

# ATLAS Cloud R&D

**Sergey Panitkin<sup>1</sup>, Fernando Barreiro Megino<sup>2</sup>, Jose Caballero Bejar<sup>1</sup>,  
Doug Benjamin<sup>3</sup>, Alessandro Di Girolamo<sup>2</sup>, Ian Gable<sup>4</sup>, Val  
Hendrix<sup>5</sup>, John Hover<sup>1</sup>, Katarzyna Kucharczyk<sup>2</sup>, Ramon Medrano  
Llamas<sup>2</sup>, Peter Love<sup>6</sup>, Henrik Ohman<sup>7</sup>, Michael Paterson<sup>4</sup>, Randall  
Sobie<sup>4</sup>, Ryan Taylor<sup>4</sup>, Rodney Walker<sup>8</sup>, Alexander Zaytsev<sup>1</sup>  
on behalf of the ATLAS Collaboration**

1: Brookhaven National Laboratory, Upton, USA

2: CERN, Geneva, Switzerland

3: Duke University, Durham, USA

4: University of Victoria, Victoria, Canada

5: Lawrence Berkeley National Lab, Berkeley, USA

6: Lancaster University, Lancaster, UK

7: Uppsala University, Uppsala, Sweden

8: Ludwig-Maximillan Universitat, Munich, Germany

**Abstract.** The computing model of the ATLAS experiment was designed around the concept of grid computing and, since the start of data taking, this model has proven very successful. However, new cloud computing technologies bring attractive features to improve the operations and elasticity of scientific distributed computing. ATLAS sees grid and cloud computing as complementary technologies that will coexist at different levels of resource abstraction, and two years ago created an R&D working group to investigate the different integration scenarios. The ATLAS Cloud Computing R&D has been able to demonstrate the feasibility of offloading work from grid to cloud sites and, as of today, is able to integrate transparently various cloud resources into the PanDA workload management system. The ATLAS Cloud Computing R&D is operating various PanDA queues on private and public resources and has provided several hundred thousand CPU days to the experiment. As a result, the ATLAS Cloud Computing R&D group has gained a significant insight into the cloud computing landscape and has identified points that still need to be addressed in order to fully utilize this technology.

This contribution will explain the cloud integration models that are being evaluated and will discuss ATLAS' learning during the collaboration with leading commercial and academic cloud providers.

## 1. Introduction

The ATLAS experiment [1] at the Large Hadron Collider (LHC) is designed to explore the fundamental properties of matter for the next decade. Since LHC start-up in 2009, the experiment has produced and distributed hundreds of petabytes of data worldwide among more than 100 computer centers. Thousands of physicists are engaged in analyzing these billions of events. The ATLAS Computing model [2] is based on a Grid paradigm [3], with multilevel, hierarchically distributed computing and storage resources. However, new cloud computing technologies bring attractive features to improve the operations and elasticity of scientific distributed computing. ATLAS sees grid and cloud computing as complementary technologies



that will coexist at different levels of resource abstraction, and two years ago created an R&D working group to investigate the different integration scenarios.

Earlier ATLAS studies in virtualization and cloud computing were reported elsewhere [4]. In this paper we will describe some of the cloud utilization models that are being evaluated as well as ATLAS' experience in collaboration with leading commercial and academic cloud providers. ATLAS utilizes the PanDA workload management system [5] (WMS) for distributed data processing and analysis.

## 2. Distributed cloud computing - Grid of Clouds

ATLAS has been evaluating a distributed cloud computing system for running Monte Carlo simulation jobs. A distributed cloud computing system provides a number of attractive features. Virtualization shields the application suite from changing technologies and reduces the need for systems personnel to be knowledgeable about the user application, allowing resource centers that do not have expertise in grid computing or ATLAS software to nevertheless contribute to ATLAS distributed computing. Also, Infrastructure as a Service (IaaS) clouds provide a simple way to dynamically manage the load between multiple projects within a single center, and a distributed cloud aggregates heterogeneous clouds into a unified resource with a single entry point for users (namely, a PanDA queue). The idea of a distributed cloud was initially suggested by Keahey *et al* [6], who called it “sky computing”; it could also be termed a “grid of clouds”.

The scheduling of jobs is performed by HTCondor [7] and the management of virtual machines (VMs) is done by Cloud Scheduler [8]. HTCondor is a well-known job scheduler that is designed as a cycle scavenger, making it an ideal job scheduler for a dynamic environment where VM resources appear and disappear dynamically, based on demand. The management of the VMs in the distributed cloud computing system is done by a general-purpose component called Cloud Scheduler. Cloud Scheduler periodically reviews the requirements of the jobs in the HTCondor batch queue and makes requests to boot user-specified VM images on any of the IaaS clouds which meet those requirements. Using Cloud Scheduler minimizes the overhead of booting and shutting down VMs, since VMs are reused to continue running jobs until there are no matching jobs remaining. CernVM [9] images are used (both Xen and KVM flavors), and Puppet [10] is used to manage the system configuration of the VMs.

The distributed cloud computing system currently uses a total of approximately 15 academic IaaS clouds, located in Canada (Victoria, Edmonton, Quebec and Ottawa), CERN, the United States (FutureGrid [11] clouds in San Diego and Chicago), Australia (Melbourne and Queensland), and the United Kingdom. The Victoria and FutureGrid clouds run Nimbus [12] while the other clouds use OpenStack [13]. Several different PanDA queues are used, each corresponding to a general geographic region. Since April 2012, the system has completed over 1,000,000 ATLAS jobs (running up to 1,000 at a time) in these clouds, located on three different continents. A full description of the system can be found in [14].

## 3. Cloud platform on the ATLAS HLT Farm

With the LHC collider at CERN currently going through the period of Long Shutdown 1 (LS1) there is a valuable opportunity to use the computing resources of the large trigger farms of the experiments for other data processing activities. In the case of the ATLAS experiment the TDAQ High Level Trigger (HLT) farm [15, 16, 17], consisting of more than 1500 compute nodes deployed in the SDX1 area or LHC Point 1, is particularly suitable for running Monte Carlo production jobs. In 2013 ATLAS initiated design and deployment of a virtualized platform running on the ATLAS TDAQ computing resources and using it to run the large groups of CernVM based virtual machines operating as a single CERN-P1 WLCG site. This platform has been designed to avoid interference with TDAQ usage of the farm and to guarantee the

security and the usability of the ATLAS private network; OpenStack has been chosen to provide a cloud management layer. During the period of February–September 2013 the project has gone through all the stages of deployment, from prototyping up to the full scale deployment in the environment of ATLAS TDAQ HLT farm located at the LHC Point 1. The full scale deployment was finalized in mid-2013 and resulted in demonstration of the capabilities of operating the virtualized infrastructure deployed on top of the ATLAS HLT farm at the level of 17.1k single CPU cores exported to the CERN-P1 site by 2.1k virtual machines. It was also demonstrated that it is feasible to operate such a group of VMs with a single instance of the centralized OpenStack components deployed locally within the farm, with 45 minutes needed to switch between the TDAQ and Sim@P1 state and about 10 minutes needed for switching back to TDAQ from the Sim@P1 state. Both operations are typically performed via Puppet with no requirement for rebooting the farm in the process. Over the follow up period of July–September 2013 the first production operations for Sim@P1 project were carried out, resulting in successfully finishing more than 1.4 million of both event generator and detector simulation jobs for ATLAS production and delivering more than 1.4k CPU-years of computing power to CERN-P1 WLCG site. That corresponds to 7.4% of the resources used for ATLAS MC production on all of the WLCG sites during the period of June 1, 2013 Sep 18, 2013. Thus on the basis of experience obtained while operating Sim@P1 infrastructure so far one can conclude that the virtualized infrastructure deployed on top of the ATLAS HLT farm is capable of contributing to the ATLAS MC production on a level of computing power comparable to that of a large Tier-1 WLCG site, despite the opportunistic nature of the underlying computing resources being used.

## 4. Experience with commercial cloud platforms

### 4.1. Project with Amazon EC2

In 2013, US ATLAS and the Brookhaven Lab RHIC/ATLAS Computing Facility (RACF) group have worked on both the utilization of commercial Cloud services, and on advancing the usage of dedicated local resources via the IaaS approach. Our system consisted of several components which worked together to create an integrated system. These components included Boxgrinder [18] (to automate and generalize virtual machine image creation and customization), HTCondor (both to submit virtual machines to cloud platforms, and to handle batch job scheduling on the virtual cluster), the Open Science Grid [19] worker node middleware ( for authorization and file transfer), network file system CVMFS [20] (to provide globally-visible application software), and our custom pilot submission utility, AutoPyFactory (APF), which handled the coordination of both local batch scheduling and VM invocation. We utilized Amazon EC2 [21], using the spot pricing mechanism and deployed OpenStack to provision local, dedicated resources so that they could be used transparently alongside EC2 resources. Much IaaS usage has focussed on replicating traditional Grid/Virtual Organization (VO) entities on virtual machines running on Cloud platforms. Rather than deal with the complexity and heterogeneity of these systems, we adopted a hybrid approach, with facility-based central services combined with Cloud-based cluster execute hosts, all relatively homogeneous.

Since the components of the system were used in other ATLAS cloud projects will describe them in some detail below.

The HTCondor batch system has been used for years in ATLAS for local batch cluster management and scheduling and for traditional grid submission. In the last two years, HTCondor added a full HTCondor-G interface to any resource provided via the Amazon EC2 API, allowing the invocation of EC2 VMs which can then be queried and terminated like any job. We also used HTCondor as the execute host batch system, with VM execute hosts connecting back to a central manager at BNL to retrieve pilot jobs, which in turn retrieve work from the ATLAS central system at CERN.

In 2013, OSG moved to an RPM-based distribution of its middleware. This greatly streamlined

its usage on our worker node (WN) images generated by BoxGrinder (since BoxGrinder installs software via YUM utility). The advent of CVMFS (an HTTP-based, secure, distributed, read-only filesystem) was the development which enabled our hybrid approach to IaaS usage, since no shared filesystem need be maintained in the Cloud.

The dynamic management capability of AutoPyFactory (APF) version 2 was a new addition in late 2013. We had already been using APF for pilot submission to US and European grid sites via HTCondor-G (and in several cases via local cluster submission). It was logical to extend this usage to EC2 VM submission. Further combining this with local batch submission of pilots, from the same submit host, is what enabled well-integrated management of VM-based batch clusters. By combining information about execute hosts state (idle, busy, retiring) with the state of the VM job hosting the execute node, we can dynamically scale the size of a Cloud-based cluster up and down depending on whatever parameters we choose (the amount of waiting work to be done, spot price, etc.).

*4.1.1. Amazon EC2 Spot Market and OpenStack* Aside from special, unusual cases, standard commercial clouds initially were too expensive for most scientific computing. The advent of EC2 spot pricing, which trades a willingness to tolerate VM terminations for significantly reduced hourly prices, made IaaS an economically viable competitor to site-based dedicated resources. At the same time, the maturation of software platforms for provisioning local, physical resources via cloud interface (e.g. OpenStack) offered the possibility of combining owned, local resources with remote, commercial resources in a flexible manner. Our system fully realizes this possibility. BNL created a 200 VM-scale local OpenStack (v4) cloud which was used alongside EC2 spot-priced resources during a scaling test performed in 2013. The 5000 VMs were invoked and managed using APF, with all the execute hosts joining a single virtual cluster that spanned BNL, and Amazon us-east, us-west1, and us-west2 zones. This cluster performed standard ATLAS production simulation jobs (high CPU, low I/O) which staged in from, and out to, the standard Grid storage interfaces at BNL.

#### *4.2. Project with Google Compute Engine*

ATLAS was invited to participate in the Google Compute Engine (GCE) [22] trial period in August 2012. At that time Google was rolling out a new IaaS cloud platform with modern hardware and a brand new API.

After several months of preparation and testing Google agreed to allocate additional resources for ATLAS, about 5 million core hours, on 4000 cores. The idea of this project was to run a production site on GCE, on a scale similar to a typical ATLAS Tier 2 site, that is continuously operational for several weeks. Resources were organized as HTCondor based PanDA queue, similar to the setup employed for the EC2 run, described earlier in this paper. That allowed for utilization of common components and transparent inclusion of cloud resources into the ATLAS Computational Grid.

Since GCE at that time did not allow to run user provided VM images, we built a custom image based on the Google provided CentOS 6 [23] image. Initially a set of custom scripts was used for image building, later on a set of Puppet recipes was developed to standardize the procedure. In total we ran on GCE for about 8 weeks in March–April of 2013. Two weeks were planned for start up, debugging and scale up to full size. We ran computationally intensive workloads - physics event generators, fast detector simulations and full detector simulations. Produced data were automatically transferred to ATLAS Grid storage for subsequent processing and analysis. We completed 458,000 jobs, generated and processed about 214 M events.

We would like to note that during the whole running period GCE was very stable in all aspects of operations, quite a remarkable feat for a brand new cloud platform. Overall job error rate on GCE was about 6%, over the whole period of running, with most of the errors occurring during

start up and system debugging period.

## 5. Summary

In this paper we give an overview of some of the recent activities of the ATLAS Virtualization and Cloud Computing R&D project for the past year and a half. ATLAS continues to actively explore cloud technology using private and public cloud platforms. Many of the projects are characterized by a shift from pure R&D to production level projects, where cloud resources delivered to ATLAS are comparable to Tier 2 or even Tier 1 centers on the Grid. Also active engagement with large commercial cloud providers like Amazon and Google was a hallmark of the period since CHEP 2012.

## References

- [1] ATLAS Collaboration, J. Inst. 3 (2008) S08003.
- [2] Adams D, Barberis D, Bee C, Hawkings R, Jarpe S, Jones R, Malon D, Poggioli L, Poulard, G, Quarrie D and Wenaus T 2005 The ATLAS computing model Tech. rep. CERN CERN-LHCC-2004-037/G-085
- [3] Foster I and Kesselman C (eds) 1998 The Grid: Blueprint for a New Computing Infrastructure (Morgan-Kaufmann)
- [4] F. Barreiro Megino et. al., Exploiting Virtualization and Cloud Computing in ATLAS, Proc. of the 19th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2012)
- [5] Maeno T., Overview of ATLAS PanDA Workload Management, Proc. of the 18th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2010)
- [6] Keahey K., Tsugawa M., Matsunaga A. and Fortes J., Sky Computing., Internet Computing, IEEE, 13(5), 43-51, 2009
- [7] HTCondor Project <http://research.cs.wisc.edu/htcondor>
- [8] Armstrong P. et al., Cloud Scheduler: a resource manager for a distributed compute cloud. arXiv:1007.0050v1 [cs.DC]
- [9] CernVM Portal <http://www.cernvm.ch/portal>
- [10] Puppet Labs <http://puppetlabs.com>
- [11] FutureGrid Portal <https://portal.futuregrid.org>
- [12] Nimbus Project <http://www.nimbusproject.org/>
- [13] OpenStack Project <http://www.openstack.org>
- [14] Sobie R. et al., HTC Scientific Computing in a Distributed Cloud Environment, Science Cloud '13 Proceedings of the 4th ACM workshop on Scientific cloud computing
- [15] J. Zhang et al., 2010. ATLAS Data Acquisition. 16th IEEE NPSS Real Time Conference 2009, Beijing, China, 10-15 May 2009
- [16] A. Adeel-Ur-Rehman et. al., 2010. System Administration of ATLAS TDAQ Computing Environment. J. Phys.: Conf. Ser. 219 022048
- [17] M. Dobson et al., IT Infrastructure Design and Implementation Considerations for the ATLAS TDAQ System. In J.A. Moinhos Cordeiro, M. Virvou, and B. Shishkov (Eds.), ICSoft 1, 206-209, SciTePress, 2010
- [18] Boxgrinder Project <http://www.boxgrinder.org>
- [19] Open Science Grid Portal <http://www.opensciencegrid.org>
- [20] CVMFS Portal <http://cernvm.cern.ch/portal/filesystem>
- [21] Amazon EC2 <http://aws.amazon.com>
- [22] Google Compute Engine Portal <https://cloud.google.com/products/compute-engine>
- [23] CentOS Project <http://www.centos.org>