Mastering Opportunistic Computing Resources for HEP

M J Schnepf, C Heidecker, M Fischer, M Giffels, E Kuehn, A Heiss, A Petzold and G Quast

KIT - Karlsruhe Institute of Technology, Germany

E-mail: {matthias.schnepf, christoph.heidecker, max.fischer, manuel.giffels, eileen.kuehn, andreas.heiss, andreas.petzold, guenter.quast}@kit.edu

Abstract. As results of the excellent LHC performance in 2016, more data than expected has been recorded leading to a higher demand for computing resources. It is already foreseeable that for the current and upcoming run periods a flat computing budget and the expected technology advance will not be sufficient to meet the future requirements. This results in a growing gap between supplied and demanded resources.

One option to reduce the emerging lack of computing resources is the utilization of opportunistic resources such as local university clusters, public and commercial cloud providers, HPC centers and volunteer computing. However, to use opportunistic resources additional challenges have to be tackled. At the Karlsruhe Institute of Technology (KIT) an infrastructure to dynamically use opportunistic resources is built up. In this paper tools, experiences, future plans and possible improvements are discussed.

1. Introduction

Computing resources for the LHC experiments are usually provided statically by the Worldwide LHC Computing Grid (WLCG) [1]. However, the demand for computing resources varies over the time due to periods of data taking and conferences. This results in either over- or under-commitment of available resources, both having its drawbacks. In addition, the current deployment strategy of WLCG sites is relying on a flat budget and the expected technology advance. However, this will not be sufficient to cover future resource needs for the LHC computing leading to a gap between supplied and demanded resources [2].

Dynamic integration of additional computing resources for High Energy Physics (HEP) could help to reduce this gap and would allow more flexibility of the computing models. Therefore, one of the research topics at the Karlsruhe Institute of Technology (KIT) is the development of an infrastructure for a transparent and dynamic provisioning of computing resources not dedicated to HEP, so-called opportunistic resources provided for example by High Performance Computing (HPC) Centers, scientific and commercial cloud providers.

In this paper the developed infrastructure, experiences, future plans and possible improvements are discussed.

2. Integration of Opportunistic Resources

The integration of opportunistic resources can be grouped into three different categories as shown in Fig. 1. One category is the extension of resources for a longer period of time as

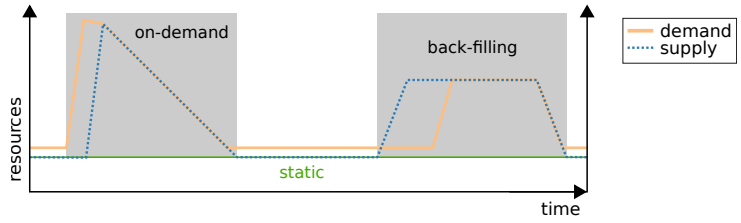

Figure 1: Three different ways to extend statically available HEP compute resources (green line). The constant capacity expansion (black line), the dynamic extension of resources (depicted on the left side) that depends on the current demand (orange line) and the back-filling approach (shown on the right side) which is independent of the current demand. The dotted blue line shows to total amount of compute resources available over the time.

static resources. The other two categories are the dynamic on-demand extension of resources to cover peak-loads and the back-filling approach. Since the extension with static resources is comparable to the deployment of new dedicated HEP computing resources, we will focus on the dynamic approaches in this paper. Typically HEP computing resources are provided to the experiments utilizing the concepts of traditional Grid computing. In other words, it means that resource providers have to maintain a complex framework with special dependencies in terms of the operating system and the HEP software requirements to make their resources available to the HEP community. However, this is usually not feasible on opportunistic resources, since it is often incompatible with the intended purpose of those resources.

To overcome this challenge it is necessary to utilize modern virtualization and container technology following the "Infrastructure as a Service" approach. This leads to a task sharing between the resource provider, who solely operates the infrastructure and HEP that manages its complex software environment using virtual machine images or containers. This approach allows HEP to utilize additional resources in a dynamic fashion from various providers like scientific and commercial clouds as well as High Performance Computing centers, given that a virtualized environment or a container solution is offered for resource provisioning.

Many tools utilized in HEP computing do already support modern cloud technologies and therefore provide a profound base for making use of opportunistic resources. The HTTP based CERN Virtual Machine File System (CVMFS) [3] is well suited for the world-wide distribution of experiment software. Using conventional caching HTTP proxies enables a scalable on-demand provisioning of software. This makes it to an adequate choice to deploy software on opportunistic resources. The XRootD protocol [4] and the data federations of the experiments completes this technology and provides transparent access to the experiment data located on traditional Grid storage systems. In addition, the batch system HTCondor [5] which is commonly used in HEP is also perfectly suited to integrate dynamic and opportunistic resources due to its pull workload mechanism as well as its native support of running jobs in `docker` [6] and `singularity` [7] containers.

3. Resource Management

In addition to the software stack describe above, the cloud scheduler `ROCED` (**R**esponsive **O**n-**D**emand **C**loud-**e**nabled **D**eployment) [8] has been developed to address one of the remaining challenges, namely the dynamic management of resources depending on the demand. `ROCED` monitors the demand for computing resources by checking the job queue of a batch system as well as the available resources. Based on this information it can either request new or release unused resources by starting or terminating virtual machines, respectively. The workflow of `ROCED` is depicted in Fig.2. `ROCED` provides interfaces to various cloud APIs and batch systems, to allocate

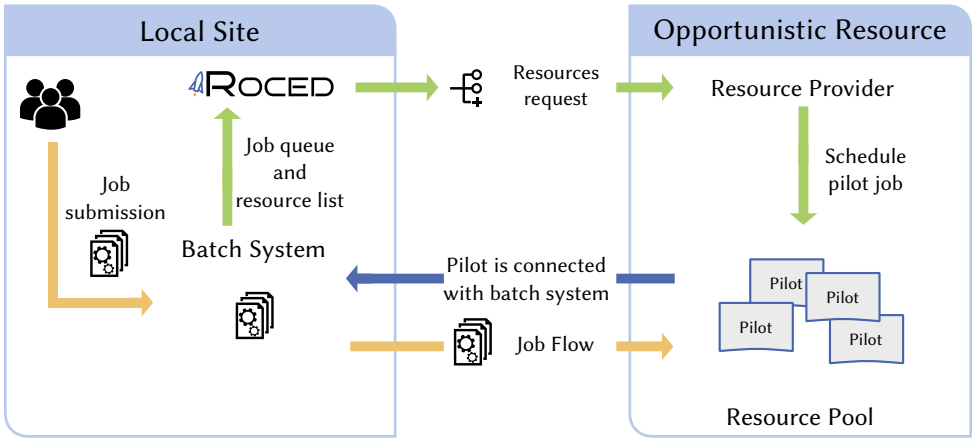

Figure 2: The resource manager `ROCED` monitors the job queue and the available resources of the batch system. Based on this information it can request additional resources from a supplier like a HPC center or a cloud provider if necessary. Once the resources are provisioned, the pilots connect to the batch system and new jobs can be assigned.

and integrate resources from different providers. Nevertheless, support for additional batch systems and cloud APIs can be easily added thanks to its modular design. `ROCED` can handle several different cloud resources at once and implements a basic cloud brokerage functionality based on cost.

4. Experiences with Opportunistic Resources

The development of `ROCED` started back in 2011, since then the group has gained longterm experience with the acquisition, integration, and utilization of different kind of opportunistic resources for HEP computing, which will be described in the following using illustrative examples. The following resources have been integrated transparently for the users into the same `HTCondor` batch system.

4.1. Desktop Cloud

The institutes desktop PCs are primarily used for office applications like email, web browsing, text editing, etc. For this purpose, the users rely on a modern operating system like Ubuntu Linux supporting state-of-the-art versions of office applications. On the other side, recently procured desktop PCs equipped with powerful eight core Intel i7 CPUs, 16 GB RAM and SSDs are well suited to run HEP applications. Bearing in mind the strict requirements of the HEP software stack based on a mature operating system, both use cases seem incompatible at the first glance. However, a recently added feature of the `HTCondor` batch system allows the execution of jobs inside `docker` containers. By utilizing this feature, modern container technology could be used to manage the balancing act between office use and HEP applications. In total around 300 additional job slots could be added to the batch system more than doubling the locally available computing power for HEP. In order to avoid interferences with the daily users of the desktop PCs, compute jobs are automatically suspended and resumed depending on the local utilization of the PCs. Altogether the utilization of the desktop PCs as well as the available HEP computing power could be increased by using modern container technologies without unreasonably great effort and expense.

4.2. Commercial Cloud Providers

The usage of commercial cloud providers to cover peak loads has been evaluated as well. In this case, the `ROCED` cloud scheduler was used for the on-demand resource provisioning and integration in the locally operated `HTCondor` batch system. The provisioning of resources on the infrastructure of commercial cloud providers still relies mostly on traditional virtualization techniques even more and more providers started to support container technologies as well. Therefore, virtual images based on SL6 and `CVMFS` were used in order to supply the HEP software environment. In order to allow efficient access to conditions data and `CVMFS` repositories at CERN, service infrastructure like HTTP proxies need to be deployed on the cloud sites as well. So far, the integration of resources from commercial cloud providers has been tested based on grants and the experiences gained are mixed. For global players, almost everything works out of the box, whereas for smaller vendors often special adaptions are needed. The cost of virtual resources procured from commercial providers is nearly dropping every year, however the cost of network traffic still remains very high. Nowadays the integration of commercial providers in scientific computing is driven by the Helix Nebula Science Cloud (HNSciCloud) project [10] Founded by the European Union. Further test as part of the HNSciCloud project are currently ongoing.

4.3. High Performance Computing Center NEMO

In the context of the bwHPC project [9] of the state of Baden-Württemberg, HEP is also entitled to use HPC resources at the bwFORCluster NEMO at the University of Freiburg [11]. NEMO is a shared HPC cluster between neuroscience, microsystem engineering and particle physics. Each community has different needs and requires a very different software stack. The only viable solution is to operate a classic HPC infrastructure on bare metal in combination with a virtualized infrastructure on the same cluster [12]. For this purpose a hypervisor is installed on each compute node and the virtual infrastructure is managed by OpenStack. However, the needs for an accounting and fairshare mechanism on the cluster leads to a more complex setup compared to the commercial cloud providers described above. The resource allocation needs to be done via the HPC batch system. Therefore, a batch job containing a small script is submitted, which takes care of blocking a job slot on a corresponding worker node, as well as requests the provisioning of a virtual machine on the same worker node via the OpenStack API. Using this setup up to 8000 cores can be requested in parallel and the scalability of the solution has been proven up to 11000 cores. The NEMO cluster is currently largest supplier for opportunistic resources integrated into HEP computing at KIT. The resource utilization of the KIT HEP group at NEMO is shown in Fig.3.

5. Outlook

The usage of large amounts of opportunistic resources comes along with new challenges. Unlike traditional Grid computing sites, suppliers of opportunistic resources usually do not provide permanent storage systems or do not offer access to dedicated WLCG networks. Thus, it is recommended to use opportunistic resources only for CPU intensive tasks so far. Otherwise the remote data access to Grid storage systems could overload the network and the storage systems as well. As a result I/O intensive tasks would suffer from a low CPU utilization. However, in the following concepts to use opportunistic resources also for I/O intensive tasks are presented.

One solution could be transparent caching of data locally at the site, for example by using shared storage available at HPC centers. This concept is described in more detail in [13].

Another supplementing approach could be the improvement of the job scheduling based on the network utilization and the available network bandwidth as shown in Fig. 4. HEP workflows usually consist of several thousands jobs running the same application with different input parameters or files. Thus, it can be assumed that the network access pattern of different

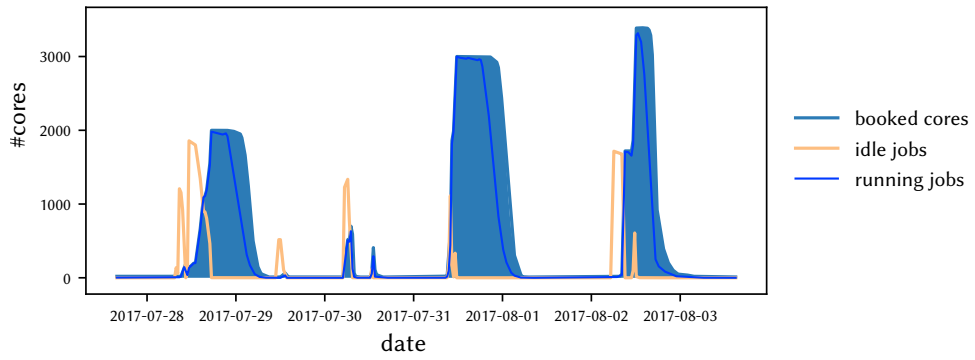

Figure 3: Resource utilization of the KIT HEP group at the NEMO HPC center. The orange line shows the number of idle jobs which corresponds to the demand for additional resources. The blue area represents the number of cores integrated in our batch system. The blue line shows the number of used CPU cores. Depending on the utilization of the NEMO cluster, the number of integrated cores could increases slower than the number of demanded cores. Due to draining effects idle job slots can only be released once all jobs running inside one VM are finished.

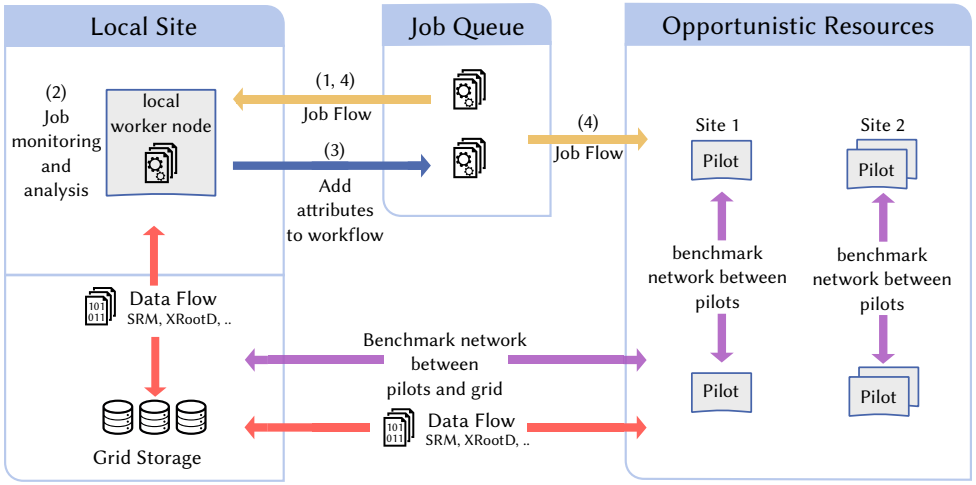

Figure 4: Improved scheduling using information about estimated network utilization of jobs and available bandwidth between sites. (1) Run sample jobs of a given workflow on local resources. (2) Monitor and analyze the network I/O behaviour of these jobs. (3) Add network I/O related attributes to the workflow description. (4) Evaluate additional attributes for the scheduling of jobs on opportunistic resources. In addition, coordinated network benchmarks between the local site and the oppportunistic resource are performed to determine the available network bandwidth.

jobs in the same workflow are very similar. The network I/O behaviour can be estimated by running sample jobs of a workflow on local resources. Afterwards network I/O related attributes are added to the remaining jobs of the workflow.

In addition, coordinated network benchmarks between the Grid sites and the opportunistic resources are performed to determine the available bandwidth using for example the perfSONAR [14] toolkit.

By taking into account the available bandwidth and the estimated network I/O requirements of the jobs in a given workflow, the scheduling process can be improved leading to a more efficient

utilization of the available network bandwidth and therefore to a better CPU utilization on the opportunistic resources. Both concept are currently in development and will be evaluated in the future.

6. Summary

This paper presents the current state of the integration of opportunistic resources into the KIT HEP computing resource pool. To deploy the required HEP software and operating system stack on those resources modern virtualization and container technologies are utilized. The dynamic on-demand resource management is done using the cloud scheduler `ROCED` developed at KIT. All acquired resources are integrated into the same `HTCondor` batch system to allow transparent access to the users. Further concepts to tackle the challenge of efficient remote data access arising from the usage of opportunistic resources by taking into account additional information about network utilization and available bandwidth during the job scheduling process are presented as well. Overall the ability of the dynamic integration of opportunistic resources into HEP computing has been demonstrated.

Acknowledgment

The authors acknowledge the support by the DFG-funded Doctoral School "Karlsruhe School of Elementary and Astroparticle Physics: Science and Technology", the German Helmholtz Association and the state of Baden-Württemberg through bwHPC.

References

[1] Eck C et al. 2005 LHC computing Grid : Technical Design Report *Technical Design Report LCG*, https://cds.cern.ch/record/840543

[2] Bird I et al. 2016 Status of the WLCG Project, including Financial Status, CERN-RRB-2016-124, https://cds.cern.ch/record/2210410 [accessed 2017-10-12]

[3] Buncic P et al. "CVMFS" [software], version 2.X.Y, http://iopscience.iop.org/1742-6596/219/4/042003

[4] XRootD project, "XRootD" [software], http://xrootd.org/

[5] HTCondor project, "HTCondor" [software], version 8.6.1, Available from https://research.cs.wisc.edu/htcondor/downloads/ [accessed 2017-10-12]

[6] Docker inc., "docker" [software], version 17.05.0-ce, Available from https://www.docker.com/ [accessed 2017-10-12]

[7] Kurtzer GM, Sochat V, Bauer MW, "singularity" [software], version 2.2.1, 11 May 2017. https://doi.org/10.1371/journal.pone.0177459

[8] ROCED project, "ROCED" [software], version 1.0.0, Available from https://github.com/roced-scheduler/ROCED [accessed 2017-10-12]

[9] bwHPC-C5: Coordinated Compute Cluster Competence Centers, http://www.bwhpc-c5.de/ [accessed 2017-10-12]

[10] Helix Nebula Science Cloud, http://www.hnscicloud.eu/ [accessed 2017-04-02]

[11] Meier K et al. 2016 Dynamic provisioning of a HEP computing infrastructure on a shared hybrid HPC system, *Journal of Physics: Conference Series* Volume 762 012012

[12] Fischer F et al. 2017 On-demand provisioning of HEP compute resources on cloud sites and shared HPC centers, *Journal of Physics: Conference Series* Proceedings of 22st International Conference on Computing in High Energy and Nuclear Physics (to be published)

[13] Heidecker C et al. 2017 Provisioning of data locality for HEP analysis workflows, *Journal of Physics: Conference Series* Proceedings of 18th International workshop of Advanced Computing and Analysis Techniques (to be published)

[14] perfSONAR collaboration, perfSONAR [software], Available from https://www.perfsonar.net/ [accessed 2017-10-12]