian systems, with RedHat systems having a much higher priority.*

  The project initially selected CentOS 5.5 and Ubuntu 10.04 for the standard supported operating systems, because they have long-term support and cover the two major branches in the linux world. Recently the project switch from CentOS 5.5 to Fedora 14 to avoid having to work around problems associated with KVM and the older kernel in CentOS 5.5. Fedora 14 is an RPM-based system that feeds into the RedHat distributions. All of the StratusLab services install and function on Fedora 14.

  Many of the services also build and install on Ubuntu; however, there is no systematic testing of this and packages for Ubuntu are not routinely pro-

duced. Moving forward from StratusLab v1.0, an emphasis should be placed on achieving the same level of support on Ubuntu as currently exists for Fedora 14.

- *Integration of the cloud distribution with automated site configuration and management tools should be demonstrated with Quattor and/or Puppet, with Quattor being the more popular.*

  Given that Quattor is more widely used in the European Grid Infrastructure than Puppet and that the StratusLab partners have experience with it, Quattor was chosen to demonstrate the compatibility of the StratusLab distribution with site configuration and management tools. The project maintains the Quattor-based installation in parallel with the scripted manual installation via the stratus-install command.

  However, testing of the Quattor installation still requires a significant amount of manual intervention. This should be automated to ensure the same level of confidence in the Quattor installation method as with the stratus-install method.

- *Demonstrations of grid services over the cloud should initially target core services of the gLite middleware.*

  WP5 has deployed a complete, certified grid site on top of the StratusLab distribution, demonstrating the ability of the cloud to support grid services. The deployed services include the Computing Element (CE and WorkerNode), Storage Element, BDII (information system), APEL (accounting)–the core site services.

- *The cloud distribution must supply stock images for popular Red-Hat and Debian-based systems.*

  The project produces and maintains base virtual machine images for:

  - ttylinux: A small linux distribution ideal for testing
  - CentOS: A RedHat-derived operating system used by many communities
  - Ubuntu: A Debian-derived operating system also used by many communities

  This will be expanded to include also Scientific Linux (on which the grid images are based) and OpenSuSE.

  Although this requirement is satisfied by the project, it would help to automate the production of these images so that they can be easily kept up-to-date with security patches released by the maintainers.

- *The cloud infrastructure must be as operating system neutral (with respect to running virtual machines) as possible to maximize its utility.*

The StratusLab distribution places minimal constraints on the operating systems running in virtual machines. They must simply mount an ISO image with the contextualization information and use that information to configure the system. Network information is found using DHCP, which has nearly universal support in common operating systems. Coupled with the use of hardware-supported virtualization, the constraints pose no real hurdles for running a wide range of operating systems within virtual machines.

- *The application benchmarks must cover all of these types of applications: sequential, multi-threaded, shared memory, and parallel.*
- *The application benchmarks should include workflow and master/worker applications.*
- *The application benchmarks must be parameterized to allow a wide range of input sizes, output sizes, and running times to be evaluated.*

  These three requirements are all satisfied by the current StratusLab benchmarks. These are all routinely tested via the continuous integration server.

- *The StratusLab cloud implementation must include access control mechanisms for stored data and must permit the use of encrypted data.*

  Storage services in the StratusLab distribution only cover a persistent disk services and then only at a prototype level. No access control is provided and there is no support for encryption. More work needs to be done to satisfy this requirement and on the storage services in general.

- *The cloud must allow both file and block access to data, although file access is by far more important.*

  The primary target user community, the existing EGI community, already has access to file-based access through the grid services. Consequently, work concentrated instead on a disk-based (block access) abstraction to storage. The prototype persistent disk service provides an implementation, which will obviously need to evolve into a production service.

- *The cloud must allow access to data stored in object/relational databases.*

  The StratusLab distribution permits outbound internet access to all running virtual machines and thus has no impediments for machines accessing object or relational databases. With the prototype persistent disk service, it is also possible to host such databases within the cloud.

- *Short-term work (<12 months) should concentrate on developments for deploying cloud infrastructures and longer-term work should concentrate on their use.*

  Two methods for installing, configuring, and maintaining a StratusLab cloud are currently supported: script-assisted manual installation and Quattor-based

installation. They are both a mature part of the distribution and attention has turned to the systematic testing of a wide range of use cases.

- *The StratusLab distribution must be simple enough for users themselves to configure their own resources as a cloud.*

  The script-assisted manual installation is intended to make the installation of a StratusLab cloud as simple as possible. However, the distribution requires a specific network setup to support the three classes of network IP addresses (public, local, and private). Researchers often to not have the required access to the computing infrastructure to do this configuration themselves. Nonetheless, if such access is possible, the complete installation of StratusLab is fairly simple.

- *The StratusLab distribution must allow both full-virtualization and para-virtualization to be used.*

  The virtual machine management services in StratusLab revolve around OpenNebula, which supports of a range of popular hypervisors (e.g. KVM, XEN, and VMware) which in turn, support both full- and para-virtualization. Although both are supported by OpenNebula, the project has only used full-virtualization and the supplied machine images require full-virtualization.

  If the project wants to support para-virtualization, then such configurations should be supported by the installation tools and should be tested via the continuous integration processes of the project.

- *The cloud service must have a command line interface and a programmable API.*

  The project provides a command line interface (CLI) for the core part of the distribution. Exceptions include the prototype persistent disk service and the registration service. The project plans to include commands for the prototype services as they evolve into production services. Interactions with the registration service are expected to happen through the web interface, so a CLI is not really useful.

  All of the service APIs are either based on REST or XML-RPC and are simple enough to be used directly. The existing StratusLab commands (stratus-*) are written in Python and provide an example for accessing the services. No explicit programmable API has been provided.

- *The cloud distribution must allow a broad range of grid and standard services to be run.*

  The running of a certified EGI grid site over the StratusLab cloud demonstrates the versatility of the cloud platform. The sole issue with the distribution is the support for persistent storage (for user data and for service state), which exists as only a prototype in the v1.0 release.

- *Quantitative performance evaluations must be done to understand the penalties in using virtualization.*

  The project has not done any systematic evaluations concerning the performance penalties of virtualization. It has instead relied on general external studies which show that CPU and network penalties are negligible (order of a few percent) but that disk IO can have a significant penalty of 10-20% depending on the configuration. Work within the project has shown that shared file systems like NFS are a bottleneck and has investigated alternatives such as Ceph and GlusterFS, which have unfortunately not provided an adequate solution.

- *The project must determine the criteria by which administrators and users can trust machine images.*

  The current design of the Marketplace as a registry of virtual machine and disk image metadata grew out of this requirement. The StratusLab distribution allows system administrators to define a policy for evaluating and trusting requested images based on the image metadata in the Marketplace. Administrators can use any of the information in the metadata, such as the endorser of the image, the operating system, version of the operating system, and network requirements. The system is easily extensible to allow arbitrary policies to be implemented.

- *The project should consider all features listed in the surveys as valid requirements.*

  The requirements listed in the surveys covered the full range of "Infrastructure as a Service" (IaaS) services as implemented by Amazon and other commercial providers. The project does consider all of those requirements as valid and is working to implement as many as possible given manpower and time constraints.

- *Integration with site management tools is a critical short-term requirement.*

  The Quattor-based installation was developed in parallel with the distribution itself, ensuring that the distribution remained compatible with site management tools. The largest problem with this support was the switch from CentOS to Fedora 14 which required significant changes to the configuration and some changes within Quattor itself.

- *The cloud implementation must scale to O(10000) virtual machines.*

  The project has not worked on testing the scalability of the distribution. With the release of v1.0, this, along with performance measures, will become more important.

- *The implementation must sufficiently sandbox running machine images to prevent unintended or malicious behavior from affecting other machines/-tasks.*

Placement of virtual machines on the cloud infrastructure uses the resource requirements defined when starting the machine. At this level, there is little room for interference between the running virtual machines, except if there is a significant over-subscription of CPU resources. The machines, however, do not run within a dedicated VLAN, so there can be interference at the network level between machines. Future work on the networking (dynamic VLANs and dynamic firewalls) should reduce this interference.

- *The project must create application benchmarks (CPU-Intensive, Simulation, Analysis, Filtering, Shared Memory, Parallel, and Workflow) to measure quantitatively the performance of the cloud implementation for realistic applications.*

  These benchmarks have been created. However, they are not yet run systematically to allow quantitative measurements on the performance of different types of applications or of different cloud configurations. This, as with the other performance measures, needs to become a higher priority in the second year of the project.

- *Performance benchmarks should also be created using packages like HEP-SPEC, Iozone, and iperf for CPU, disk IO, and network performance, respectively.*

  A few of these are used within the application benchmarks, but a complete set of performance benchmarks needs to be created.

5.2 Gaps

Generally, the project has done a good job in satisfying the requirements identified by the surveys and following the given recommendations. The areas which need further attention in the second year of the project include:

- Performance and scalability: As the project has done with testing use cases, it also needs to develop an infrastructure to run performance and scalability metrics and to track these as the distribution evolves.
- Storage: Services that provide storage must evolve into production-level services and expanded to include file-based access as well.
- Network Services: These need to expand to provide better sandboxing of virtual machines, particularly via dynamic VLANs and dynamic firewalls.
- Commercial users: More effort needs to be made to contact commercial users to ensure that their needs are met and to see if they are interested in adopting the StratusLab distribution.

Making improvements in these areas will make future versions of the StratusLab cloud distribution even more appealing to users and system administrators.

6 EGI User Virtualization Workshop

The European Grid Infrastructure (EGI) hosted its first User Virtualization Workshop on the 12-13 May 2010 in Amsterdam. EGI is investigating the use of virtualization and cloud technologies in order to improve the flexibility and efficiency of the current infrastructure and ultimately empower virtual research communities to control directly the environment they offer their users. StratusLab had a significant presence at the workshop.

The model for integrating virtualization and cloud technologies in the infrastructure that emerged from the discussions is very similar to the layered, grid-over-cloud architecture that is the basis of the StratusLab project. EGI was already identified as a critical "customer" for the StratusLab distribution; this workshop has only served to reinforce that view.

Given the importance of the EGI community for the widespread use of the StratusLab distribution and ultimately its sustainability, it is worthwhile to evaluate the current StratusLab distribution in terms of EGI's scenarios for integrating virtualization and requirements emerging from the workshop's breakout sessions for monitoring, accounting, virtual machine management, and information services. A summary [3] and more detailed minutes [2] are available for the workshop.

6.1 Scenarios

EGI defined a minimal set of six scenarios that provide a basis for use of virtualization and cloud technologies within the infrastructure and that promote an incremental and evolutionary transition from the current infrastructure. These six scenarios are defined in the "EGI Cloud Integration Profile" document [4]. Each of the following section describes one scenario and provides commentary of StratusLab's ability to satisfy the scenario.

6.1.1 Running a pre-defined VM image

This scenario describes a remote user being able to select a pre-existing image and launch a virtual machine instance from the selected image. The document identified three services necessary achieve the goal of this scenario: Management Interface, Authentication & Authorization, and Remote Network Access.

This is the most basic cloud use case describing the usual start, use, stop lifecycle of a virtual machine. StratusLab can fully achieve the functional requirements

of this scenario using credentials currently used on the European Grid Infrastructure.

However, the detailed description of the scenario also mentions the use of standard interfaces: OCCI for the cloud API and SAML/XACML for the authentication and authorization. These standards are not yet supported by the StratusLab distribution, although OCCI is a planned enhancement. The use of SAML/XACML seems a bit premature as even EGI itself does not use those standards.

6.1.2 Running my VM image (with my data)

This scenario expands on the first scenario by allowing the user to define her own VM image and use that on the infrastructure. Additionally, the image operates on data that resides outside of the image where the connection between the image and the data is specified when the VM is started. The document describes two additional required services: Data staging, Instance/image configuration.

StratusLab already allows users to define their own virtual machines and run them. It even facilitates this by automating the process with the stratus-create-image command. The distribution already stages images as necessary on the cloud infrastructure and a well-defined contextualization mechanism allows machines to be configured when instantiated.

If the data are fixed, then they can be put into a read-only disk image and managed through the Marketplace. Access to these data can be configured when a machine is instantiated, with the disk image handled similarly to virtual machine images by the infrastructure. Persistent data can be handled through the prototype persistent disk service, where again, the connection between the data and image can be specified when a machine is instantiated.

Currently disk and data staging are handled via the http(s) protocol, but it would be fairly straightforward to add others, including the GridFTP standard mentioned in the document. The document mentions also the CDMI and SRM protocols for data management. CDMI is being discussed within the project; SRM is unlikely to be directly supported.

6.1.3 Deciding which virtualized resource to use

EGI is composed of a collection of distinct resource providers. This scenario describes simple federation of those providers by having the capabilities of a resource center published. Users can then use the published information to select an appropriate provider. This is already done for grid resources; it must also be done for cloud resources. The document identifies an additional service: Service Description.

The current StratusLab distribution assumes that users have access to at most a few well-known cloud providers and become aware of each providers capabilities through “out-of-band” communication.

Satisfying this requirement necessitates having summary information of the hardware capabilities of the site (e.g. the maximum number of CPUs, amount of RAM, and network bandwidth permitted) as well as some information on the cur-

rent load of the site. All of this information is easy to collect and already available to some extent through existing tools. It also requires publishing this information, but this should be trivial to do if the grid information system is reused and an agreement can be reached on the information schema, which is likely to be based on the GLUE2 standard.

6.1.4 Accounting across resource providers

Accounting information from a cloud instance needs to be reported to a central accounting system or to multiple accounting systems, giving users a global view of their resource use. This scenario requires an accounting infrastructure to transport and correlate information and an standard format for reporting resource utilization.

Currently, StratusLab only provides limited accounting information that can be extracted from the OpenNebula database. Work must be done to provide accounting information for all types of resources and also to provide incremental accounting reports, as services may run for long periods of time. This work must be done independently of whether a cloud infrastructure is federated with others or not. The information must also be published in a format compatible with the overall accounting system, probably based on the OGF's Usage Record (UR) format.

Work within StratusLab will likely be limited to providing the accounting information in the correct format. Accounting and billing services will hopefully be provided through a collaboration with other projects, notably VENUS-C. We expect that the overall accounting framework to collect and correlate information from different sites will be provided by EGI itself.

6.1.5 Reliability/availability of the resource

Information relating to the reliability, availability, and current status of a remote virtualized resource must be available. This scenario requires monitoring of the cloud infrastructure, storing of monitoring information to analyze trends, and reporting aggregated information and metrics. The two additional services are Monitoring and Reporting.

StratusLab currently deploys Ganglia during the installation. This allows standard information about the physical machines to be collected. StratusLab has also developed Ganglia probes that also provide information about the running virtual machines. This information needs to be aggregated and analyzed to provide metrics on reliability and availability of a particular cloud infrastructure. Presumably, EGI would provide the infrastructure for publishing this information, probably through the standard grid information system.

6.1.6 State change notification from the VM manager

When the status of the instance changes, the user should be notified. This requires the deployment of a Notification infrastructure.

StratusLab does not provide any mechanism for users to receive notification of state changes of their machine instances. However, in the context of monitoring

and accounting a messaging system is being considered. If this is included and deployed, it would be fairly straightforward to add user notifications to the system.

6.2 Identified Requirements

During the workshop four breakout sessions were held (Monitoring, Accounting, Virtual Machine Management, and Information Services), allowing more detailed discussion of each topic. The summary presentation from each breakout contained a list of requirements related to the use of virtualization and cloud technologies on the infrastructure. (See the workshop agenda [1] for the presentations.) The sections below describe the identified requirements and provide commentary on how StratusLab does or could meet them.

6.2.1 Monitoring

The purpose of monitoring is to answer the question "Is the service functioning correctly?" There were three areas identified during the breakout session for monitoring: service level monitoring, external monitoring of the system, and security monitoring.

For service level monitoring standard site tools like ganglia and nagios can be used to determine if the (cloud) service is operating correctly. This can be verified via external monitoring where an outside agent launches virtual machines and ensures that they start correctly. This is akin to the current job-based monitoring carried out by the EGI nagios infrastructure. The conclusion was that this infrastructure should be reused for virtualization and cloud services.

For security monitoring, it was less clear what should be done. Virtual machines are essentially just applications and no special security monitoring is done for the application on the grid now. The recommendation was to inform ourselves concerning best practices and determine if something more needs to be done on a virtualized infrastructure.

For StratusLab, the principal need is to provide a mechanism for monitoring the cloud services running on the infrastructure. Ganglia is already in place for this, although the project should consider developing probes to determine the health of each StratusLab service.

6.2.2 Accounting

The accounting session identified that at least compute, storage, and network use should be tracked and reported. The following are important points from the summary presentation:

- The footprint (duration $\times$ number of cores) of the virtual machine should be accounted for, rather than the CPU utilization.
- The existing APEL infrastructure in EGI can be expanded to new record types.

- Information can be gathered from logs (e.g. like in OpenNebula or Eucalyptus).
- Extend the current OGF User Record standard for other types of resources.
- There are a range of implementation issues concerning identity management, normalization, fairness, etc. that are important but should not impede getting a reporting infrastructure in place.

Commercial providers already provide accounting and billing information, so it is clearly possible for it to be done in an academic setting.

For StratusLab, the important points are first, to ensure that accounting information can be provided by all of the StratusLab services and second, to participate in the discussions concerning the usage record format.

6.2.3 Virtual Machine Management

The virtual machine management breakout enumerated a list of requirements and prioritized them (high, medium, low):

- Deployment
  - Parameters to instantiate a single VM (high)
  - API should expose supported hypervisors (high)
  - Mechanisms for deploying VM landscapes (medium)
  - Specification of QoS (via SLA) (low)
- Management
  - Bulk operations (high)
  - State view (provider/user) (high)
  - Expiry and revocation of images (medium)
  - Snapshot taking (medium)
- Security
  - Support traditional (X.509/VOMS), but consider others (SAML, Shibboleth, eduGAIN) (high)
  - Provider should be able to understand which running VM is based which image (for revocation) (medium)
- Capacity planning
  - Scheduling capabilities in the interface (low)
  - "How long will it take from request to instantiation" (both for single and bulk submissions) (low)

Concerning the deployment requirements, StratusLab already allows VM instances to be parameterized both in terms of required resources and by providing a contextualization mechanism for customizing a particular instance. StratusLab currently has only tested KVM, but other hypervisors supported by OpenNebula should be easy to use; exposing the supported hypervisors at a site would be trivial to do. The other two requirements are lower priority and less clear in terms of implementation.

For the management requirements, Claudia provides a mechanism for the bulk deployment of virtual machines. The web monitor provides a system administrator with an overview of the machines running on the infrastructure; users can get an overview through the stratus-describe-instance command. Use of Marketplace metadata allows images to expire and be revoked, although active revocation would require an additional daemon that monitors and kills running instances based on revoked images. The stratus-create-image command allows limited snapshotting of images.

For the security requirements, StratusLab already satisfies both requirements. Standard grid credentials can already be used and the service can easily be extended to support other types of authentication. The image used to start a machine is easily gathered from the logs, although this should be made more apparent from the command line interfaces as well.

StratusLab does not intend to introduce queueing semantics into its cloud distribution. Doing so goes against the way clouds are intended to work and introduces many complications. However, it will be important to understand how the illusion of "infinite capacity" can be preserved on (very) finite resources.

6.2.4 Information Services

The information services discussion was more wide ranging and less focused than the other breakout sessions, largely because the information system is closely related to many other concerns: accounting, monitoring, state information, etc. In the end, a set of priorities was presented:

1. Determine what capabilities need to be represented in the information model? (compute, storage, and network)

2. Work on the 'transport of information' via useful systems

3. Understand overlaps with monitoring and set the boundaries

This set of priorities also provides a rough roadmap of the work necessary to integrate cloud services in an information system.

For StratusLab, we need to be very practical in this area, publishing a minimal amount of information about the cloud services into the existing grid information system. Once this is accomplished, then the information can be expanded as necessary to include more details about the capacities of the services.

6.3 Summary

Overall, the User Virtualization Workshop broadly validated the existing work plan of the project. The StratusLab distribution already satisfies a significant number of stated requirements and handles the “standalone” scenarios.

The scenarios and requirements that StratusLab does not yet satisfy revolve around federating resources–the major topic for the second year of the project. This workshop has provided a concrete set of tasks which will allow the StratusLab to support federated cloud deployments.

7 Summary and Conclusions

This document has evaluated the current StratusLab v1.0 cloud distribution in three areas: 1) use cases defined in the continuous integration system, 2) requirements and recommendations identified from user and system administrator surveys conducted at the beginning of the project, and 3) scenarios and requirements from the EGI User Virtualization Workshop.

A significant number of use cases are tested automatically and systematically via the project's continuous integration server–Hudson. These include the full user tutorial, which encompasses the virtual machine lifecycle and handling images through the Marketplace, as well as tests for the supported authentication methods, image creation, registration server, quota management, image policies, and application benchmarks. These need to be expanded to include:

- Performance benchmarks
- Tests of the persistent storage
- Tests of the service manager, Claudia
- Validation of all StratusLab-provided base, grid, and bioinformatics images

All of these will be added incrementally as we learn how to better parameterize and automate these tests.

The current distribution substantially meets the requirements and recommendations identified early in the project through user and system administrator surveys and enumerated in the D2.1 deliverable [7]. The areas which need further attention in the second year of the project include:

- Performance and scalability: As the project has done with testing use cases, it also needs to develop an infrastructure to run performance and scalability metrics and to track these as the distribution evolves.
- Storage: Services that provide storage must evolve into production-level services and expanded to include file-based access as well.
- Network Services: These need to expand to provide better sandboxing of virtual machines, particularly via dynamic VLANs and dynamic firewalls.

- Commercial users: More effort needs to be made to contact commercial users to ensure that their needs are met and to see if they are interested in adopting the StratusLab distribution.

Making improvements in these areas will make future versions of the StratusLab cloud distribution even more appealing to users and system administrators.

Given the importance of the EGI community for the widespread use of the StratusLab distribution and ultimately its sustainability, it is worthwhile to evaluate the current StratusLab distribution in terms of EGI's virtualization integration scenarios and requirements emerging from the User Virtualization Workshop (12-13 May 2011) breakout sessions for monitoring, accounting, virtual machine management, and information services.

Generally, the User Virtualization Workshop broadly validated the existing work plan of the project. The StratusLab distribution already satisfies a significant number of stated requirements and handles the "standalone" scenarios involving the execution of simple virtual machines.

The scenarios and requirements that StratusLab does not yet satisfy revolve around federating resources–the major topic for the second year of the project. This workshop has provided a concrete set of tasks in the areas of accounting, monitoring, and publishing of information, which will allow the StratusLab to support federated cloud deployments.

Overall, the distribution responds well to the identified use cases and requirements. Areas which need improvement had already been identified and tentatively added to the project's second year work plan. This evaluation will help solidify that work plan leading into the second year of the project.

Glossary

APEL	Accounting Processor for Event Logs (EGI accounting tool)
Appliance	Virtual machine containing preconfigured software or services
CDMI	Cloud Data Management Interface (from SNIA)
CE	Computing Element in EGI
DCI	Distributed Computing Infrastructure
DMTF	Distributed Management Task Force
EGEE	Enabling Grids for E-sciencE
EGI	European Grid Infrastructure
EGI-TF	EGI Technical Forum
GPFS	General Parallel File System by IBM
Hybrid Cloud	Cloud infrastructure that federates resources between organizations
IaaS	Infrastructure as a Service
iSGTW	International Science Grid This Week
KPI	Key Performance Indicator
LB	Load Balancer
LRMS	Local Resource Management System
MoU	Memorandum of Understanding
NFS	Network File System
NGI	National Grid Initiative
OCCI	Open Cloud Computing Interface
OVF	Open Virtualization Format
Public Cloud	Cloud infrastructure accessible to people outside of the provider's organization
Private Cloud	Cloud infrastructure accessible only to the provider's users
SE	Storage Element in EGI
SGE	Sun Grid Engine
SNIA	Storage Networking Industry Association
TCloud	Cloud API based on vCloud API from VMware
VM	Virtual Machine
VO	Virtual Organization
VOBOX	Grid element that permits VO-specific service to run at a resource center
Worker Node	Grid node on which jobs are executed

XMLRPC	XML-based Remote Procedure Call
YAIM	YAIM Ain't an Installation Manager (configuration utility for EGI)

References

[1] European Grid Infrastructure. Agenda of EGI 1st User Virtualisation Workshop. http://go.egi.eu/uvw1.

[2] European Grid Infrastructure. Minutes of EGI 1st User Virtualisation Workshop. http://go.egi.eu/535.

[3] European Grid Infrastructure. Summary of EGI 1st User Virtualisation Workshop. http://go.egi.eu/559.

[4] S. Newhouse and M. Drescher. EGI Cloud Integration Profile. https://www.egi.eu/indico/materialDisplay.py?materialId=1&confId=415.

[5] OpenNebula Project. OpenNebula. http://opennebula.org/.

[6] Oracle. Hudson. http://hudson-ci.org/.

[7] StratusLab. Review of the Use of Cloud and Virtualization Technologies in Grid Infrastructures. http://stratuslab.eu/lib/exe/fetch.php/documents:stratuslab-d2.1-v1.2.pdf.

[8] StratusLab. StratusLab Toolkit 1.0. http://stratuslab.eu/doku.php/deliverables.

[9] Telefónica. Claudia. http://claudia.morfeo-project.org/.