HPC usage experience EU

- Rod Walker, LMU Munich
28th Jan 2016

- Munich centric view
- Not mentioning
  - long-running Nordic HPC
  - potential usage (UK, FR)
  - China
- General comments on HPC for ATLAS

Munich HPC

- LRZ SuperMUC
  - Phase 1: 150k cores, Sandybridge
  - Phase 2: 86k cores, Haswell
  - 10Mcore hours used from 20M allocation
    - effectively open-ended allocation if preempt-only
- Max Planck Institute computer centre: Hydra
  - 83k Sandybridge

ATLAS ProdSys integration

- Benefit from ND middleware and experience
- ARC CE designed for non-intrusive integration
  - aCT, stage-in/out data, BS interface(LoadLeveler)
  - added ability to have remote CE access cluster via ssh
- ATLAS SW available by rsync of cvmfs and relocation, more recently parrot.
  - SLES11 workarounds(openssl naming convention)
  - no outbound IP → no Frontier → only sim
  - only whole-node scheduled → AthenaMP

ARC CE via ssh

- Not allowed service on HPC login node
- Key-base ssh allowed
- Mount shared FS using Fuse(sshfs)
- Interact with BS using ssh to run commands
  - important details solved by Michi(Bern, for CSCS)
- Remarkably stable
- Not optimal for data movement (ok for sim)

Parrot-CVMFS for HPC

- CVMFS needs no introduction
  - needs a local cache,... and Stratum-0 source
  - needs WN root mount, or at least FUSE
  - needs outbound IP connectivity
- HPC fails on all counts
  - no local disk, no (local)cache
  - no root, no fuse
  - no connectivity

http://cernvm.cern.ch/portal/filesystem/parrot

Parrot-cvmfs

- Parrot is part of the cctools suite
  - http://ccl.cse.nd.edu/software/
  - much history and collaboration with cvmfs(Blomer)
- Wrapper around command/script/binary to intercept FS operations and do something
  - inc. HTTP, FTP, GridFTP, iRODS, CVMFS, Chirp
  - access to /cvmfs handled by plugin from Jakob
- Still requires outbound IP and proxy.

Parrot fun

Cvmfs anywhere

```
[aipanda121] cctools $ ls /cvmfs/atlas.cern.ch
ls: cannot access /cvmfs/atlas.cern.ch: No such file or directory
[aipanda121] cctools $ cctools-5.3.4-x86_64-redhat6/bin/parrot_run bash
[aipanda121] cctools $ ls /cvmfs/atlas.cern.ch
repo
[aipanda121] cctools $
```

Make sure TRF does not need AFS

```
[aipanda121] cctools $ ls -d /afs/cern.ch
/afs/cern.ch
[aipanda121] cctools $ cctools-5.3.4-x86_64-redhat6/bin/parrot_run --mount=/afs=/dummy bash
bash-4.1$ ls -d /afs/cern.ch
ls: cannot access /afs/cern.ch: No such file or directory
bash-4.1$
```

Alien cache

- Cvmfs cache can be on a shared FS
  - used by all clients, but still needs outbound IP
- Cvmfs cache can be pre-loaded
  - copy of stratum-0, 100% cache hits
  - no outbound IP required → HPC
- Pre-loading can choose directories
  - anything containing .cvmfscatalog file
  - eg. base releases, DBReleases
  - faster than rsync
- Parrot ptrace style intercepts not without difficulty
  - several problems found and quickly fixed by cctools dev
    - argument ignored, seg fault, tar for log fails (on SLES)

```
> export PARROT_CVMFS_ALIEN_CACHE=/gpfs/work/pr58be/ri32buz2/cvmfs_preload
```

Optimized FS access

- Particular SuperMUC Phase1 problem
  - GPFS client configuration not good for ATLAS
    - inode cache too small(1000) - delays on file access
    - G4 accesses O(1000) data files → thrashing
- cvmfs has some internal caching
  - fewer GPFS inode lookup operations
  - effect is dramatic …
    - G4 Initialization: 32mins → 5mins
    - time per event: 115s → 35s
    - both comparable to native cvmfs

Current usage

- ARC CE each for Phase 1 and 2
- Asked to run in preempt queue on phase 1
  - this part has the larger jobs → backfill potential
- Running 200 whole-node jobs (3200 cores), 4hr wall limit
  - usually at 200 limit.
    - occasionally drains a little. Rarely O(20) jobs preempted.
  - cannot delay 'proper' HPC job
  - negotiating increased limit
    - usually >1000 nodes idle
- 10M core hours running standard production G4
  - Reco-pile needs CD in DBRelease: no Frontier
- MPI Hydra also running in production ~60 nodes
- Looking forward to ES -ARC integration
  - negate preempt loss, maximize backfill

General HPC use for ATLAS

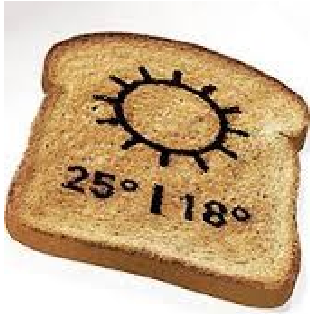
- Non-x86 covered by US talk
- CSCS intend to provide T2 inside general HPC
- Potential to contribute to HPC bid and design
  - efficient way to provide cpu power to science
    - single facility for capability computing and HTC
  - as a stake holder, HEP can ensure pledge
- Needs an attitude shift from HPC

Hardware choices

- Like: Linux & x86 – maybe MIC too
- Agnostic about: fast network, Batch system.
- Can live with: OS & low RAM/core
  - prefer container-based virtualization(Docker, Shifter in US)
- Unhappy: Lack of compute node disk
  - computer is cpu, RAM, storage
  - OS lives in RAM, no swap
  - no local scratch for high io or caching(cvmfs)
  - disk adds little $, and does not hurt HPC

Policy

- Outbound connectivity
  - no self-respecting HPC code would need the Internet
    - HEP code does: Frontier, cvmfs, wget, ...
    - even toasters have Internet!



  - assumption that users and intruders are queuing up to DoS attack a litigious bank
    - destinations controlled and throttled by firewall/NAT rules
    - no danger

Policy(2)

- Only multi-node jobs
  - HEP has almost no need – wonderfully parallel
    - exception some evgen integration(Mira)
  - fragmentation of resources
    - scheduling question. Only short or preemptable jobs.
  - batch system load
    - only whole-node jobs implies 10k max – OK.
  - SuperMUC and Hydra accept single-node jobs
    - makes perfect sense with preemption enabled

Policy(3)

- No gateway
  - or not useful GT5, UNICORE
- Must login to headnode to submit jobs
  - key-based ssh if lucky, or securID code if in US
- HEP needs a gateway
  - integration to *automatic* production system
  - data in/out , job submit, monitor
  - real HPC users would benefit too

Conclusion

- Persistence overcomes HPC hostility
  - masters and admins are often positive and helpful
    - but feel inhibited by funding and tradition
  - takes time, pressure from above
    - SuperMUC, Hydra in production with compromises
  - challenge each policy decision, for justification
- HEP stake in new HPC will change this
  - make cluster useful for more workloads