Virtualization of the ATLAS software environment on a shared HPC system

Anton Josef Gamel\textsuperscript{1,2}, Ulrike Schnoor\textsuperscript{1}, Konrad Meier\textsuperscript{1,2}, Felix Bührer\textsuperscript{1}, Markus Schumacher\textsuperscript{1}
Albert-Ludwigs-Universität Freiburg,
\textsuperscript{1}Physikalisches Institut, Hermann-Herder-Strasse 3
\textsuperscript{2}Rechenzentrum, Hermann-Herder-Strasse 10
79104 Freiburg im Breisgau, Germany
E-mail: anton.gamel@physik.uni-freiburg.de

Abstract. The shared HPC cluster NEMO at the University of Freiburg has been made available to local ATLAS users through the provisioning of virtual machines incorporating the ATLAS software environment analogously to a WLCG center. This concept allows to run both data analysis and production on the HPC host system which is connected to the existing Tier2/Tier3 infrastructure. Schedulers of the two clusters were integrated in a dynamic, on-demand way. An automatically generated, fully functional virtual machine image provides access to the local user environment. The performance in the virtualized environment is evaluated for typical High-Energy Physics applications.

1. Introduction
High-Performance Computing (HPC) and other research cluster computing resources provided by universities can be useful supplements to the ATLAS [1] collaboration’s own WLCG [2] computing resources for data analysis and production of simulated event samples. Known difficulties to use these opportunistic resources are incompatibilities in network, batch schedulers, operating system, software installed and no access to external resources required. There exist a vast variety of different and very individual approaches to make opportunistic resources available, mainly for ATLAS production [3][4][5]. In order to use the University of Freiburg HPC-cluster NEMO[6] (388/TOP500[7], 287280 HEP-SPEC) we further pursued the local approach already started and described here [8], taking advantage of the preexisting OpenStack[9] layer, allowing virtual machines (VMs) wrapped as jobs to be started on top of the NEMO bare metal installation.
2. **Tasks**
A transparent setup that makes NEMO resources available for users and that is as well smooth and flexible for administrators had to include the following:

- Full ATLAS / WLCG analysis and production Tier2/Tier3 environment
- Full local user environment including access to dCache\[10\] and cluster filesystem BeeGFS\[11\]
- Easy generation of up-to-date VM images and if possible in an automated procedure
- Integration of local Tier2/Tier3 batch system slurm\[12\] and NEMO scheduler Torque/Moab\[13][14\]
- Start VMs on-demand using separate batch queues on Tier2/Tier3 slurm scheduler.

3. **Generation of Virtual Machines using Packer**
For the diskless installations of workernodes on the Tier2/Tier3 cluster a template machine is used that is a complete and fully functional workernode installation residing on VMware\[15\]. This template is configured by Puppet\[16\]. Exactly this Puppet configuration can be used to create virtual images with the identical setup the current workernodes on the Tier2/Tier3 cluster also have. Packer\[17\] installs a minimal SL6.8 from iso-file in a nested KVM on the OpenStack Service-Environment. After software update and reboot, Packer injects a valid Puppet certificate into the VM and connects the Puppet client with the Puppet master. Full configuration by Puppet takes some time since all packages are installed one by one. At this stage also the kernel modules are built to access the cluster filesystem later. After configuration has finished a rootfs-resize\[18\] optimizes the size of the image and Packer creates a qcow2 image under version control ready to be uploaded to NEMO OpenStack.

4. **Integration of batch systems**
The VMs that are now available via OpenStack have to be started on request i.e. if a job is queued in the corresponding slurm partition. Unfortunately the Elastic Computing feature of slurm that may start and stop physical machines on-demand only offers very limited control over VMs on remote systems. We therefore adapted ROCED\[19\] which is a modular python program interfacing different batch systems to different cloud sites and managing virtual machines on these sites. Moreover ROCED provides the administrator with handles to monitor the status of the queues as well as to analyze the usage. It was initially designed for HTCondor\[20\] and was now adapted to be used with slurm.

The ATLAS user submits the Tier3 job as usual (Fig. 1) (1) but to a dedicated queue that reflects for fairshare reasons the group affiliation. ROCED demons, on a separate VM, monitor for incoming jobs (2) and trigger (3) the start of a job-wrapper in the NEMO UI & Moab environment (4). Once the job is launched by Moab, the job wrapper requests a VM from OpenStack (5) that starts into the job-wrappers space (6). After the start of the VM, the most important features of the environment are automatically checked, e.g. file system access and mounts as well as network access. If the tests are passed, the VM contacts the slurm control demon on the slurm server and is ready to receive jobs from the slurm queue (7). Several ROCED services can be run without interference in order to serve different slurm partitions. Watchdog processes shutdown the VM if idle for a certain period or set it to be drained in time before the end of walltime approaches. Since the VMs have no cvmfs cache available neither on virtual disk nor on the host, within the NEMO network a dedicated squid proxy VM was installed that handles all the requests.
5. **Performance in the virtualized environment**

NEMO and the Tier2/Tier3 cluster partly share the identical hardware configuration: 4in1 INTEL S2600KPR board with 2x INTEL CPU E5-2630v4 2.20GHz 40cores hyperthreaded, 128GB RAM, SSD. It was therefore possible to compare the performance of jobs in the VMs (4-core, CentOS7 host) with the performance of jobs running on bare metal (multicore, diskless install). For the evaluation two applications were chosen. On the one hand a Powheg/Pythia8\(^{[21]}\)[22] event generation that used ATLAS software via cmvfs (Fig. 2A), on the other hand the HEP-SPEC06 benchmark application \(^{[23]}\) [24]. As a reference HEP-SPEC06 benchmarks were also run on reserved machines on bare metal using different numbers of cores (Fig. 2B). In both applications the jobs in the VMs perform better than the jobs on bare metal: no loss of performance in the VMs can be determined (Fig. 2A,B). The difference in performance VMs versus bare metal was unexpected and is under further investigation.

---

Fig. 1 Overview over NEMO cluster receiving jobs from local Tier2/Tier3 cluster via ROCED node

![Diagram of NEMO and Tier2/Tier3 cluster interaction](image)

Fig. 2A VM nodes better performing compared to bare metal

![Fraction of jobs (%) vs. run time](image)

Fig. 2B HEP-SPEC06 on 4-core VMs similar to optimal conditions on bare metal

![HEP-SPC0s per CPU vs. N(CPUs)](image)
6. **Outlook**

Meanwhile NEMO Moab received an improvement and is able to start a job with a VM in it directly from OpenStack without the need of a job wrapper. The job monitoring information ROCED makes available to Slurm or the user is still sparse and has to be enhanced. In the described setup a queue for Tier2 production can be established as well. For the transition from SL6 to CentOS7 a template to generate VMs with CentOS7 is in preparation.

7. **References**

[1] ATLAS Collaboration 2008 J. Inst. 3 S08003
[2] Shiers J 2007 Comput. Phys. Commun. 177 219–223
[3] Kennedy J A, Kluth S, Mazzafferro L and Walker R 2015 Bringing ATLAS production to HPC resources - A use case with the Hydra supercomputer of the Max Planck Society. In: 21st International Conference On Computing in High Energy and Nuclear Physics (Chep2015), Parts 1-9, Vol. 664, 092019
[4] Cameron D, Filipčič A, Guan W, Tsulaia V, Walker R and Wenaus T Exploiting opportunistic resources for ATLAS with ARC-CE and the Event Service. 22nd International Conference on Computing in High Energy and Nuclear Physics, (CHEP 2016), San Francisco, USA, 10 - 14 Oct 2016
[5] Barreiro M, Fernando H, De K, Klimentov A, Nilsson P, Oleynik D, Padolski S, Panitkin S and Wenaus, T Integration of Titan supercomputer at OLCF with ATLAS production system. 22nd International Conference on Computing in High Energy and Nuclear Physics, (CHEP 2016), San Francisco, USA, 10 - 14 Oct 2016
[6] http://www.hpc.uni-freiburg.de/nemo [accessed 2017-09-27]
[7] https://www.top500.org/list/2017/06/?page=4 [accessed 2017-09-27]
[8] Konrad M et al 2016 J. Phys.: Conf. Ser. 762 012012
[9] OpenStack [software] https://www.openstack.org/ [accessed 2017-09-27]
[10] dCache [software] https://www.dcachex.org/index.shtml [accessed 2017-09-27]
[11] BeeGFS [software] https://www.beegfs.io/content/ [accessed 2017-09-27]
[12] Slurm [software] https://slurm.schedmd.com/ [accessed 2017-09-27]
[13] Torque [software] http://www.adaptivecomputing.com/products/open-source/torque/ [accessed 2017-09-27]
[14] Moab [software] http://www.adaptivecomputing.com/products/hpc-products/moab-hpc-basic-edition/ [accessed 2017-09-27]
[15] https://www.rz.uni-freiburg.de/services-en/serverdienste_en/virtualisierung?set_language=en [accessed 2017-09-27]
[16] Puppet [software] https://puppet.com/ [accessed 2017-09-27]
[17] Packer [software] https://www.packer.io/ [accessed 2017-09-27]
[18] rootfs-resize [software] https://github.com/ctyler/rootfs-resize [accessed 2017-09-27]
[19] Erli, G; Fleig, G; Hauth, T; Riedel, S “ROCED” [software] Available from https://github.com/roced-scheduler/ROCED [accessed 2017-09-27]
[20] HTCondor [software] http://research.cs.wisc.edu/htcondor [accessed 2017-09-27]
[21] Nason P, JHEP 0411 (2004) 040, hep-ph/0409146
[22] Sjöstrand T, Mrenna S and Skands P, JHEP05 (2006) 026, Comput. Phys. Comm. 178 (2008) 852
[23] SPEC® CPU2006 [software] https://www.spec.org/cpu2006/ [accessed 2017-11-13]
[24] http://w3.hepix.org/benchmarks/benchmarking.html [accessed 2017-11-13]