Journal of Physics: Conference Series **513** (2014) 032103 doi:10.1088/1742-6596/513/3/032103

Exploring virtualisation tools with a new virtualisation provisioning method to test dynamic grid environments for ALICE grid jobs over ARC grid middleware

B Wagner[1], B Kileng[2], for the ALICE Collaboration
[1] Universitetet i Bergen
[2] Høgskolen i Bergen

E-mail: Boris.Wagner@uni.no, Bjarte.Kileng@hib.no

Abstract. The Nordic Tier-1 centre for LHC is distributed over several computing centres. It uses ARC as the internal computing grid middleware. ALICE uses its own grid middleware AliEn to distribute jobs and the necessary software application stack. To make use of most of the AliEn infrastructure and software deployment methods for running ALICE grid jobs on ARC, we are investigating different possible virtualisation technologies. For this a testbed and possible framework for bridging different middleware systems is under development. It allows us to test a variety of virtualisation methods and software deployment technologies in the form of different virtual machines.

1. Introduction

Grid computing is the central idea to allow analysis of the huge amount of data generated (more than 10 PB/year) by the LHC [1] at CERN. In order to to deal with this amount of data several grid middleware systems have been developed. The most common one for Worldwide LHC Computing Grid (WLCG [2]) is gLite [3], but also other ones like OSG [4] in the USA, or ARC in Scandinavia, are in use.

1.1. The AliEn middleware

ALICE [5] has developed its own middleware platform called AliEn [6]. This can be used as middleware on its own, but is also applied by ALICE as a common interface to a range of underlying middleware systems. Contrary to the decentralised ideal of the grid, the AliEn system relies on the availability of central services, located at CERN, containing a centralised task queue and a central meta data and file catalogue. This improves overall resource allocation and job optimisation. In this system users authenticate centrally and submit jobs to the task queue. As a central design aspect of AliEn the jobs will follow the data. Jobs are also pulled by job agents to free resources instead of getting pushed. They can request automatic installation of necessary software versions on the local system prior to starting.

In each cluster providing computing resources a separate machine called VObox[7] acts as the entry point. It is used to submit jobs to the local batch system, for monitoring, and also serves as a communication proxy between the running jobs and the central services.

1.2. The Nordic distributed Tier-1 centre

The WLCG is organised along the hierarchical Monarc model [8], specifying a tiered computing structure. The Nordic countries created a distributed Tier-1 centre organised in the Nordic Data Grid Facility (NDGF [9]), now operated by the Nordic e-Infrastructure Collaboration NeIC [10]. ARC (Advanced Resource Connector) [11], has been developed in Scandinavia and it is used as internal middleware for computing resources. The distributed Storage Element (SE) is based on the dCache [12] storage middleware. This makes the distributed storage appear as one entity. The AliEn system does not have a transparent interface to the ARC job management, so the internal centres in NDGF appear as separate entities in the ALICE grid. This project is an approach to create such an interface, so that the NDGF Computing Element (CE) could appear as one, analogous to what is already implemented for the SE.

2. Software deployment on Grid

Computing grids were supposed to be heterogenous. This made the task of software deployment extremely difficult and led to a large spectrum of different solutions in all grid middleware systems. Today's automatic deployment systems assume homogenous platforms with little variation.

2.1. Virtualisation

During the recent decade, the hardware support for virtualisation has become more and more prevalent in commodity hardware. Virtual machines today can run without large performance loss on regular low-cost multicore machines. Virtualisation helps to run tasks that need a complex software environment and allows to dynamically provide resources to different users, in order to optimise the capacity utilisation of a computing cluster. It also provides good separation between different user jobs and minimises side effects and interferences, that can develop between user jobs that run on the same hardware.

Today, cloud computing frameworks can provide Infrastructure-as-a-Service (IaaS) by managing the virtual machines as well as authorisation and virtual network infrastructure, usually with a web interface for administration.

2.2. CernVM

CernVM [13] is a project to provide a software environment for all CERN experiments. It consists of a small core virtual machine available for many virtualisation technologies like Xen or KVM. The second component is the CernVM file system (CernVMFS). It is a read-only network filesystem that uses http as underlying transport protocol. It uses caching on the local worker node and can also use other common web caching methods like proxies. Implemented as a FUSE module [14], it appears to the OS that all files in the mounted directory containing the experiment specific software stack is available locally, but only when a file is accessed it gets fetched and cached. A batch computing environment will profit significantly from the caching. Also new software versions are installed centrally and are visible everywhere immediately.

CernVMFS is now used by AliEn for software provisioning. It is also used by other LHC experiments when running on ARC.

2.3. Virtual Machines with AliEn

In order to do some small scale tests of different virtualisation technologies in a grid computing or batch processing context a testbed for a local cluster has been developed. During development we realised the potential for wider spread usage. For smaller sites it could be used to enable the virtualisation method of choice without much interference to the local setup. The method is distribution agnostic and its setup is less complex than a full private cloud infrastructure. Those computing clouds are normally less suited to have short running VMs in a batch computing setup.

2.4. Testbed with Vmbatch

For the testbed a system called Vmbatch has been developed (source available at [15]). It is based on the vibatch system [16], that creates virtual machines for each running batch job in a queue. In our Vmbatch setup we use mainly Xen as virtualisation method, but it is based on the libvirt library [17] and it is able to make use of other hypervisors.

The local cluster uses TORQUE as the cluster resource management system [18], from which the system uses its prologue and epilogue scripts to start and stop the guest. Similar functionality can be achieved in other resource management systems as well.

A shared NFS file system is used for data exchange between a host and its guest systems. A shared user home directory and a shared site specific software directory is very common in clusters. Our site has such a setup, so no modifications are needed. Those requirements can be eliminated in future versions by using the existing staging mechanism in TORQUE. Apart from the use of a shared NFS filesystem, our system has no other requirements to the network setup at the site.

Each virtual machine (or application stack) needs a dedicated queue. The Vmbatch system will in a first step create a virtual machine for each job and once the VM is up and running, inject the job script into the VM via ssh and run it. As an alternative to ssh, Vmbatch can also run the job in a pseudo terminal with the master on the guest connected to the slave on the host. After job completion the result will be transferred out of the VM which then will be destroyed. Options to use one virtual machine per node rather than one per job or longer living virtual machines were considered, but for the typical ALICE grid job which uses one core and runs 48 hours, it wasn't necessary.

The worker node needs a local directory named vmbatch for storing disk images, logs and lock files, and in our test setup it used a Xen kernel. We also provided a default network via libvirt to enable network address translation (NAT) between the host and the guest. In this case the NFS mount option "insecure" must be used and in order to allow proper user id mappings for the NFS share the service rpcidmapd must be running in the guest system.

In order to prepare a raw disk image to be used in our setup (Xen on Scientific Linux) several modifications had to be done. Vmbatch provides command line utilities for this. The /boot partition must be separate and mounted read-only. The rest of the OS and the software stack has to reside in a Qemu Copy On Write disk image (qcow) [19] created by Xen tools. Then a new qcow disk image can be created using an existing qcow image as backing image. This allows the Vmbatch system to copy large disk images very fast. All disk writes will go to the new image, but unmodified reads will come from the backing image. Other site specific modifications are e.g. the start of the NFS user mapping daemon rpcidmapd, the set up of Vmbatch specific system services inside the guest and configuring the network of the guest. If a pseudo terminal is used to run the job, the guest will be configured to do an automatic login as the submitting user.

When the guest is started, the Vmbatch service will create the submitting user, mount the NFS share and create a specific file on the NFS share as a signal to the prologue script that the guest has started. A different program, called remoteshell, will connect with the guest and run the job inside the guest. When the job is finished a TORQUE epilogue script will destroy the guest, delete the disk images and clean up lock files.

Vmbatch uses an xml file for configuring guests. Different TORQUE queues can use different xml files. The xml startup file is a libvirt domain xml file with some additional xml tags. These tags specify the disk layout and how to create the disk images from the templates.

2.5. Vmbatch Extension for CernVM

CernVM supports several methods of contextualisation. The first implementation used by the Vmbatch system uses the cdrom method by generating a small custom ISO image and attaching

it to the VM via additional cdrom tags in the domain-xml configuration file. When CernVM starts it will execute a script prolog.sh if it exists on a mounted cdrom. We use this to copy a prepared tar archive into a predefined directory. Next a contextualise script (context.sh) is executed and after CernVM is up and running the epilog.sh script does the finalising steps.

3. Test Run

As a first test job to compare the performance of different virtualisation methods with our Vmbatch system, we chose PbPbbench from the ALICE software analysis framework. It simulates and reconstructs lead-lead collisions and is one of the verification steps of new ALICE software package versions. This is a typical CPU intensive program without much I/O.

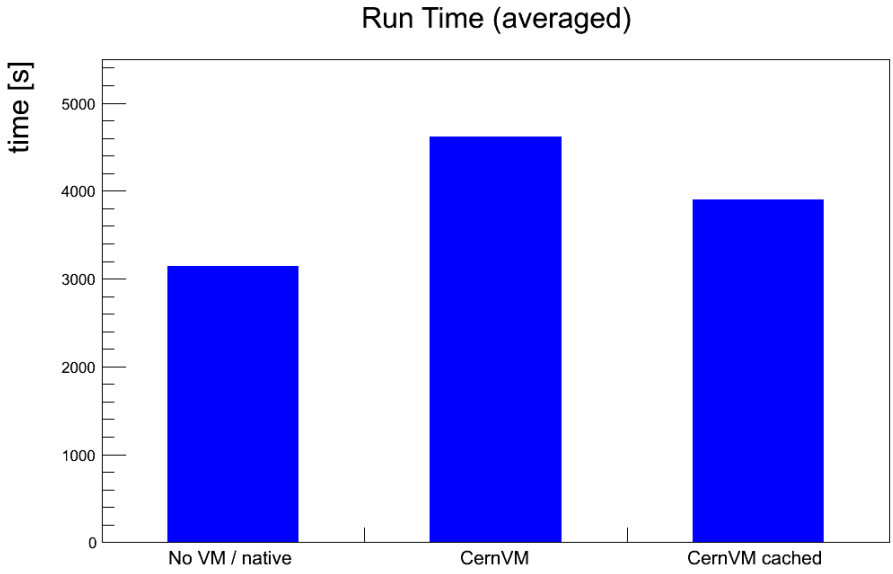

Figure 1. Comparison of run times of PbPbbench. Native is run on the host directly with all files on disk. CernVM is the first run of the benchmark and includes fetching the needed files from remote storage. CernVM cached is a run where the files are already cached in the virtual machine

4. Results and Discussion

First performance numbers for different configurations were done with regular Unix tools like time and date. One important aspect is the setup time of the virtual machine in comparison to the run time of the job. Disk image creation and VM setup takes ≈ 70 seconds and is therefore negligible compared to an overall running time of 6000 seconds. Another important aspect is the run time overhead generated by the chosen virtualisation method. Setup and overall run time also depends on the software provisioning method. For CernVM one has to distinguish between the setup/startup time of the first VM which has to fetch the needed files from remote storage and the consecutive jobs that will fetch most of the files from the caches. Figure 1 shows the difference between native run times, CernVM fetching the data from remote storage and CernVM which has most of the files cached in the VM. The more realistic scenario of cached files increases the overall runtime around 24%, including the virtualisation overhead.

5. Conclusion

The first results of the Vmbatch system as a lightweight tool to use dynamic virtualisation in a batch processing setup have been very promising. It can be installed on a site with minor adjustments to the configuration, since the default settings reflect common compute cluster setups. The extremely fast copy on write method negates one of the major drawbacks of dynamic virtualisation, which is the setup phase in comparison to the run time of a job. It has proven to be easily extendable to different underlying virtualisation methods, so a site could find the proper balance between performance and convenience.

In future work more fine grained performance measurements are needed to investigate bottlenecks induced by software provisioning, network access and data caching on different levels. Also different grid job work patterns have to be investigated, especially how disk and network I/O affects the different virtualisation technologies.

References

[1] `http://home.web.cern.ch/about/computing`
[2] I. Bird, Annual Review of Nuclear and Particle Science, Vol. 61: 99-118, November 2011
[3] E. Laure et al., Computational Methods in Science and Technology 12(1), 33-45 (2006), `http://cds.cern.ch/record/936685/files/egee-tr-2006-001.pdf`
[4] `http://www.opensciencegrid.org/`
[5] K. Aamodt et al., 2008 JINST **3** S08002
[6] S. Bagnasco et al., J. Phys.: Conf. Ser. **119** (2008) 062012
[7] J. Zhu et al., ICACT 2012, 14th International Conference on Advanced Communication Technology, p. 1209-1214, ISBN: 978-1-4673-0150-3
[8] H. Newman (ed.), CERN 24 March 2000, CERN/LCB 2000-001
[9] `http://www.ndgf.org`
[10] `http://neic.nordforsk.org/`
[11] M. Ellert et al., Future Generation Computer Systems 23 (2007) 219 2013240
[12] G. Behrmann et al, 2008 J. Phys.: Conf. Ser. **119** 062014
[13] P. Buncic et al, 2010 J. Phys.: Conf. Ser. **219** 042003
[14] `http://fuse.sourceforge.net/`
[15] Subversion repository `http://eple.hib.no/svn/vmbatch/tags/latest/`
[16] `https://ekptrac.physik.uni-karlsruhe.de/trac/BatchVirt/wiki/ViBatch`
[17] `http://libvirt.org/index.html`
[18] G. Staples SC '06 Proceedings of the 2006 ACM/IEEE conference on Supercomputing, 2006
[19] `http://www.linux-kvm.org/page/Main_Page`