Enhancing Grid Infrastructures with
Virtualization and Cloud Technologies

Survey of Targeted Communities Concerning StratusLab

Deliverable D2.3 (V1.0)
12 August 2011

Abstract

The purposes of this document were 1) to update the user requirements gathered at the beginning of the project and 2) to identify concrete use cases to demonstrate the utility of the StratusLab cloud distribution, both through interactions with the scientific, engineering, and commercial communities targeted by StratusLab. Based on these interactions and feedback from the project's reviewers, a general strategy has been developed to guide our work with users in the second year of the project.

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

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

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

Contributors

Name	Partner	Sections
Mohammed Airaj	CNRS/LAL	All sections.
Christophe Blanchet	CNRS/IBCP	Bioinformatics use cases.
Charles Loomis	CNRS/LAL	All sections.
Thérèse Malliavin	Institut-Pasteur Paris	TOSCANI use case.
Henar Muñoz	TID	Commercial use case.
Michael Nilges	Institut-Pasteur Paris	TOSCANI use case.
Mariusz Sterzel	CYFRONET	Chemistry use cases.

Document History

Version	Date	Comment
0.1	6 July 2011	Initial skeleton.
0.2	10 August 2011	Complete version for review.
1.0	12 August 2011	Final draft after review.

Contents

List of Figures

1 Executive Summary

The purposes of this document were to 1) update the user requirements collected at the beginning of the project and 2) identify concrete use cases that demonstrate the utility of the StratusLab cloud distribution. The use cases are an important component of the project's overall strategy for increasing the number of users of the StratusLab distribution.

Unfortunately the survey created for this study, targeted at current users of the StratusLab reference cloud, had very few responses (5 of 30 current users). Consequently the results were useful neither for updating the user requirements nor for identifying new use cases. The project will continue to use the previously identified requirements to guide the program of work. The activity may conduct a limited number of interviews to better understand the users' experience with the StratusLab software.

A set of preliminary use cases, which will be the initial focus of the StratusLab porting activities, has been identified. There are seven use cases in total touching bioinformatics, computational chemistry, and software engineering. Integration with the European Grid Infrastructure has also been identified, as this provides an opportunity for greatly expanding the number of StratusLab deployments and contacts with communities that already use distributed computing resources.

The overall strategy for increasing the number and variety of users of the StratusLab infrastructure is to create a "virtuous cycle", starting with porting of a use case and publicizing it to generate more interest. The project will also continue with its hands-on tutorials to generate interest and to train people to use the technology. A "StratusLab User Workshop" will also be planned allowing current users to show their results and new users to receive one-on-one consulting for their applications.

The document provides more detail on these use cases. Readers can consult those descriptions to better understand the capabilities of the StratusLab cloud distribution and to guide their own use.

2 Introduction

The purposes of this document were 1) to update the user requirements gathered at the beginning of the project and 2) to identify concrete use cases to demonstrate the utility of the StratusLab cloud distribution. StratusLab participants interacted with the scientific, engineering, and commercial communities targeted by StratusLab in order to achieve these objectives. The survey of the targeted communities is described, followed by the identified initial use cases.

Based on these interactions and feedback from the project's reviewers, a general strategy has been developed to guide the work of the WP2 activity with respect to the StratusLab users in the second year of the project. This document describes that strategy and related events.

3 Updated Requirements

Two detailed surveys were conducted at the beginning of the project asking system administrators and users for their requirements concerning cloud technologies. The results of these surveys are available in the deliverable "Review of the Use of Cloud and Virtualization Technologies in Grid Infrastructures" [14]. The responses to this survey have informed the project's roadmap over the first year of the project. These requirements are still valid and will continue to guide the project during the second year as well.

Nonetheless, there was a desire to update these requirements focusing on the current users of the StratusLab reference infrastructure. Consequently, a survey was created to ask these users about their requirements, use cases, and experiences (to date) with the reference infrastructure.

As for the previous surveys, the Zoomerang [16] service was used to create and conduct the survey. The survey consisted of five sections:

- Background Information
- StratusLab Cloud Experience and Plans
- StratusLab Appliance Repository & Marketplace
- Application Characteristics & Use Cases
- Use Cases to be deployed and Encountered Problems

The detailed questionnaire can be found in the Appendix.

The survey was open for three weeks in July and August 2011, with the link being sent to the 30 users of the reference infrastructure. During this time there were 5 complete responses, 2 incomplete responses, and 18 unique views of the survey. This means that around 25 of the 30 users visited the survey, but most decided not to respond. As the number of completed surveys was extremely poor, no real conclusions could be drawn from the responses. Worse, none of the respondents provided a description of a use case.

The poor response rate could be explained by any number of factors: the holiday period, the survey was too long or complicated, or users do not have a scientific profile (hence scientific use case). In any case, the survey results were not useful for updating the user requirements or identifying new application use cases.

Because of this, the activity will take a more hands-on, individualized approach when interacting with the community. In addition to the tutorials run by the activity, application porting workshops will be organized that will provide one-to-one, hands-on consulting for porting identified applications to the StratusLab cloud.

The activity will also consider conducting phone or personal interviews with users of the reference infrastucture. Although only a small number of these could be done, they would likely identify places where the software and operations could be improved.

4 Use Cases

4.1 Introduction

Demonstrating a small number of use cases that highlight the capabilities of the StratusLab cloud and offer opportunities for "marketing" is an important task for the second year of the project.

This chapter identifies seven different use cases. They cover bioinformatics, computational chemistry, and software engineering. These are marked as either "internal" or "external" use cases. The internal use cases involve mainly people within the StratusLab project and can be scheduled more flexibly. External use cases involve collaboration with outside people and may have timing constraints based on their availability.

The descriptions below are preliminary and intended only to give an idea of the scope and target of the use case. Further refinements of these use cases will have to be done before work on them begins. Work will be planned through the normal sprint process and tracked via JIRA.

We will work with the dissemination work package so that a press release and other marketing activities are done at the conclusion of each use case. This will allow the project to take maximum advantage of the effort invested.

4.2 Bioinformatics Web Services (INTERNAL)

4.2.1 Context

Several experimental technologies have been improved to such a degree that obtaining data is easy, causing a deluge of data for the bioinformatics community. The challenge is to be able to analyze efficiently these data with the relevant applications. Many projects are working on the genome sequence of different organisms, continuously providing new sequences for analysis. Some bioinformatics algorithms for that analysis like BLAST, FastA or ClustalW are data-intensive, processing gigabytes of data stored in flat-file databases like UNIPROT, EMBL or PDBseq via a shared filesystem. Others like Abyss, BWA, or Ray are CPU- and memory-intensive.

The adoption of clouds for bioinformatics applications will be strongly correlated to the capability of cloud infrastructures to provide ease-of-use and access to reference biological databases and common bioinformatics tools. In the context of

the StratusLab project, two bioinformatics virtual appliances [13] have been built as a first step to fulfill the need for efficient analysis. However, these appliances are usable only through a remote shell display, potentially limiting wider use. More needs to be done.

4.2.2 Use Case

The goal of this use case is to create bioinformatics appliances containing the previous applications that scientists and engineers can deploy on demand.

Because bioinformatics applications require access to reference databases to process their analyses, these appliances will need POSIX access (like NFS) to the storage volumes in the cloud repository containing the biological databases.

Biologists and bioinformaticians are regularly combining multiple software packages to study their data via analysis pipelines. For a long time, they have been accustomed to accessing these tools from web portals not only for the ease of use but also for the power of composing these tools into a pipeline. This power clearly depends on the portal design itself; for several years, the focus has been on providing such composable services via web service technologies. This has been done at a pure Web Service Resource Framework (WSRF) level (e.g. European NoE EMBRACE [7, 4]) leading to online public services like the ones listed in the IBCP services repository[1] or at a level combining web services and grid facilities [1].

To meet researcher's expectations, these appliances must present a standard programmatic, public, web service interface, permitting users to combine the different bioinformatics methods in useful analysis pipelines.

4.2.3 Analysis

Cloud technologies provide scientists with the flexibility to deploy bioinformatics applications on different virtual machines. But clouds have to be connected with the existing public bioinformatics infrastructures. In that sense a cloud infrastructure should:

- Provide scientists with bioinformatics appliances to deploy on academic or commercial datacenters, or on their own computer or private cloud.
- Make the cloud infrastructure tightly connected to the storage of the the biological data.
- Ease the procedure of access by using the community's existing authentication methods (for example, single sign-on across portals and web services with Shibboleth [3] technology).
- Help bioinformaticians to build and to deploy single machines, clusters, or web service infrastructures to run a complete analysis pipeline.

[1] http://gbio-pbil.ibcp.fr/ws

The foreseen work consists of two major tasks:

1. Build bioinformatics appliances with web service interfaces: a) select the representative bioinformatics tools, b) build the Web service framework, c) integrate the tools in the virtual machine, and d) study a simple way of integrating new ones on-demand.

2. Prepare for scientific usage by the community: a) make the appliances publicly available and keep them up-to-date, b) provide user with Shibboleth access to the cloud, and c) evaluate the usage by the community in terms of resources consumption, flexibility and elasticity brought by the cloud.

The StratusLab distribution should already provide all of the cloud services necessary to implement this use case. In particular, this use case will intensively use the image creation and management services as well as the data management facilities.

4.3 TOSCANI (EXTERNAL)

4.3.1 Context

TOSCANI: TOwards StruCtural AssignmeNt Improvement is a project to improve the determination of protein structures based on Nuclear Magnetic Resonance (NMR) information. This concerns the scientific disciplines around (i) molecular and structural biology (determination of biomolecular structures up to atomic resolution) and (ii) bioinformatics, which includes the ensemble of computer algorithms for treating the data from biological systems.

In the domain of NMR, the protein structures are calculated by iterative approaches, based on a generation of family of molecular conformations allowing a filtering of the inconsistent distance constraints. The ambiguity is unavoidable in NMR data, as any proximity information measured on NMR spectra involve an undefined set of protons. The iterative approach presented above reduces the set of protons involved, to obtain a set of constraints consistent with a set of converged molecular conformations.

The problem of determining protein structures from incomplete and fuzzy spatial constraints has been largely explored over the last few years in the field of NMR. This knowledge could be, in principle, extended to other domains as many constraints obtained from biophysical and biochemical experiments describe a spatial proximity of protein regions.

The programs ARIA [9] and ISD [10, 11] will be used to calculate the structures. ARIA uses a simulated annealing procedure for the generation of the conformers and statistical analysis for the constraints filtering. ISD implements the first structure calculation approach fully based on Bayesian probability theory. The ARIA algorithm requires heavy computational resources in the case of large systems and/or high level of ambiguity in the NMR data; ISD is also quite demanding in terms of computational power.

Because of these large computational needs, an NMR laboratory not specially involved in bioinformatics developments will not invest in building a cluster of about 100 nodes to be able to run NMR structure calculations with ARIA or ISD. We propose here to demonstrate the flexibility of the cloud to deploy the different bioinformatics tools required to accelerate such a procedure.

4.3.2 Use Case

The test case is the re-calculation of NMR protein structures which were proposed as targets in the CASD-NMR contest [12]. Available NMR constraints concerning distances and angles in the structures, will be used for the calculation, and the obtained structures will be compared to the structure of these proteins deposited in the Protein Data Bank.

4.3.3 Analysis

ARIA uses a computational model where the master schedules the structural computation on a number of worker nodes according to the complexity of the structure to analyze and the precision required for the result.

The first step for implementing this use case is the definition of the virtual machines required to run ARIA on the cloud infrastructure.

- The first machine will be the ARIA master node where users connect, prepare their datasets, run the computation, and analyze their results. This machine will require remote graphical access (e.g. NX/FreeNX), large memory (8GB+), mounting the user storage space where the data is generated (cloud infrastructure persistent storage) and sharing a workspace filesystem with the worker nodes where the temporary files are stored and seen by all the actors.

- The second machine to be defined is the CNS worker node that performs the structure computations under constraints defined by the ARIA master. This machine requires low memory, a fast CPU, mounting the shared workspace exported by the master, and capable of fast transitions between sleep and running modes.

These two base machines will be registered in the Marketplace and available to users to be deployed on the cloud resources.

A significant increase in the number of calculated protein conformations improves the statistics on the NMR conformations and can help to overcome the ambiguity bottleneck. The large computing power required for this is concentrated in the simulated annealing procedure with the CNS software. Thus the elasticity of the cloud could be an advantage by waking the CNS VMs only during the simulation periods and putting them in sleep mode the rest of the time.

The definition of the ARIA infrastructure will be done with the OVF description rules and the Claudia service deployment system available in StratusLab. This will allow one-shot deployment of a full analysis system.

There is a commercial interest is providing such tools to structural biologists on a "pay as you go" basis. Development will be required in StratusLab to provide ARIA software providers with a monitoring and accounting system to record the user's resource consumption to permit billing for this service. Of course the authentication and authorization system of StratusLab should also be able to constrain access only to authorized users.

The work for this use case can be divided into two tasks:

- Implementation of ARIA on StratusLab: a) creation of the two required virtual machines, b) definition of the ARIA infrastructure combining one master VM and several worker VMs using OVF and Claudia, and c) validating the infrastructure usability with a set of well-known data.
- Scientific use: a) preparation of input for validation calculation with CASD-NMR data, b) evaluate the obtained results with respect to the model already known, and c) evaluate the robustness of StratusLab implementation and the flexibility and elasticity brought by the cloud.

As for the previous use case, this one relies heavily on the image management and storage facilities of StratusLab. In addition it also requires Claudia, the service manager, for the deployment and management of the entire system.

4.4 GAMESS Software (EXTERNAL)

4.4.1 Context

General Atomic and Molecular Electronic Structure System or GAMESS is a general *ab initio* quantum chemistry package. GAMESS is maintained by the members of the Gordon research group at Iowa State University. GAMESS was put together from several existing quantum chemistry programs. A wide range of quantum chemical computations are possible using GAMESS, and many of these calculations may be performed in *parallel*.

By 2005, GAMESS[2] had grown to roughly 650,000 lines of **FORTRAN**. It includes analytic hessian computation, electron correlation, perturbation theory, density functional approaches, and Coupled-Cluster approaches. Development of GAMESS continued by including the nuclear gradient code for MCP in 2007, also the ZFK family of model core potentials for p-block elements was added to GAMESS in 2010. Many other codes were interfaced to GAMESS.

This package is standard software used within the computational chemistry community. Having a virtual machine available with this software would facilitate the use of StratusLab cloud infrastructures by chemists.

4.4.2 Use Case

CYFRONET is planning to work first with the GAMESS, compiling it with Open-MPI to perform parallel computations.

[2]More information about GAMESS can be found at http://www.msg.ameslab.gov/gamess/.

CYFRONET is planning to implement a simple use case consisting of:

- Deploying a virtual machine with the desired resources (number of CPUs, memory, etc.) and attaching a persistent disk to it.
- Logging in to the machine and download the computation input file when the machine starts.
- Running GAMESS via its **rungms** script.
- Post-processing and visualizing its outputs.

Once this is working, deployments consisting of multiple machines can be investigated, for instance using a cluster of virtual machines to speed computations.

4.4.3 Analysis

The StratusLab cloud distribution (v1.0) satisfies the requirements needed for deploying the GAMESS software. In addition, StratusLab benchmarks already cover generic OpenMP and OpenMPI parallel applications. Consequently, this should be a fairly straight-forward use case that would expand interest in StratusLab within the computational chemistry community.

4.5 Gaussian Software (EXTERNAL)

4.5.1 Context

Scientists in many disciplines rely heavily on licensed software to complete their data analysis. For these scientists to take full advantage of cloud platforms, they must be able to use this licensed software on the cloud, while of course, respecting the limits of the license.

Gaussian [2] is a common commercial package used in computational chemistry research. CYFRONET has experience in running Gaussian on a grid infrastructure and this will help when trying to do the same on a cloud infrastructure.

MATLAB [15] is another package used in many scientific disciplines. Members of the project also have experience with running this package on a grid infrastructure. This may be a good candidate for a demonstration of using licensed software on the cloud, if for any reason Gaussian cannot be used.

4.5.2 Use Case Description

The core of this use case is a demonstration that licensed software can be installed within a customized virtual machine and that access to that customized virtual machine is controlled in a manner consistent with the license.

Gaussian should be the first choice for this use case as there is already an identified community behind it and CYFRONET has agreed to work with StratusLab on this topic. Using MATLAB can be considered a backup. MathWorks has been very collaborative with previous grid projects and it is likely that they would welcome such a collaboration with StratusLab.

4.5.3 Analysis

At a technical level, there is little to be done: simply create customized virtual machines with the licensed software (Gaussian or MATLAB) installed. The software will need to be validated, but this should not be a significant issue.

Much more work needs to be done to ensure that the software licenses can be enforced. Questions that will need to be investigated are:

- What types of licenses are compatible with cloud use?
- Is it necessary, desirable, and/or possible to control access to the virtual machine images themselves?
- Can licenses be deployed through the standard contextualization scheme?
- Do dedicated license servers need to be deployed to enforce licenses?
- Is accounting a necessary prerequisite for running licensed software (i.e. knowing who accessed and used the software)?

Answering these questions will involve some thought on how licensed software can be deployed on the cloud.

The outcomes from this use case should be 1) a demonstration of a commonly-used commercial (licensed) software package running legally on the cloud and 2) a document with recommendations for making licensed software available on the cloud.

4.6 Commercial Application (INTERNAL)

4.6.1 Context

Enterprises invest huge amounts of money in computing infrastructure, and its maintenance. Not only is this is a major cost but it requires significant effort for maintenance; all for something that is often not a core part of the business. Cloud services potentially allow enterprises to outsource their storage and processing needs so that they do not have to invest in the infrastructure, converting capital expenses into operating expenses.

Application providers can host their applications in the cloud, instead of using their own installations, potentially offering them as a Platform as a Service (PaaS) or Software as a Service (SaaS) application, optionally with a "pay for use" access scheme. Such a deployment is attractive as it can scale depending on the enterprises' computing, storage, and networking requirements.

Modern internet applications are complex software implemented in several layers. In addition to the usual computing, storage, and networking resource expected in any IaaS cloud, they also require high-level services. These include:

- Services driven by Key Performance Indicators (KPI): Services must provide a specific Quality of Service (QoS) to its users. Allocation (or reallocation) of resources must respect this QoS with respect to given KPIs.

- Scalability: In order to handle peaks of demand and guarantee the QoS, some tier in the service should be scaled. This means checking which tier is the bottleneck and providing more resources to handle the load.
- Multi-tier application management: Reallocation of resources in one tier can affect the performance in others because they are highly coupled and heterogeneous. Analysis must take place for the application as a whole.
- Security: This is a principal concern of all enterprises. This requires advanced services like dynamic Virtual Local Area Network and firewall management.

Running such a modern application on the StratusLab cloud is a good demonstration of its capabilities.

4.6.2 Use Case

The goal of this use case is to deploy a typical commercial application into a StratusLab cloud. The example which will be deployed is the Rice University Bidding System (RUBiS) [8]. This is an auction site prototype modeled after eBay.com used to evaluate design patterns and application server performance. This contains all of the elements of a typical enterprise application and potentially allows us to benchmark the performance.

The architecture of RUBiS has the typical three-tier configuration (Figure 4.1):

1. Presentation tier: The presentation layer represents the interface between the user and the rest of the application. In order to guarantee performance during peak demand, this layer will scale.
2. Business Logic tier: It has the business logic of the application. This layer can be scaled if needed, consequently, a load balancer will be included.
3. Back-end tier: All the data in the database or in the storage service belong to this tier. In order to get persistence, scalability is not possible at this level.

The demand for any application fluctuates, consequently the infrastructure must identify bottlenecks (idleness) at the various tiers and deploy (remove) resources, ideally automatically and consistently with any defined Service Level Agreements (SLA).

4.6.3 Analysis

This use case consists of running a complete, multi-tier commercial application on the StratusLab cloud. It will require the full range of capabilities provided by StratusLab. In particular, it will require:

1. Computation Capabilities: The services to deploy virtual machines.
2. Storage services: The data are going to be stored in the storage services that StratusLab offers.

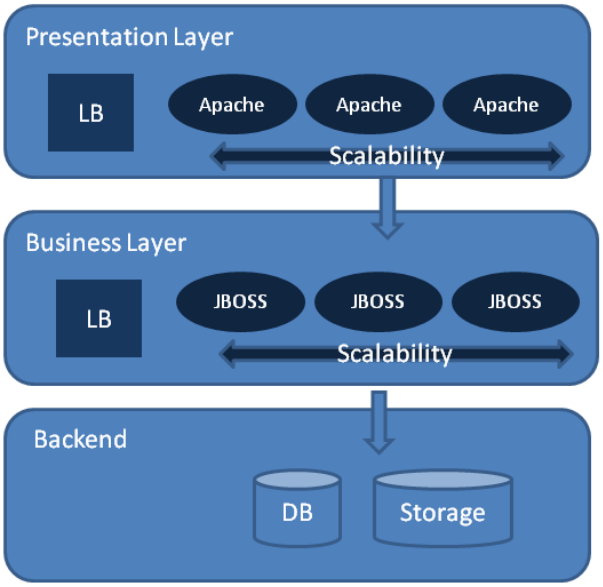

Figure 4.1: *Standard Three-Tier Archtecture*

3. Networking services: Networking capabilities are going to be provided by the networking services.

4. Service manager: The StratusLab service manager is in charge of managing application multi-tier and scalability at the service layer.

5. Authentication proxy: The authentication proxy provides authentication and authorization mechanism.

6. Monitoring: The monitoring of KPI is required in the monitoring system. However, the StratusLab monitoring system has focused on hardware information, and should be extended to take into account KPIs.

It will also require customized virtual machines to be created that contain the various services of the RUBiS application.

4.7 Software Development PaaS (INTERNAL)

4.7.1 Context

Efficient software development requires a large number of supporting services–code versioning systems, bug trackers, and continuous integration servers, for example. If more than a single programmer is involved, then a number of communication services (mailing lists, web sites, etc.) are also required.

Groups developing software can benefit by deploying these in the cloud. As for other services, the cloud provides more dynamic management, easier failover,

etc. It also allows customized services to be deployed which use a common authentication and authorization service to support a particular group of developers. This is important for short-lived projects and communities.

StratusLab itself falls into this category and should be the initial focus of this use case. Doing this will also provide valuable feedback to the project itself about the stability of its software and how easily they can be used to manage services.

4.7.2 Use Case Description

Running production services fully on the StratusLab cloud requires storage services as well as network and computing services. Persistent storage is required in order to save the state of the services, often held in databases or files. StratusLab 1.0 has this support and it should now be possible to host all of the project's services within the StratusLab reference infrastructure.

As the first part of this use case, StratusLab should become "self-supporting". That is, all of the project's infrastructure services should be hosted within a StratusLab cloud.

As a second part of this use case, a study should be done to see if a consistent set of software development services could be easily bundled and deployed. This would allow short-term projects and groups to quickly deploy their infrastructure. This would essentially be providing a Software Development PaaS.

4.7.3 Analysis

The existing StratusLab services (and current location) that should be migrated are:

- git repository (AWS)
- JIRA issue tracking (AWS)
- LDAP for authentication (AWS)
- web server (LAL)
- yum repository for packages (LAL)
- Nexus repository for maven artifacts (LAL)
- hudson continuous integration server (GRNET)

Those already running in AWS should be easy to migrate to a StratusLab cloud. The other services will need to be studied in more detail to determine how to migrate them to the StratusLab cloud.

Migration of Hudson slaves will require more analysis. Already some jobs are run on virtualized slaves. However deployments and tests which rely on KVM will probably still need to run on dedicated hardware.

Overall, the migration at the technical level should be fairly straight-forward. It will also require a discussion with the operations people on how to manage these machines once they are virtualized.

4.8 EGI Integration (EXTERNAL)

4.8.1 Context

The European Grid Infrastructure (EGI) is an existing e-Infrastructure having over 13000 users and consisting of over 300 sites throughout Europe and the world. High-energy physics researchers dominate the use of this infrastructure, although there is also use from other communities such as biomedicine, earth science, and humanities.

StratusLab has already demonstrated that gLite grid services can run in production within a StratusLab cloud. Using this work as a model for general deployment on EGI would allow broader deployment of StratusLab and thus a broader impact within the European e-Science community. Broader deployment, however, requires better integration with the operations procedures and tools of EGI.

4.8.2 Use Case Description

EGI has realized that the adoption of virtualization and cloud technologies is critical to improve the flexibility and efficiency of the current infrastructure and ultimately to empower virtual research communities to control directly the environment they offer their users.

The EGI "Cloud Integration Profile" [5] lists six scenarios for the integration of virtualization technologies on the infrastructure:

- Running a pre-defined VM image
- Running my VM image (with my data)
- Deciding which virtualized resource to use
- Accounting across resource providers
- Reliability/availability of the resource
- State change notification from the VM manager

These scenarios are a basis for the integration of StratusLab into EGI, with an analysis of them showing what new features need to be developed.

4.8.3 Analysis

The computing, storage, networking, and image management services already allow the first two scenarios (running a pre-defined VM image, running my VM image (with my data)) to be fully satisfied. The last four scenarios require tighter integration with EGI's information, accounting, and operations systems.

"Deciding which virtualized resource to use" essentially requires publishing the cloud's endpoints and available resources into the EGI information system. This information system is based on LDAP and uses the GLUE 2.0 Schema [6]. For StratusLab, this requires collecting the information about available resources,

formatting this information according to the schema, and pushing this into the information system. The core part of this work is an analysis of the current schema to see how cloud resources can be published. Actual publication requires some minor modifications to the StratusLab services to provide information. Pushing the information into the system is trivial, but may require running a part of the EGI information system (BDII) on the cloud.

"Accounting across resource providers" requires some significant changes to the StratusLab services to ensure that all of the required usage records are provided. The work in this area should concentrate on finding an agreement on the formats of the usage record formats. Once an agreement has been reached, a single record type should be collected from the StratusLab cloud and published into the EGI accounting system. This will likely be a CPU metric as other resource types will require significant changes in the EGI services (which are unlikely to happen in the lifetime of the StratusLab project).

"Reliability/availability of the resource" requires integration with the resource monitoring and testing system of EGI. This system is based on Nagios. This system sends tests (e.g. running a simple job) and collects statistics on success or failure. StratusLab integration means developing a set of simple tests which can be launched by the system. Given the large number of existing tests in the StratusLab continuous integration system, it should be straight-forward to adapt the existing tests to work with the EGI system.

"State change notification from the VM manager" requires messages to be sent from the StratusLab services to the user. EGI has its own messaging infrastructure and there are some public (test) infrastructures as well (e.g. for RabbitMQ). OpenNebula, the StratusLab virtual machine manager, already allows hooks to be defined when virtual machines enter a particular state. A trivial implementation would use these hooks to send a message. StratusLab should provide this implementation and then work with EGI and the users to determine the best method to get these to users. One open question, for example, would be whether to have a channel for each VM, each user, or some other granularity.

These scenarios provide a good framework for the EGI integration use case. This will require some minor modification of services or their configuration; however, it should be feasible to meet the minimums of these scenarios within a few sprints.

5 Strategy

The strategy for attracting new users and new user communities to the StratusLab distribution revolves around setting up a "virtuous cycle". The cycle starts with the porting of specific applications via one-on-one interaction with members of the activity and project. When the application has been ported, the people porting the application will interact with the dissemination activity to make a "media event" around the successful porting (e.g. press releases, technical articles, videos, etc.). This will lead to more interest in the StratusLab distribution and more opportunities to port new applications.

Work on this document has identified a number of potential use cases to be considered for initial porting. These include use cases for bioinformatics, computational chemistry, and software engineering. They also mix "internal" use cases which can be accomplished with just StratusLab personnel and "external" use cases which require the interaction and participation of outside people. All of these use cases will need to be defined in more detail as they are scheduled for implementation. The schedule will be determined via discussions with all of the WP2 participants; work on scheduled use cases will be handled through the standard sprints.

To increase the visibility of the software and to provide the technical basis for its use, the activity will continue to host StratusLab tutorials. EGI Technical and User fora will be targeted, but standalone tutorials will also be planned as well. In addition, "StratusLab User Workshops" will be planned in which current users can present their accomplishments and new users can obtain one-on-one consulting for porting their applications.

6 Summary and Conclusions

One of two detailed surveys conducted at the beginning of the project asked users for their requirements concerning cloud technologies. The results of this survey are available in the deliverable "Review of the Use of Cloud and Virtualization Technologies in Grid Infrastructures" [14]. Although these responses are still valid, the activity wanted to update them based on the experience and feedback of the users of the StratusLab reference cloud infrastructure. Unfortunately, the response to the survey was poor with only 5 of 30 users providing a complete response and none providing use case descriptions.

Because of this, the activity will take a more hands-on, individualized approach when dealing with the targeted communities, relying on tutorials and application porting "camps" for one-on-one interactions. The activity may consider a limited number of phone or personal interviews with users of the reference infrastructure to get feedback on problems with the software or infrastructure.

A set of preliminary use cases have been identified, which will be the initial focus of the StratusLab porting activities. There are seven use cases in total, touching bioinformatics, computational chemistry, and software engineering. Integration with the European Grid Infrastructure has also been identified, as this provides an opportunity for greatly expanding the number of StratusLab deployments and contacts with communities that already use distributed computing resources.

These use cases will have to be analyzed in more detail as they are scheduled for implementation. The use cases have been marked as "internal" or "external" depending on whether interactions with people outside of the project are required. The personnel of WP2 will discuss the scheduling of these use cases depending on their availability and the availability of the outside collaborators. All of the work will be managed through the normal sprint process.

The overall strategy for increasing the number of users and variety of users of StratusLab consists of setting up a "virtuous cycle". The overall idea is to concentrate on a limited number of use cases, providing one-on-one support as necessary for the porting. After each use case is completed, a dissemination effort with press releases, videos, announcement of scientific results, etc. will be done in collaboration with the dissemination work package. This will increase interest in StratusLab. The activity will also continue its tutorials and will work on hosting a "StratusLab User Workshop" to allow current users to show their results and for new users to received one-on-one consulting for their applications.

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] C. Blanchet, C. Combet, V. Daric, and G. Deleage. Web services interface to run protein sequence tools on grid, testcase of protein sequence alignment. *LNCS Biological And Medical Data Analysis*, 4345:240–249, 2006.

[2] Gaussian, Inc. Gaussian. http://www.gaussian.com/.

[3] Internet2 Middleware Initiative. Shibboleth. http://shibboleth.internet2.edu/.

[4] M. Kalas, P. Puntervoll, A. Joseph, E. Bartaseviciute, A. Töpfer, P. Venkataraman, S. Pettifer, J. Bryne, J. Ison, C. Blanchet, K. Rapacki, and I. Jonassen. BioXSD: the common data-exchange format for everyday bioinformatics web services. *Bioinformatics*, 26(18):i540–i546, 2010.

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

[6] Open Grid Forum, GLUE Working Group. GLUE Specificiation v. 2.0 (GFD-R-P.147). http://www.ogf.org/documents/GFD.147.pdf.

[7] S. Pettifer, J. Ison, M. Kalas, D. Thorne, P. McDermott, I. Jonassen, A. Liaquat, J. Fernández, J. Rodriguez, I. Partners, D. Pisano, C. Blanchet, M. Uludag, P. Rice, E. Bartaseviciute, K. Rapacki, M. Hekkelman, O. Sand, H. Stockinger, A. Clegg, E. Bongcam-Rudloff, J. Salzemann, V. Breton, T. Attwood, G. Cameron, and G. Vriend. The EMBRACE web service collection. *Nucleic Acids Res*, 1(38):Suppl:683–688, 2010.

[8] Rice University. Rice University Bidding System (RUBiS). http://rubis.ow2.org/.

[9] W. Rieping, M. Habeck, B. Bardiaux, A. Bernard, T. Malliavin, and M. Nilges. ARIA2: automated NOE assignment and data integration in NMR structure calculation. *Bioinformatics*, 23:381–382, 2007.

[10] W. Rieping, M. Habeck, and M. Nilges. Inferential structure determination. *Science*, 309:303–306, 2005.

[11] W. Rieping, M. Nilges, and M. Habeck. ISD: a software package for Bayesian NMR structure calculation. *Bioinformatics*, 24:1104–1105, 2008.

[12] A. Rosato, A. Bagaria, D. Baker, B. Bardiaux, A. Cavalli, J. Doreleijers, A. Giachetti, P. Guerry, P. Gntert, T. Herrmann, Y. Huang, H. Jonker, B. Mao, T. Malliavin, G. Montelione, M. Nilges, S. Raman, G. van der Schot, W. Vranken, G. Vuister, and A. Bonvin. CASD-NMR: critical assessment of automated structure determination by NMR. *Nat Methods*, 6:625–626, 2009.

[13] StratusLab. Creation of Virtual Appliances for Bioinformatics Community. http://stratuslab.eu/lib/exe/fetch.php/documents:stratuslab-ms3-v1.0.pdf.

[14] 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.

[15] The MathWorks, Inc. MATLAB. http://www.mathworks.com/products/matlab/.

[16] Zoomerang. Zoomerang. http://www.zoomerang.com/.

A Survey

A.1 Background Information

1. First Name
2. Last Name
3. Email Address
4. What type of institute do you work for?
    - Public Research Institute
    - Private Research Institute
    - Educational Institute
    - Government Entity
    - Large Enterprise
    - Small or Medium Enterprise
    - Not-for-profit organization
5. Country where your institute is located.
6. Specify your scientific or commercial domain of activity. (E.g. astronomy, bioinformatics, engineering, etc.)
7. Indicate the project you are affiliate to. (E.g. DCI or other European Project, etc.)

A.2 StratusLab Cloud Experience and Plans

8. How often do you use StratusLab reference cloud infrastructure? (never, once, occasionally, monthy, weekly, daily)
9. Rate the following StratusLab services (poor [1]–outstanding [5], not used)
    - Appliances Repository
    - Marketplace

- StratusLab Reference Infrastructure
- StratusLab website
- StratusLab support
- StratusLab mailing list

10. Rate the quality of the current StratusLab distribution(1.0) (poor [1]–outstanding [5], not used)
11. How relevant are the StratusLab software and cloud infrastructures for your applications? (not applicable, not at all [2]–very useful [6])
12. List important missing services in StratusLab

A.3 StratusLab Appliance Repository & Marketplace

13. To run your application in the cloud, are you using

    - Virtual machines from StratusLab Appliance Repository
    - Virtual machines metadata from StratusLab Marketplace
    - Other, please specify

14. If you are using StratusLab Appliance Repository, you are principally using:

    - ttylinux 9.7 base image
    - centos 5.5 base image
    - ubuntu 10.04 base image
    - grid appliances
    - bioinformatics appliances
    - Other, please specify

15. If you create your own virtual machines or appliances, what tool(s) do you use?

    - None (don't build customized images)
    - StratusLab stratus-create-image command
    - BitNami
    - rBuilder
    - Kameleon
    - CernVM
    - Cloud provider tools (e.g. Amazon build tools)
    - Manual creation (e.g. from OS distribution media)
    - Other, please specify

16. Are you using StratusLab Marketplace to share your virtual machines? (yes/no)

A.4 Application Characteristics & Use Case

17. Type(s) of executable(s) used in your application.
    - Sequential
    - Multi-threaded
    - Shared Memory (e.g. OpenMP)
    - Parallel (e.g. MPI)
18. Describe the overall control model for your multiple-job/task analyses.
    - Only single jobs used for complete analysis
    - Interactive control of running jobs/tasks
    - Manual submission of independent jobs/tasks (batch)
    - Automatic submission of independent jobs/tasks (master/workers, parameter sweep)
    - Automatic orchestration of interdependent tasks (workflow)
    - Parallel execution of interdependent tasks (parallel)
19. Typical size of input data for a single job/task ($<$10 kB, 100 kB, 1 MB, 10 MB, 100 MB, 1 GB, 10 GB, $>$100 GB)
20. Typical size of output data for a single job/task ($<$10 kB, 100 kB, 1 MB, 10 MB, 100 MB, 1 GB, 10 GB, $>$100 GB)
21. Approximate time needed to run a single job/task on modern machine/CPU. (1 minute, 10 minutes, 1 hour, 10 hours, 1 day, $>$1 day)
22. Do you have privacy, contractual, or confidentiality constraints for data stored in the cloud? (yes/no) If yes, indicate the type of protection needed (access control, encryption, etc.).
23. What are the most important mechanisms for data access in your application? (Leave blank if you don't know.)
    - POSIX file access
    - Remote File I/O API
    - Relational database
    - Object database
    - Block storage
    - Other, please specify

A.5 Use Cases to be deployed and Encountered Problems

24. Please, describe your use case(s) for porting your application(s) to StratusLab Cloud reference.

25. Describe any major problems or issues you have encountered when using StratusLab reference infrastructure.

26. StratusLab project is willing to work closely with you to port your application to the cloud. When are you available for one-on-one interaction for working on getting your use case to run?

    - Available Now
    - within 1 month
    - within 2 months
    - within 3 months
    - within 4 months
    - After 4 months

27. Are you willing to do a press release or other scientific dissemination activity when use case runs? (yes/no) Any additional comment?

28. Are you willing to participate in a StratusLab User Workshop? (yes/no)

29. Additional Comments