Exploring Issues in Event-Based HPC Cloudbursting

Lev Lafayette
Senior HPC Support and Training Officer
HPC Advisory Council, Fremantle, 2018
HPC and Cloud Use Cases

Standard HPC architecture provides exceptional performance for complex computational tasks and will continue to dominate supercomputing metrics. Numerous studies over time have shown that there is modest overhead in cloud implementations [1,2], along with cost issues [3].

Whilst improvements in overhead are ongoing the major advantage of cloud computing is flexibility and opportunity cost. The particular example of Spartan at the University of Melbourne - where single-node computational tasks dominated - is illustrative [4] for the principles of proportionality and incremental design.
Hybrid HPC-Cloud Models

Two major models are available for internal HPC-cloud hybrids; the chimera model (e.g., University of Melbourne) and the cyborg model (e.g., University of Freiburg). In the former cloud images are deployed as virtual machines and become an HPC partition. In the latter, the resource allocation from job submission allows for user-managed virtual machines [5].

In the chimera model, an option is available for cloudbursting ("elastic computing"), where additional compute can be allocated as part of the job submission system to a specified partition. The Slurm Workload Manager [6], for example, makes use of its a subset of its power-saving feature where the ResumeProgram script equates with bursting.
Internal Cloudbursting

Because the majority of the University of Melbourne's cloud partition infrastructure was sourced from the local allocation of the NeCTAR research cloud [7] it provided a very good test case to experiment with Slurm's elastic computing model; same data-centre, same underlying hardware, same provisioning scripts. Initial testing allowed the entire vHPC partition to be converted to a cloud-bursting model.

Unfortunately, with Spartan's version of Slurm at the time (16.05.3) we discovered a bug [8] whereby the slurmctld would lose the list of excluded nodes and partitions on a `scontrol reconfigure` command and then will treat all nodes as being eligible for power control, neglecting the SuspendExcNodes and starts to suspend them, putting the excluded nodes into a power saving state. Although this was resolved by the Slurm team fairly quickly, it was decided that whilst the cloud-bursting experiment was successful, it was also a risk with additional variables.
External Cloudbursting

A further investigation was carried out for external cloud-bursting, where jobs launch on a specific partition (e.g., `burst`) could be attached to a public cloud. We were approached by the usual suspects to encourage this implementation. However to achieve this completely, a replication of the existing HPCs environment would be required (remote mounting is not feasible), including home or project directories, the Slurm controller daemon and database, LDAP, specialised application installations, etc. Others have noted the need for homogeneous configurations between host HPCs and remote clouds [9].

Proof of concept of external cloud bursting is available [10], and in production it is plausible as subset of an HPC architecture. The method we have explored would involved a wrapper script would contact the active public cloud, spin up a node, copy the necessary data, submit a job on the remote system (using an identical version the application and toolchain), and copy the data back to the HPC, and powerdown the node. Another feasible alternative is the transfer of pre-configured Singularity containers.
Another option for external cloud-bursting is Moab/NODUS cloudbursting by Adaptive Computing [11]. The principles used by NODUS are similar to the exploration here; a workload queue has an elastic trigger that uses an API key to an template image which then invokes nodes, deploys upon them and potentially cluster nodes simultaneously, completes the job, and transfers the data as necessary. Note that the same issues are raised in this environment that are elsewhere; partitionining has to be strict, replication and transfer times is essential. Notably, Adaptive Computing specifically argues for external cloud-bursting for business reasons rather than technical reasons.
Cloudbursting Ideals vs Reality

If demand is sufficient pre-building specific partitions of worker nodes that are matched can be justified. Specific cases include when capacity is low and a longer-running job (i.e., initialisation is a low percentage of overall runtime) is required with a specific application (low administrative burden). Another usage is minimal replications of an environment specifically for training and teaching purposes, to avoid existing queue-wait times and to ensure that novice users do not interrupt serious research.

The promise of cloudbursting from HPC, of a transparent means of adding on-demand compute nodes with equivalent performance, is very far from from the reality. Not all research and practise provides the successes originally desired, but the discovery of limits and dead-ends is valuable to reduce the change of others following the same path again, and to provide a successful path, via negativa, through a process of elimination.
References


https://slurm.schedmd.com/elastic_computing.html


[8] Chris Samuel, Bug in node suspend/resume config code with scontrol reconfigure in 16.05.x (bugzilla #3078), Slurm-Dev Mailing List, 25 September, 2016.


http://www.adaptivecomputing.com/moab-nodus-cloud-bursting/
THANKS FOR WATCHING & LISTENING PATIENTLY