HEP software on Supercomputers

- Rod Walker, LMU Munich
6th Jun 2016

- ATLAS & Munich centric view
- The Problem
- HPC integration
- Parrot-cvmfs
- Multi-node jobs
- Current usage and future

What is the problem?

- HEP access SW on a range of resources
  - Grid clusters, interactive, boinc
- NFS was ok
  - scaling issues due to large and complex SW
  - distribution and sync painful
- CVMFS is better
- Problem for HPC
  - no outbound IP connectivity
  - no root privileged service or kernel module
  - no local disk
- Together rule out many of the cvmfs alternatives
  - plain, over-NFS, via-Fuse, Parrot
  - left with shared-FS, and associated problems

Munich HPC

- LRZ SuperMUC: pure x86, no acccelerators
  - Phase 1: 150k cores, Sandybridge
  - Phase 2: 86k cores, Haswell
  - ATLAS has 20Mcorehours allocation
    - effectively open-ended allocation if preempt-only
  - Next Generation planned for 2018
- Max Planck Institute computer centre: Hydra
  - 83k Sandybridge

HPC integration to Production

- Panda 'pull' pilot model no longer flies
  - no Internet connection
  - submit real jobs(pre-loaded pilots)
- Benefit from Nordugrid middleware and experience
- ARC CE designed for non-intrusive integration
  - stage-in/out data on shared FS, BS interface(LoadLeveler)
- Since HPC admins fear change and utility → no ARC CE available
  - added ability to have remote ARC CE, access cluster via ssh/ssh-fuse
- ATLAS SW available via rsync of cvmfs
  - to shared-FS directory, needing relocation of hard paths
  - more recently via parrot-cvmfs....

Parrot-CVMFS for HPC

- CVMFS needs no introduction
  - needs a local cache,... and Stratum-0 source
  - needs WN root privilege, 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

Parrot-cvmfs

- Parrot is part of the cctools suite
  - http://ccl.cse.nd.edu/software/
  - much history and collaboration with cvmfs
- Wrapper around command/script/binary to intercept FS operations and do something
  - inc. HTTP, FTP, GridFTP, iRODS, Chirp,CVMFS
  - 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 $
```

Test grid job without 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$
```

Parrot alien cache

- Cvmfs cache can be on a shared FS
  - used and populated 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
- Pre-loading can choose directories, to speed-up/save space
  - anything containing .cvmfscatalog file
  - eg. base releases, DBReleases
  - much 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
```

Bonus: 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
    - can ramp-up SuperMUC usage ...

Multi-node jobs

- Typical HPC workload uses many nodes in a single job
  - ATLAS don't need such jobs: already parallelized
    - no efficiency gain, can only lose
    - preempt whole multinode job rather than just few single node jobs
  - but often a site policy requires it
- Yoda developed to fill this artificial need
  - use mpi to distribute events to AthenaMP processes on multiple nodes
  - mpi ranks inherit the environment of rank 0
    - single SW setup, but must then see same paths on each node
    - means parrot must run in each rank

Parrot for MPI jobs

- Rank 0
  - parrot_run; setup sw env; poe athenaMP --args
- Rank N
  - env is inherited
  - poe parrot_run athenaMP –args
- For Cray, replace poe(IBM) with aprun

Current usage

- SuperMUC: running 300 whole-node jobs (4800 cores)
  - usually at 300 limit.
    - often drains a little. Occasionally O(50) jobs preempted.
  - cannot delay 'proper' HPC job
  - negotiating increased limit
    - usually >1000 nodes idle
  - ran 18M core hours of standard production G4
- UK HPC Archer using ARC CE and Parrot-cvmfs
  - in progress
- CSCS, Prague – interest but not sure of status?

HPC policy

- Outbound connectivity
  - no self-respecting HPC code would need the Internet
    - HEP code does: Frontier(Db), cvmfs, wget, ...
    - much effort invested in making this efficient
- machines funded for wider user community, including HEP – we can hope for ….
  - container VMs
  - allow outbound IP, web proxy cache
  - native cvmfs or Fuse permissions
  - useful gatekeeper(?)

Future & wishlist

- Will need to safely remove directories from pre-loaded cache – to free space.
- Could get Fuse allowed so Parrot/ptrace part not needed
  - tool to start cvmfs this way
  - pre-loaded alien cache supported by plain cvmfs?
- SLES12 rpms please
- Commit node memory for cvmfs cache
  - how about shared memory as cvmfs cache?
    - could be genuine reason to run muli-node jobs

Conclusions

- Parrot with cvmfs an important bridging technology
  - can use existing HPC resources
- NG likely to offer containers, Fuse, outbound IP
  - good that cvmfs dev already thinking about this

ARC CE via ssh

- Not allowed a service on HPC login node
- Key-base ssh is 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
- HPC Cluster has gateway outside their control
  - on VM at LMU – data transfer path not optimal, scaling
  - HPC should provide ARC CE