ISSN 0002-306X. Изв. НАН РА и ГИУА. Сер. ТН. 2010. Т. LXIII, № 1.

UDC 530.15: 004.032.26 COMPUTER SCIENCE AND INFORMATICS

A.T. HARUTYUNYAN

# DEVELOPMENT OF TWO MODELS OF DYNAMIC INTEGRATION OF CLOUD COMPUTING RESOURCES WITH ALIEN GRID

The work on enabling one click deployment of CERN ALICE experiment Grid sites on the Infrastructure as a Service cloud is described. Two models of dynamic virtual AliEn Grid site deployment are developed.

***Keywords*:** grid, cloud computing, automated resource integration, virtualization.

**1. Introduction.** Infrastructure as a Service (IaaS) providers allow users to easily acquire on-demand computing and storage resources. They provide each user with an isolated environment in the form of Virtual Machines which can be used to run services and deploy applications. This approach, also known as 'cloud computing', has proved to be viable for a variety of commercial applications. Currently there are many IaaS providers on the market. The biggest of them is Amazon with its 'Amazon Elastic Computing Cloud (Amazon EC2)' service [1].

The question arises whether we can dynamically provide cloud resources and elastically integrate them into the pool of resources available to CERN ALICE experiment [2] to satisfy the time-varying needs of scientists. Furthermore, can we do it in such a way that no change is visible to the user, i.e. the users do not need to change the ways in which they use the system?

In this contribution we show how cloud computing resources can be used within the AliEn Grid framework [3, 4], developed by CERN ALICE experiment, for performing simulation, reconstruction and analysis of physics data. We deploy baseline virtual software appliance for the Large Hadron Collider (LHC) experiments developed by the CernVM project [5] on the resources of Science Clouds that use the Nimbus project [6, 7] to enable deployment of virtual machines (VMs) on remote resources. We further also use Nimbus tools for one click deployment of dynamically configurable AliEn Grid site on the Science Cloud of the University of Chicago.

An introduction to AliEn, Nimbus and CernVM is given in Chapter 2, Chapter 3 presents the problem definition, Chapter 4 describes two models of automated integration of IaaS resources to AliEn, in Chapter 5 we present timing measurements of dynamic site deployment, and in Chapter 6 the summary is given.

## 2. AliEn, Nimbus and CernVM

*2.1. AliEn.* AliEn is a lightweight framework developed by the ALICE experiment to handle execution of jobs performing simulation, reconstruction and analysis of physics data. It operates like a global queue system which schedules jobs to about 100 sites distributed all over the world.

The framework is built from open source components using a combination of Web Services and distributed agents that communicate with each other using the SOAP protocol [8]. AliEn services can be divided into 2 categories: central services and site services. Central services perform Virtual Organization (VO) wide tasks such as user authentication (Proxy

Service), job management (Job Manager Service), job scheduling (Job Broker Service), etc. Site services provide interfaces to computing and storage resources on the sites as well as perform auxiliary tasks (e.g. monitoring of those resources). Examples of site services are Computing Element (CE) and application package manager (PacMan). Site services must be deployed on all sites serving the needs of the ALICE experiment, whereas central services must be deployed only in one place. For the detailed description of AliEn framework please refer to [3] and [4].

*2.2 Nimbus.* The Nimbus project provides an open source, extensible IaaS implementation supporting Web Service Resource Framework (WSRF) [9] as well as the Amazon EC2 interfaces (the Virtual Workspace Service (VWS)) as well as end-user services that make cloud computing easy-to-use for the end user. An example of the latter is the Context Broker service [7] that enables secure establishment of security and configuration context over hosts spanning a distributed environment. The project mainly targets scientific community and offers a platform for experimentation with features for scientific needs. Detailed information on Nimbus project can be found in [6] and [7].

*2.3. CernVM.* The CernVM Virtual Software Appliance is a thin Virtual Machine that contains just enough of Operating System to run the application frameworks of the four LHC experiments. Its Operating System, based on rPath Linux2, fits into a compressed file smaller than 100 MB. The experiment software stack is brought into appliance by means of a file system specifically designed for an efficient and 'just in time' software distribution. In this model, the client downloads only necessary binaries and libraries as they get referenced for the first time. By doing that, the amount of software that has to be downloaded in order to run the typical experiment tasks in the Virtual Machine is reduced by an order of magnitude. For detailed information about CernVM please refer to [5].

**3. Problem definition.** The developed models enable integration of IaaS cloud resources with AliEn Grid infrastructure for the solution of the following problems:

1) Allow users to customize the environment in which their jobs are executed - currently the control over the operating system is handled by the site administrators and the application software is controlled by the VO administrators.

2) Allow the job execution environment to remain consistent across multiple executions – there are over a hundred sites where AliEn is deployed, the software version and configuration varies from site to site, and thus the users have no possibility to test their programs in the environment which will be identical to that of the worker node. This is very important for the debugging of those programs since the behaviour of the same program which was run in different environments is likely to vary.

3) Minimize the efforts of resource providers in supporting different VOs/user communities – most of the sites which provide resources to ALICE also provide resources to other scientific communities. So in addition to maintaining the Grid middleware site administrators have to also maintain application software stacks required by the communities they support. This process requires significant efforts since as the rule the software packages required by the scientists are complex, have a lot of external dependencies, as well as require frequent reinstallation since they are in the phase of active development.

4) Allow AliEn users to use the resources offered by scientific IaaS providers and in case of a high system load or urgent need of computing and storage capacity to lease for short or long term computing and storage resources from commercial IaaS providers.

**4. 'Classic' and 'Co-Pilot' models for the Deployment of AliEn site on an IaaS cloud**

*4.1. 'Classic' model.* To deploy a virtual site on a cloud one needs to deploy following services:

- ClusterMonitor - routes messages from AliEn site services to central services. All site services communicate to central services and the configuration database through the site Cluster Monitor. (runs on services node)
- PackMan (Package Manager) – installs application packages (e.g. ROOT, AliRoot) required by the jobs on the site (runs on services node)
- MonaLisa – monitors site services (runs on services node)
- JobAgent (JA) – delivers the input of the data required by the job, starts the job and registers the output of the job in the AliEn file catalogue. (runs on WNs)

There is no need to run a CE service, which serves as an interface to the local batch system and is typically used to start the JA on the WNs. In our setup JAs are automatically started on the WNs whenever WNs are booted. We do not deploy Storage Element (SE), a service for accessing local mass storage systems, because the whole site running on a cloud is supposed to be started and stopped very often, and thus the data which potentially could be kept in the SE would not be available most of the time. That is why the data produced on the virtual site is kept on the SEs of other sites. For the detailed description of AliEn services see [4]. The step by step site deployment procedure is as follows:

1. Deploy the workspaces (1 services node, which will run ClusterMonitor, PackMan and MonaLisa and required number of WNs, which will run JA) using the image provided by CernVM.
2. Transfer credentials (X.509 certificate + private key) to all newly created workspaces
3. Transfer site configuration information (e.g. name of the SE where the data produced by the site must be kept) to the nodes.
4. Configure services node (e.g., NFS server, which must export the directory where the PackMan service will install the application software).
5. Configure WNs (e.g., add to configuration files the address of the services node)
6. Start the ClusterMonitor, MonaLisa and PackMan services
7. Start JA service on WNs

The Nimbus Context Broker [7] allows to perform all these steps (and thus deploy an AliEn site) with a single command. The command gets as an input a cluster description file and the VM image which must be used to start each of the nodes. Cluster description file is written in XML and specifies number and type ('service' or 'WN') of nodes as well as site configuration parameters.

When the site deployment command is executed Nimbus launches the VMs and 'contextualizes' them to the type specified in cluster description file. Contextualization is done in the following way:

Upon launching VMs a lightweight agent running on each VM contacts Nimbus Context Broker and gets the contextualization information. This information is passed to the set of scripts (so-called contextualization scripts), which are launched to configure appropriate services inside VMs and start them.

Once the deployment and contextualization are completed, JAs contacts the Job Broker central service which fetches jobs from the ALICE task queue and sends them to JAs for execution.

*4.2. 'Co-Pilot' model.* In this case AliEn site services are not deployed on a cloud. Instead we take the approach of deploying just the worker node VMs on the cloud. To implement that we have created CernVM Co-Pilot Agent and CernVM Co-Pilot Adapter services. Co-Pilot Agent runs on the WNs and uses Jabber/XMPP instant messaging protocol [10] to communicate with Co-Pilot Adapter, which runs outside the cloud and provides the functionality previously provided by AliEn site services. Co-Pilot Agent and Adapter were developed specifically to ease deployment of Grid sites on computing clouds. The use of Jabber/XMPP allows scaling of the system in case of a high load on the Adapter node by just adding new Adapter node(s) to the existing messaging network.

The steps for site deployment are following:

1. Deploy workspaces using an image provided by CernVM
2. Transfer credentials (Jabber ID and password of the agent) to the newly created workspaces
3. Start Co-Pilot Agent service on the workspaces

Like in case of a 'classic' site, here we also use Nimbus Context Broker to deploy AliEn site with a single command. The command gets as an input cluster description file and the VM image which must be used to start the nodes. Cluster description file is written in XML and specifies number of nodes which need to be started. The Agent is preconfigured with the Jabber server hostname and the Jabber ID of Co-Pilot Adapter, which are the only configuration parameters it needs. Co-Pilot Agent then contacts Co-Pilot Adapter service and requests a job to execute. Upon receiving job request from an Agent, Adapter contacts AliEn Job Broker central service which fetches jobs from ALICE task queue and sends them to Adapter, which in turn forwards jobs to Agents. When the job is done, Agent reports results to Adapter which in turn forwards them to AliEn central services. Fig. 1 represents the AliEn site setup following the 'Co-Pilot' model.

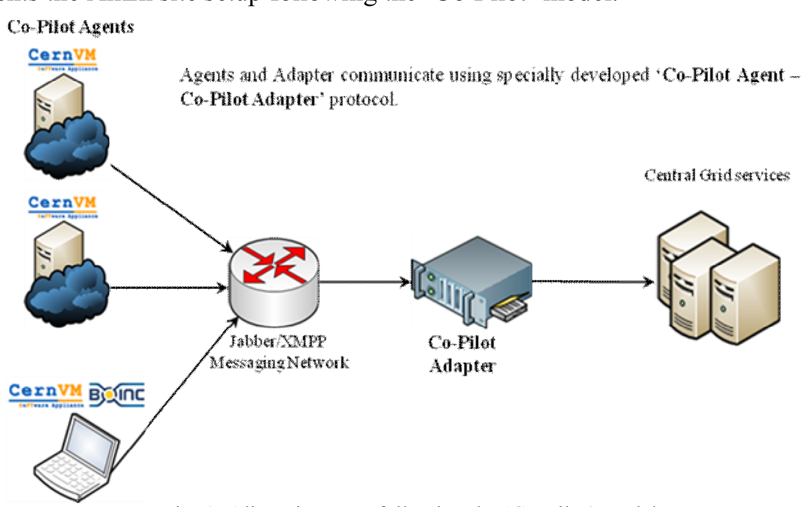


Fig. 1. AliEn site setup following the 'Co-Pilot' model

*4.3. Comparison of the models.* The 'classic' approach is very straightforward and does not require any new development neither in AliEn nor in Nimbus. However, it has some drawbacks. The first thing is that one needs a separate workspace (and in case of high load few of them) to run site services. Their deployment is time consuming and if we could exclude them from the setup we would potentially have more workspaces for running WNs. Another thing is that in this case the application software is brought to the cloud by PackMan service and is made available to WNs through NFS server. This is not optimal because CernVM image already provides the application software and it would be cleverer and faster if WNs could use it (that would also allow elimination of PackMan service from the deployment), but that would require modification of AliEn JobAgent service code. The 'classic' approach can be used to easily integrate cloud resources into other Grid frameworks.

The Co-Pilot approach does not require deployment of service nodes and thus potentially allows running more jobs on the same number of virtual workspaces (since some of them will be used as worker nodes rather than service nodes). Besides it uses application software available from CernVM and does not require any package management from Grid middleware whatsoever. Currently Co-Pilot Adapter can fetch jobs only from AliEn. However it can be extended to communicate with any pilot job framework, e.g. PanDA - distributed production and analysis system [11] used by CERN ATLAS [12] experiment, or Dirac [13] – Grid solution used by CERN LHCb experiment [14]. Co-Pilot agent does not have anything AliEn-specific and is written in a way, that running jobs fetched from other frameworks should not require extra development.

The 'Co-Pilot' approach enables [15] the integration of not only IaaS cloud resources but also the resources provided by volunteer computing systems like BOINC (Berkeley Open Infrastructure for Network Computing) [16].

One of the noteworthy characteristics of both models is that the integration of IaaS cloud resources into the existing Grid infrastructure is completely transparent for the end users and therefore does not change their perception of the system.

**5. Cloud site deployment timing measurements.** Measurements were performed during the series of deployments on the Nimbus Scientific Computing Cloud of the University of Chicago and Argonne National Laboratory. The measurements have revealed that:

1) The time of worker node deployment on the cloud is proportional to the number of virtual machines being launched (Figure 5).

2) For the same number of the worker nodes, the start-up duration, that is the time period from the issuance of site deployment command to the booting of the OS of nodes is longer in 'Classic' model (Figure 5, light green and light orange bars): this is because in the case of deployment following 'Classic' model one launches an additional node for running AliEn site services.

3) The time span between worker node start-up and job request does not depend on the number of worker nodes and varies slightly (Figure 5, dark green and dark orange bars). It is about 400 seconds in case of 'Classic' model and about 15 seconds in case of 'Co-Pilot' model. This time is spent to start and initialize site services on the services node and the Alien Job Agent service on worker nodes in first case, and to start and initialize Co-Pilot Agent in second case.

4) The mean values of the time which elapses between requesting of job(s) by the worker nodes and assignment of jobs to them by AliEn do not exceed 5 seconds in case of 'Classic' model, and vary from 3 to 30 seconds in case of 'Co-Pilot' model. The reason for this is that the current version of Co-Pilot Adapter is serving requests sequentially, whereas AliEn site services are processing multiple requests in parallel.

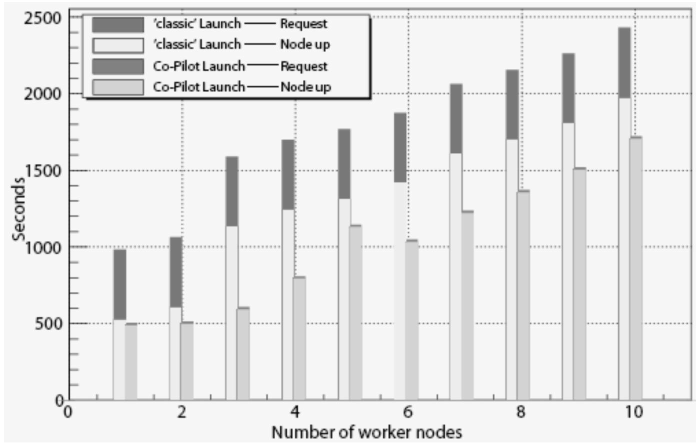

Fig. 2. Virtual site deployment timings

**6. Summary.** Both models provide solution to all four problems outlined in Chapter 3:

- Using the CernVM virtual appliance AliEn users can prepare the virtual machine images which will contain the customized environment necessary for execution of their jobs. This environment will not depend in any way either on the configuration or on the hardware platform of the site where the jobs will be executed. After testing the configuration on the images on workstation or the laptop of the user they can be used to deploy virtual machines and execute jobs on IaaS cloud and volunteer computing resources. This provides solution to problems 1 and 2
- The use of IaaS approach for the provision of computing and storage resources simplifies significantly the job of site administrators which are responsible for supporting Grid middleware and experiment specific application software. Instead of duplicating the efforts of managing and maintaining complex software applications and Grid middleware for different scientific communities they will just need to support an infrastructure implementing IaaS cloud (e.g. Nimbus). Using the implementations of the developed models such an infrastructure can be transparently integrated with the Grid and solve problem number 3
- Both models allow AliEn users to exploit the resources offered by scientific IaaS cloud providers and volunteer computing projects and in case of a high system load or urgent need of computing and storage capacity to lease for short or long term computing and storage resources from commercial IaaS providers such as Amazon

**Acknowledgements.** This work has been supported by Google Summer of Code 2008, as well as by Swiss Fonds "Kidagan" and Calouste Gulbenkyan Foundation.

## REFERENCES


1. Amazon Elastic Compute Cloud (Amazon EC2) http://aws.amazon.com/ec2/
2. The ALICE experiment at the CERN LHC / **K. Aamodt**, **Quintana A. Abrahantes, R. Achenbach,** et al **[ALICE Collaboration]** // Journal of Instrumentation. – 2008.- 3 S08002.
3. The AliEn system, status and perspectives / **P. Buncic, A.J. Peters, P. Saiz** // ECONF C0303241 MOAT004. - 2003.
4. The architecture of the AliEn system / **P. Buncic, A.J. Peters, P. Saiz** et al // CHEP 2004, Interlaken, Switzerland 440. - 2004.
5. CernVM - a virtual appliance for LHC applications / **P. Buncic, Sanchez, C. Aguado, J. Bloomer,** et al // Proceedings of Science, PoS (ACAT08)012. - 2008.
6. Sky Computing / **K. Keahey, M. Tsugawa, A. Matsunaga, et al** // IEEE Internet Computing journal. - September/October, 2009. - Vol. 13, N.5.
7. **Keahey K., Freeman T.** Contextualization: Providing One-Click Virtual Clusters// Proceedings of eScience 2008 conference. - Indianapolis, USA, 2008.
8. SOAP – Simple Object Access Protocol http://www.w3.org/TR/soap/
9. WSRF – Web Service Resource Framework http://www.globus.org/wsrf/
10. XMPP – Extensible Messaging and Presence Protocol http://www.xmpp.org
11. **Maeno T.** PanDA: distributed production and distributed analysis system for ATLAS// J. Phys. Conf. Ser. 119 062036. - 2008.
12. The ATLAS Experiment at the CERN Large Hadron Collider / **G. Aad, E. Abat, J. Abdallah** et al // JINST 3 S08003. - 2008.
13. DIRAC Review Report / **J.P. Baud, P. Charpentier, J. Closier,** et al // Technical Report LHCb-2006-04 COMP, CERN. - 2006.
14. **[The LHCb Collaboration]** The LHCb Detector at the LHC/ **Alves Jr. A. Augusto, Filho L.M. Andrade, A.F. Barbosa,** et al // 2008 JINST 3 S08005 doi: 10.1088/1748-0221/3/08/S08005. -2008.
15. Building a Volunteer Cloud / **B. Segal, Quintas D. Garcia, Gonzalez D. Lombrana,** et al // Proc. of Latin American Conference on High Performance Computing. - Merida, Venezuela, September, 2009.
16. **Anderson D.** BOINC: A System for Public-Resource Computing and Storage// 5th IEEE/ACM International Workshop on Grid Computing. - Nov. 8, 2004. - Pittsburgh, PA, 2004. - P. 365-372.


SEUA. The material is received 25.01.2010.

**А.Т. АРУТЮНЯН**

## РАЗРАБОТКА ДВУХ МОДЕЛЕЙ ДИНАМИЧЕСКОЙ ИНТЕГРАЦИИ РЕСУРСОВ КОМПЬЮТЕРНЫХ ОБЛАК В ГРИД ИНФРАСТРУКТУРУ ALIEN

Описана работа по автоматизации развертывания служб грид сайтов на компьютерном облаке типа “Инфраструктура в качестве Сервиса (ИвкС)”. Разработаны две модели, позволяющие динамически интегрировать ресурсы, предоставляемые облаками ИвкС в грид инфраструктуру эксперимента ALICE, проводимого на большом адронном коллайдере в ЦЕРН (Женева, Швейцария).

***Ключевые слова:*** грид, облачный компьютинг, автоматизированная интеграция ресурсов, виртуализация.

**Ա.Տ. ՀԱՐՈՒԹՅՈՒՆՅԱՆ**

## ALIEN ԳՐԻԴ ԻՆՖՐԱԿԱՌՈՒՑՎԱԾՔԻՆ ԱՄՊԱՅԻՆ ՔՈՄՓՅՈՒԹԵՐԱՅԻՆ ՌԵՍՈՒՐՍՆԵՐԻ ԻՆՏԵԳՐՄԱՆ ԵՐԿՈՒ ՄՈԴԵԼԻ ՄՇԱԿՈՒՄ

Նկարագրված է ‘Ինֆրակառուցվածք որպես Ծառայություն (ԻռԾ)’ տիպի քոմփյութերային ամպերի վրա Գրիդ կայքային ծառայությունների տեղադրման ավտոմատացման աշխատանքը: Մշակված են ԻռԾ ամպերի տրամադրած ռեսուրսները ՑԵՌՆ-ի (Ժնև, Շվեյցարիա) Մեծ Հադրոնային Արագացուցչի ALICE գիտափորձի Գրիդ ինֆրակառուցվածքի մեջ ինտեգրման երկու մոդելներ:

***Առանցքային բառեր.*** գրիդ, ամպային քոմփյութերային, ռեսուրսների ինտեգրման ավտոմատացում, վիրտուալացում: