Processing LHC Workloads on Archer

Andrew Washbrook
University of Edinburgh
GridPP 35, Liverpool
11th September 2015

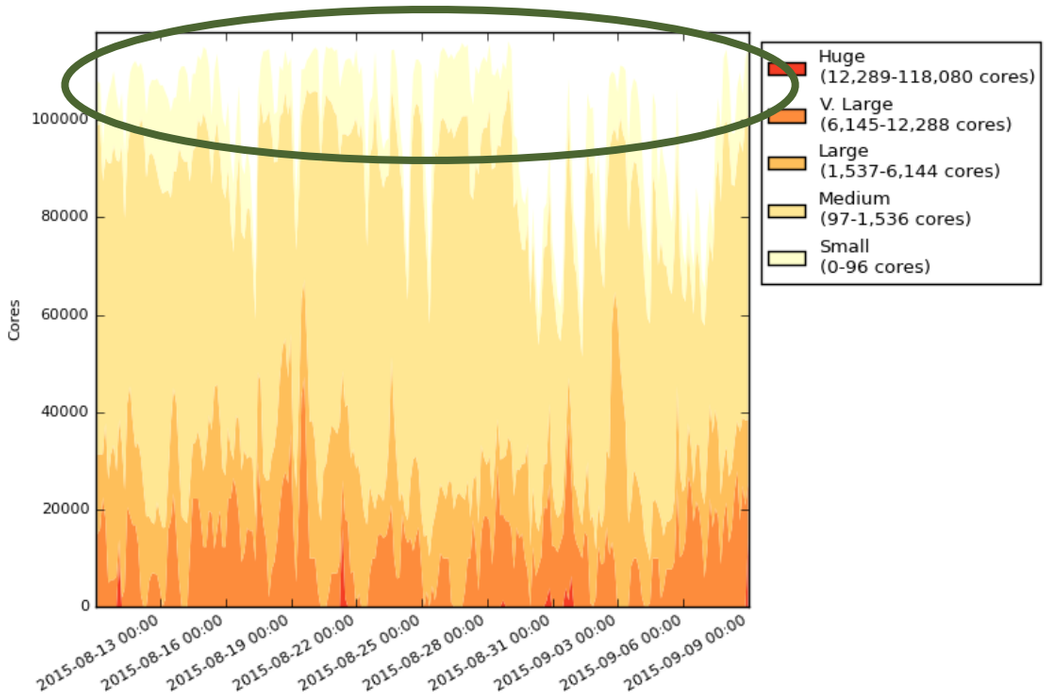
With thanks to: Sergey Panitkin, Michl Hostettler and Dirk Hufangel for providing material

Motivation

Why are HPC facilities being considered as a LHC computing resource?

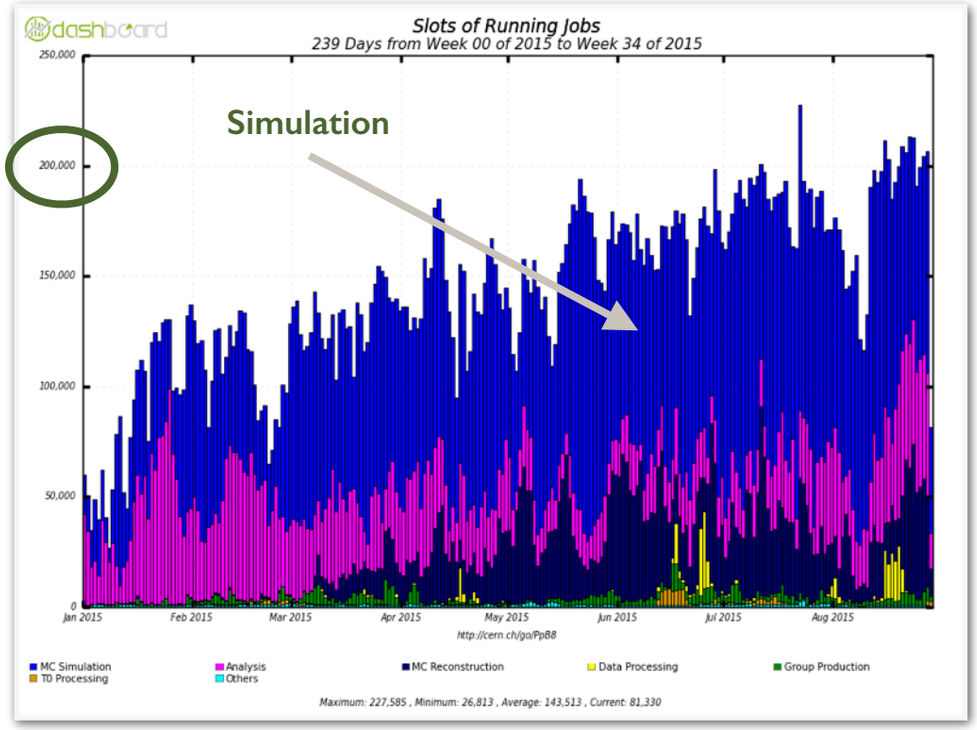
- LHC experiments are looking to exploit opportunistic resources (clouds, volunteer computing and HPCs)
- HPC and Leadership Class Facilities (LCFs) offer a potentially large source of computing power
- Typical HPC facility on average at **90%** occupancy
  - On Titan this translates to over **300 M** unused core hours per year
- LHC workload is well placed to take advantage of idle CPU cycles
- Can co-exist with other, bigger HPC workloads with minimal impact on job scheduling

10,000 free CPU cores



Archer Utilisation (past month)

200,000 cores



ATLAS total slots of running jobs by job type (2015)

HPC facilities

RANK	SITE	SYSTEM	CORES	RMAX (TFLOP/S)	RPEAK (TFLOP/S)	POWER (KW)
1	National Super Computer Center in Guangzhou China	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3,120,000	33,862.7	54,902.4	17,808
2	DOE/SC/Oak Ridge National Laboratory United States	**Titan** - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560,640	17,590.0	27,112.5	8,209
3	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1,572,864	17,173.2	20,132.7	7,890
4	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705,024	10,510.0	11,280.4	12,660
5	DOE/SC/Argonne National Laboratory United States	**Mira** - BlueGene/Q, Power BQC 16C 1.60GHz, Custom IBM	786,432	8,586.6	10,066.3	3,945
6	Swiss National Supercomputing Centre (CSCS) Switzerland	**Piz Daint** - Cray XC30, Xeon E5-2670 8C 2.600GHz, Aries interconnect , NVIDIA K20x Cray Inc.	115,984	6,271.0	7,788.9	2,325
7	King Abdullah University of Science and Technology Saudi Arabia	**Shaheen II** - Cray XC40, Xeon E5-2698v3 16C 2.3GHz, Aries interconnect Cray Inc.	196,608	5,537.0	7,235.2	2,834
8	Texas Advanced Computing Center/Univ. of Texas United States	**Stampede** - PowerEdge C8220, Xeon E5-2680 8C 2.700GHz, Infiniband FDR, Intel Xeon Phi SE10P Dell	462,462	5,168.1	8,520.1	4,510
9	Forschungszentrum Juelich (FZJ) Germany	**JUQUEEN** - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect IBM	458,752	5,008.9	5,872.0	2,301
10	DOE/NNSA/LLNL United States	**Vulcan** - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect IBM	393,216	4,293.3	5,033.2	1,972

- Selected other sites: Edison (NERSC), Archer (UK), SuperMUC (Germany), Kurchatov Institute (Russia)

LHC experiments have access to some of the world's largest HPC facilities

Facility	Current Award (CPU hours)
MIRA	50M
Titan	10M
NERSC	2M

Allocations for ATLAS activities at US HPC Facilities

- MIRA is PowerPC architecture - allocation being used for MC generator workload

Top 10 Supercomputers (June 2015)

Challenges

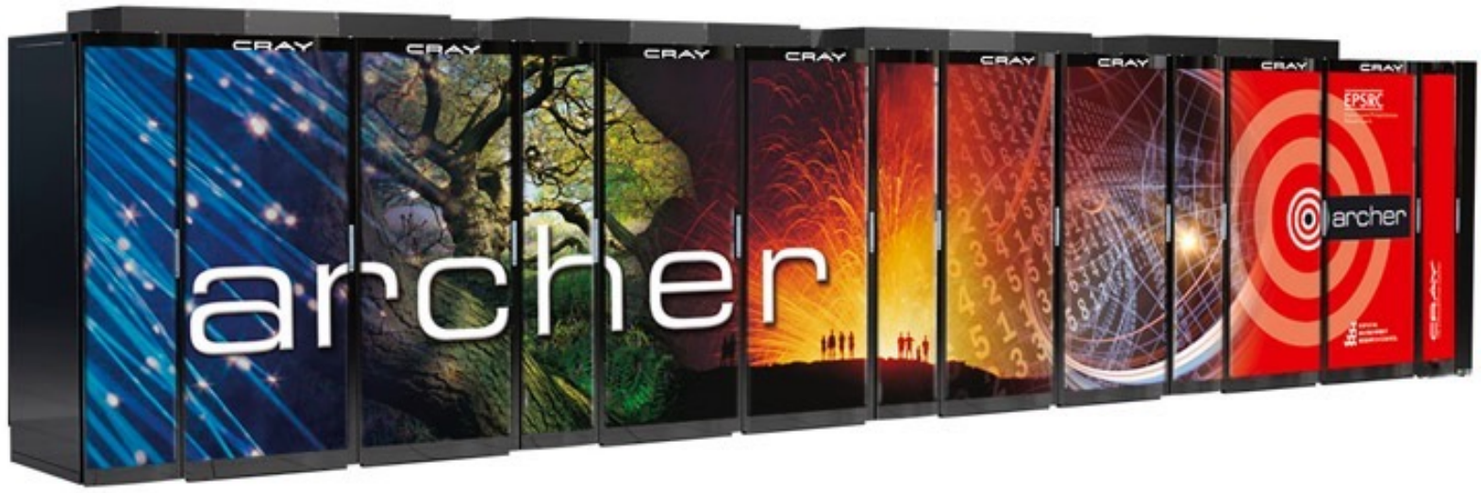
- It is not possible to extend the Grid computing model to HPC facilities
- No unified solution can be applied - need to address different issues for each HPC facility

Challenges common to HPC facilities

- Limited to no external network connectivity from compute nodes
  - No access to Grid storage or CVMFS
- Specialised Operating Systam and software stack
  - More restricted set of packages and tools on (diskless) compute node
- No pool accounts and no persistent grid services allowed at the facility
- Per user limit on the number of submitted and running jobs in the batch queue
  - Large submissions of single node pilot jobs is ruled out
- Job submission model is different - a job executes on a scheduling node and then requests compute resources

Archer

- Archer is the UK's primary academic research supercomputer
- Operational since Nov 2013
- Phase 2 upgrade completed in Nov 2014



Access to Archer possible via a nominal allocation pledged to University of Edinburgh researchers

- Cray XC30 system
- Each compute node comprises of:
- 2 x 12-core 2.7 GHz Ivy Bridge processors
- At least 64 GB of DDR3-1833 MHz main memory
- Cray Aries interconnect (multi-tier all-to-all connectivity)
- 4.4 PB scratch storage (Lustre)
- ~~3008~~ **4920** compute nodes ~~72,192~~ **118,080** cores
- ~~1.56~~ >2 Petaflops of theoretical peak performance.

Running LHC Workloads on Archer

Access

- Use my personal account (SAFE login) to manage environment and job submission

Connectivity

- No external network connectivity from compute nodes
  - Since Phase 2 can use new Cray Realm Specific IP addressing (RSIP) from compute nodes
- Assume WAN access only from login or gateway nodes

Software Delivery

- CVMFS not available from compute nodes
- External filesystem resident on edge server could be mounted on scheduling nodes
- Options: CVMFS rsync from external server, Pacman, Parrot

OS and Package Dependencies

- Cray Linux Environment OS (based on SUSE Linux)
- Some missing packages and libraries have to be incorporated
- Paths to common tools and libraries inherited from default environment have to overridden (e.g. gcc, Python)
  - Archer uses TCL modules to define library and application setup

Job Environment

- Provide grid environment script and worker node "tarball solution" (similar to model for other shared sites)
- HEP specific libraries need to be installed and referenced before job execution

Job Throughput

- Archer queue limitations: maximum **16** queued jobs, **8** running jobs per user
- Virtually no restrictions on number of computing nodes requested

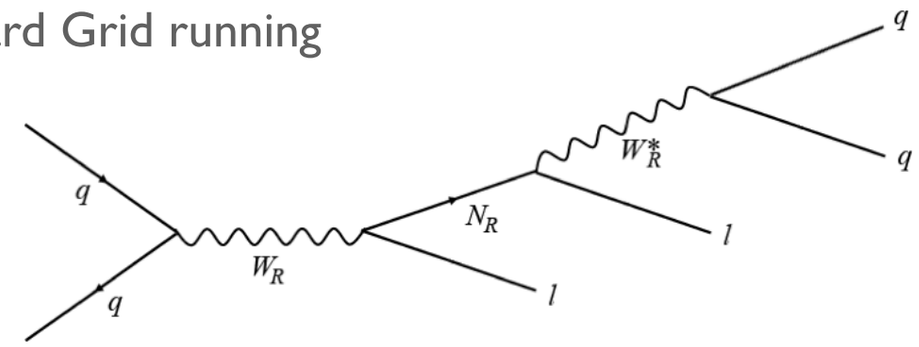
Simulation Demonstrator

Illustrate the steps needed to get ATLAS software running at scale on a HPC facility with a realistic simulation workload

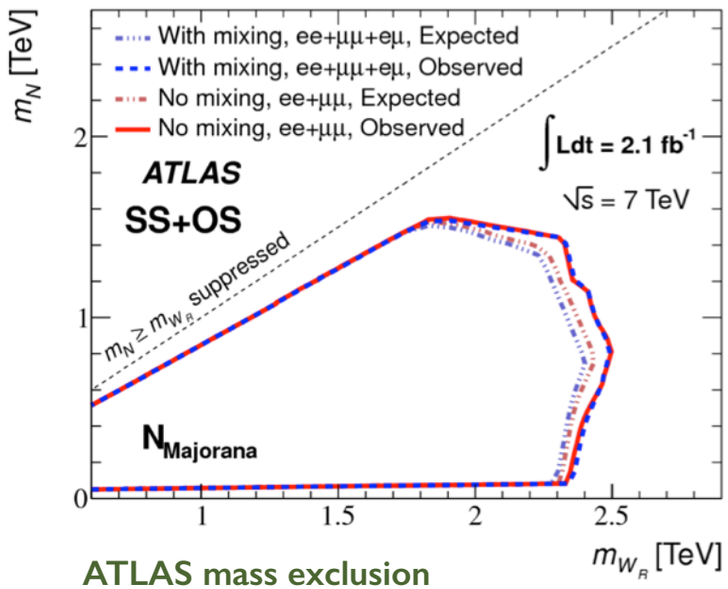
- Generate **1.1M fully** simulated events for an ATLAS exotic search analysis (**20k** events at **56** particle mass combinations)
- **10-15** minutes to generate **one** event - in total over **200k** (unnormalised) CPU hours
- Identify any showstoppers or inefficiencies compared to standard Grid running

Physics Motivation

- Improve searches for right-handed versions of W bosons and neutrinos
- Experimental signature includes two jets or an overlapping as a single "fat jet"
- Full simulation allows new jet substructure analysis not possible with fast simulation
  - Better set of kinematic properties for simulated events
  - More discriminatory power against the background
  - Enables machine learning techniques (Neural Nets, BDTs) to be applied for more rigorous analysis



N_R modelled by MadGraph 5 event generator



ATLAS mass exclusion
Eur.Phys.J. C72 (2012) 2056

Setup Experiences

Release Setup

- Provided software release by rsync CVMFS over ssh
- Software release directory is not relocatable
  - Sanitised directory of any absolute paths - over **67,000** files modified
- Also copy over software releases, gcc libraries, conditions database files

Workflow Construction

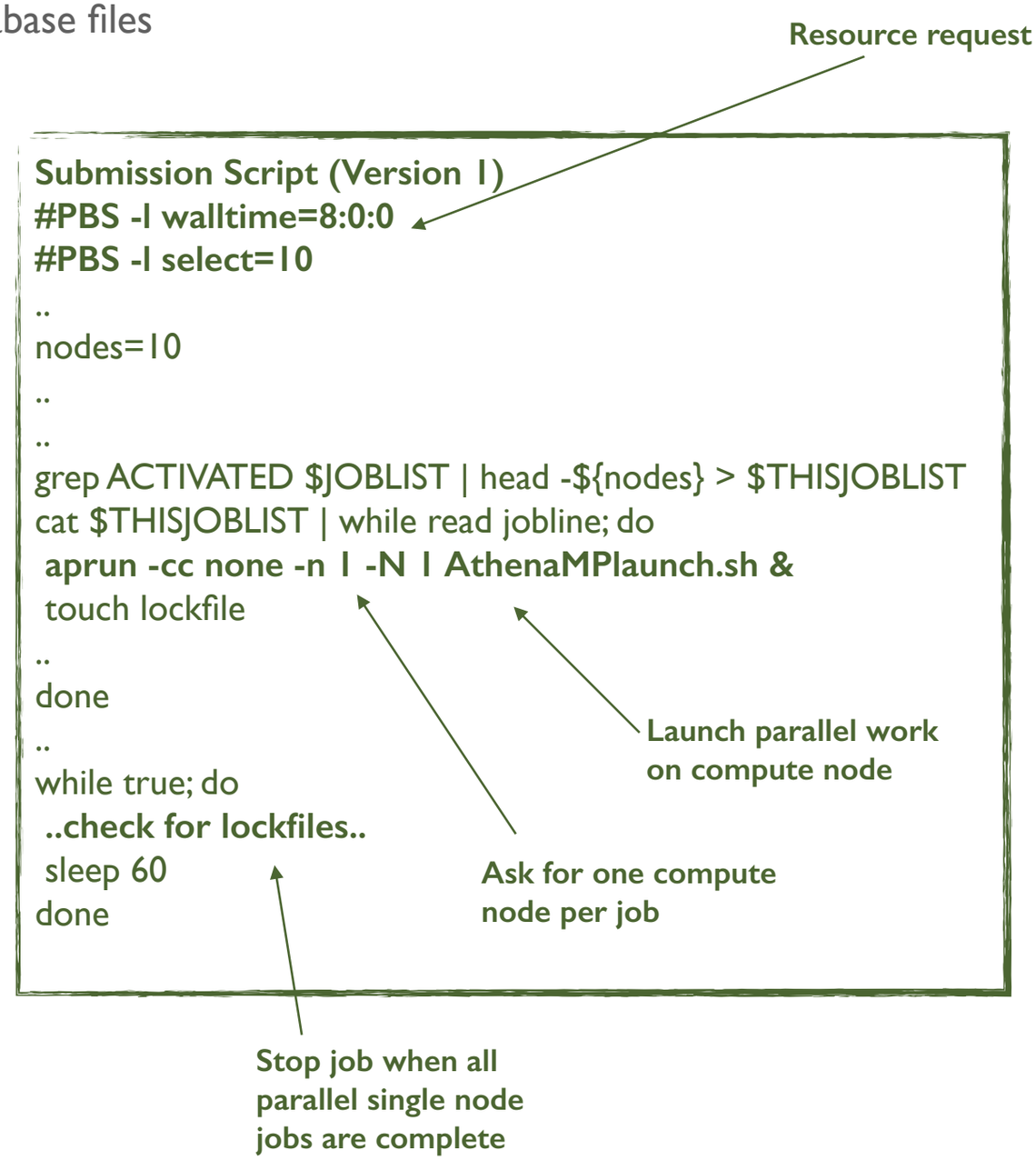
- MC event generation input files downloaded from Grid storage
- Divided into thousands of single node simulation tasks
- Single node tasks grouped into fixed size sets launched in parallel from one batch job
- Events validated and reconstructed elsewhere
  - No technical barriers to running full event processing chain (generation, simulation, reconstruction)

Job Environment and Dependencies

- Referencing correct python libraries was a persistent issue
  - Different environment for staging and compute nodes
  - Software setup should override local configuration but some dependencies hidden in the ATLAS code
- Required *libcrypt* and *openssl* packages

Environment Testing

- Testing enabled through short queue and interactive jobs



Running Experiences

Job Efficiency

- Hyperthreading enabled by default on compute nodes
  - AthenaMP unchecked grabbed **48** worker processes per node
  - Scaled down to 1.5x (**36** cores)
- Environment setup of multiple compute nodes increased total job time considerably
  - Moved this step onto the staging node
- Input file validation of a 1GB input file took 3 hours (!)
  - Fortunately there was a job option was available to skip this check

Scheduling and Job size

- Opted for *short* jobs (**8** hours) with *modest* resource requirements (**10 - 100** compute nodes) to minimise queue time
- Quickly scaled up to running on **290** HPC nodes = **6,960** cores (**10,440** worker threads) running within one hour of job submission
  - No reason why this could not have been much higher
- File I/O slowed job lifetime
  - Job size needs to be tuned
  - Need to identify reasons for slowdown

Submission Script (Version 2)

```
#PBS -l walltime=8:0:0
#PBS -l select=10
..
nodes=10
cores=36
..
module swap anaconda python-compute
export LD_LIBRARY_PATH=..
..Athena setup..
..
grep ACTIVATED $JOBLIST | head -${nodes} > $THISJOBLIST
cat $THISJOBLIST | while read jobline; do
 aprun -cc none -n 1 -N 1 AthenaMPlaunch.sh
 touch lockfile
..
done
..
while true; do
 ..check for lockfiles..
 sleep 60
done
```

Fix number of cores

Change python environment

Move Athena setup outside of parallel job submission

Include dependencies

PanDA Integration

The next step is to fully integrate the facility into ATLAS distributed computing operations

- Two methods being used at other HPC facilities providing resources to ATLAS
- **Both** are currently being evaluated on Archer

Method 1 - HPC Pilot Wrapper

- Developed for use on Titan
- Runs modified PanDA pilot on a HPC login node
- Custom light-weight MPI wrapper scripts to orchestrate single node workloads in parallel across hundreds (if not thousands) of compute nodes
- Communicates with PanDA server using long lasting grid proxy certificate (with a production role)

Prerequisites

- ATLAS software available somewhere on HPC shared filesystem
- Access to a Grid storage endpoint from HPC login node
- Grid software on shared filesystem to allow proxy generation

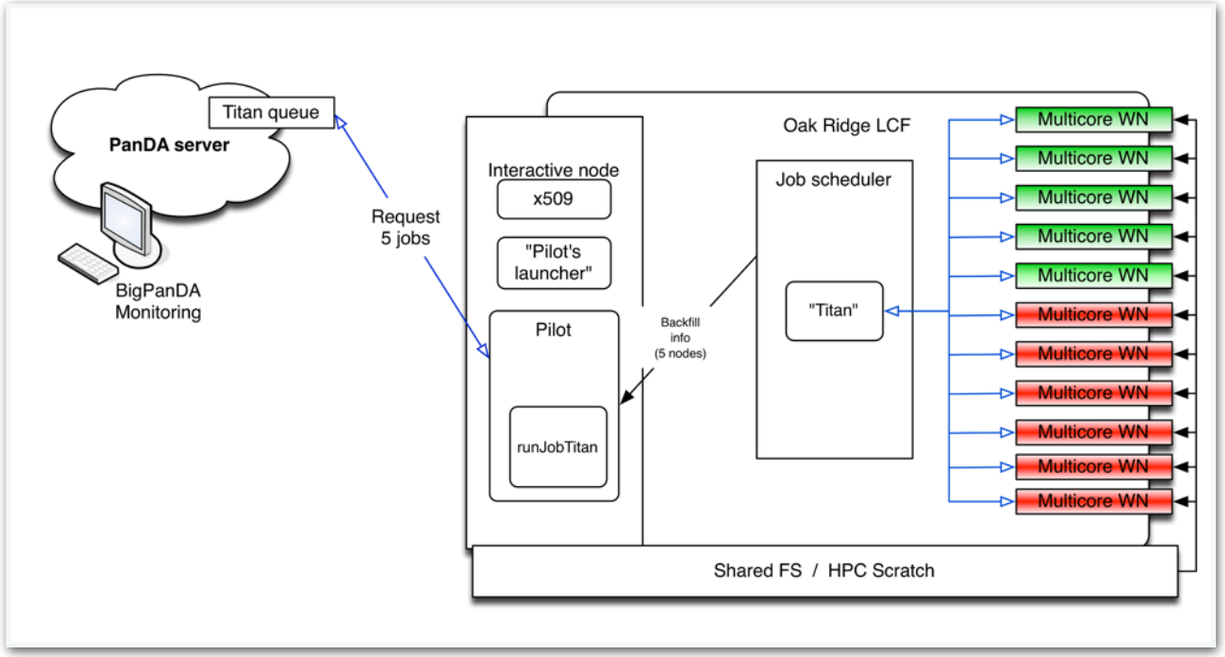
Archer Status

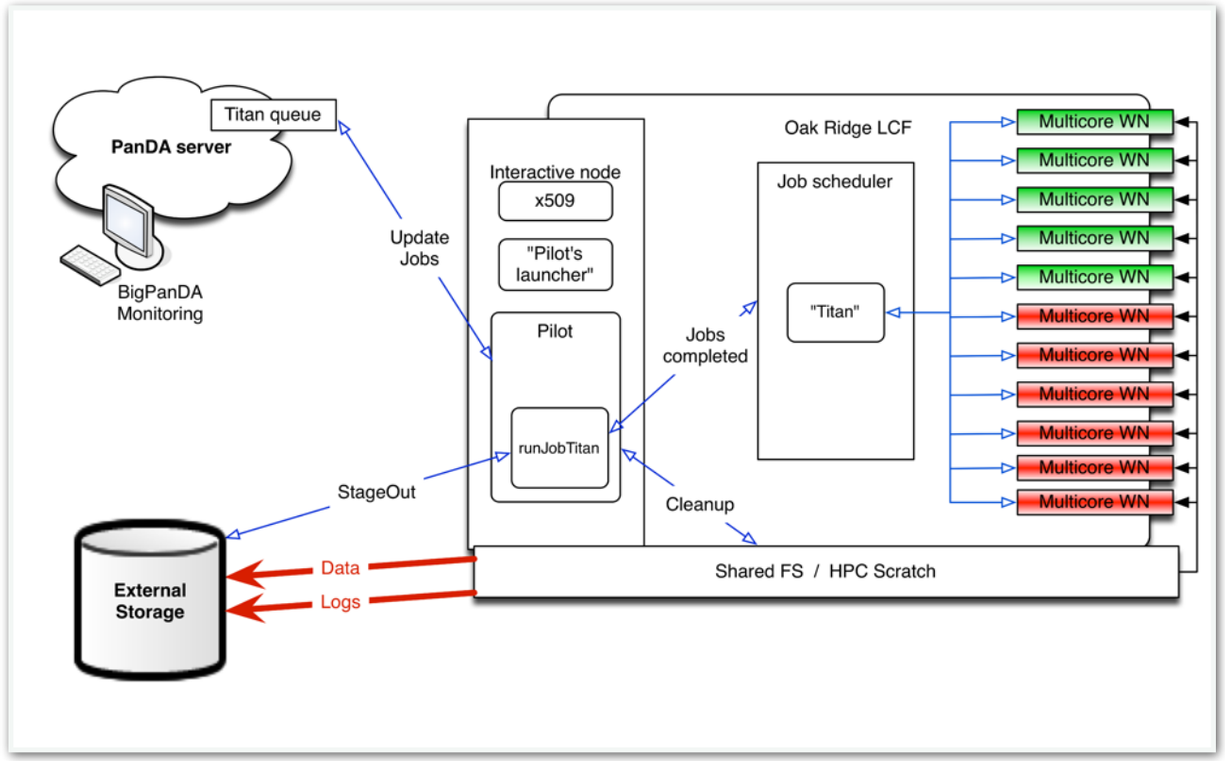
- HPC pilots developed for Titan being modified for Archer
- PanDA HPC queue to be created for test jobs

HPC Pilot Workflow

Pilot Lifecycle

- **Pilot requests information about available resources**
- Requests appropriate number of jobs from JEDI
- Stages in data and prepares job environment before MPI submission
- Rechecks resource availability scaling down job size if needed
- Job submitted (one job per MPI rank) and updates status
- Pilot monitors running jobs
- **After completion, pilot performs data stage-out and cleanup**



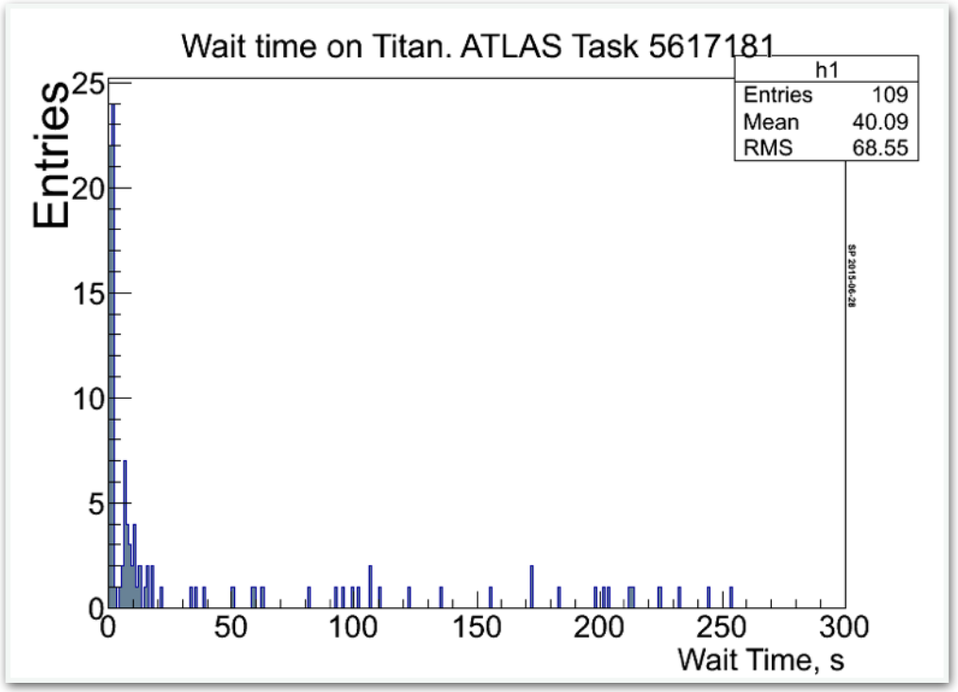


Adaptive Resource Requests

How many resources should a HPC pilot ask for?

HPC Pilot Wrapper

- The HPC pilot is able to determine optimum number of compute nodes to request with minimum execution latency
- On Titan retrieve this value by directly polling backfilling information provided by scheduler (**showbf**)
- For Archer backfilling information is not exposed (PBS pro vs Cray ALPS)
  - Could be derived indirectly from other client tools (**apstat, qstat, xtnodestat**)
  - Or could determine by analytics on queue load and average wait times (work in progress)
  - A static value may be good enough for most purposes



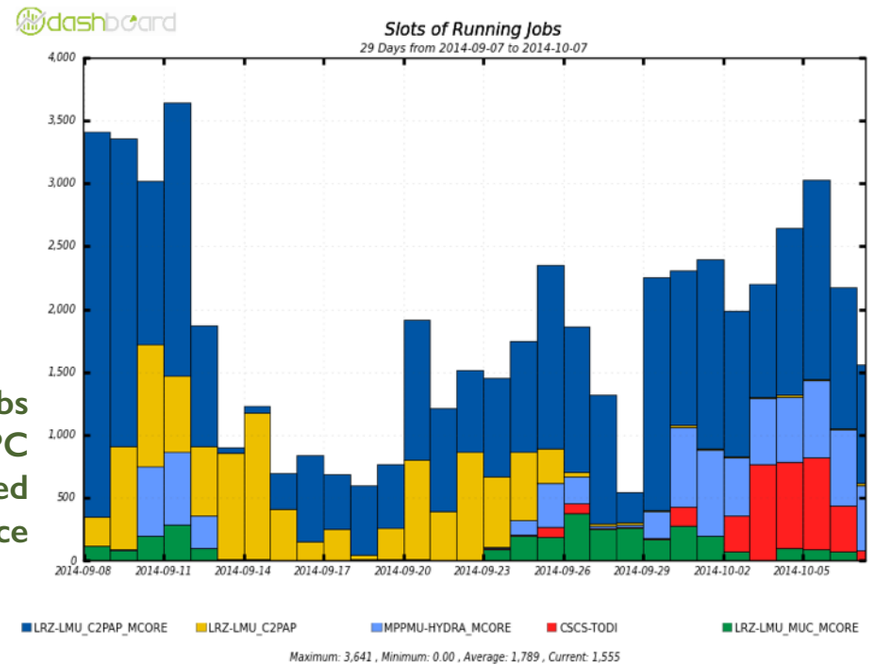
Average wait time on Titan with sample ATLAS task

HPC access using ARC

Method 2 - ARC

- ARC service accepts and forwards PanDA jobs (via ACT) to run on a remote HPC system
- Handles both staging in and out (including ability to cache files)
- Assumed that ARC server is not resident at facility
- Access to the system for job submission and data management is only provided over ssh
- ARC directories shared over sshfs
- Deployed successfully at NorduGrid HPC centres and EU HPC facilites: SuperMUC, C2PAP, Hydra and CSCS (Todi, PizDaint)

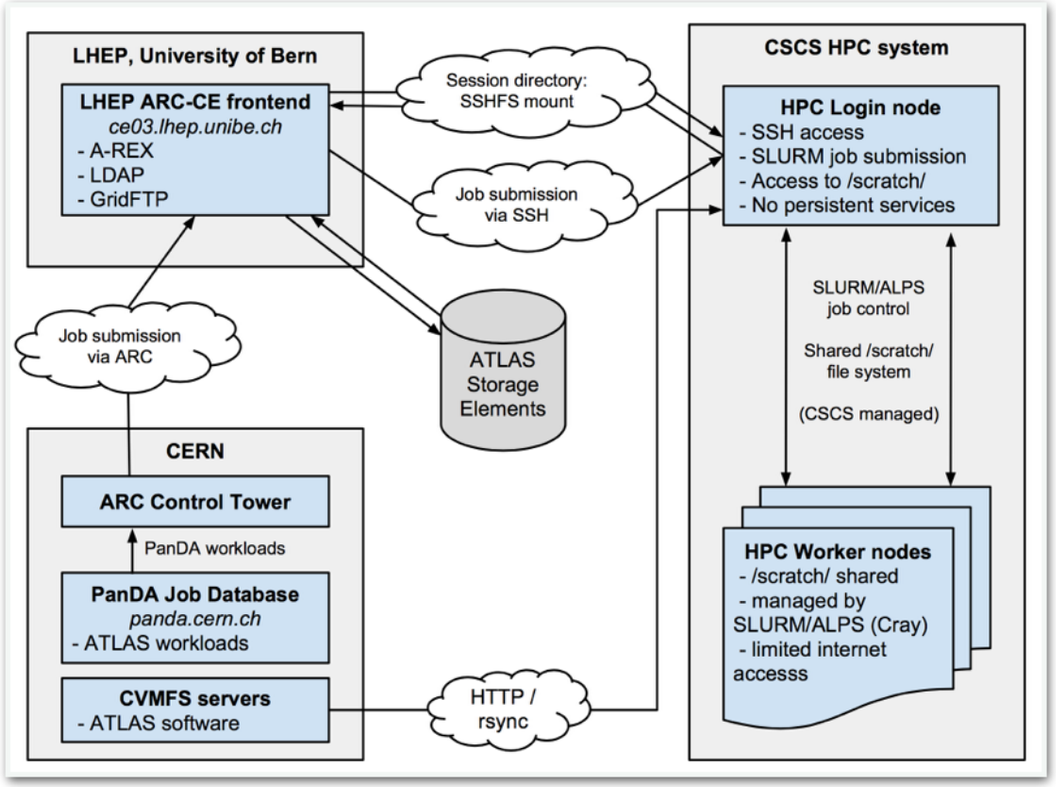
ATLAS jobs running at HPC centres connected by ARC service



Archer Status

- Dedicated ARC CE installed at Tier-2 (ECDF) to connect to Archer login nodes
- ssh passwordless hooks for PBS to be developed
- Issues with persistent passwordless sshfs connection to Archer

HPC model developed for CSCS



Optimisation Steps

Most HPC sites have a set quota allocated usually by wallclock time - strong incentive to minimise job inefficiencies

Data Staging

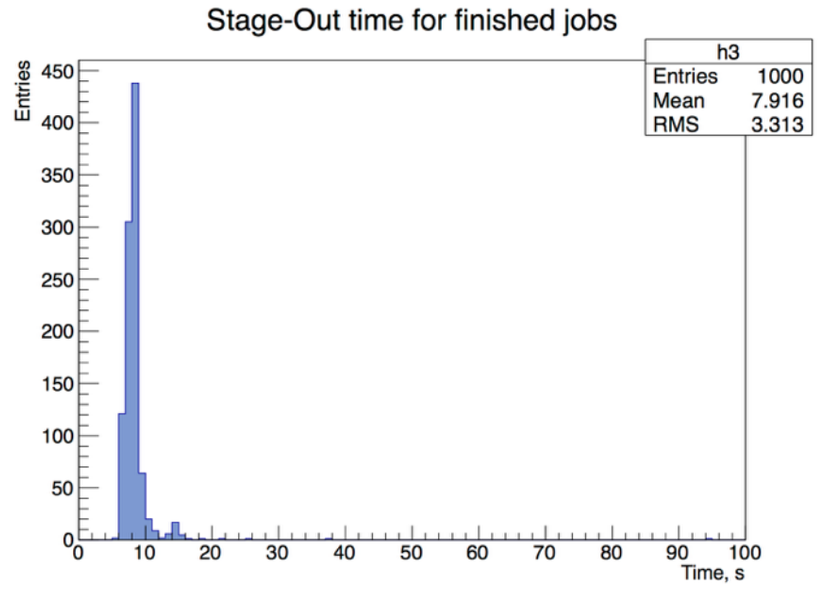
- Files already staged in outside of job execution
- Staging out to grid storage still done within job
  - Asynchronous and parallelised stage-out being considered
  - On Archer could move file transfer to post-processing queue

File Operations

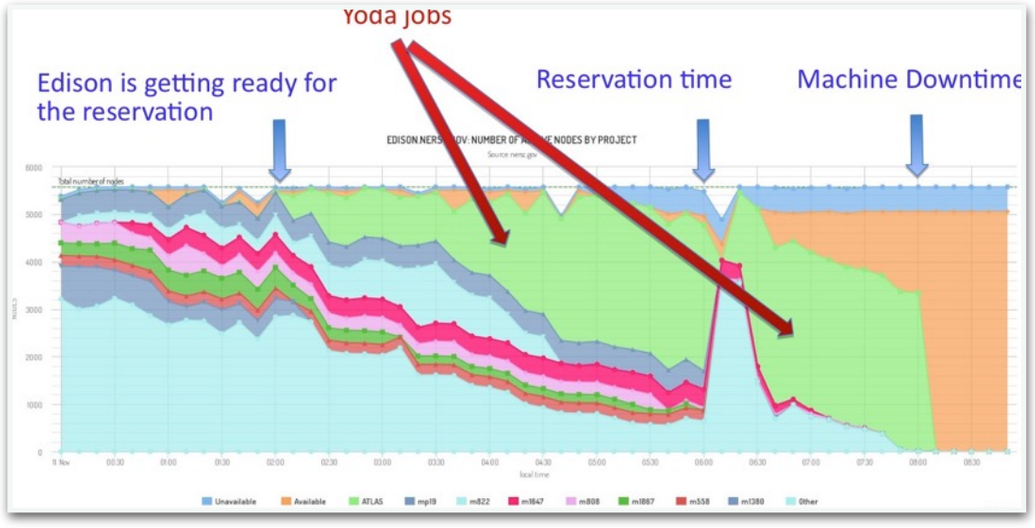
- Constant file IO is costly within a HPC job lifetime
- Operations on small files on high performance file systems (e.g. job environment setup) need to be reduced
  - Need to fully investigate bottlenecks (job profiling with **Darshan**)
- Small self-contained ATLAS software release being developed to reside in RAM on the compute node

Job Size and Preemption

- Event Service model allows job preemption (without checkpointing) with only a small loss of consumed CPU cycles
- Model adapted for HPC (Yoda)
- Demonstrated on Edison with plans for deployment on other HPC sites
- Very useful model to demonstrate the use of opportunistic resources



File Staging out time for jobs on Titan



Opportunistic data simulation on the Edision HPC facility (NERSC from SC14)

Other HPC Effort

Not just ATLAS! Other LHC experiments have been exploring how to best make use of HPC computing resources

CMS

- Using Gordon (SDSC) for production activities
- Allocation on Carver and Edison (NERSC)
- Carver has been working for a while
- **Using docker containers with CMS specific container image**

ALICE

- **Multithreaded** Geant4 simulation
- ALICE simulation workloads can be launched via PanDA on Titan using EC2 PanDA server
- Integration with ALICE production system is in progress

Conclusions

- HPC facilities are a powerful computing resource LHC experiments are starting to exploit
- Computing model rules being relaxed to incorporate HPC resources
  - Cannot be used as a general Grid resource
  - For now ATLAS liaisons need to embedded to resolve issues specific to the facility operations

- Private MC simulation for ATLAS working on Archer
  - Significant resources for dedicated tasks obtained relatively quickly
  - More effort needed in optimising job performance
  - Integrating Archer into ATLAS distributed computing operations

- A HPC event service model - coupled with pilots that provide adaptive resource requests - is a promising solution
- A demonstration of the effective use of opportunistic slots would help bolster the case to request additional resources