THE UNIVERSITY
of EDINBURGH

ATLAS HPC

Andrew Washbrook
University of Edinburgh
Centre of Excellence in Physics at Extreme Scales
WP5/A2: Exploitation of HPC Clusters for LHC data intensive workflows and analysis applications
19th September 2014

ATLAS Computing Model

Central management

- PanDA: workload management system, tracking job execution
- DDM: distribution of data among storage sites, data catalogs

Software Distribution

- Installed on centrally managed CVMFS filesystem
  - HTTP-based read-only file system
  - Fuse-mounted on worker node
- Assumes network access from worker node
- Installed by trusted persons, no write access for ATLAS users
- 10GB needed for installation of an ATLAS release

Job Submission

- Pilot-based model - jobs are submitted by pilot factories to a persistent Computing Element service at each site
- Allows for self-regulation of workload based on site load conditions
- Payload not known until job execution
- Data access assumed from worker node

Typical workload

- *Monte-Carlo Geant4 simulation:* 1 hour on 16-cores, 4GB of memory 50MB input, 50MB output
- *Data or Monte-Carlo Reconstruction:* 30 minutes on 16-core, up to 32GB of memory 1GB input, 1GB output

ATLAS HPC Interest

- ATLAS could potentially run simulation on HPC site very efficiently
  - Workload can be parallelised and adapted to match available HPC resources
  - Relatively low I/O compared to reconstruction jobs

- ATLAS HPC working group has been running for some time to investigate the use of HPC for MC event generation and simulation production

- ATLAS has members with access to several leading High Performance Computing facilities:
  - US: Titan, Mira, Stampede, Edison
  - EU: Piz Daint, SuperMUC, **Archer**
  - Several other smaller HPC facilities worldwide

Titan

- Awarded **10M** CPU-hours on Oak Ridge Leadership Computing Facilities (TITAN)
- Not just for ATLAS - collaboration with multiple groups and experiments (ALICE, LSST)

27 PFlops (Peak)
18,688 compute nodes with GPUs
299,008 CPU cores
AMD Opteron 6200 @2.2 GHz (16 cores)
32 GB RAM per node
Nvidia TESLA K20x GPU per node
32 PB disk storage (Luster)
29 PB HPSS tape archive

Sergey Panitkin (BNL)
Danila Oleynik (UTA)

Proposed Solutions

- SAGA (Simple API for Grid Applications) framework used as interface to HPC batch schedulers
  - http://www.ogf.org/documents/GFD.90.pdf
- MPI wrapper/overlay scripts that allow multiple “single node” workload instances in parallel
  - Transparent to ATLAS workload
- Backfill functionality in pilot
  - Collection of information about currently available free resources on HPC machine
  - If job exceeds wait time limit, pilot cancels the job and repeats the cycle

Titan

Testing

- Have conducted 2 x 24 hour continuous job submission tests this year
- MPI wrappers for two workloads from ATLAS and ALICE
- In last test was able to secure ~145k core hours with only short wait times (~2 minutes)

Software Distribution

- Individual release installation using pacman tool
- CVMFS+Parrot did not work for Titan (compare with EU HPC sites)

NERSC

- Awarded 1.1M CPU-hours on NERSC facilities
- PanDA pilot developed for Titan tested on Cray machines at NERSC (Hopper, Edison)
- Batch policies of NERSC machines meant relatively little backfill opportunities on available

Mira

- 786k 1.6GHz **PowerPC A2** cores
- 16 cores/node
- 16GB/node
- 512 nodes/job minimum
- 10 Petaflop max
- 6.8 billion CPU-hours available

- Awarded a **50M** CPU-hour allocation over 12 months (have to burn through 1M CPU-hours each week)

Taylor Childers, Tom LeCompte, Tom Uram (Argonne)
Doug Benjamin (Duke)

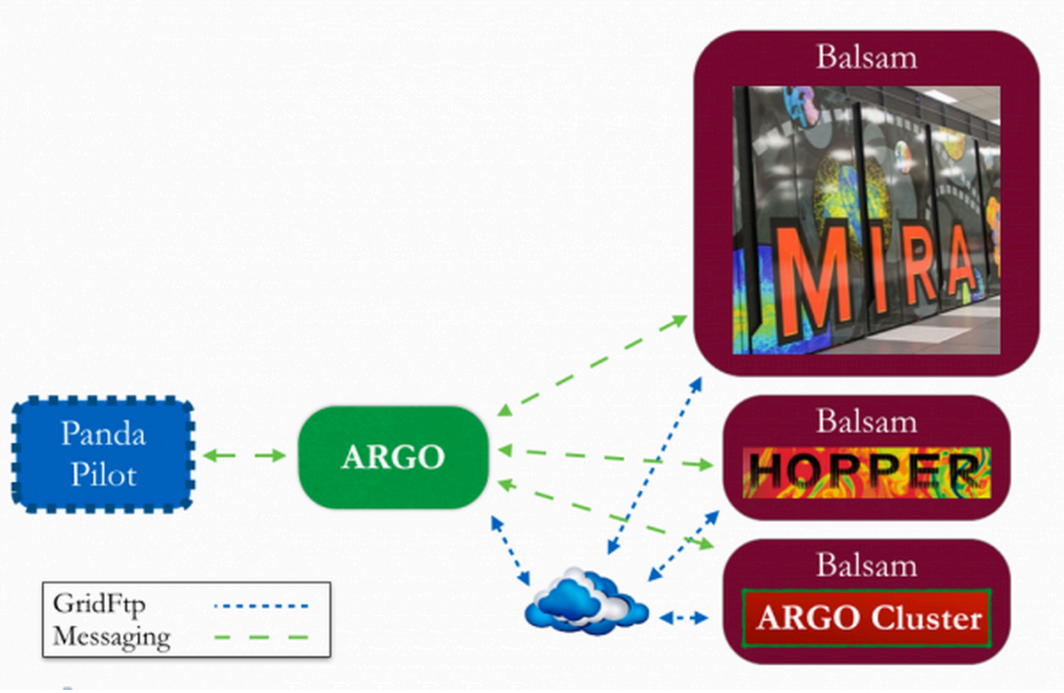
- Focus is on MC event generation e.g. generating Alpgen W/Z+3,4,5jets @13TeV
- Already demonstrated scaling to ~260,000 parallel processes
- Jobs are submitted outside of ATLAS distributed computing system (for now)
- PowerPC architecture results in problems not encountered in x86 architectures (e.g. assigns past end of arrays)
- Dedicated effort required to debug generator code

Mira

Proposed solution

- *Balsam*: A platform-agnostic software that is run on each resource and handles job submission, monitoring, and data motion
  - Run on MIRA and Hopper HPC resources
  - Will be extended to report current queue status
- *ARGO* Workflow Manager:
  - Manages job workflow, data motion, and user interface
  - Can submit job to any connected Balsam resource
  - Backfill enabling: dynamically submit custom job sizes as queues open up

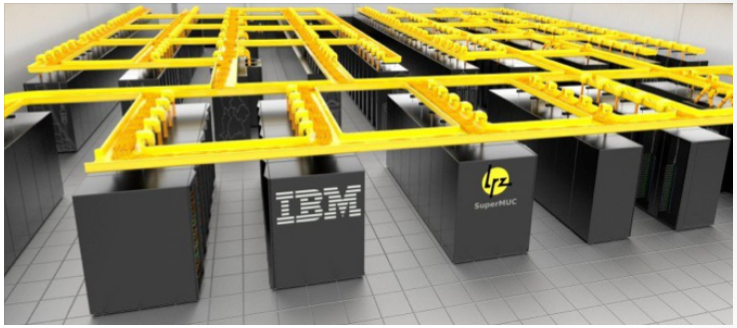
See: https://atlaswww.hep.anl.gov/asc/wikidoc/doku.php?id=hpc:argobalsam



EU HPCs

- Interested EU-based HPC sites have been looking at common solutions
- All x86 based HPCs

- Modified ARC CE middleware to interface with HPC batch systems
- Developed modified ssh backend
- Software distribution by rsync copy of CVMFS or using parrot for CVMFS access
- Possible use of pre-emption queues with short lifetime jobs (~1hr)
- Current workload is mostly multicore simulation production



SuperMUC

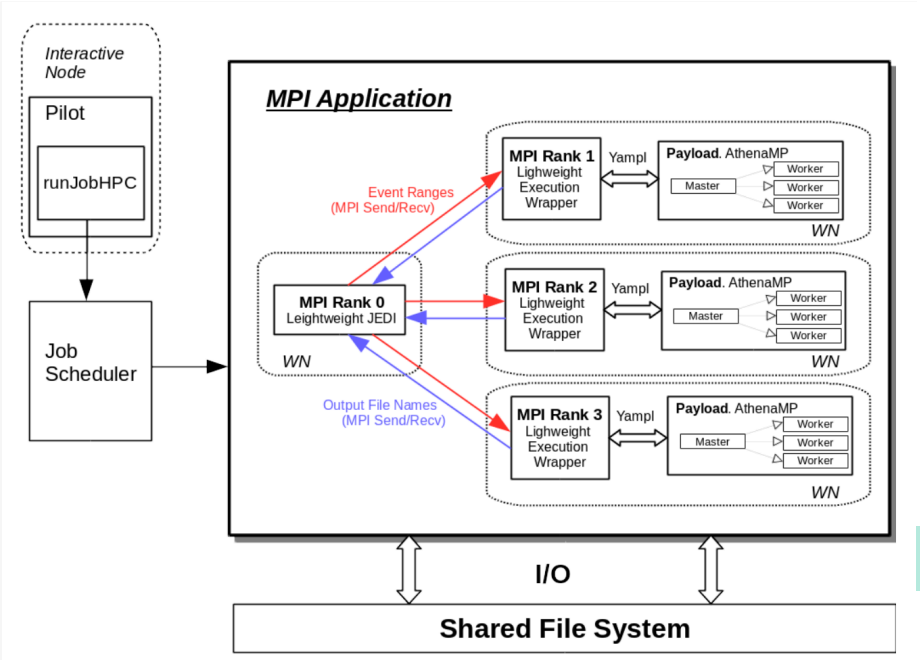
Piz Daint

- Makes sense for Archer to follow this model where applicable
- We will need a PBS module in ARC to allow ssh-based job management
- Collaborating with Titan on HPC pilot development

Solutions: MPI Event Service

- Event service: long jobs run faster in a true parallel mode with efficient checkpointing at a smaller time scale O(mins)
  - Master job: managing the distribution of events
  - Slave jobs: event processing
- Known HPC limitations require this model to be redesigned and also to scale out across HPC resources.
- Ongoing development to turn Event Service into a MPI-application (*Yoda*)
- Job is submitted to the batch system like a regular MPI job (eg. using `aprun` syntax)

- See: https://twiki.cern.ch/twiki/pub/PanDA/EventServer/Yoda.pdf



Vakho Tsulaia (LBNL)