Enhancing Grid Infrastructures with
Virtualization and Cloud Technologies

Installing and operating a production grid site in the StratusLab cloud: Experience and issues

Technical Note TN-GridOverCloud (V0.2)
29 April 2011

Abstract

StratusLab provides a complete, open-source solution for deploying an "Infrastructure as a Service" cloud infrastructure. Deployment and operation of a grid site on top of this IaaS poses a number of challenges if we wish to take full advantage of the Cloud service capabilities. In this technical note we report are own experience gained from the installation of a production grid site on top of StratusLab's reference cloud service and we provide various suggestions for areas that need improvement.

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

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

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

Contributors

Name	Partner	Sections
Evangelos FLOROS	GRNET	All
Athanasia ASIKI	GRNET	All
Kyriakos GINIS	GRNET	All

Document History

Version	Date	Comment
0.1	07 May 2011	Initial version for comment.
0.2	19 May 2011	Reviewed internally in GRNET

Contents

1 Introduction

StratusLab provides a complete, open-source solution for deploying an "Infrastructure as a Service" cloud infrastructure. One of the main use cases we wish to support is the operation of grid sites on top of IaaS cloud services. During the first phases of the project we experimented extensively with the installation and operation of grid sites on top of cloud services. In parallel, we prepared a number of VM appliances for the basic node roles of a gLite-based grid site and namely: the Computing Element, the Storage Element, the Worker Node, the User Interface and the APEL service (used for site accounting). All these images are available from the appliance repository http://appliances.stratuslab.eu. These appliances currently follow the evolution of gLite middleware; with every new release a new image snapshot is created and is uploaded to the repository.

In order to fully evaluate the capability of cloud services to support the operation of grid sites we deployed a production-level grid site on top of the project's reference cloud service running in GRNET. The site named HG-07-StratusLab was certified within the GRNET NGI (the Greek National Grid Initiative) and has joined the Greek national grid infrastructure (HellasGrid). The site offers a CE and 8 dual-core WNs thus providing a total capacity of 16 cores for job submission. The site supports MPICH-2 and OpenMPI parallel jobs. Each WN is configured with 4GB of main memory. The site also provides a SE that offers a total storage space of 2TB. It should be noted at this point that the storage is configured directly as an NFS mount-point from the local storage server and is not yet virtualized (i.e. it cannot be managed as a persistent block storage service from the StratusLab command line tools).

The GStat page with all the details of the site as they are reported from the Site-BDII are available at http://gstat-prod.cern.ch/gstat/site/HG-07-StratusLab/.

This technical note summarizes our findings from the installation of HG-07-StratusLab and the experience we have gained so far from operating the site within the EGI pan-european grid infrastructure. In particular we consider issues related to the installation, configuration and daily operation of the site and we recognize potential impediments and issues that according to our opinion forbid the optimal exploitation of cloud technologies for the provision of grid services.

2 Installation

The installation process in a cloud environment depends on the availability of appropriate pre-configured VM images. Typically in this case grid software will come pre-packaged in virtual appliances. It is expected that the appliance providers will maintain images with the base OS required by grid services (e.g. SL 5.5 or CentOS 5.5) as well as the RPM packages needed for a specific grid node like a Computing Element, Storage Element, Worker Node etc. Additionally the appliance provider should have configured the necessary yum repositories that enable the quick update of the VM instance installed software. As a good practice, it is recommended that the grid administrator using these VMs should issue a run update command upon the instantiation and the first but of the VM.

In the context of StratusLab we have prepared appliances for the basic grid site nodes (CE, SE, WN, UI and APEL). We'd expect that this task will be officially taken over by the cloud middleware providers (e.g. EMI) or the grid infrastructure supervisors (EGI, NGIs etc). These appliance providers should make sure that the VMs follow the evolution of grid middleware and that the appliances are validated to be secure and bug-free before their release.

3 Configuration

3.1 Service configuration

Grid services are configured using the yaim tool [3]. Yaim uses a set of configuration files in order to fine-tune all the aspects of a grid site such as the number of worker nodes, the creation of user accounts, the number of VOs supported by the site and the information that the site broadcasts to the central grid information system.

Overall the yaim system is very static and assumes a very homogenous hardware setup where all Worker Nodes expose same capabilities and hardware characteristics. The administrator has to complete a large number of information and to make configuration decisions at the setup of the grid site that remain unchanged. In order to make any changes in the existing configured setup, the grid admin has to edit the respective configuration files and re-issue the yaim command.

3.1.1 Site locality

Geographical information defined by SITE_LAT and SITE_LONG macros might not be available to the grid site admin due to lack of knowledge about the location of the cloud datacenter. Even if this information is available it might not even remain the same during the lifetime of the grid site. If we consider an architecture of federated cloud providers distributed in different regions in the same country or across europe it might be very probable that, due to various reasons like physical node maintenance or in order to move workload to different locations, this VM might be migrated to a different location. In this case the information reported by the above variable will be outdated. The grid site admin might not be even aware of this incident thus he/she is not capable of updating them manually.

3.1.2 Hardware information

CE_CPU_MODEL, CE_CPU_VENDOR, CE_CPU_SPEED, CE_OS_ARCH might not be known, may be difficult to define or finally may not be relevant at all in a cloud environment. In case of VM migration and considering a scenario that a datacenter is build from non-homogenous physical nodes (different vendor, different architecture) these values may change and in some cases quite often.

CE_MINPHYSMEM, CE_MINVIRTMEM, CE_PHYSCPU, CE_LOGCPU and CE_SMPSIZE assume a static, homogenous cluster where all the WNs share the

same physical characteristics. Notice that this has been an issue also in the past with traditional grid cluster, but now with the cloud this assumption is even more restrictive since in the cloud the grid admin has the ability to customize very easily WNs with different hardware profiles. Latest versions of yaim configuration support the glite-CLUSTER nodetype which allows a more fine grain separation of the physical resources. Still the nodes in a cluster are considered to be of the same type.

CE_PHYSCPU and CE_LOGCPU are difficult to be separated in a virtualized environment. In particular CE_PHYSCPU may not be easy to be defined at all since there is always a chance that a VM is not assigned a dedicated CPU but a subset of its time (e.g. 80%). Moreover the static definition of the above macros makes it very cumbersome to take advantage of the elasticity characteristics of the cloud. This values should be able to change on the fly whenever either the grid site admin adds WNs by hand to the site or when a automated mechanism is used to adjust the size of the site adjusting to workload fluctuations and forecasted resource demands from grid jobs.

3.2 End user software

Currently grid sites follow a rather inelastic way of installing and advertising available software from site admins and VO managers. User should be able to create their own WN VM images with their software pr-installed and attach it on-demand to an existing grid site that supports their VO. The StratusLab Marketplace could come into play in this scenario. Grid sites would act as endorsers of these VMs based on the specific VOs they support. This of course requires the establishment of the appropriate policies for VM endorsement on a EU-wide level. Again EGI could play an important coordination role in this area.

4 Operations

We consider two aspects of grid operations that are immediately impacted from the cloud: the initial certification of the grid site and the site monitoring.

4.1 Site certification

Currently a site is certified through a formal process within the hosting NGI. It remains an open question who will be responsible for implementing the certification process in the case that the virtualized grid site is hosted in a cloud provider residing outside the grid site NGI. Will it be the NGI of the cloud provider? Will it still be the responsibility of the grid site NGI? Will it be delegated to a centralized authority (a team within EGI?).

One other form of certification is the one required for issuing the digital certificates necessary by many services in a grid site (e.g. CE, SE). Typically the grid site admin will have to generate a certificate request for the service and email it to the Certification Authority responsible for his/her country. One of the requirements that the CA will check in the request will be the domain name of the service that has to reside within its area of authority (e.g. .gr for Greece). In the case of grid sites over clouds it is very probable that the cloud service might reside in a different country thus the allocated virtual machines will have a top level domain in a country different than the one in the area of responsibility of the CA. This will probably forbid the CA from signing the certificate. Obviously for this to work the CA policy has to be altered and allow signing of certificates for servers residing in foreign countries. Otherwise grid sites can take advantage of only same-country cloud providers.

Moreover, if we consider a federated cloud environment in which resource providers from different countries collaborate to provide cloud services, there is always a change that part or all of the grid site might migrate to a different country or split among 2 or more countries in respective cloud service providers. Who will be the hosting NGI in this case? Will the digital certificates have to be reissued from a different CA? Obviously, we have to reconsider to certification and monitoring procedures in order to take into account the characteristics of cloud environments.

4.2 Site monitoring

Currently information about a specific site is collected by a service running in the site itself and broadcasted to a centralized service in EGI. These services are using LDAP to collect, organize and provide access to information. LDAP by design targets at providing a system for infrequent updates and frequent queries; thus LDAP does not apply as an adequate solution. A similar problem has been already noticed in traditional infrastructures, for example in updating the information about available CPUs, the total available storage, etc. In a cloud environment, this problem is amplified since a site may be structurally altered (virtually expanded or contracted) exploiting the elasticity and flexibility of the underlying cloud.

4.3 Accounting

The scenario we had so far in grids was this of computing resources offered to scientists free of charge or at least with no immediate charging (but rather the charges where managed centrally by the government authorities). Thus the grid accounting system was targeted mainly to support the collection of statistics and the centralized workload management systems of grid infrastructures. In the case of cloud e-Infrastructures it is not clear yet how the costs will be mitigated and who will be responsible for paying them. If we consider though a typical scenario where commercial cloud providers will offer resources to scientists or hybrid clouds where government funded clouds will burst to commercial clouds in order to handle peek workloads, it is crucial that the accounting system collects detailed information on VO and individual user level.

As mentioned, in our production site we have used glite-APEL [1] for site accounting. APEL collects only a limited number of information such us the number of jobs submitted or total CPU time consumed per site/user/VO. Integration with cloud services will require a much more detailed report regarding resource usage including network bandwidth, storage space and potentially consumption of software licenses (wherever applicable). On the other hand the cloud layer should be able to re-use this information in order to charge costs or/and to apply quota limitations.

An alternative solution to APEL is DGAS (Distributed Grid Accounting System) [2] developed by INFN. DGAS offers the ability to collect accounting information for a broader range of metrics including Economic Accounting. According to our knowledge this last capability of DGAS has not been exploited so far by any of the production EGI sites. It may still be the case that this functionality will be useful in the cloud computing context.

4.4 Site elasticity

Resource elasticity and flexibility is one of the most well known benefits that cloud computing brings to e-Infrastructures. Grid sites should be able to capitalize on this capability by being able to dynamically adjust their dimensions based on temporal

demands. Typical dimensions of a grid site are:

- Processing capacity: being able to modify the size the cluster by adding or removing WNs on demand.
- Processing capability: being able to modify the profile of the WNs by adding more CPU cores, local storage and
- Storage capacity: being able to modify the available storage provided by the SE node.

Currently StratusLab is working on grid site elasticity functionality. The idea is to integrate the Service Manager (Claudia) with the LRMS (Local Resource Manager System - e.g. Torque) in order to modify the size of the site based on grid admin rules. E.g:

- Increase the size of the site by 10% if the job queues become 80% full
- Decrease processing capacity (remove WNs) by 20% if the utilization of the job queue falls below 20%

This dynamic behavior of grid sites on the other hand may cause inconsistency on the global level if the information about the site's new capabilities are not announced promptly to the top level information systems (e.g. top-level BDII). Otherwise centralized job management services like WMS (Workload Management System) will not be able to make the appropriate decisions for job scheduling.

References

[1] APEL. Accounting Processor for Event Logs. https://wiki.egi.eu/wiki/APEL.

[2] DGAS. Distributed Grid Accounting System. http://www.to.infn.it/dgas/.

[3] YAIM. YAIM Ain't and Installation Manager. https://twiki.cern.ch/twiki/bin/view/EGEE/YAIM.