

**Lxcloud**: a prototype for an internal cloud in HEP.

Experiences and lessons learned

Sebastien Goasguen, Belmiro Moreira, Ewan Roche, Ulrich Schwickerath

CERN IT, route de Meyrin, 1211 Geneva 23, Switzerland

Abstract.

Born out of the desire to virtualize our batch compute farm CERN has developed an internal cloud known as *lxcloud*. Since December 2010 it has been used to run a small but sufficient part of our batch workload thus allowing operational and development experience to be gained. Recently, this service has evolved to a public cloud allowing selected physics users an alternate way of accessing resources.

1. Introduction

In 2008 CERN launched a virtualization project aiming at the virtualization of batch compute resources. It was setup in a way which strictly distinguishes between infrastructure and guests and is thus able to serve as an Infrastructure as a Service (IaaS) cloud. The system was put into production on a small scale in December 2010 and has grown to almost 500 virtual machines in spring 2011. It was opened to selected users deploying CERNVM [1] images on it in 2011, thereby allowing new possibilities for the delivery of computing resources to users by exploiting a cloud like way of working. This paper provides an overview over the project, its evolution and growth, as well as the different use cases encountered. Operational experiences and issues seen are reported and discussed.

2. Project history and overview

In early 2008 discussions took place at CERN about possible ways to improve the efficiency of its batch resources. At the time it was noted that some experiments were unhappy with the software installations they found at various sites, because it was not up to date or simply because they required newer versions than those found on the worker nodes. In addition, experiments started to submit so called pilot jobs to the sites. A pilot job is an empty job wrapper which examines the environment on the worker node where it starts up and only then receives the user job from a central, experiment controlled, job queue. In many cases this creates an additional and possibly unnecessary overhead. A situation where sites are able to launch experiment specific virtual machines which process tasks delivered directly by the relevant user community was felt to be more efficient. A prerequisite to such a scenario is a scalable infrastructure capable of providing resources to launch images. At the same time it became clear that virtualization is a way to improve resource usage in the services area: experiments run dedicated servers which are located in the computer center and which are typically managed (operating system wise) by IT. At this time, physical nodes were allocated for such use cases, resulting in a proliferation
of underutilized machines. It was decided that these two use cases are sufficiently distinct to launch different projects aimed at their specific requirements.

Batch virtualization use case:
- build on cheap hardware
- low virtualization overhead in CPU, I/O and networking needed
- use of local disk for performance
- easy scalable to a large number of virtual guests, O(10K-100K) and hypervisors
- optimized for performance rather than reliability

Specifically, there is no need for live migration capabilities, snapshotting or shared file systems in this scenario.

Service consolidation use case:
- dedicated hardware
- fail safe disk layout (i.e. remote storage, RAID, ...)
- performance less important
- scalability in terms of number of guests less important, O(1K)
- full support for live migration and high availability

Service consolidation is common practice today. Batch virtualization is only now being discussed again in the context of clouds.

3. Historical evolution of the project
A first proof of concept was presented in June 2009 [2], based on Platform Computing’s Virtual Machine Orchestrator (VMO) [3] tool. Scalability tests were performed in August 2010 when 480 servers were allocated to the project for a limited amount of time to perform those tests. Both Platform’s Infrastructure Sharing Facility (ISF) [4], the successor of VMO, and OpenNebula [5] were used. With OpenNebula it was possible to manage up to 16,000 virtual machines which corresponds to the number of available IPv4 addresses for the tests. The fast ramp up was made possible by integrating the hypervisors with the central management tool Quattor [6] and organizing them in a new host cluster called lxcloud. Scalability of ISF was limited to 10,000 virtual machines which was understood only later when the test nodes were no longer available.

Apart from testing the virtualization provisioning tools, the purpose of the scalability tests was to find the limits of the Platform’s Load Sharing Facility (LSF) batch system [7] which is used at CERN to provide batch services. These tests allowed us to increase the number of worker nodes and jobs in the system by a factor of 4 over what was available with physical systems thus allowing service evolution to be planned. The virtual machines dynamically joined a LSF test instance. Monitoring of the available virtual machines (VMs) was performed via the LSF communication layer rather than just relying on the provisioning system.

In December 2010 the system went into production by providing the first 96 virtual batch nodes in CERN’s production batch farm. This important milestone went entirely unnoticed by the users. During the following year virtual batch resources were ramped up to 432 virtual machines and allowed us to slightly over commit resources. The ramp up was done using OpenNebula only and later on ISF was entirely dropped for the sake of simplicity of the system. Thanks to virtualization lxcloud became the first production cluster at CERN to be migrated to Scientific Linux CERN (SLC) [8] Version 6 as an operating system.

Although not originally designed as an Infrastructure as a Service (IaaS) cloud, lxcloud has many features of a private cloud installation [9]. This allowed CERN to use OpenNebula’s Elastic Cloud compatible interface (ECONE) to offer restricted access to selected users outside IT as well as to perform tests of different cloud computing models.
OpenNebula econe server

OpenNebula server

Image repository

physical resources

Application manager/User

Figure 1. Schematic of the lxcloud infrastructure. Application managers (or end users) send their requests for virtual machines to the EC2 interface of OpenNebula. The virtual machine request is processed by the OpenNebula server and dispatched to the physical resources which consist of a cluster of hypervisors supporting Kernel Based Virtual Machines (KVM) [11]. The images are stored in a central image repository from where they are distributed to the hypervisors using rtorrent [12].

4. Design decisions
When virtualizing batch resources one can think of using the batch farm to schedule the virtual machines directly on the hypervisors and to launch virtual machines as part of the job itself. The advantage of this approach is that one has all the scheduling capabilities of the batch system and that one can profit from the already existing accounting capabilities. This approach has been taken by INFN [10].

For lxcloud we explicitly decided to separate the batch farm from the resource provisioning layer. On lxcloud the software installed on the hypervisors is limited to that which is necessary to host virtual machines. Batch resources are provided as virtual machines running the batch client software. Software wise, our virtual batch nodes run exactly the same software stack as physical worker nodes.

Aiming at batch like applications it was decided not to support live migration or snapshotting features. Instead, the system was designed to be fast in order to optimize the resource usage.

5. IaaS infrastructure
The layout of lxcloud is shown in figure 1.
5.1. Hypervisors: The lxcloud cluster

The hypervisors currently in use are off-the-shelf two socket servers with two or three local hard disks. They run Scientific Linux CERN (SLC) version 6.3, with KVM as virtualization technology. The operating system is installed on the local disk and all remaining disk space is organized in a single large logical volume. This logical volume provides the required storage for virtual machine images, instances and their scratch disks. Instances are created as snapshots of locally stored images. This approach required a few patches to the OpenNebula, which have been partially contributed to the project but provides a very rapid way of launching virtual machines. Lxcloud hypervisors, as well as all other machines running the required services, are fully integrated into CERN’s central management system. This includes software and configuration management as well as monitoring and alarming. CERN uses the Quattor tool kit [6] to centrally manage resources.

5.2. Image creation

Thanks to the generic setup of the lxcloud infrastructure it is possible to import any virtual machine image and run it. This feature has been tested with a variety of images, including CERNVM.

A special way to create virtual machine images is to use so called golden nodes. The idea is described in figure 2. A golden node is a virtual machine which lives outside the IaaS infrastructure and which is fully integrated into the central management. Therefore, it automatically receives all updates for the cluster to which it belongs. It does not run any production service though and can be stopped at any time to create an updated image.

The golden node approach is specifically useful for virtual batch resources where it is required that virtual and physical resources are synchronized in terms of their software setup and configuration. Virtualized batch resources mix seamlessly with physical resources and execute normal payload just like any other worker node in the public batch resources.

An alternate way to synchronize virtual and physical batch resources is to centrally manage all of them. The disadvantage of this approach is that it implies a fairly static setup which makes it difficult for virtual machines to be reused for different purposes.

5.3. Image repository

Images are stored in a central image repository to which access is restricted. End users have no access to it so only administrators can upload and deploy images. Virtual machines can only be instantiated from images in the central repository thus ensuring that only trusted images are used on the infrastructure.

The image repository is at the same time a local Virtual Machine Image Catalog (VMIC) [13]. Image exchange between sites is under development within the HEPiX [14] community.

5.4. Image distribution

Images are distributed from the central image repository to the hypervisors using the bit-torrent protocol. Measurements of the performance have been presented in [12]. The images are stored persistently on the hypervisors which allows for a very rapid instantiation of virtual machines. Special care has to be taken to ensure that running virtual machines are not affected if the base image is updated. Our home grown image distribution system takes care of this by making use of the renaming capabilities of the logical volume manager (LVM). It also ensures the integrity of the transferred images before deployment by calculating a MD5 check-sum of the image and comparing the result with the (signed) meta data.
Golden nodes are one possible way to create trusted images. A golden node is a statically allocated virtual machine which is fully Quattor managed but does not run any production service. Whenever a major update of the virtual machines is required, the golden node is stopped, and a snapshot of it is put into the central image repository. After the distribution of this new image, newly created instances will start from the updated image.

5.5. Contextualization support
Contextualization provides a way to customize virtual machines at boot time. OpenNebula creates a contextualization ISO image which is attached to the instance and which contains a set of scripts. If the virtual machine image supports it these scripts can be executed at boot time. For the virtual batch application, contextualization is used to synchronize the software setup and node configuration with the current status of the golden node. It also sets up scratch space and AFS cache space, if needed, on a local disk for better performance.

Other applications, notably CERNVM images, use the contextualization to process user provided parameters and data. In this manner the same image can be used by different user communities.

5.6. Networking
It was necessary to integrate lxcloud with CERNs networking infrastructure.

Virtual machines are pre-registered in the networking database so that MACs and IPs are known to the central DHCP servers. OpenNebula is provided with a lease file which contains a list of valid MAC-IP pairs allowing the scheduler to choose a valid pair when a new virtual machine is to be created. In addition, it has a list of hypervisor candidates to which it can dispatch the request. The selection is based on the monitoring information and scheduling requirements which need to be matched.

Hypervisors publish a list of MACs which they support to OpenNebula through the host
monitoring and virtual machine templates contain a requirement telling the scheduler that the MAC of the virtual machine must match that list. This trick is CERN specific and necessary to ensure that virtual machines are started on hypervisors in the correct network segment, else network traffic to and from the virtual machine will not be routed.

5.7. External disks
In the initial setup, virtual machines on lxcloud were bound to specific hypervisors. Once the MAC was selected from a lease file, the hypervisor was fixed. The advantage of doing this is that it is possible to support persistent external disks per virtual machine without having the need for a shared file system. The main use cases are short lived virtual machines which require large and fast cache files such as AFS [15] or CVMFS [1]. Refilling these caches with potentially identical contents after recreation of the virtual machine (VM) is not efficient.

Persistent disks were created at the installation time of the hypervisor, one set of disks per virtual machine. Each disk is identified by the MAC of the virtual machine. The persistent disks are configured in the virtual machine template file for OpenNebula. The additional disks can then be mounted during the contextualization process.

The disadvantage of this approach is that it leaves no room for scheduling of virtual machines. Once a MAC is chosen there is only one hypervisor candidate. If this specific machine is not available the VM stays pending. Therefore, in spring 2012 the static binding of VMs to hypervisor was relaxed. The mechanism is now used to ensure that virtual machines are dispatched to a hypervisor in a proper network segment. Disks are no longer persistent but are created locally by OpenNebula at instantiation time of the virtual machine and partitioned and configured during the contextualization. In order to maintain an efficient system, the average allowed life time of the virtual batch nodes was increased from 2 days to 2 weeks which is considered to be a more realistic scenario in the long term.

6. Resources and Experiences

6.1. Hardware
Lxcloud currently consists of

- 48 servers (Dual L5520 CPU, 24GB Memory, 2x 500 GB local disks, SLC6.2 OS) for production purpose
- 13 servers (Dual L5520 CPU, 24GB Memory, 3x 1 TB local disks, SLC6.2 OS) for public cloud testing
- 1 dedicated hypervisor for image creation
- 1 disk server, used as image repository

It is controlled by a single OpenNebula instance (version 3.2). As mentioned earlier, all hardware is Quattor managed, monitored and alarmed.

6.2. Virtual machine slots
All lxcloud hypervisors have in total 8 physical cores. Each hypervisor has 9 slots for virtual machines by default which are entered into the network database at the installation time of the hypervisor. The additional slot allows for a slight over-commitment of resources. Scheduling requirements based on the CPU usage, memory, and the number of running virtual machines guarantee that virtual machines are dispatched in an optimal way.

An external disk is attached to each virtual machine. It is visible as a block device and can be configured during contextualization time, to provide for example local scratch space or AFS cache space.
6.3. User registration

Lxcloud is not integrated into CERN’s Single Sign On (SSO) mechanisms, therefore users have to explicitly request access. Upon agreement, they receive a personal user name and password hash combination which allows them to authenticate to the system. In addition, the system is only visible to users inside CERN. Access from the Internet is blocked by firewall rules so that only CERN registered users can sign up for the service. Currently about 20 users are registered. This includes IT and support staff.

6.4. Applications

Lxcloud is currently in use for the following applications:

- virtualized batch resources
- build servers for CERN’s physics applications group (SFT)
- development servers running various flavors of Fedora
- CERNVM based tests of new computing models based on IaaS (performed by several user communities at CERN)

6.4.1. Virtual batch

48 hypervisors with 1 TB local disk space are dedicated to the virtual batch farm application. Virtual batch resources are requested from a special account through the Elastic Cloud compatible interface of the OpenNebula server. A cloud factory script running as a cron job every 10 minutes fills up the resources with virtual machine requests until it runs over quota. This simple rule ensures that all resources are always fully used. Currently only single core SLC5 worker nodes are supported in the virtual batch farm. In a future large scale virtualized batch system it is planned to extend this script to support several flavors of worker nodes. One possible scenario is that the script checks the current demand by checking pending requests in the batch farm. It will then start virtual machine flavors to match the current demand. This approach only really works if the batch system supports dynamic adding and removing of hosts which will change their flavors and features every time. While this is in principle possible with LSF it was found not to be stable enough for a production deployment at CERN.

Virtual batch worker nodes mix with physical resources and run public jobs both from local and grid submissions. Each virtual batch worker node runs only one job at a time and provides one virtual CPU. A mechanism has been added to these instances which puts them into draining mode after a certain amount of time (currently two weeks) and destroys the instance via a call back to OpenNebula after the last job has finished. The virtual machine is restarted automatically by the cloud factory script and, as images can change at any time, new instances are started from the new image. Thanks to this approach:

- intrusive interventions are entirely automated
- draining of hypervisors is easy and does not need any human intervention

If a hypervisor needs to be emptied a system administrator takes it out of production using standard CERN tools. This procedure automatically disables the hypervisor in OpenNebula thus preventing new VMs from being dispatched to it. From the system administrator perspective, the procedure to take a hypervisor out of production is identical to the procedure for physical batch nodes.

6.4.2. CERNVM

A CERNVM image is provided on lxcloud for testing new computing models. Instances are configured during the contextualization using the user data which is passed in when the request for a virtual machine is launched. From the operational point of view these tests
run in the background transparently. Thanks to contextualization only one image is needed for all interested user communities which results in these tests being operationally cheap.

6.4.3. Other applications  Other early adopters have provided images which have been endorsed and distributed to lxcloud resources. The endorsement process generally requires several iterations until all required virtual machines behave as expected.

6.5. Experiences  CERN is running 432 of its public worker nodes as virtualized resources on lxcloud in a manner which is entirely transparent to the users. Since initial deployment into production the system has been updated several times by deploying new versions of OpenNebula as well as the base operating system on the hypervisors.

The core of the system, the infrastructure itself, turns out to be very resilient against changes. Most issues have been spotted on the application side. As an example, CERNs virtual batch worker nodes require AFS because the local users have their home directory on AFS. Issues have been seen depending on the work load of the virtual machines and the kind of application which is run on it. This is not specific to virtual batch worker nodes and has been observed on physical batch worker nodes as well, albeit at a lower rate. While on physical worker nodes all associated jobs are lost in case of a system failure, for virtual batch nodes other VMs running on the same hypervisor continue to work.

From the operational aspect, the system runs for a long period of time essentially unattended without major issues. Most problems which we have seen were related to system and software upgrades, or hardware failures, specifically disk issues.

Recent updates to Scientific Linux 6.3 introduced a change in the behavior of LVM. This unexpectedly affected running instances during the deployment of a new image version. This issue required a work around in the image distribution system and is still under investigation.

7. A vision of virtualization and clouds in high energy physics
Virtualization technologies can help sites to improve their resource usage and reduce operational costs. Lxcloud was designed with this idea in mind, with the focus on large batch processing farms.

It is believed by many people that sites can profit from organizing their computing resources in the form of a private cloud allowing resource sharing. It is less clear though if and why such infrastructures, once they are in place, should be opened to end users, or not, as the case might be.

Some of the lxcloud resources have been opened to selected users from different physics communities at CERN to test if they can exploit such resources at sites. Based on this, a new computing model could look like that shown in figure 3. The model assumes that each user group has an image maintainer who is trusted by the different sites running an IaaS cloud. These image maintainers provide virtual machine images which obey site restrictions and which work for their Virtual Organization (VO). Images are imported by the sites and distributed internally so that the VOs can instantiate virtual machines. As it is the case now end users submit their jobs to the experiment task queue (rather than the batch system or the grid) from where virtual machine instances fetch and execute it.

While the proof of concept of such a scenario has been done successfully, including tests on lxcloud, many details still need to be clarified. For example root access for users to instances is likely to be disallowed by policy at some sites. This can be for technical reasons, such as the networking infrastructure which may not be easy to change. Consequently, not all IaaS clouds at all sites will support the upload of user owned images as is the case for commercial providers. Instead, the image life cycle needs to be controlled and a chain of trust has to be
established to allow individual images to be instantiated at different sites. This issue has been discussed in the HEPIX virtualization working group and has been picked up by the European Grid Infrastructure project, EGI [16]. Further discussion is required on how to secure the image life cycle which consists of the following steps:

- image creation
- image audit and endorsement
- image update publication
- image import by sites
- image instantiation
- image contextualization
- image revocation

Other open questions include

- authentication and authorization in a federated world of clouds
- accounting and billing models
- draining and freeing of resources

### 7.1. Authentication and Authorization

Traceability of user activity, a key requirement in the grid world, remains a source of concern in a cloud world. In the described scenario much of the authentication and authorization of users and user payloads is pushed to the experiment pilot job frameworks. The person requesting the

![Figure 3. Exploitation of resources in IaaS clouds. A possible scenario.](image)
virtual machines at sites replaces the pilot job submission role. In the case of a misbehaving user job sites will have to rely on the experiments co-operation to identify and eventually ban the problematic user.

7.2. Accounting and billing
Accounting and billing models are rather different in a cloud world with respect to what is in place in the high energy physics world today. At present accounting is based on the CPU usage of batch jobs. Experiments receive an allocation and don’t directly pay for resources. In the cloud world the paradigm is to pay for what you get whether or not it is used. If such an economic model is suitable for the high energy physics world is a question which needs to be addressed.

8. Conclusions
Originally designed for virtualizing unreliable resources at a large scale, lxcloud has evolved into a reliable and scalable infrastructure which has all features of an IaaS cloud. The infrastructure is mainly in use for providing virtualized batch nodes which mix with physical batch nodes at CERN and has been used since December 2010. Experiences with the infrastructure are rather positive and a small part of it has been opened for testing direct IaaS access by three major physics user communities at CERN. IaaS clouds open possibilities for entirely new computing models which are expected to be further explored in the coming years.

It is planned that lxcloud will be absorbed in a new project which has been launched in 2011 at CERN and which is built on the idea of providing all resources on top of a single private IaaS infrastructure.

[1] http://cernvm.cern.ch/portal
[2] Workshop on adapting applications and computing services to multi-core and virtualization, June 24-26 2009, CERN
[3] http://www.platform.com/press-releases/2008/platform-computing-announces-vm-orchestrator-4
[4] http://www.platform.com/Products/isf
[5] http://opennebula.org
[6] http://quattor.org
[7] http://www-03.ibm.com/systems/technicalcomputing/platformcomputing/products/lsf
[8] http://cern.ch/linux
[9] http://en.wikipedia.org/wiki/Infrastructure_as_a_service
[10] http://web.infn.it/wnodes/index.php/wnodes
[11] http://www.linux-kvm.org
[12] Image Distribution Mechanisms in Large Scale Cloud Providers, R. Wartel et al, Proceedings of 2nd IEEE International Conference on Cloud Computing Technology and Science, Nov. 2010, Indianapolis, USA
[13] https://twiki.cern.ch/twiki/bin/view/HEPIX/VMIC
[14] http://hepix.org
[15] http://openafs.org
[16] https://wiki.egi.eu/wiki/SPG:Drafts:Virtualisation_Policy