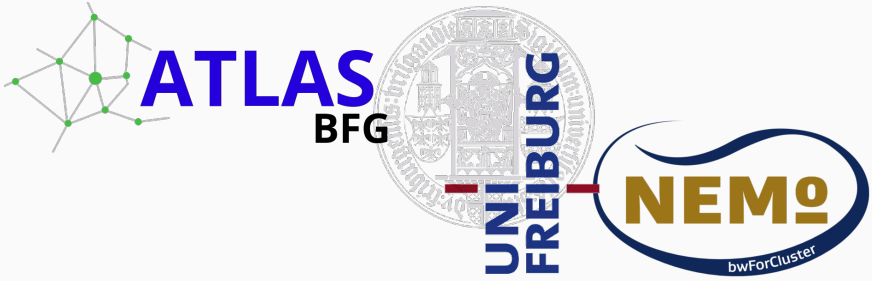
Virtualization of the ATLAS software environment on a shared HPC system: Lessons learned and what's next

Felix Bührer, Anton Gamel, Ulrike Schnoor, Markus Schumacher

Annual Helmholtz Alliance Meeting - Tuesday 28th November, 2017



The situation

- Since beginning of 2017:
  Exclusive System for experimental HEP
- Tier 2 / Tier 3 of the WLCG
- ~3000 CPU cores
- ~1.7 PB dCache storage
- SLURM scheduler
- No file system for local usage (anymore)

- Shared HPC cluster:
  State-wide resource
- Hybrid HPC and cloud approach
- 17.6k CPU cores
- 750 TB parallel storage
- MOAB/Torque scheduler
- OpenStack instance for VMs

Goal: Provide the full ATLAS software stack on NEMO in a virtualized solution for local users
→ opportunistic extension of the ATLAS Tier 3

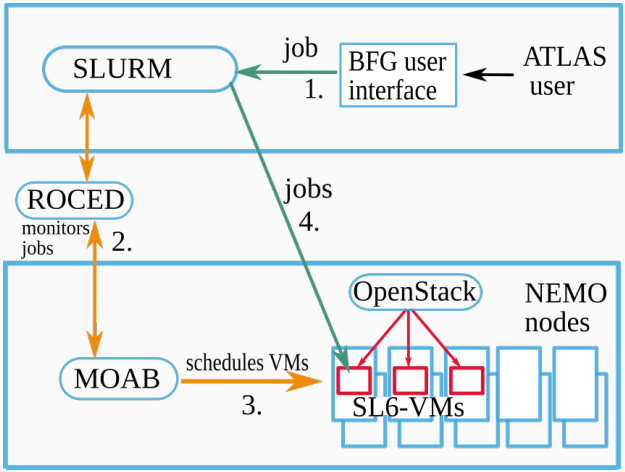
Provisioning of VM image

So far: Providing SLC6 image only (working on CentOS 7 image)

- Using packer together with puppet on a base SLC6 iso to build a qcow image for OpenStack
  → all Freiburg-specific configuration done in puppet
  → same set-up for VM images and bare-metal machines
- Until now: Maintaining two separate images - Working on being able to use VM image for bare-metal machines
- Generating images with packer gives flexibility for further developments (e.g. can output Docker container instead of Openstack image)
- Updated images can be provided, tested and deployed with no delay or downtime
  → create and upload new image, test it and set it as default when happy

Next step: How can users on one system (ATLAS-bfg) trigger VMs on the second (shared) system?

Using VMs for user jobs



- Using ROCED (https://github.com/roced-scheduler/ROCED) as an intermediate layer between the two batch systems:
- More detailed information on ROCED in the talk by Matthias later
- Userjobs are sent to special NEMO queues
- ROCED calculates the number of VMs needed
- Submits jobs that start VMs to MOAB on NEMO
- When job starts and VM is ready: SLURM sends job to this node
- Time for job to start depends on current usage of NEMO and group priorities

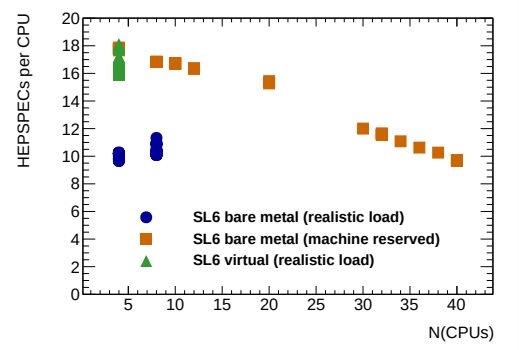
Lessons learned so far

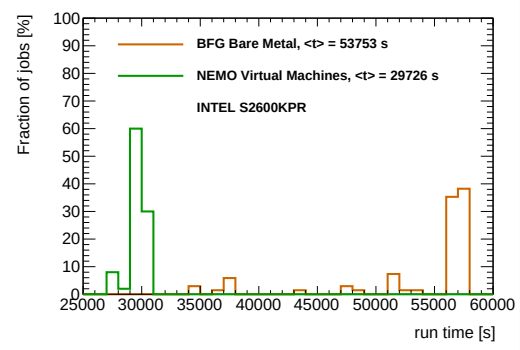
Went from testing to user-ready in October

1. Users will use resources that become available (especially resources that don't run production jobs)
2. SLURM performance deteriorates greatly when adding hundreds of (potential) nodes
   → Timeouts leading to ROCED not being able to monitor queues
   → Jobs not starting
3. Current 4-core machines lead to a lot of overhead when large job-arrays are submitted (hundreds of VMs starting in a short time)
   → Will move to larger VMs as baseline
4. Using a front-end scheduler makes it possible to reproduce user-fairshares even though the backend is only aware of group-shares
   → ROCED requests VMs for all queued jobs
   → If not enough resources are available: SLURM prioritises according to user fairshare

First benchmarks

During provisioning and testing phase: Ran benchmarks comparing virtual to bare metal machines





All benchmarks ran on identical hardware. More tests are ongoing.

So far: virtualization does not seem to have a big effect on performance

What's next

First weeks of usage already highlighted potential fields of further development:

1. Unified queue
   - So far: Users need to decide at submission if jobs are sent to bare-metal or VM queue
   - Need to check current usage of backend scheduler via website
   - ROCED node has access to both systems: In case of free resources on NEMO VMs can be requested and jobs moved to VMs
   - Real opportunistic resource
2. Error propagation
   - In case of problems with VMs or Openstack users don't get any feedback why jobs crashed
   - Need to properly announce / propagate errors from one system back to the other (error output from MOAB job that starts the VM)
3. Monitoring of current status
   - ROCED is an invisible layer to the user
   - Also hard to quickly monitor if everything is running properly for admins
   - Processing information about current status (Machines running / number of jobs to start VMs queued / Error messages) and present them in an easy to digest way (User & Administrator view)
   - Monitoring can also be used for simple automatic measuring of job attributes