

Testing as a Service with HammerCloud

Ramón Medrano Llamas^α, Quentin Barrand^α, Johannes Elmsheuser^β, Federica Legger^β, Gianfranco Sciacca^γ, Andrea Sciabà^α, Daniel van der Ster^α

^α: CERN, Geneva, Switzerland

^β: Ludwig-Maximilians-Universität München, Munich, Germany

^γ: Universität Bern, Bern, Switzerland

E-mail: ramon.medrano@cern.ch

Abstract. HammerCloud was designed and born under the needs of the grid community to test the resources and automate operations from a user perspective. The recent developments in the IT space propose a shift to the software defined data centres, in which every layer of the infrastructure can be offered as a service.

Testing and monitoring is an integral part of the development, validation and operations of big systems, like the grid. This area is not escaping the paradigm shift and we are starting to perceive as natural the Testing as a Service (TaaS) offerings, which allow testing any infrastructure service, such as the Infrastructure as a Service (IaaS) platforms being deployed in many grid sites, both from the functional and stressing perspectives.

This work will review the recent developments in HammerCloud and its evolution to a TaaS conception, in particular its deployment on the Agile Infrastructure platform at CERN and the testing of many IaaS providers across Europe in the context of experiment requirements. The first section will review the architectural changes that a service running in the cloud needs, such an orchestration service or new storage requirements in order to provide functional and stress testing. The second section will review the first tests of infrastructure providers on the perspective of the challenges discovered from the architectural point of view. Finally, the third section will evaluate future requirements of scalability and features to increase testing productivity.

1. Introduction

HammerCloud has quickly become the de-facto tool for grid site testing, both from the functional point of view and for the throughput and performance measurement. It is being used by ATLAS, CMS and LHCb, which are requesting more than 18,000 tests per year. These tests have generated around 40 million grid jobs in 2013, with their associated 2 billion performance metrics [1].

Also, it is clear that the need of an infrastructure testing service is rising: not only we need to test and validate the grid from a functional perspective, but also infrastructure related services, such as Infrastructure as a Service (IaaS) endpoints. This testing has to be done on-demand, has to offer real time results and has to serve as the first information to debug complex problems, such as performance issues on a very complex software stack.

HammerCloud already has in place the basics for this kind of testing, which we call Testing as a Service (TaaS), and has been extended to be based on an elastic infrastructure which will guarantee the sustainability and scalability of the service.



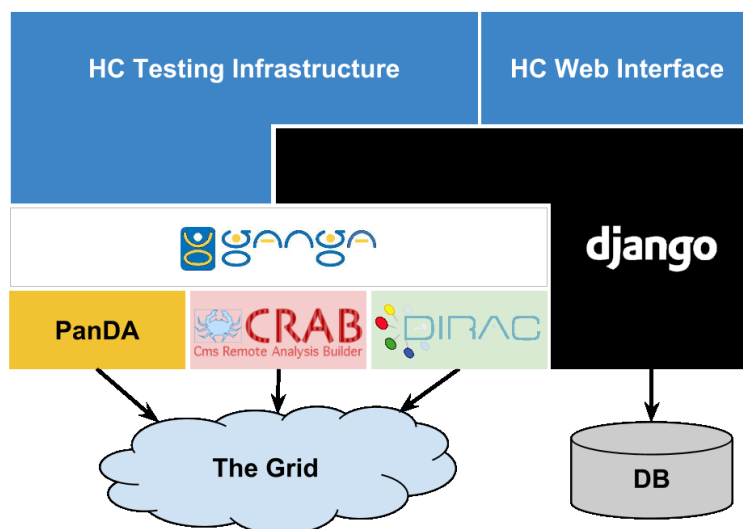


Figure 1: HammerCloud architecture diagram

As can be seen on the Figure 1, the architecture of HammerCloud is based on plugins for the Ganga project, which allows abstracting all of the specifics of the grid back ends. Also, this plugin architecture can be used to add any infrastructure back end that is not a traditional grid framework, such as local batch systems, storage test beds (such as DPM or CVMFS) or cloud interfaces to access IaaS offerings [2].

Currently, there is an on-going discussion to add a fourth instance of HammerCloud devoted to the CERN IT Department, which will make use of the testing infrastructure in order to submit stress test to development instances of DPM and possibly to functionally probe the OpenStack cloud.

This work presents the infrastructure foundations changes that HammerCloud is overcoming in order to support all this new testing profiles. The rest of the introduction will introduce the motivation of the deployment changes. The second section is focused on the architectural changes that HammerCloud is receiving, after a review on the infrastructure tests performed at CERN is presented, along with guidelines to make efficient use of a dynamic infrastructure. Finally, an outlook of future scalability challenges is presented. The paper ends with an analysis of conclusions and a review of the future work needed.

1.1. The Agile Infrastructure

Two years ago, the CERN IT Department started a project to modernize the tool chain used within the department and to ease the management of the local computer centre plus the new remote centre in Budapest.

The infrastructure is now based on an OpenStack managed private IaaS cloud, plus some other tools for the configuration management, such as Puppet and the Foreman. These tools allow a very easy management for the applications, their automatic scalability and reduce the need of managing duplicated services, such as monitoring tools or issue tracking systems.

The cloud infrastructure is now managing more than 60,000 cores in the computer centre and on the High Level Trigger (HLT) farms of ATLAS and CMS, both managed using OpenStack [3].

Also, there is a set of now Platform as a Service (PaaS) services, being the most notable in the case of HammerCloud the Database on Demand (DBoD) service. The current storage backend of HammerCloud is composed of four MySQL databases running on this service, which store more than 100 GiB of raw data.

In general all services are encouraged to move to the new model, and although it requires some adaptation time for the development and deployment of services in a cloud fashion, the benefits in terms of reliability, scalability and reduced operations are expected to be high.

1.2. HammerCloud cluster management

HammerCloud was deployed on a static cluster since its inception, composed of a mixture of about 20 physical and virtual machines provided by the collaborations (known as *voboxes*). The management of the machines is done by the central services of each collaboration using configuration management tools such as Puppet or Quattor.

The central services only provide the platform for the deployment of the machines (namely, an OS and several packages), while the deployment of the applications and services has to be done by the service providers.

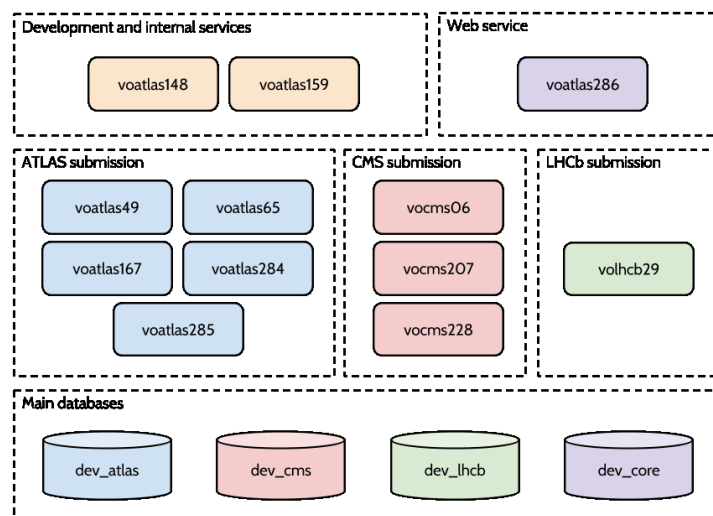


Figure 2: HammerCloud cluster deployment on late 2012

The main problem with this approach is the lack of agility in order to cope with new load: computing resources must be requested to the collaboration, we need to wait for the provisioning and then handle the long life time of a machine. These servers have to be maintained, upgraded and optimized. Also, having custom versions of software is cumbersome (such a specific versions of Python modules) due to the lack of flexibility of the fabric management of the *voboxes*.

We need to be more dynamic requesting the computing resources while also requesting the exact amount of machines we need. Also, we need to be flexible enough to have our custom software in order to provide the best service possible.

1.3. New test cases and scenarios

Since the original design and inception of HammerCloud, a number of new test cases have arose, a part of the traditional functional and stress testing cases [4]:

- Analysis and Production Functional Testing: these are special version of the functional tests deployed for ATLAS whose results are used to blacklist and whitelist the PanDA sites.
- Benchmarking suite: a special functional test that is used to crosscheck site performance in terms of event rate. This test submits longer jobs (full simulation) and is submitted in bunches of 100 jobs (in order to avoid scheduling artefacts due to the different hardware generations).
- Athena Nightly Build System: this test suite allows the deployment of nightly builds of Athena on grid sites and cross check the new software version on different setups and storage element types.

- XRootD federation (FAX) testing: this test needed to modify how jobs are scheduled in order to force different redirector access. PanDA is now allowing overriding some of the queue configurations and HammerCloud allows selecting different redirector from the local and global redirector table.
- Cloud resource validation: the deployments on the Agile Infrastructure at CERN, on several providers such as Rackspace and on the Helix Nebula consortium have been carried out by using HammerCloud as submission tool, performing stress testing for several weeks before opening the clusters to general purpose processing.
- ROOT I/O and WAN tests: these tests allow crosschecking the performance of basic tools, such as ROOT, whose performance is crucial to the overall analysis performance.

All these new testing profiles have as consequence the fact that the submission profile is composed of a predictable background submission (coming from regular functional tests) and an unpredictable submission requests coming from on demand testing and other functional tests which variable profile.

2. Architectural changes

The main change that is needed to make HammerCloud an elastic service that can handle on demand load without making use of excessive resources is to request the computing resources on demand. An IaaS service is then a perfect fit for this task, since the on demand self-service capabilities to get computing resources are the base of an elastic service on top.

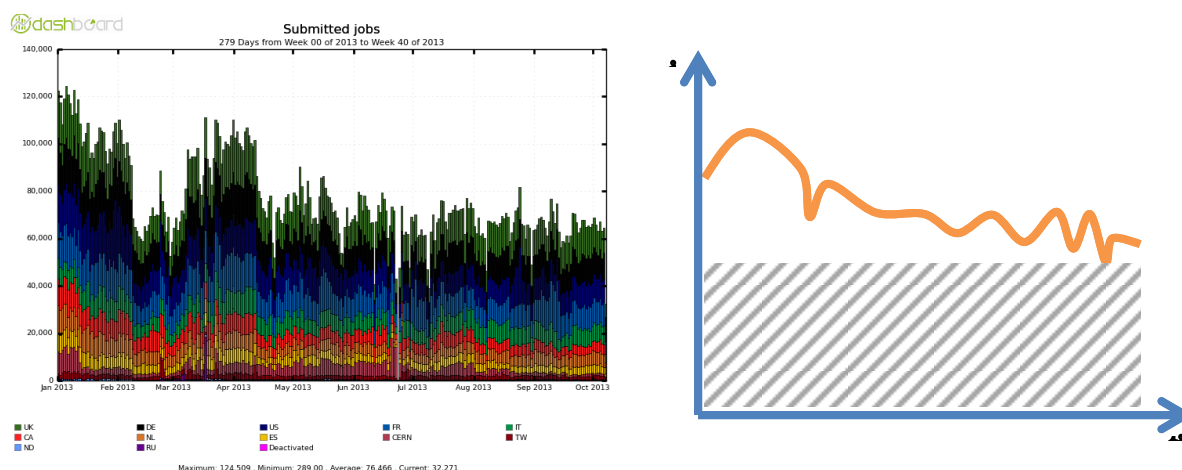


Figure 3: Schematic of the job submission profiles and the actual submission for 2013

The test generation and scheduling had to be rewritten, as well as the local storage of log files. Test generation, for instance, needed to schedule the test on a submit host of the HammerCloud cluster; while now, it will need to get a virtual machine on demand when the test is close to start. Test scheduling is then as simple as matching one virtual machine to a test process.

In the classic version of HammerCloud, tests were assigned to a submission host on the request time, based on some load balancing guidelines; with the new deployment model, the submission host is not existing at this time yet and will have to be requested when the test is close to start in order to be able to start the test on the desired time. This is a major change on the scheduling algorithm of HammerCloud and imposes some new challenges, like delaying virtual machine requests until the last moment, which can be difficult on an infrastructure whose performance and response time is unpredictable.

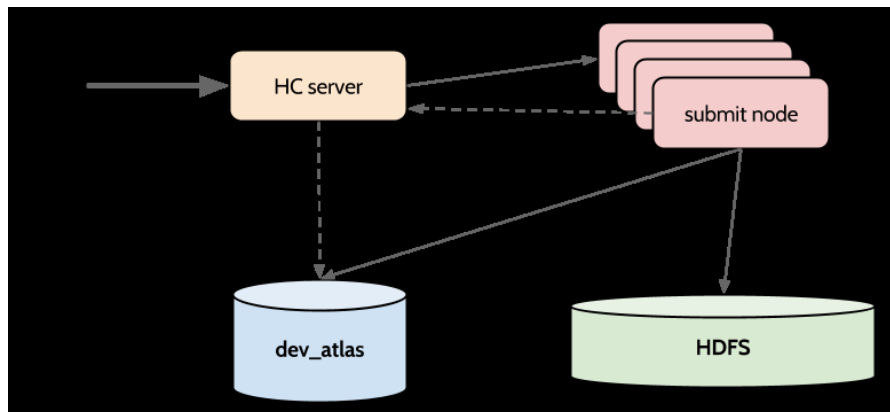


Figure 4: HammerCloud cluster deployment on the Agile Infrastructure, with the orchestrator

2.1. Infrastructure orchestration

The new HammerCloud model only has a long live server: the infrastructure orchestrator. This node, as part of serving as the web server (this responsibility can eventually be moved to another machine), will handle the test scheduling, machine creation and log storage. The scheduling of a test is now a six step process:

1. A test is requested, either by a user, a tool or is generated automatically by HammerCloud.
2. The test gets scheduled for the desired start time.
3. Sometime before the scheduled time, the virtual machine is requested for the test.
4. The machine is bootstrapped and the test is physically scheduled and started.
5. On the test ending event, all the log files are stored in a central location.
6. The machine is killed and cleaned up.

This mechanism allows a very tight resource usage, while providing isolation between tests, horizontal scalability and reliability in case of cluster partition.

2.2. Provisioning optimization

It is very important to request the virtual machines some time in advance in order to fulfil the test schedule time. It is also important to not do this in much time in advance in order to not waste computational resources.

A study was carried out on the CERN private cloud in order to estimate the time needed to boot a machine, resulting in a predictable distribution.

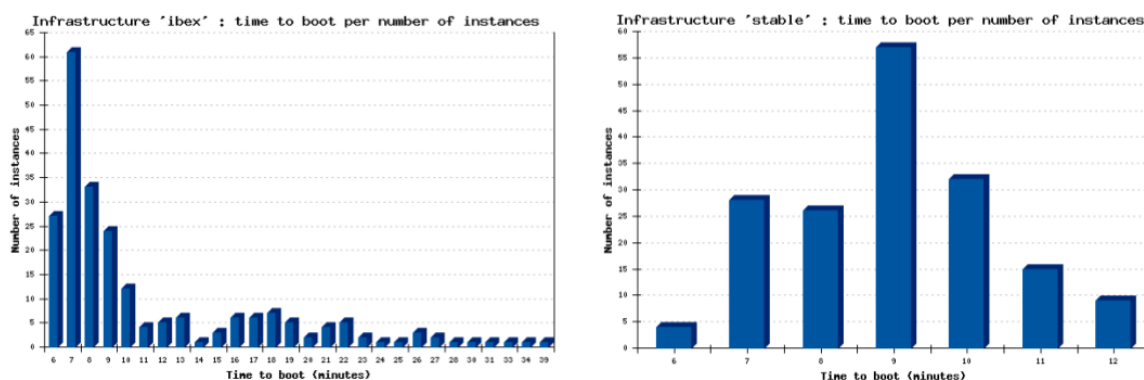


Figure 5: Provisioning time distribution for two versions of the cloud infrastructure at CERN

Since the time to provision virtual machines depends on the load on the infrastructure and the middleware, this distribution will vary over time. For instance, with the deployment of the stable version of the cloud infrastructure, the provisioning time improved notably.

To mitigate the effect of these changes, HammerCloud will know and adapt the distribution over time with the performance that is measuring from the machines it is requesting.

2.3. Shared log storage

Apart of the performance metrics obtained by the submitted jobs, which are stored on the sharded MySQL databases, the log files obtained from the Ganga plugins are also very important, since they allow tracing and debugging the errors and issues by the site admins and the test requestors.

On the classic deployment, the logs are kept for 30 days and are stored on the submission nodes, which need to have a web server in order to access them. This situation was suboptimal due to the limited amount of time of log retrieval and the increased service load on the submission hosts.

On the new infrastructure, a shared file system has been deployed in order to store the logs centrally for a unified and comfortable access. The first iteration is based on an NFS export to the infrastructure orchestrator. On future versions, the logs can be stored on the central HDFS service for monitoring information.

2.4. Programmatic test request

Another important feature in order to provide the testing on demand behaviour is to allow other tools, such as the Workload Management Systems (WMS) to request tests programmatically. This will require an API to request tests which is under development.

3. Infrastructure as a Service providers testing

At the same time of the development of this new model, some testing on IaaS providers was requested. HammerCloud was used this year to test the CERN Agile Infrastructure, using both ATLAS and CMS workloads, to test some resources at Rackspace and to test the providers which are part of the Helix Nebula consortium, namely Atos, CloudSigma, Interoute and T-Systems.

In grand total, more than 1.2 million jobs were submitted to the cloud backed batch queues, based on PanDA and glideinWMS. These WMS are using the same techniques as HammerCloud in order to request computational resources from the cloud infrastructures [5].

Furthermore, this validation was carried out to test resources kindly provided by Rackspace, which could lead to nice future projects, such as cloud federation.

4. Scaling HammerCloud further

The main issue for further scalability for HammerCloud will presumably be the database engine. For the moment, MySQL has behaved correctly and has coped with the entire load being requested to support.

However, the metric storage is becoming bigger and bigger and usually prevents the experiments to request storing more than 20 metrics per job. A new schema or storage engine should be found to allow richer job descriptions and further performance improvements. Also, other options for metric storage must be found, such as NoSQL approaches.

5. Conclusions and future work

The new deployment is a big step forward towards a service that can serve different users and different workloads in a transparent way in relation with the infrastructure underneath. The improvements are already paying off, reducing the amount of time dedicated to operations, optimizing the resources used and, more importantly, making the service more sustainable.

Also, there several other benefits, such as the ability to deploy submission hosts in other clouds and the ease to introduce new automation on the deployment of HammerCloud itself.

On the future, there has to be effort on the shared log storage, in order to allow other tools and systems to profit from correlations and crossed analysis of data coming from different tools. Also, some new testing back ends are in development for HammerCloud, to allow testing not only batch like resources, but any other infrastructure provider.

References

- [1] Improving ATLAS grid site reliability with functional tests using HammerCloud. Legger, Federica and Elmsheuser, Johannes and Medrano, Ramon and van der Ster, Daniel. Computing in High Energy and Nuclear Physics (CHEP). 2012.
- [2] Experience in Grid Site Testing for ATLAS, CMS and LHCb with HammerCloud. van der Ster, Daniel and Elmsheuser, Johannes and Medrano, Ramon and Legger, Federica and Sciaba, Andrea. Computing in High Energy and Nuclear Physics (CHEP), 2012.
- [3] Production Large Scale Cloud Infrastructure Experiences at CERN. Bell, Tim and van Eldik, Jan and Schwickerath, Ulrich and Castro Leon, Jose and Moreira, Belmiro. Computing in High Energy and Nuclear Physics (CHEP), 2013.
- [4] Grid Site Testing for ATLAS with HammerCloud. Legger, Federica and Elmsheuser, Johannes and Medrano, Ramon and van der Ster, Daniel and Sciacca, Francesco. Computing in High Energy and Nuclear Physics (CHEP), 2013.
- [5] Commissioning the CERN IT Agile Infrastructure with experiment workloads. Barreiro Megino, Fernando and Cinquilli, Mattia and Denis, Marek and Kurcharczyk, Katarzyna and Medrano Llamas, Ramon. Computing in High Energy and Nuclear Physics (CHEP), 2013.