Flexible HPC: bwForCluster NEMO

Bernd Wiebelt, Konrad Meier, Michael Janczyk, and Dirk von Suchodoletz

Rechenzentrum, Universität Freiburg

Traditional High Performance Computing on *bare metal* is based on the paradigm of *maximum usage of resources* whereas Cloud Computing relies on virtualization and is considered as a shift towards *flexible usage of resources*. The bwForCluster NEMO is a HPC system that offers virtualized research environments as an additional mode of operation, thus spanning a bridge between both paradigms. The important achievement is that for the HPC scheduler, a virtual machine instance is *just another job*. There is no static partitioning necessary and the HPC scheduler keeps control over accounting and fair-share.

1 Motivation

The bwForCluster NEMO, part of the bwHPC initiative of the state of Baden-Württemberg, was designed and procured as a High Performance Computing system serving the scientific domains of Elementary Particle Physics, Neurosciene and Microsystems Engineering. During the procurement period of the system it became clear that the classic HPC model of operation would not be sufficient to satisfy the demands of all those scientific communities. In particular, the Particle Physics community had requirements regarding reproducibility that would have been hard to implement and sustain by installing, maintaining, running and ultimately binding the software directly to the underlying hardware. Therefore, while procuring the final HPC system, research was done in the framework of the bwHPC-C5 project [1] whether virtualization technology could be applied and integrated to allow additional usage profiles for a HPC system.

2 Virtualized Research Environments

In the classic HPC model the software environment is heavily optimized towards the acquired HPC hardware. This is done to achieve the maximum possible performance for the individual compute tasks to be accomplished. In turn, usage of the optimized software environment without the HPC hardware becomes cumbersome or impractical. On a different hardware, the software environment has to be completely redeployed. Optimizations that benefited the software running on the original hardware might turn out to yield inferior results on the new hardware. In any case, since the newer hardware typically needs newer versions of the operating system and system libraries, even by using the same version of the application software, reproducible results cannot be guaranteed.

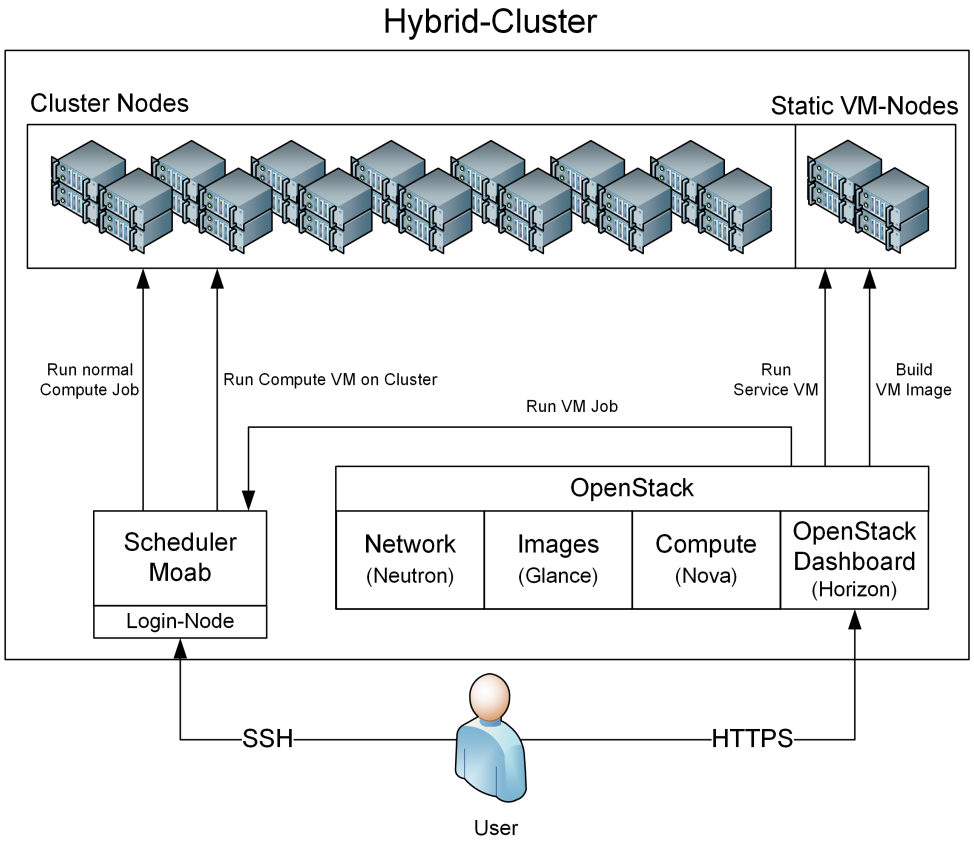

Figure 1: NEMO HPC/Cloud interface [3].

Cloud Computing, by abstracting the software environment from the underlying hardware, is known to solve these problems. Virtualized research environments can be created as virtual machine images. Once created, they are ready to run inside the virtualization framework, yielding the same results no matter how often the underlying hardware changes. As immediate additional benefits, virtualized research environments can be easily versioned, copied, archived and referenced.

The caveat is that there is a performance loss due to the virtualization overhead. However, this is perfectly acceptable if the benefits of virtualized research environments outweigh the loss in performance for an individual use case or an entire scientific work flow.

General feasibility and provisioning of virtualized research environments for scientific communities is a research topic currently investigated by the Baden-Württemberg VICE project [2].

3 NEMO HPC/Cloud

The bwForCluster NEMO is running virtualized research environments on the HPC hardware orchestrated by an OpenStack server. For the HPC scheduler, virtual machine instances are ”just another job”. There is no static partitioning necessary and the HPC scheduler keeps control over accounting and fair-share. Scientists who would like to use the virtualization feature need to provide their own custom VM images. They modify the script *startVM.py* and add it as an

instruction in their job description file. The script allocates the job resources, starts the virtual machine instance via the OpenStack nova interface and then waits for the virtual machine instance to be finished, e.g. by the user issuing a halt operation inside the virtual machine [3].

4 Future HPC Virtualization Concepts

Using virtualization inside an HPC system opens up the possibilities for several interesting features. While their implementation would require tighter integration between HPC scheduler and virtualization framework, they could solve several classic problems with HPC systems, especially those designated for novice HPC users.

Snapshot and migration functionality for running virtual machine instances are a typical feature of virtualization frameworks. This means that running processes can be stopped, possibly moved to a different node in the virtualization cluster and then resumed. For an HPC system, this would be practical for two use cases. The first one concerns long running monolithic jobs. These are, for very practical reasons, non favored jobs in HPC environments, assuming they are permitted in the first place. However, the costs to adapt a particular workflow based on such monolithic tasks to a HPC system, e.g. by parallelizing and partitioning it manually, may sometimes exceed the practical use of the resulting solution. If the monolithic job could automatically be stopped, checkpointed and resumed at regular intervals, this might very well constitute a more economic procedure. In the second use case, if there is a mix of pleasingly parallel high throughput jobs (using only single cores or nodes) and massively parallel high performance jobs (using several nodes), the second class of jobs should be concentrated on nodes that share optimal high performance network communication paths. Typically this is accomplished by high investments in the network topology or sophisticated tuning of the job queue. However, if jobs could be moved around the physical machines (i.e. "de-fragmented"), optimal high performance network communication paths can be guaranteed by concentrating massively parallel jobs on the same or adjacent high performance network switches.

Last but not least, to make an HPC system capable of processing sensitive data, the usual strategy is to isolate it from other systems. However, this also means that the HPC resources cannot easily be shared. Virtualization of HPC resources could in principle enable the creation of isolated safe data center partitions inside a shared HPC system.

References

[1] http://www.bwhpc-c5.de

[2] https://www.alwr-bw.de/kooperationen/vice

[3] K. Meier, G. Fleig, T. Hauth, M. Janczyk, G. Quast, D. von Suchodoletz, and B. Wiebelt "Dynamic provisioning of a HEP computing infrastructure on a shared hybrid HPC system." Journal of Physics: Conference Series, Volume 762(1):012012, 2016