Spartan and NEMO: Two HPC-Cloud Hybrid Implementations

Lev Lafayette
Department of Infrastructure
University of Melbourne
Melbourne, Australia
lev.lafayette@unimelb.edu.au

Bernd Wiebelt
High Performance Computing
Albert-Ludwigs-Universität
Freiburg, Germany
bernd.wiebelt@rz.uni-freiburg.de

***Abstract*—High Performance Computing systems offer excellent metrics for speed and efficiency when using bare metal hardware, a high speed interconnect, and parallel applications. This however does not represent a significant portion of scientific computational tasks. In contrast cloud computing has provided management and implementation flexibility at a cost of performance. We therefore suggest two approaches to make HPC resources available in a dynamically reconfigurable hybrid HPC/Cloud architecture. Both can can be achieved with few modifications to existing HPC/Cloud environments. The first approach, from the University of Melbourne, generates a consistent compute node operating system image with variation in the virtual hardware specification. The second approach, from the University of Freiburg, deploys a cloud-client on the HPC compute nodes, so the HPC hardware can run Cloud-Workloads for backfilling free compute slots.**

Keywords—high performance computing, cloud computing, high throughput computing

I. Introduction : The Problem Stated

Massively parallel workloads need a large proportion of system resources which and can lead to under-utilization. In contrast compute cloud workloads typically produce use small fractions of the compute resources, but often in large quantities. In addition HPC systems are designed with a largely predefined operating environment, whereas the typical cloud compute environment provides flexibility in operating system or extended image sets. A cloud environment offers more flexibility at the expense of the virtualization overhead and a loss in efficient multi-node communication. For a lot of high throughput workloads, the performance loss is negligible and will be completely compensated by the gain in flexibility in designing the software environment.

The question raised is whether it is possible to overcome the dichotomy of cloud versus HPC a single cohesive system. Is it possible to provide the performance of an HPC system which also providing the flexibility of a cloud compute environment? Further, can such a single system provide the best possible result for overall throughput and a better use of computational resources? This is no mere fancy. Applications and datasets are optimal for different computational workflows and therefore different computational architectures. It is certainly preferable from a user's perspective that a single system is capable of adapting to these diverse requirements, rather than having to migrate data to different systems according to task.

II. Hybrid Architectures

A. HPC with Compute Nodes as Cloud VMs

The University of Melbourne approach is to have a traditional HPC cluster with a high speed interconnect in one partition or queue, and an alternative partition or queue which makes use of a collection of virtual machines managed through OpenStack as compute nodes. In this case the virtual machines on the partition use a common image just like the traditional HPC compute nodes but with variant virtual hardware according to the results of user job profiling either based on prior use or specific requests.

Jobs are submitted to the Slurm workload manager scheduler specifying which partition that they wish to operate on. In addition the login and management nodes are also deployed as virtual machines. The collection of virtual machines comes from the Melbourne share of the Australian-wide NeCTAR research cloud. The virtual machines can be configured flexibly into different partitions with virtual hardware specifications in accordance to user needs.

Of particular importance is assuring that the HPC “physical” partition has a high-speed interconnect. Mellanox 2100 switches with 16 x 100Gb ports with a mixture of 25/50/100Gb, maximum of 64x25Gb connections with RDMA over ethernet and Cumulus Linux OS. An MPI ping pong test was conducted between two compute nodes on separate infrastructure, with 40Gbe RDMA over ethernet receiving better latency results than 1.15 than 56Gb Infiniband FDR14 on a comparable system.

B. HPC with Cloud VMs on Compute Nodes

The cloud setup at the University of Freiburg uses a HPC cluster with standard components like a high speed interconnect. On top of the HPC configuration it enables users to run virtual machines as standard compute jobs (VM jobs). To run virtual machines on every compute node on the cluster the KVM hypervisor is on each of these nodes. This architecture enables users to run compute jobs on bare metal through the resource manager (bare metal jobs) or inside a virtual machine (VM job) without partitioning the cluster into two parts.

For the Management of the virtual machines on the cluster the OpenStack framework is used. If a compute job is designed to run in a virtual environment (VM) the Moab scheduler is

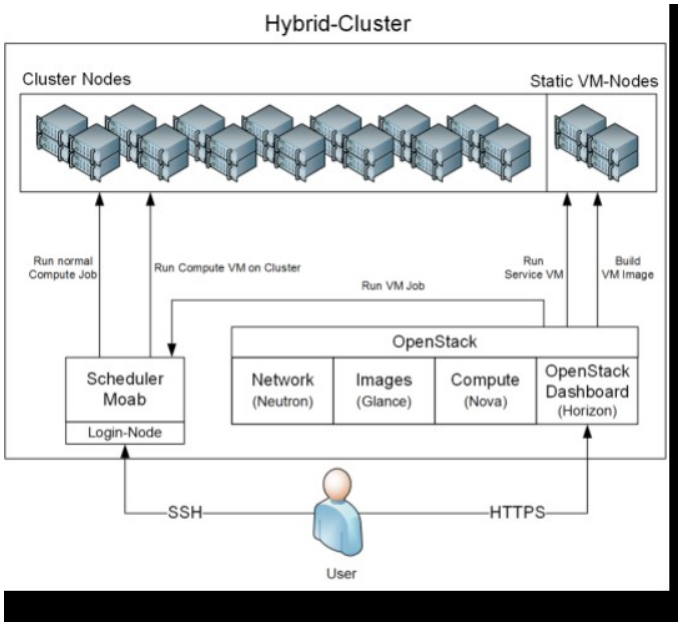
configured to start this VM through the OpenStack API. A special VM environment which currently consists of four compute nodes is used for preparing VM images and for running service. These compute nodes and virtual machines are not under control of the Moab scheduler and are not part of the compute resources of the cluster.

III. Workflow

The University of Melbourne uses a traditional HPC workflow where job Submission with Slurm Workfload Manager occurs on different partitions according to whether they are based on physical or cloud architectures. At the University of Freiburg three workloads; job Submission via Moab scheduler without running a resource manager client in the VM., job Submission via Moab scheduler with a resource manager client (Torque) running in the VM., job Submission via OpenStack Dashboard/API.

A. Job Submission with Slurm for Different Partitions

In the University of Melbourne model, the Slurm workload manager acts as the job scheduler and resource manager. Different partitions refer to the queues which a user may submit jobs and are varied by the physical or virtual hardware that have been provisioned. For single node jobs, whether single or multicore, a low speed network and virtual machines are suitable, whereas for multinode jobs with a high-speed interconnect the physical partition is used. Deployment of compute node according to partition is carried out with a simple script which invokes the OpenStack Nova service to deploy specific images. Thus, the physical architecture is optimised for the type of computational task required, increasing overall throughput and more efficient resource allocation.



B. Job Submission via Moab Scheduler without running a resource manager client in the VM

At the University of Freiburg, there is a one use case where users provide their VM images themselves. These VM images cannot be trusted and therefore they are not allowed to access cluster resources like parallel storage or user home directories. These VM images have no Torque client running so it is expected that the user is working with an external resource manager or is using the cloud-init procedure to start compute jobs within the VM.

C. Job Submission via Moab scheduler with a resource manager client (Torque) running in the VM

A second use case at the University of Freiburg is when the user submits classic compute jobs to a different software environment on the cluster. The software environment is represented by a VM in this case. This makes it necessary to install and run a torque client in the VM.

D. Job Submission via OpenStack Dashboard/API

The third use case from the University of Freiburg is when the user submits compute jobs simply by creating a VM via the OpenStack web interface (Horizon) or OpenStack API. These VMs then should be represented as a compute job in the Moab scheduler. The compute job script is injected via cloud-init into the VM during boot and is executed in the VM after the boot process is finished.

IV. Conclusions

The two models - HPC with Cloud VMs on Compute Nodes, and HPC with Compute Nodes as Cloud VMs - represent different hybrid systems to solve different problems. In effect, the University of Freiburg model provides a “cyborg”, where the HPC compute nodes are replaced with cloud virtual machines, whereas the University of Melbourne model provides a “chimera”, a multi-headed beast where the virtual machines have become new cloud nodes. In the former case there was a desire to make existing compute nodes available to researchers for their particular configurations. In the latter case there was a desire to make virtual machines accessible to an HPC system to provide a cost-efficiencies and improved throughput. The two approaches illustrate the importance of HPC-Cloud hybrids in the provision of general purpose research computing.

For future development, the University of Melbourne's model provides the ability to include cloudburstting to external providers (e.g., Amazon, Azure), as well as hosting highly varied architectures on the same system. For the University of Freiburg's model, mapping Moab commands to OpenStack commands allows to pause/hibernate and resume the virtual machine for preemption or maintenance, rather than killing a job. In addition the possibility of mapping to Moab the live migration of virtual machine during runtime gives the opportunity to migrate compute jobs during runtime to optimize the overall cluster utilization.

Acknowledgments

Lev Lafayette would like to thank Bernard Meade, Linh Vu, Daniel Tosello, and Greg Sauter for their contributions to this document. Bernd Wiebelt would like to thank Konrad Meier, Michael Janczyk, and Dirk von Suchodoletz, for their contributions to this document