PROCEEDINGS OF SCIENCE

LHC Cloud Computing with CernVM

Ben Segal[1]
CERN
1211 Geneva 23, Switzerland
E-mail: b.segal@cern.ch

Predrag Buncic
CERN
E-mail: predrag.buncic@cern.ch

13th International Workshop on Advanced Computing and Analysis Techniques in Physics Research - ACAT 2010
Jaipur, India
February 22–27 2010

[1] Speaker

PoS(ACAT2010)004

PROCEEDINGS OF SCIENCE

David Garcia Quintas
CERN
E-mail: david.garcia.quintas@cern.ch

Carlos Aguado Sanchez
CERN
E-mail: carlos.aguado.sanchez@cern.ch

Jakob Blomer
CERN
E-mail: jakob.blomer@cern.ch

Pere Mato
CERN
E-mail: pere.mato@cern.ch

Artem Harutyunyan
Yerevan Physics Institute, Armenia
E-mail: artem.harutyunyan@cern.ch

Jarno Rantala
Tampere University of Technology, Finland
E-mail: jarno.rantala@tut.fi

David J. Weir
Imperial College, London, UK
E-mail: david.weir03@imperial.ac.uk

Yushu Yao
Lawrence Berkeley Laboratory, USA
E-mail: yao.yushu@gmail.com

13th International Workshop on Advanced Computing and Analysis Techniques in Physics Research - ACAT 2010
Jaipur, India

PoS(ACAT2010)004

PROCEEDINGS OF SCIENCE

February 22–27 2010

Abstract

Using virtualization technology, the entire application environment of an LHC experiment, including its Linux operating system and the experiment's code, libraries and support utilities, can be incorporated into a virtual image and executed under suitable hypervisors installed on a choice of target host platforms.

The Virtualization R&D project at CERN is developing CernVM, a virtual machine designed to support the full range of LHC physics computing on a wide range of hypervisors and platforms including end-user laptops, Grid and cluster nodes, volunteer PC's running BOINC, and nodes on the Amazon Elastic Compute Cloud (EC2). CernVM interfaces to the LHC experiments' code repositories by means of a specially tuned network file system CVMFS, ensuring complete compatibility of the application with the developers' native version. CernVM provides mechanisms to minimize virtual machine image sizes and to keep images efficiently up to date when code changes.

We also describe Co-Pilot, an interface to the LHC experiments' job submission and workload management systems (e.g. ATLAS/PanDA and ALICE/AliEn), allowing clouds of CernVM-equipped worker nodes to be accessed by the experiments without changing their job production procedures. Currently supported clouds include Amazon EC2, private clusters, Tier3 sites, and a cloud of BOINC volunteer PC's which represents a very large potential resource, so far untapped by the LHC experiments.

This paper presents the current state of development of CernVM and Co-Pilot support for LHC cloud computing.

13th International Workshop on Advanced Computing and Analysis Techniques in Physics Research - ACAT 2010
Jaipur, India
February 22–27 2010

PoS(ACAT2010)004

PROCEEDINGS OF SCIENCE

1. Introduction: Grids and Clouds

A traditional means of supplying computing power to research projects has been the development of "research Grids". These are essentially distributed federations of large computing clusters belonging to research institutes and operated by them. An alternative for smaller research groups or institutes has been the installation and support of dedicated local computing clusters, with all the operational and financial overheads that this entails.

A recent change to this traditional model has seen the emergence of "computing clouds". These allow efficient leverage of the installed capacity of large computing centers by offering attractive rented chunks of processor power and storage to consumers over the Internet. Providers like Amazon, Google, IBM and Microsoft can benefit from the economies of scale associated with their highly optimized installations, and customers can benefit by simply using resources when it suits them and incurring no overheads or wasted cycles when they are unused.

2. Virtualization

A key technology that has enabled such cloud development is virtualization. This permits logical separation of an underlying physical computing fabric, installed in a computing centre, from the users' view of the computing resources provided by this fabric. In particular, underlying physical platform characteristics such as operating system, memory and CPU configuration, and I/O connectivity can be abstracted into virtual machines of chosen standard type(s) which can then be further custom-configured for (or by) the end-users for the period of their resource occupation. For an example of such a service, see details [1] of the Amazon Web Services Elastic Compute Cloud (EC2) and Simple Storage Service (S3).

Apart from its benefits for providers, virtualization is extremely useful for users, by enabling cross-platform support of project applications across a wide range of computing elements. In high energy physics, the quantity of code involved and the frequency of code changes makes porting extremely arduous. Using virtualization, the entire application environment including operating system, code, libraries and support utilities can be incorporated into a virtual image, which can then be executed under suitable hypervisors installed on the worker nodes, ensuring complete compatibility of the application with the developers' native version. However, the large size of such images (around 8-10 GB), and the need to rebuild them completely when anything changes, has been a major problem.

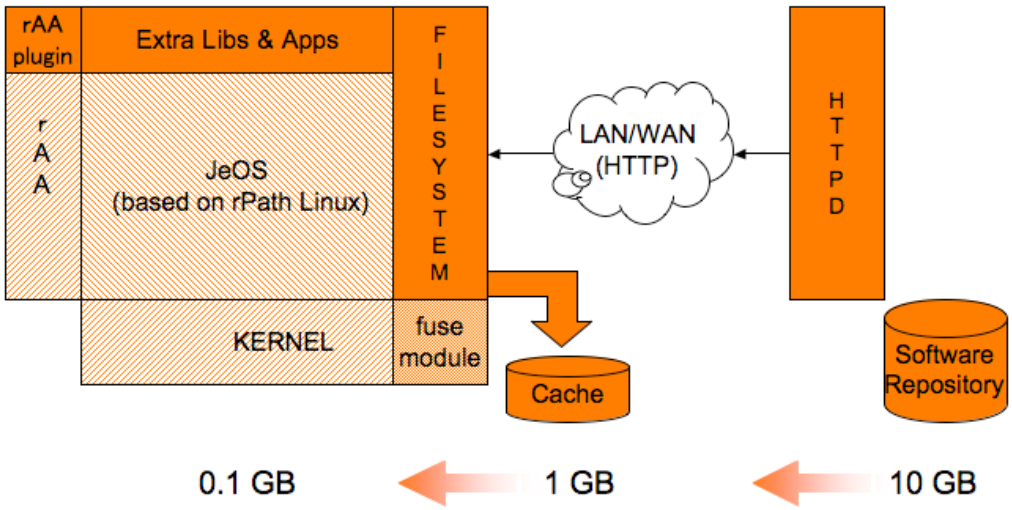
3. The CernVM Project

In 2008, a CERN R&D project called CernVM [2] was launched by Predrag Buncic, offering a general solution to the problem of virtual image management for LHC physics

PoS(ACAT2010)004

computing. Instead of loading each running virtual machine with a full image containing all the code and libraries for an experiment's applications (typically around 10 GB in size), only a basic "thin appliance" of about 100 MB is loaded initially, and the experiment specific binary code will be loaded and cached as needed from a software repository. The resulting working images are typically under 1 GB in size (see Figure 1). Updates to images after code changes are made automatically via the CernVM File System CVMFS (Figure 2), which keeps updated versions of all LHC experiment libraries in its repository. The file caching by the virtual machines minimizes access to the repository until changes appear in the physics code or a new type of application needs to be executed.

Figure 1: CernVM "thin" software appliance

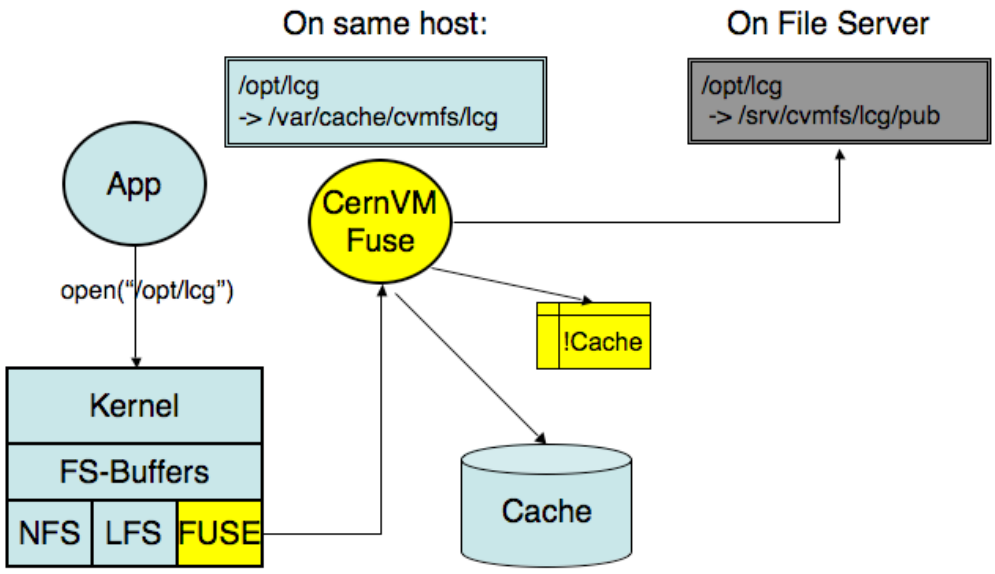


Not only has CernVM solved the problems of virtual image size and of image updating, but it also satisfies the physicists' requirement of requiring absolutely minimal changes to their working habits.

The virtual image formats that can be produced by CernVM are extremely varied. Support exists for basically every known hypervisor. In particular, physicists appreciate its support for running images under VMware, VirtualBox or Parallels, allowing them to test and develop large physics packages on their Windows or MacOSX laptops.

PoS(ACAT2010)004

Figure 2: CernVM File System CVMFS



4. CernVM and Clouds

Another very fruitful option opened to them by CernVM is that, with no further changes, they can run their applications on a production scale on computing clouds such as the Amazon EC2. Even though such commercial cloud offerings must be paid for, this has already affected the outlook of the physics communities and has put pressure on the LHC Grid community which currently adresses physicists' needs using an older and sometimes less convenient infrastructure. The actual price of an EC2-like solution is not so much more today than a Grid solution, if all the overheads are fully accounted for. Of course, not all LHC physics computing is suitable for cloud operation, particularly that which is I/O intensive or needs widely distributed data sets; but a significant proportion can be, depending on market prices of the respective service offerings.

5. Job Interface to Clouds

In order to provide additional computing resources to the LHC experiments, it is important not to expect the physicists who run job production to make changes to their existing scripts or procedures. We have observed that a large amount of LHC job production is being done using a

PoS(ACAT2010)004

"pilot job" approach: rather than submit jobs to the Grid or to in-house clusters using existing schedulers, the LHC experiments have developed their own job submission and scheduling systems (e.g. [3], [4], [5]). These send pilot agents into a fabric and use them to work out the best scheduling strategies to use at any given time. These systems also correct for failures of jobs or compute nodes, considering the fabric as an unreliable resource. This corresponds perfectly to the situation with a collection of mixed quality cloud resources which may appear, disappear, or run intermittently.

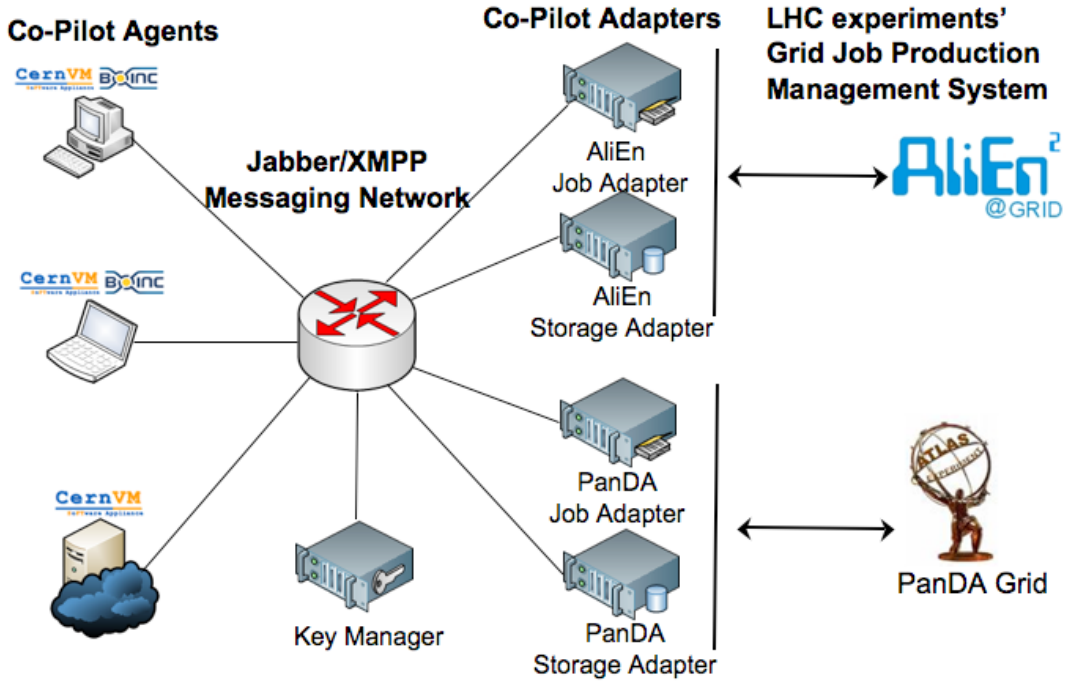
We therefore decided to interface to these pilot-job systems, and chose to use a generic interface called Co-Pilot [6, 7] which offers a gateway to the differing pilot-job implementations of the LHC experiments (see Figure 3).

On each experiment's side of the gateway, a software package called a "Co-Pilot Adapter" is required. Currently such adapers have been developed for both ALICE and ATLAS and tested with ALICE and ATLAS jobs. Adapters for LHCb and CMS should be produced in order to complete the system. All the code needed to support the Co-Pilot agents, and thus to communicate with the LHC pilot job schedulers, is included in the CernVM images that we use. No changes are needed to the LHC experiments' job production procedures.

In effect Co-Pilot allows us to interface a wide choice of extra computing resources as “dynamically configurable clouds”, via the existing LHC pilot job schedulers, including Amazon EC2, private clusters and Tier3 Grid sites.

PoS(ACAT2010)004

Figure 3: CernVM Co-Pilot architecture



6.Security and Authorization Considerations

In some cases, virtual machines running CernVM may be hosted in insecurely managed fabrics, or they could run in completely untrusted environments such as BOINC volunteer PC's. For this reason we do not want important Grid or other credentials to reside on such VM's, but restrict their presence to the Co-Pilot adapter itself, which should be hosted on a managed system.

ALICE/AliEn case (Figure 4):

The Co-Pilot agent running in CernVM is only able to talk to the Co-Pilot's Job-adapter or Storage-adapter components; only the Job-adapter can read and write to the AliEn Scheduler and Task queue, and only the Storage-adapter can read/write AliEn files.

ATLAS/PanDA case (Figure 5):

The Co-Pilot agent can only obtain a unique short-lived proxy token allowing it just to run the PanDA Pilot Agent and request and execute PanDa jobs. Job result files can only be sent to the trusted Storage-adapter which forwards them on to the Grid-SE using its own VOMS certificate. Communications between the PanDa Pilot and Server are encrypted, and the PanDA service certificate itself is invalid for any Grid services except PanDA.

Figure 4: CernVM Co-Pilot job execution (ALICE)

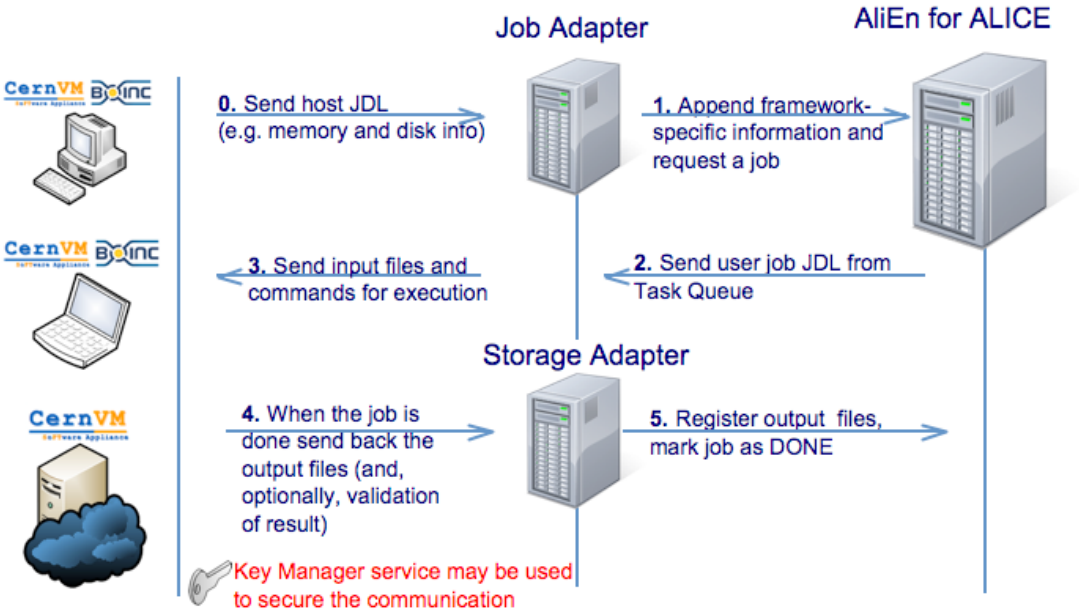
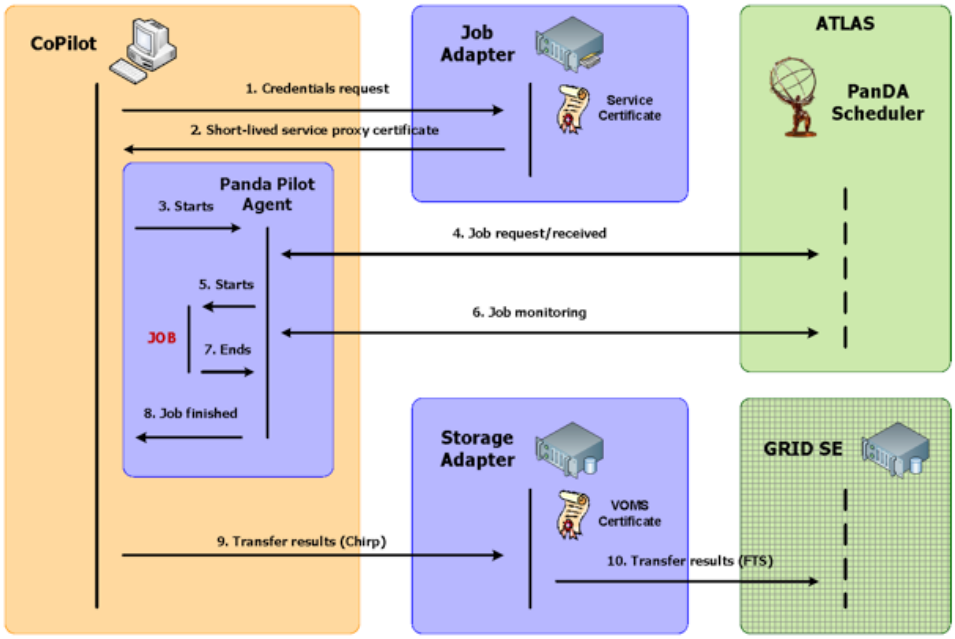


Figure 5: CernVM Co-Pilot job execution (ATLAS)



5

7. Volunteer Clouds

Apart from conventional cloud systems, we also support a cloud of volunteer PC's controlled by BOINC [8]. This very large potential resource is so far untapped by the LHC experiments: BOINC nodes are expected to be suitable for running simulation, event generation, and event reconstruction, with an emphasis on CPU intensive rather than data intensive problems.

We chose a method requiring little or no changes to the standard BOINC client or server infrastructure. This method is based on the "Wrapper" technique used for porting "legacy applications" to BOINC. The standard BOINC Wrapper simply forks and executes the binary of a legacy application and then communicates on behalf of this running application process with the BOINC core client code which also runs in the volunteer host. Our new BOINC-VM wrapper allows application processes to run optionally in guest virtual machines instead of on the host itself.

Work was first needed to provide support for hypervisors and VM's in BOINC. A general-purpose "VM controller" layer [9] was written by David Garcia Quintas. This in fact provides more than simple BOINC support. It allows a host to start and stop VM's, load and save running images, and communicate with the guest processes in the VM's, with full asynchronous communication among host and guest entities, and files and other process information able to be exchanged between the host and guest layers. Written in Python for platform independence, its architecture is shown in Figure 6. Generic support for various hypervisors is incorporated but

limited to those such as VMware and VirtualBox which expose full-function API's. VirtualBox was chosen for the intensive testing which followed.

Next, the new wrapper called "VMwrapper" [10] was written by Jarno Rantala**.** It uses a subset of the VM controller services to support hypervisors and VM's in BOINC, and is capable of running CernVM or other virtual images as guest VM processes under control of the BOINC core client in the host machine. VMwrapper is also written in Python using BOINC API Python bindings written by David J. Weir [11]. To configure the new BOINC-VM applications, VMwrapper supports XML files with formats based on the standard BOINC job.xml files but with additional tags to support the new functions associated with VM and guest process control. VMwrapper is also functionally backward-compatible with the standard BOINC Wrapper, and will run an application in the host as before, if provided with a standard BOINC job.xml file. The VMwrapper architecture is shown in Figure 7.

With our approach, no changes to either the BOINC infrastructure or to any LHC experiment code or job submission procedures are needed. The resulting "Volunteer Cloud" for LHC computing will shortly be beta-tested for the ALICE and ATLAS collaborations. More details of this work have been reported elsewhere [12].

Figure 6: Host to VM Guest communication

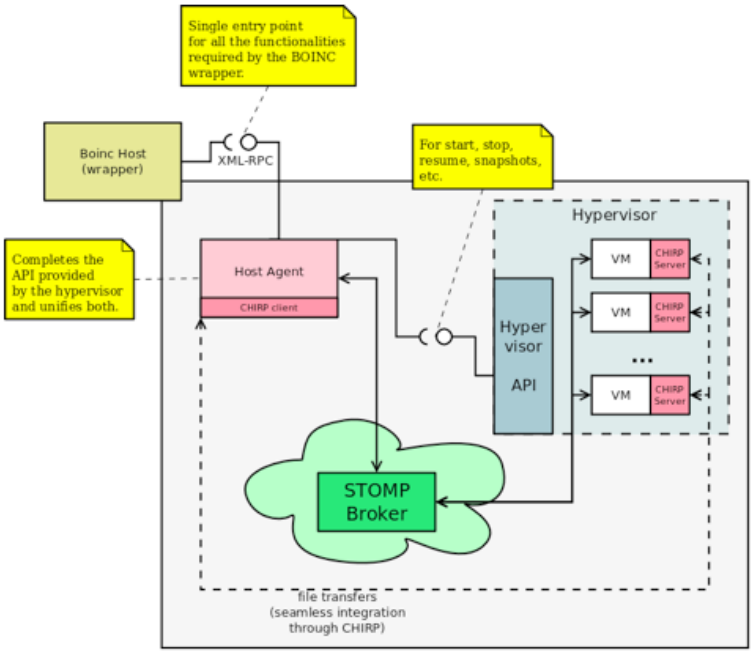
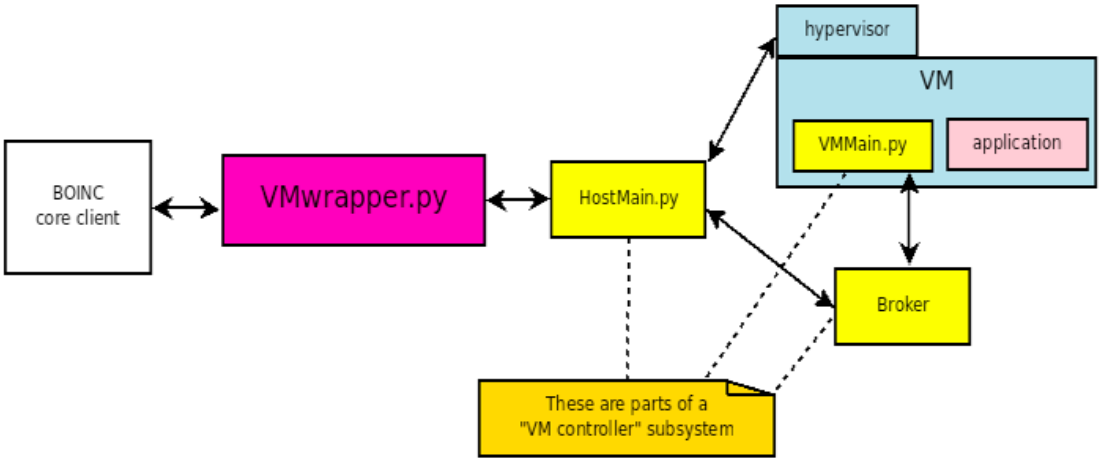


Figure 7: BOINC VMwrapper architecture



8. Acknowledgements

The authors would like to thank the ATLAS and ALICE experiments, the Tampere University of Technology, and the CERN Summer Student and CERN openlab student programs for their financial and technical support of this work.

9. References

[1] Amazon Web Services, *Elastic Compute Cloud EC2 and Simple Storage Service S3*, http://aws.amazon.com/ec2/ , http://aws.amazon.com/s3/ .

[2] Predrag Buncic et al., *CernVM*, http://cernvm.cern.ch/cernvm/ .

[3] ATLAS Collaboration, *PANDA*, https://twiki.cern.ch/twiki/bin/view/Atlas/Panda .

[4] LHCb Collaboration, *DIRAC, A Community Grid Solution,* Conference on Computing in High Energy and Nuclear Physics (CHEP'07).

[5] Buncic, P. et al., *"The Architecture of the AliEn System",* Proceedings of the Conference on Computing in High Energy and Nuclear Physics (CHEP'04), Interlaken, Switzerland.

[6] A. Harutyunyan, P. Buncic, T. Freeman, and K. Keahey, *"Dynamic Virtual AliEn Grid Sites on Nimbus with CernVM",* Computing in High Energy and Nuclear Physics (CHEP'09).

[7] Artem Harutyunyan, *CoPilot Protocol Specification,* https://cernvm.cern.ch/project/trac/cernvm/wiki/CoPilotProtocol .

[8] David Anderson et al., *BOINC – Berkeley Open Interface for Network Computing,* http://boinc.berkeley.edu .

[9] David Garcia Quintas, *A host <-> guest VM communication system for Virtual Box,* http://boinc.berkeley.edu/trac/wiki/VirtualBox .

[10] Jarno Rantala, *"VMwrapper",* http://boinc.berkeley.edu/trac/wiki/VmApps

[11] David J. Weir, *A Python API for BOINC,* http://plato.tp.ph.ic.ac.uk/~djw03/boinc-python/documentation-0.3.1/ .

[12] Ben Segal et al., *"Building a Volunteer Cloud",* Acta Cientifica Venezolana, (Proceedings of CLCAR-2009, Mérida, Venezuela, Sept. 2009: to be published). http://www.cern.ch/ben/Ven_abs.pdf .