Status and Roadmap of CernVM

D Berzano, J Blomer, P Buncic, I Charalampidis, G Ganis, R Meusel
CERN, CH-1211 Genève 23, Switzerland
E-mail: gerardo.ganis@cern.ch

Abstract. Cloud resources nowadays contribute an essential share of resources for computing in high-energy physics. Such resources can be either provided by private or public IaaS clouds (e.g. OpenStack, Amazon EC2, Google Compute Engine) or by volunteers computers (e.g. LHC@Home 2.0). In any case, experiments need to prepare a virtual machine image that provides the execution environment for the physics application at hand. The CernVM virtual machine since version 3 is a minimal and versatile virtual machine image capable of booting different operating systems. The virtual machine image is less than 20 megabyte in size. The actual operating system is delivered on demand by the CernVM File System. CernVM 3 has matured from a prototype to a production environment. It is used, for instance, to run LHC applications in the cloud, to tune event generators using a network of volunteer computers, and as a container for the historic Scientific Linux 5 and Scientific Linux 4 based software environments in the course of long-term data preservation efforts of the ALICE, CMS, and ALEPH experiments. We present experience and lessons learned from the use of CernVM at scale. We also provide an outlook on the upcoming developments. These developments include adding support for Scientific Linux 7, the use of container virtualization, such as provided by Docker, and the streamlining of virtual machine contextualization towards the cloud-init industry standard.

1. Introduction
The way computing resources are today becoming available to high-energy physics (HEP) experiments is multi-fold and changing. The common denominator of this moving scenario is the cloud concept in its broader term, ranging from dedicated commercial or public facilities to resources used in opportunistic way, e.g. harnessing volunteer machines or leftover cycles in large farms mainly dedicated to other tasks.

It is known that the cloud paradigm is intimately related to virtualization. Virtualization provides a way to disantangle services from hardware, making possible a flexible and elastic use of resources. It also provides a mean to reduce the weight of software portability activities, historically a prominent component of software maintenance work in the HEP experiments.

One of the main goals of the CernVM project has been to address the portability issue in efficient way, removing the need for the installation of the experiment software and minimizing the number of platforms (compiler-OS combinations) on which experiment software needs to be supported and tested.

The CernVM project evolved in ecosystem of products, addressing the issues of main relevenance for HEP: software distribution, by introducing a dedicated file system, CernVM-FS, extensively used also in other contexts [1]; and virtual machine configuration, by leveraging the concept of virtual appliance.
The CernVM appliance is widely used at LHC. From web server logs, where the virtual machines call on boot, it is possible to estimate to about 1.2 million per month the number of virtual machines reboots, about 200k out of which corresponding to fresh boots. Although it is not possible to identify exactly the context of each of these boots/reboots, from the nature of the support requests received, we can infer that the usage of the appliance is spread across all the LHC experiments.

At the last edition of this conference a new technology for creating the virtual appliances was introduced [2], resulting in a minimal and versatile virtual machine image capable of booting different operating systems, delivered on demand by the CernVM File System. The significant reduction in the effective size of the virtual machine (order of 100 MB are transferred during startup) makes the new technology particularly fitted for large scale distribution in different scenarios. The possibility to choose the operating system increased the attractiveness of the CernVM ecosystem for software environment preservation and, more generally, open data activities.

This paper focuses in particular on the status of the new bootloader technology, which in the meantime has matured to production grade, and the future steps concerning the appliance, including support for operating system level virtualization a.k.a. containers.

For completeness, we will also review briefly the status of the other components of the ecosystem.

1.1. The CernVM ecosystem
The locution CernVM ecosystem denotes the ensemble of the following products: the read-only file system (CernVM-FS); the appliance (CernVM); the contextualization web portal (CernVM-Online); and the toolkit to create computing infrastructures (CernVM-Copilot) extensively used by the LHC@Home 2.0 project [4] but now in a sort of legacy state.

CernVM-FS [3] has become the tool to distribute software and a critical service for WLCG [1]. It is a read-only file system, optimized for small files, with a single injection point and a delivery network for optimal performance. CernVM-FS embeds deduplication properties and automatic file integrity check. CernVM-FS allowed to offload from the image the experiment software. In CernVM 3 this concept is pushed further offloading from the image the operating system software. CernVM-FS has acquired therefore an even more central role in the ecosystem.

CernVM-Online is a web portal. Its purpose is collect the contextualization information (users, credentials, repositories, services, ...) and to feed it to laptop/desktop virtual machines, or to create an user data file to be used when instantiating machines on clouds. Contexts for different virtual machines can be saved and even shared in the markeplace, an area visible to all users.

The unique feature of this tool is the possibility to start cluster of machines with structured roles, such as master-worker architectures. This is at the base of the Virtual Analysis Facility, which was also presented in the previous edition of this conference [5].

2. Bootloader technology in a nutshell
A schematic view of the bootloader technology at work is presented in Figure 2. The process to create a virtual machine contextualized at the case in use starts with a minimal image, referred to as $\mu$CernVM, consisting of a Linux kernel and an init RAM disk which includes the CernVM-FS client. The Linux kernel is virtualization friendly, slimmed down to the essential components required by the handful of hypervisors around. This results in a reduction of at least one order of magnitude in size and number of modules with respect to kernels found in standard distributions, such as, for example, Scientific Linux 6.

---

1 The portal can be accessed at https://cernvm-online.cern.ch/dashboard/
To overcome the fact that CernVM-FS is read only, the µCernVM image comes with the union file system AUFS [6] which allows to provide a writable layer, stored on the local hard-disk, transparently expanding the read-only layer from CernVM-FS. The local hard-disk also stores the CernVM-FS cache.

The virtual machine root file system stack is created out of the writable scratch area and the read-only template taken from CernVM-FS; this is done by a steering script included, together with the CernVM-FS client into the initramdisk. The steering script can process contextualization information - also called “user data” - from various common sources, such as OpenStack, OpenNebula or Amazon EC2. It is at this point that the CernVM-FS repository and the respository version are selected.

The template for the Operating System (OS) to be loaded is installed on CernVM-FS using the OS package manager. New and updated packages can be installed incrementally by the package manager on top of an existent installation. The fact that CernVM-FS is a versionning file system is exploited by µCernVM to avoid silent and unwanted updates: the CernVM-FS client in the VM will stay on the snapshot chosen at boot (via the relevant contextualization directive) and stick to that.

The OS is defined by a set of packages and their dependencies, which are controlled by a meta package named cernvm-system. To account for different levels of maturity, three repositories are maintained: production, testing and development. This separation allows also the deployment of security hotfixes on the production more efficiently, with minimal impact for the testing and/or development activities.

Table 1 gives the hypervisor and cloud controller support status.

2.1. Experience with the bootloader technology

Experience with the bootloader technology was overall very positive. The choice of the 3.10 LTS kernel for µCernVM provided stability to this component, with only 5 security hotfixes required in 18 months, and proved to be suited for all the operative system templates which
Table 1. Hypervisor and cloud controller support status at the time of finalizing this paper; full support for Docker and Microsoft Azure was added after the conference.

| Hypervisor          | Support |
|---------------------|---------|
| VirtualBox          | Yes     |
| VMware              | Yes     |
| KVM                 | Yes     |
| Xen                 | Yes     |
| Microsoft Hyper-V   | Yes     |
| Parallels           | No      |
|                     | Unclear license of the guest additions |

| Could controller    | Support |
|---------------------|---------|
| Openstack           | Yes     |
| OpenNebula          | Yes     |
| Amazon EC2          | Yes     |
|                     | Only tested with ephemeral storage (no EBS) |
| Google Compute Engine| Yes    |
|                     | Only `amiconfig` contextualization |
| Docker              | Yes     |
|                     | Version 3.4 |
| Microsoft Azure     | Yes     |
|                     | Version 3.5 |

Table 2. Deployment of the CernVM default appliance based on the bootloader technology

| Version | Date        | Kernel | SL ref | Remarks                          |
|---------|-------------|--------|--------|----------------------------------|
| 3.1     | 31/1/2014   | 3.10   | 6.5    | Initial production               |
| 3.2     | 3/2014      | 3.10   | 6.5    | Consolidation. Support for GCE   |
| 3.3     | 27/5/2014   | 3.10   | 6.5    | Further consolidation            |
| 3.4     | 5/2015      | 3.10   | 6.6    | Support for containers. Downgrade|
| 3.5     | 8/2015      | 3.10   | 6.7    | Support for MS Azure, Cloudstack |
| 4.0     | 9/2015      | 3.18   | 7.1    |                                  |

were tested. CernVM users smoothly moved to the new version: by March 2015 we could count almost 27000 new instances of the CernVM 3 basically from all parts of the world inhabited countries. Since all major experiments did move to CernVM 3 for their cloud-related activities, the previous SL5-based version, CernVM 2, could be declared end-of-life by September 2014.

Table 2 shows the main steps of the deployment for the past and foreseen. As it can be seen, after a warm-up phase of about four months, during which a number of advanced users were asked to try out the new image, a stable state has been reached which is basically untouched since about one year.

The meaning of the CernVM version numbers has somehow changed with the advent of the bootloader technology, since the same μCernVM allows to boot different OS. The major and minor version number are now related to the default flavour of the OS: for example, the current default CernVM 3.3 creates an SL 6.5 compatible version virtual machine. Security hot fixes are

---

2 LHCb users start using the new version as soon as it was available and have provided invaluable feedback. The new version was also part of the virtual analysis facility [5] which was open to public in the same period.
provided following the vendors releases: for version 3.3, which proved to be remarkably stable and production grade, there have been 25 of such fixes, typically made available a few days after the official release by the Scientific Linux 6 maintainers.

As mentioned above, one of the main aspects of the bootloader technology is the size reduction. The current version of µCernVM is about 20 MB and experience shows that an additional 80-100 MB need to be transferred from CernVM-FS to complete the whole process of instantiating the VM. In both cases these numbers refer to uncompressed data, and have to be compared with the 300-400 MB of compressed data which are required by an equivalent virtual machine creation with standard techniques. One of the results of this significant reduction in data transfer is that the virtual machine starts in practice almost instantaneously, enabling possibilities not yet fully investigated. For example it is now possible to integrate the technology into a website and start the virtual machine on the click of a button [7].

2.2. Other examples of usage

Since the beginning, the CernVM appliance applicability range went beyond the initial goal of providing an effective way to recreate a running and development software environment of an experiment on a laptop/desktop or on a cloud. Data analysis preservation and volunteer computing are two domains where CernVM had already an impact. The bootloader technology has even increased the potential of CernVM in these areas.

2.2.1. Software preservation, public reach
The possibility to choose the OS flavour for the VM has a deep impact on data preservation activities where one of the main issues to be addressed is the preservation of the software environment required to process the data or re-run a given analysis. The potential of the CernVM bootloader technology has been demonstrated in two cases.

The first case is an example of completed experiment requiring an old version of the OS. The ALEPH experiment was completed in year 2000 and had its software stack lastly validated under Scientific Linux 4. The second case is an example of running experiment which took the decision of processing the data taken in a given year with the software environment available at the moment of data taking. This is the case of CMS and data taken in 2010. These data were processed and validated with a Scientific Linux 5 (SL5) and any analysis task run on them must be performed with an SL5 compatible software environment.

With the bootloader technology we successfully managed to recreate software environments for these two demonstrator cases. Scientific Linux 4 and 5 templates are available under CernVM-FS and can be used to instantiate virtual machines to run the ALEPH and CMS software stacks. Note that the SL5 template allows to recreate an environment equivalent to the one provided by CernVM version 2 but with a completely different technique.

The provision of the SL5 template was part of the CMS CERN OpenData [10] exercise. De facto CernVM turns out be one of the enabling technologies of the OpenData pilot project. The components required to run the analysis examples of the participating experiments - CMS, ALICE and LHCb - have all been encapsulated in a CernVM-based OVA image and distributed via the OpenData portal.

2.2.2. Volunteer computing
Another domain where CernVM proved to be a very relevant actor is the exploitation of opportunistic resources, typically provided by volunteers. The LHC@Home 2.0 project has shown how virtualization may help in solving the portability issues often affecting the use of volunteers resources. As reported in this conference [8], a lot of interest

3 Open Virtualization Archive, an open standard for packaging and distributing virtual appliances or more generally software to be run in virtual machines.
in using such resources is raising among the experiments. The use of CernVM-FS and CernVM in these activities is almost ubiquitous. In particular, the reduced size to be distributed is coping well and helping with the variety of network bandwidths encountered in the volunteer space.

3. The future of the CernVM appliance

The role that the CernVM appliance is having in the area of HEP computing is increasing thanks also to the diffusion of the cloud concept. This implies a commitment to keep it in phase with the requirements of the experiments, which means consolidation of the components and services mostly used and updates to the most recent versions. On the technology side, there is a lot of recent interest in understanding what containers can bring to HEP computing models. Several presentations at the conference have shown more or less advanced investigations along those lines. Given also the role that CernVM-FS has acquired, this is something that cannot be ignored in the context of CernVM.

The bootloader technology also raised a lot of interest and the request for making the OS template creation accessible in a more generic way, also to go beyond the Redhat-based distribution world, has come in several occasions and may be subject of a future development ⁴.

3.1. Appliance

On the consolidation / update side, the roadmap is defined, with the recent releases of Scientific Linux 6.6 and 7.1. Scientific Linux 6.6 is going to be the default OS for CernVM 3.4. In addition to the OS update, this version will feature the possibility to unroll updates, for example when they fail, and will add full support for containers. The latter feature opens the way to two main use cases: the possibility to effectively convert CernVM in a different Linux flavour (we could, for example, run Ubuntu 14.10 inside a Docker container); the possibility to better partition the usage of the virtual machine using containers to encapsulate different environments required by multiple tasks running on the machine. In relation to the latter, a tool called cernvm-fork [9] is available to clone in a container, via fork, the current status of the machine with a change in the running environment. An example of use is the assignment of a multi-core volunteer machine to run several projects.

Scientific Linux 7.1 will be the default OS for CernVM 4.0. This implies a more fundamental change that the mere change of OS version. In fact, starting with version 7, RHEL adopted the systemd technology as basic building block, replacing the old init. Given the role that CernVM-FS has in booting process from the early phases, this fundamental change requires the adaptation of CernVM-FS as low level storage daemon in the systemd suite. Another important change occurred in RHEL 7 is the use of Unix capabilities for permission checks instead of the setuid/setgid family; support for this is required in CernVM-FS and required implementations. Implementations for these changes are available and are at the base of the CernVM 4 prototype under development.

3.2. Leveraging container technology in CernVM

Containers have raised interest in the HEP world as a lightweight to create encapsulated running environments on top of a running system. Containers are not full replacements of virtual machines, but in some cases they can provide a more efficient way to exploit multi-core machines. As explained in [11] they could also provide a way to effectively restore elasticity in those cloud systems overcommitted or scarce in resources⁵.

⁴ The relevant codebase for the template creation process is already available publicly available in github; however documentation is lacking.

⁵ This is for example the case of CERN Openstack, as explained in [11].
When it comes to CernVM, the ideal goal of having CernVM as a container needs to solve two main issues: making the CernVM-FS repositories visible inside the container and, since the OS comes also from CernVM-FS, providing a writable overlay for the container instance.

For CernVM-FS, the situation depends on the user having administrator rights on the host machine or not. In the former case, it is possible to mount CernVM-FS in the host machine in a way that can be bind mount inside the container with the cache shared among the containers. This promising technique is being developed within the context of ALICE next generation facilities and described in extent in [11].

In the case administrator rights are not available, a pure user-space solution is provided by the Parrot [12]. A connector dedicated to CernVM-FS exists since a while, used, for example, in CMS and ATLAS activities on HPC resources. This can be used to recreate a CernVM shell on generic machines such as those of the lxplus CERN cluster. The current implementation of the Parrot connector still suffers from stability and performance issues, which are being addressed together with the Parrot developers.

Once the problem of efficiently mounting CernVM-FS inside a container is solved, the main remaining problem is to provide an writable overlay on top of the read-only CernVM-FS layer, to fully realize the CernVM technology. As described in [11] a feature of Docker containers can be of very much help here. Normal containers need also a writable overlay and it turns out that Docker provides this using AUFS, the same overlay file system that CernVM uses. It is possible to use the Docker overlay as CernVM overlay. This solution is being investigated and prototyped in the context of ALICE [11].

4. Related work
The bootloader technology has raised some interest in the community of the Linux distributions. For example, the Gentoo Fundation has proposed a project within the GSoC program to investigate the possibility to have a Gentoo image with this technology [13].

A lot of work is about the use of containers and integration of CernVM-FS was presented at this conference. See for example [14]. However, the solutions mentioned above and detailed in [11] seem to be a step forward.

5. Conclusion
CernVM 3 is based on a novel bootloader technology which has proven to be a solid and efficient way to provide tailored virtual machines. The technology has successfully demonstrated its potential addressing a variety of use cases. It allowed to recreate old version of the operating system in the context of data and analysis preservation activities and it is one of the enabling technologies of the CERN Open Data pilot project.

Current development activities include providing full support for Scientific Linux 7 and for containers. Both of these developments are well under way.

References
[1] R Meusel et al., Recent Developments in the CernVM-File System Server Backend, to appear in proceedings of ACAT 2014, Prague, Czech Republic.
[2] J Blomer et al., Micro-CernVM: slashing the cost of building and deploying virtual machines, 2014, J. Phys.: Conf. Ser. 513 032007.
[3] J Blomer et al. Distributing LHC application software and conditions databases using the CernVM file system, 2011, J. Phys.: Conf. Ser. 331 042003.
[4] http://lhcathome.web.cern.ch/vLHCathome.
[5] D Berzano et al. PROOF as a Service on the Cloud: a Virtual Analysis Facility based on the CernVM ecosystem, 2014 J. Phys.: Conf. Ser. 513 032009.
[6] C P Wright et al. Versatility and unix semantics in a fan-out unification file system, Tech. Rep. FSL-04-01b Stony Brook University.
[7] I Charalampidis et al. *CernVM WebAPI - Controlling Virtual Machines from the Web*, proceedings of this conference.

[8] M Marquina et al. *Towards a production volunteer computing infrastructure for HEP*, proceedings of this conference.

[9] I Charalampidis, http://github.com/cernvm/cernvm-fork.

[10] http://opendata.cern.ch/.

[11] D Berzano et al. *Lightweight scheduling of elastic analysis containers in a competitive cloud environment: a Docked Analysis Facility for ALICE*, proceedings of this conference.

[12] Part of Cooperative Computing tools: http://ccl.cse.nd.edu/software/parrot/

[13] Micro-Gentoo in http://wiki.gentoo.org/wiki/Google_Summer_of_Code/2015/Ideas

[14] A J Washbrook et al. *Evaluation of containers as a virtualisation alternative for HEP workloads*, proceedings of the this conference.