Big PanDA on HPC/LCF Update

Sergey Panitkin
BNL

BigPanDA Face to Face meeting. December, 19 2013

Current HPC resources for Big PanDA

- Currently have accounts at:
  - **Oak Ridge Leadership Class Facility (OLCF)**
    - Titan (our own Big PanDA project (CSC108) allocation)
    - Kraken (part of NSF XSEDE infrastructure, through UTK allocation)
  - **National Energy Research Scientific Computing Center (NERSC@LBNL)**
    - Hopper, Carver, Edison (through OSG allocation)
  - **New York Blue at BNL**
    - Blue Gene /P (our own project allocation)
- We concentrate on ORNL development right now. Danila will talk more about Titan based activities
  - Synergy with Geant 4 proposal for Titan (use of GPUs on Titan)
  - Great support and interest from OLCF management in Big PanDA
  - Significant CPU time allocation
- Parallel ports to NERSC machines
  - Similar platform to ORNL - Cray

Current resource allocation at OLCF

PanDA/OLCF meeting in Knoxville. Aug 9

- PanDA deployment at OLCF was discussed and agreed, including AIMS project component
- Cyber-Security issues were discussed both for near and longer term.
- Discussion with OLCF Operations
- OLCF management is very interested in prospects of increased efficiency of machine utilization
- After the meeting PanDA project (CSC108) allocation was increased from 10k to 500k hours on Titan
  - To compare:
    - ATLAS allocation at NERSC (m1092) - 450k hours
    - OSG allocation at NERSC (m670) - 300k hours (**new upgraded 1.1M in 2014. Thanks to M. Ernst**!)

Panda set up on HPC platforms

- Main idea - try to reuse existing PanDA components and workflow logic as much as possible
  - PanDA pilot, APF, cvmfs, etc
- PanDA connection layer runs on front end machines, in user space
- All connections to PanDA server at CERN are initiated from the front end machines
- “Pull” architecture over HTTPS connections to predefined ports on PanDA server
- For local HPC batch interface use SAGA (Simple API for Grid Applications) framework
  - http://saga-project.github.io/saga-python/
  - http://www.ogf.org/documents/GFD.90.pdf

Current status on ORNL : Integration

- Approved allocation on Titan (PI - Sergey Panitkin)
- Sergey Panitkin and Danila Oleynik have access to Titan and Kraken at ORNL
- Queues for Titan and Kraken are defined in PanDA
- Modified ATLAS pilot is running on Titan and Kraken front ends
  - SAGA-Python layer is used in pilot for interaction with local batch queues
  - Job submission chain “prun-PanDA-pilot-SAGA-PBS” tested
- CVMFS installed on Titan as /cvmfs , visible from worker nodes
  - CVMFS via Parrot . FUSE not available
  - “Live” CVMFS on interactive nodes
  - Replica on shared file system for worker nodes
- AutoPilotFactory (APF) is installed and tested on Titan FE (J. Hover)
  - Local HTCondor queue for APF installed

Workloads

- Root is ported to Titan and Hopper @NERSC
- Root from CVMFS runs out of the box on Titan
- Geant 4 port on Titan is available (Alice)
- ATLAS t-tbar analysis code ported to Titan and Hopper
  - ATLAS data (D3PD for ttbar analysis) transferred to Titan and Hopper
  - Proof-Lite mode tested on interactive batch nodes
- Started event generator ports
  - SHERPA (v. 2.0.b2 and v. 1.4.3) was ported to Titan and Hopper
  - MadGraph 5 (v. 1.5.12) was ported to Titan and Hopper
  - Simple examples and tutorials do run.
  - Need expert help for more realistic workload. Alexei discussed this with ATLAS management. Vakho Tsulaia from LBNL was contacted for Athena MP expertise
- **Will have to go through workloads validation steps!**

Limits on aTGC Calculations

- Request from Brian Lindquist (USB) came through ADC to help with his project.
- Limits setting for anomalous triple gauge coupling calculations.
  - CPU intensive
  - Single threaded job takes ~50 hours to calculate one point.
  - Typically 1000 points are needed for one set of parameters.
  - Several sets of parameters are needed for analysis.
  - C++ code
  - Code uses RooFit extension of Root.
  - Can be ran in multi-threaded mode .
- Difficult to run on the Grid. Ideal workload for HPC.

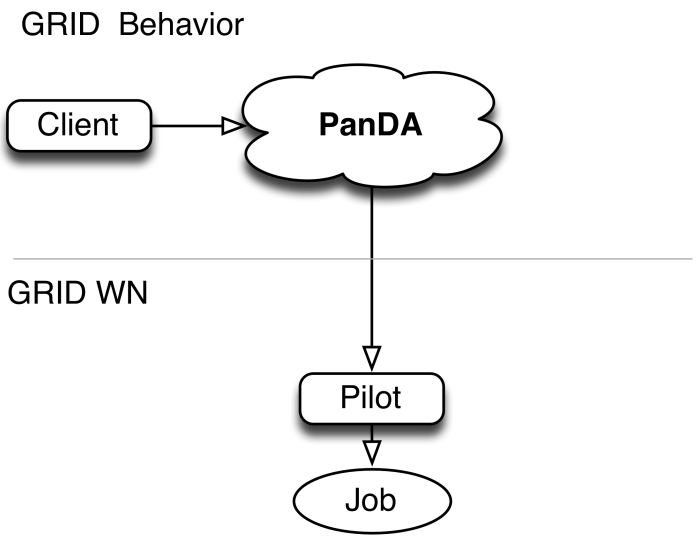
Limits on aTGC Calculations II

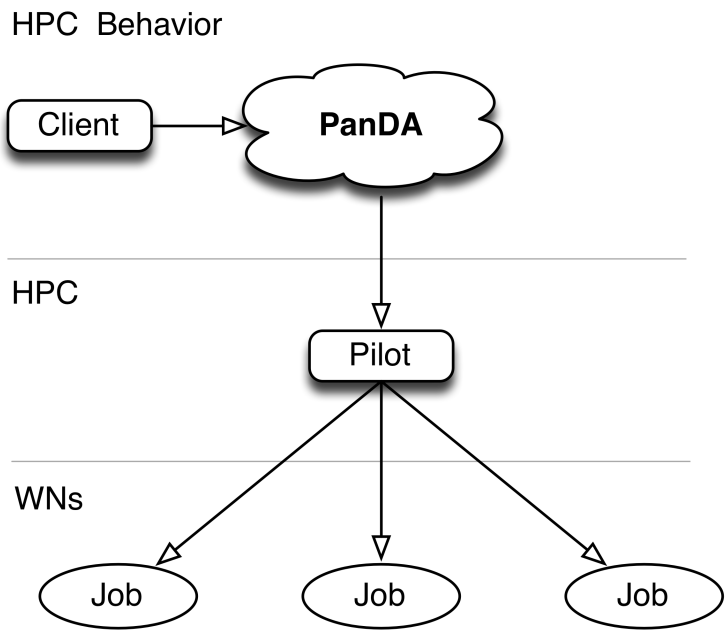
- Started ports on Titan and NERSC machines (Edison, Hopper, Carver)
- Root needed to be recompiled since RooFit was not a part of standard Root install anywhere.
- Low time limits on Titan's small queue shifted debugging to NERSC.
- Code runs on every HPC platform that we tried.
- Some issues with validation on Hopper. Under investigation.
- Converted code to use MPI
  - Message Passing Interface
  - Available on every HPC platform
- Can run parallel, multithreaded, whole node jobs
- Launched large jobs (~4000 cores) on Titan and Edison
  - Encountered huge (?) wait times ~10 days
- Switched to Carver which was less loaded at the time
- Completed first parameter set run of 1000 points, ~50k core hours

MPI benefits

- Use of MPI will allow us to run multiple independent serial jobs as an ensemble, with just one submission at time.
  - Every job knows it’s place in a group and size of the group
- Good for backfill
  - MPI allows to adjust the size of submitted jobs in a natural way.
  - The size of the available "backfillable" gap becomes MPI rank.
- MPI allows to avoid, or at least mitigate, batch queues limits on number of simultaneously submitted tasks.
- MPI allows to set unique random number seeds for simulation jobs in a natural way.
- Using MPI for our purposes does not require major surgery of the existing code base.
- It's quite transparent and I think any serial code can be turned into MPI version suitable for running on HPC.

Pilot on HPC with MPI





"One to One"

"One to Many"

Dealing with Transformations

- On a Grid worker node pilot starts a transformation to pull in and set up user payload
- From pilot's point of view transform is a part of payload.
  - When you submit a job using prun it "wraps/adds" runGen.py transformation script that pilot uses.
  - runGen.py is ~1000 lines of Python code
  - runGen.py needs internet connection (~5 wget), to DDM, to PanDA,,etc
- Problem for HPC application
  - We removed Pilot from worker node space to a place with internet connection
  - Transform still neds to be executed on worker node.
- Can't use standard grid transforms. Need a substitute of some kind.

New transforms for HPC

- Use test tool provided by Tadashi for job submission
- Substitute ATLAS transform with our custom transform script specific to Titan.
  - Sets up Titan specific environment – like appropriate modules
  - Sets up workload specific environment
  - Calls workload
- Right now every workload has it's own transform script
- Workloads are precompiled and installed on Titan
- Transforms are installed on Titan
  - Simple python scripts, potentially just shell scripts

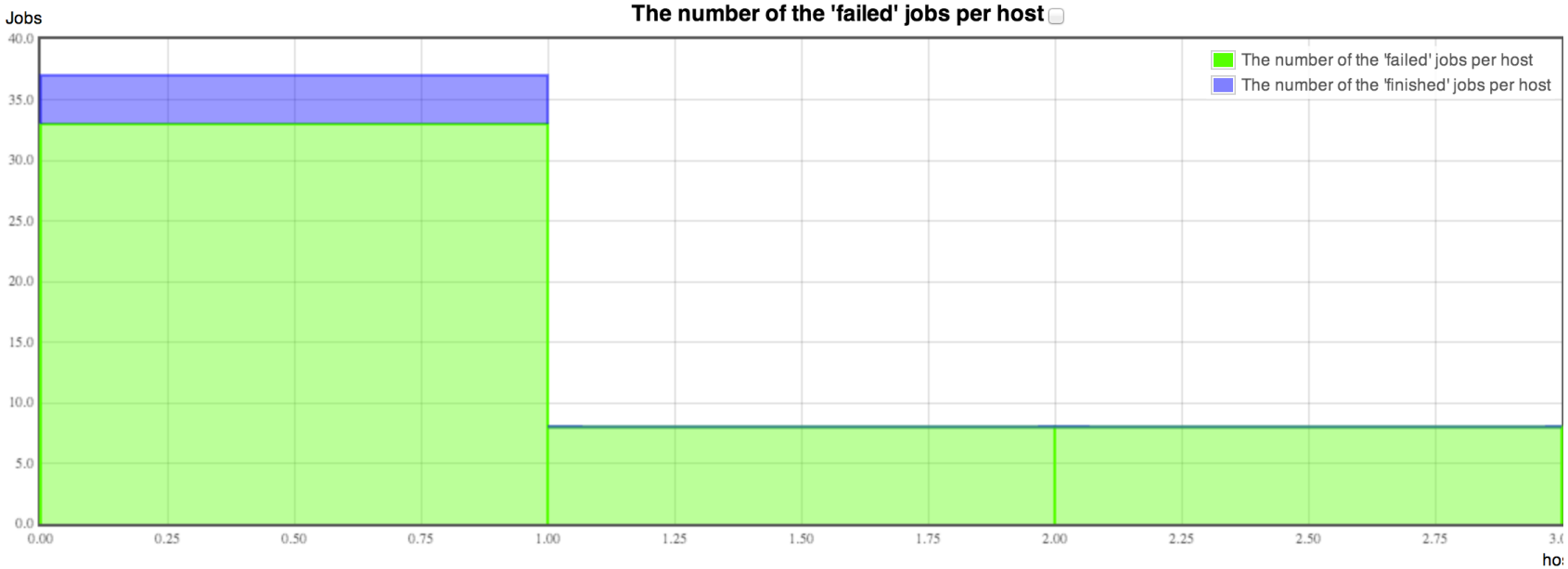
First PanDA jobs on Titan

type=None days= last 0.5

site: ANALY_ORNL_Titar days: 0.5 jobStatus: wn

The server spent **0.446** to access the CERN Oracle server

wnlist({"plot":["failed","finished"],"style":"SB","days":0.5,"jobStatus":"wn","site":"ANALY_ORNL_Titan","hostonly":"yes","jobtype":null}):



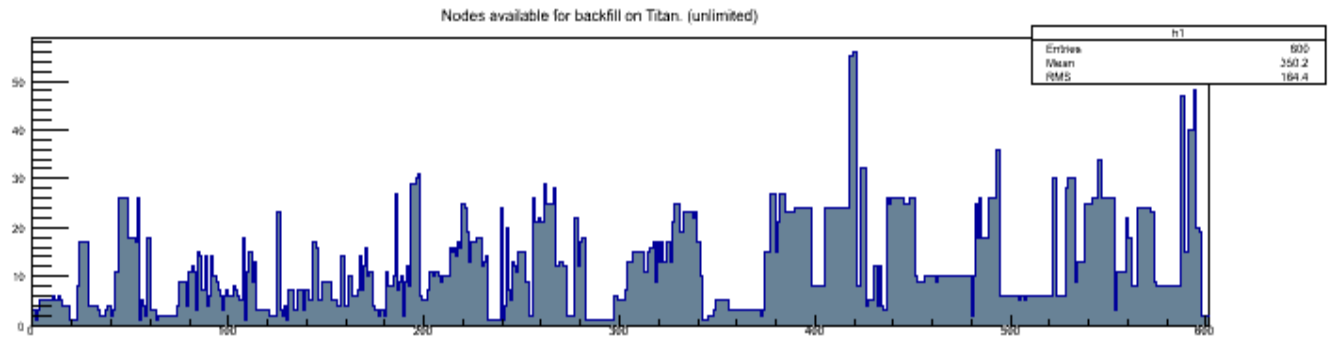
2 hosts	% fail	sent	starting	running	holding	transferring	finished	failed
titan-ext3	89%	-	-	-	-	-	4	
titan-ext4	100%	-	-	-	-	-	-	
ANALY_ORNL_Titan: 2	91%	-	-	-	-	-	4	

HPC Backfill

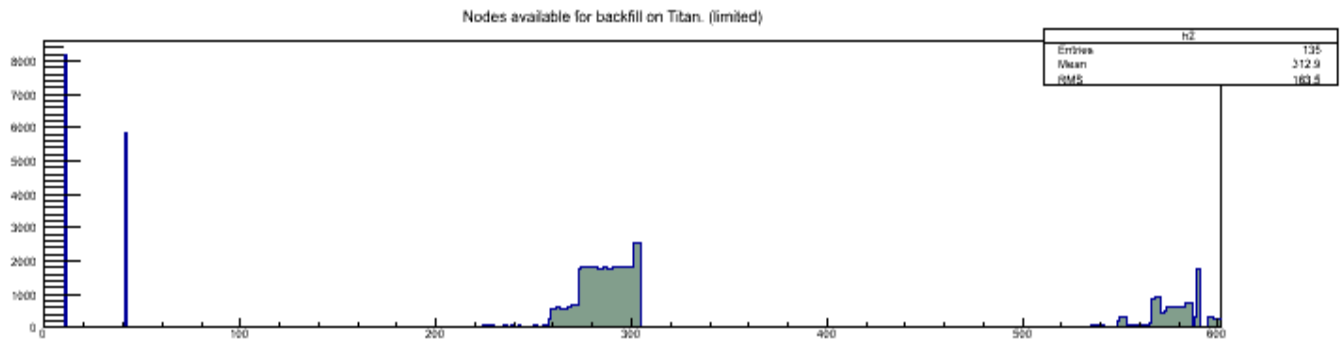
- About 10% of capacity on a typical HPC machine is unused due to mismatches between job sizes and available resources
  - By design leadership class machines are geared toward large scale jobs
  - Time allocation is competitive, review process seems to be biased toward large project
- Estimate for Titan 300M core hours per year unused
- Small jobs have a good chance of using these resources
- PanDA can be a great vehicle for harvesting opportunistic resources on HPC platforms
- One needs to deal with inner MOAB scheduler on CRAYs to get information about small unused chunks of resources

Titan Backfill 1

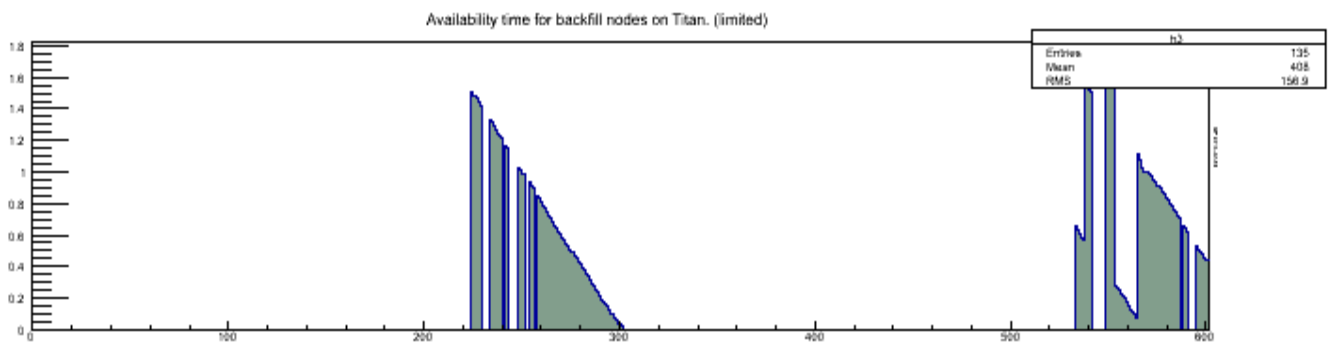
indefinite



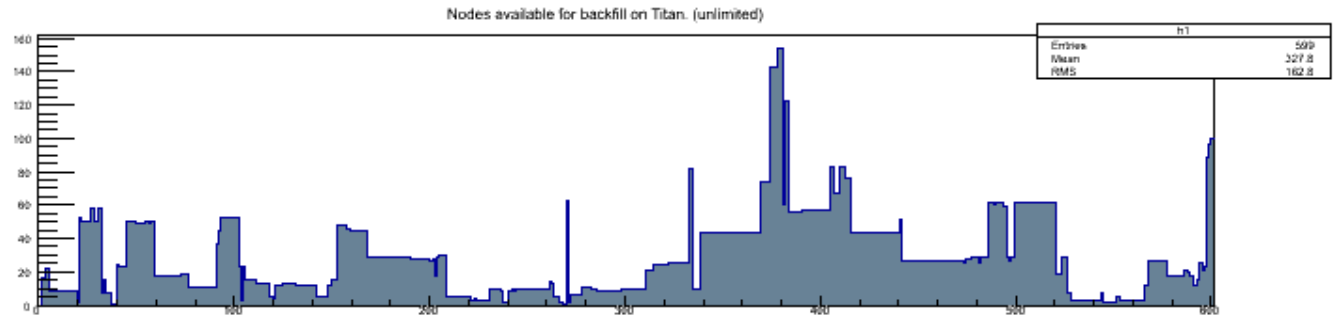
limited

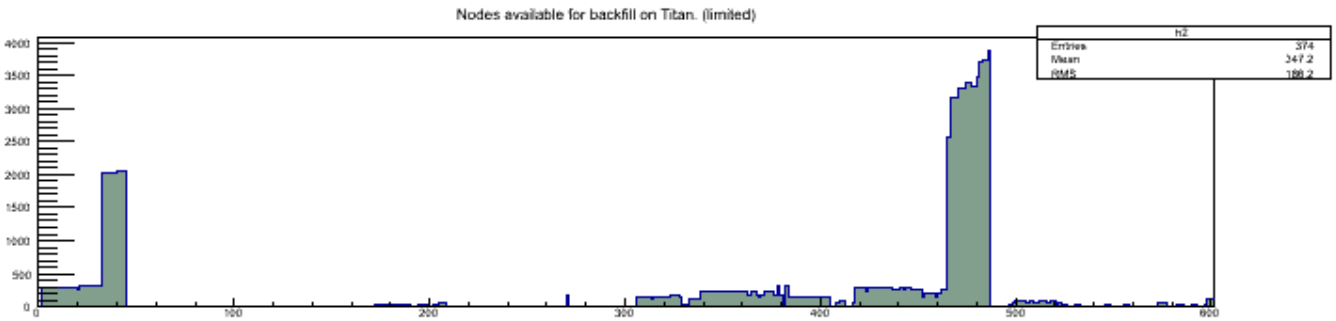


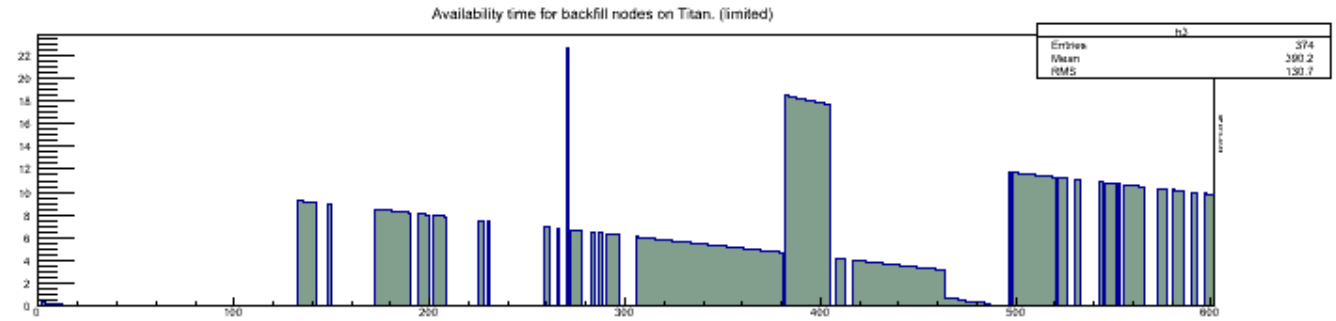
time estimate



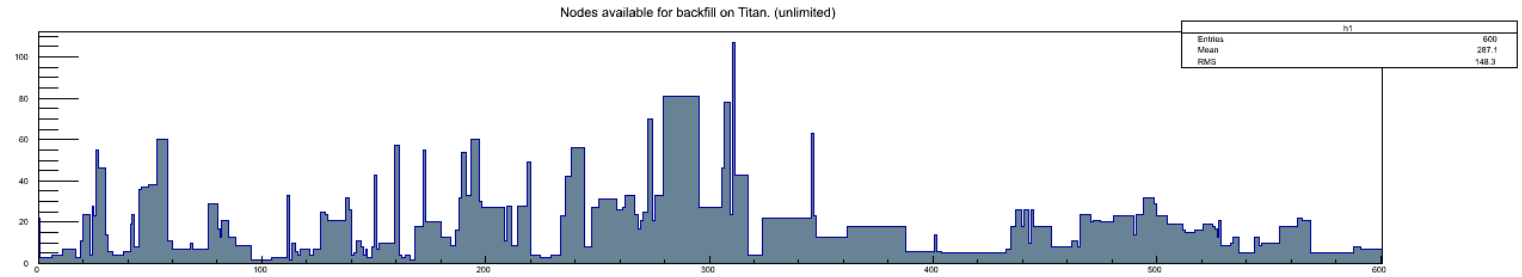
Titan Backfill 2

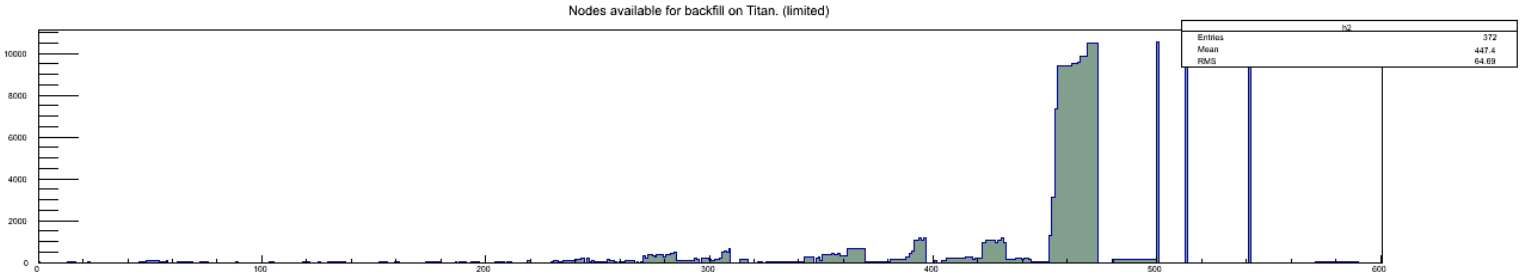


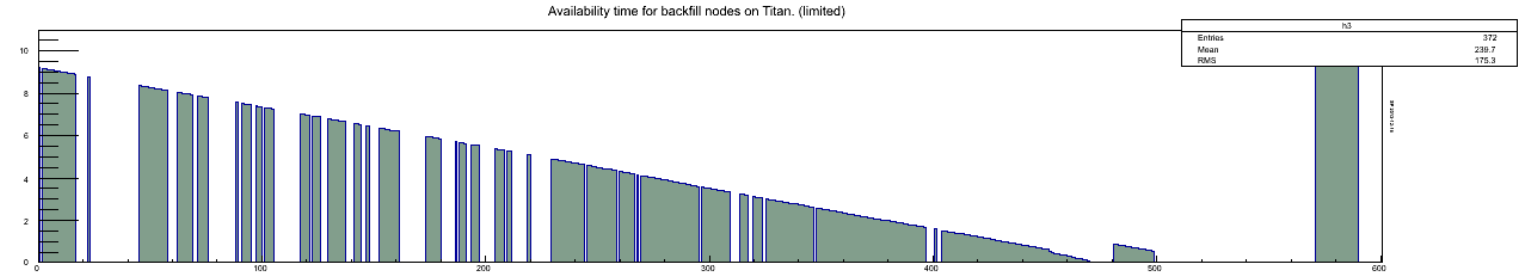




Titan Backfill 3







Summary

- Work on integration of OLCF, NERSC machines and PanDA is in progress
- Key PanDA system components ported to Titan@OLCF
- Job submission chain tested. More details in Danila's talk.
- First successful job submissions from PanDA to Titan with output transferred by pilot from ORNL to BNL T1 scratch
- First successful PanDA job completion on Titan
- Work on specialized HPC transforms started
- Workloads ports are in progress
- Conversion of ATALS code to MPI was attempted
- aTGC limits calculations performed. 50k core hours delivered