HTCondor on Titan

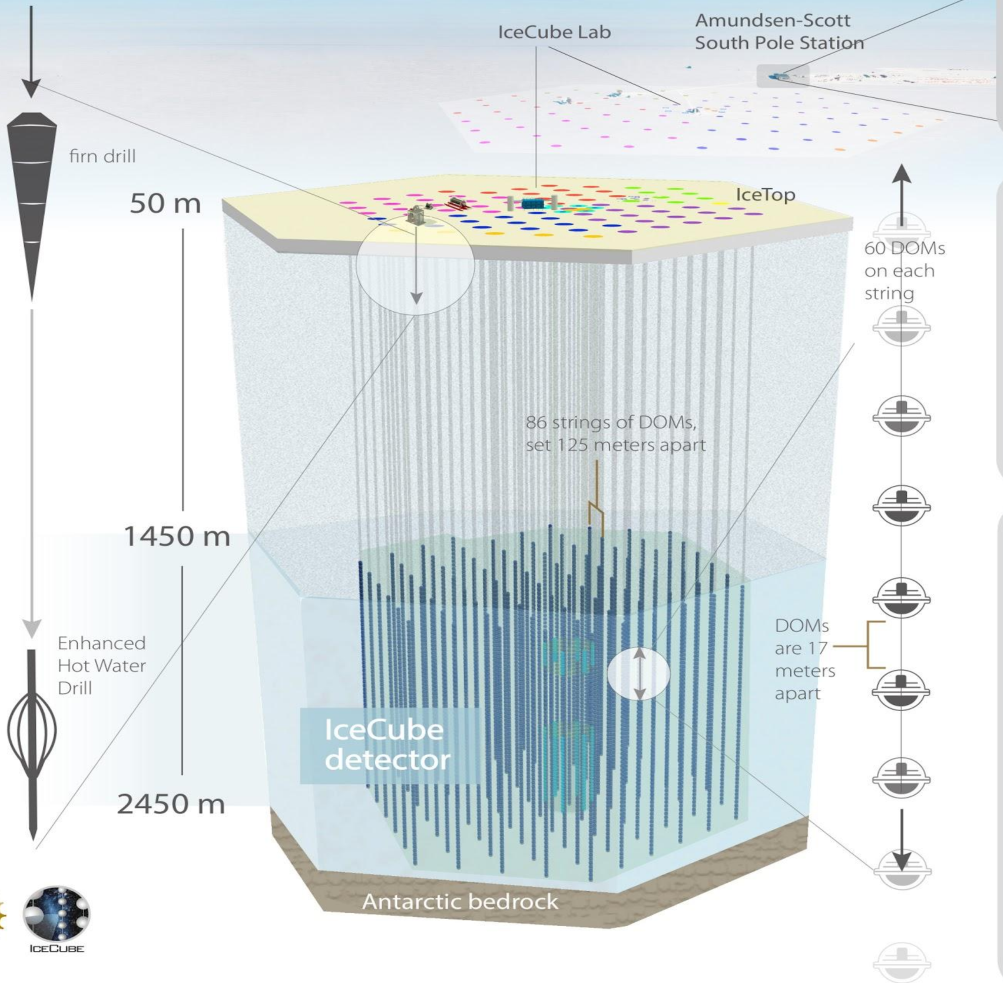
Wisconsin IceCube Particle Astrophysics Center

Vladimir Brik, David Schultz, Gonzalo Merino

CHEP 2018

The IceCube Neutrino Observatory

Design and construction

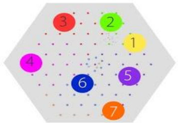


Detector Design

- 1 gigaton of instrumented ice
- 5,160 light sensors, or digital optical modules (DOMs), digitize and time-stamp signals
- 1 square kilometer surface array, IceTop, with 324 DOMs
- 00:00:02⁻⁹ 2 nanosecond time resolution
- IceCube Lab (ICL) houses data processing and storage and sends 100 GB of data north by satellite daily

Detector Construction

7 seasons of construction, 2004-2011



- 28,000 person-days to complete construction, or 77 years of continuous work
- 4.7 million pounds of cargo shipped, 1.2 million of which was the drill
- 48 hours to drill and 11 hours to deploy sensors per hole
- 4.7 megawatts of drill thermal power with 200 gallons of water per minute delivered at 88 °C and1,000 psi

Overview of Titan

- Cray XK7 Supercomputer at Oak Ridge National Lab
- Ranked #7 by TOP500 as of June 2018 (#1 when built)
- 18688 physical compute nodes
    - nVidia Kepler K20X GPU
    - 16-core AMD Opteron CPU
    - 32GB RAM
- PBS, Moab, ALPS for cluster management and operation

Challenges of using Titan for our workloads

- Connectivity restrictions
  - Worker nodes have no Internet access
  - Two factor authentication using a key fob
- Exotic ecosystem
  - Cray Linux on worker nodes
  - Titan’s Lustre file system not a good fit to hold a copy of our CVMFS repo

Challenges of using Titan for our workloads

- Titan is geared heavily toward large MPI applications
  - Scheduling and other policies are adverse to jobs that are not “leadership class” (single-job multi-node)
  - Native mechanisms alone are inadequate for dynamic node-level task scheduling
- HTCondor to the rescue!

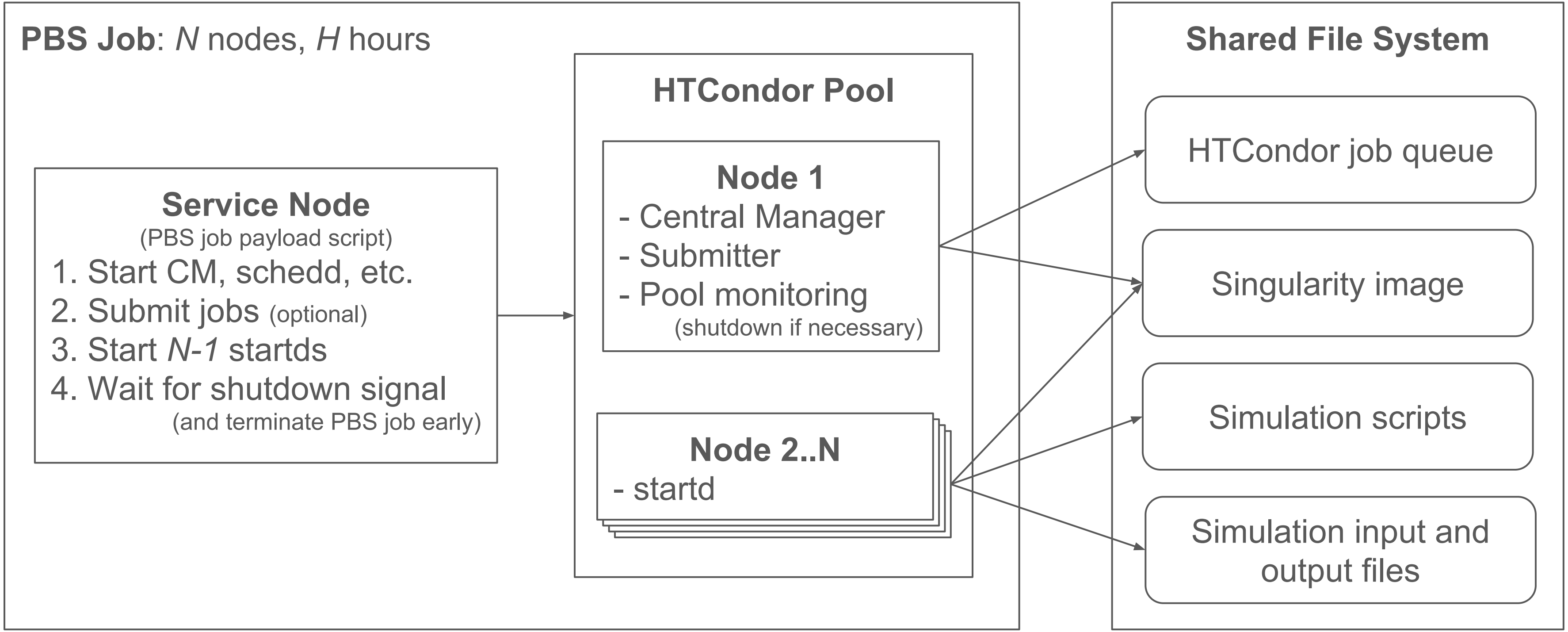
Our approach at a high level

- Transfer simulation input and output files manually
  - Just ran globus-url-copy --sync a few times during the campaign
- Package IceCube's software stack in a singularity container
  - SL6 container with Titan-specific tweaks
  - A 40GB subset /cvmfs/icecube.opensciencegrid.org
  - HTCondor

Our approach at a high level

- Use HTCondor as the second-level scheduler inside PBS
    - Start an HTCondor pool inside a PBS job, one container per worker node
    - Store/load HTCondor scheduler state from the shared file system to make pools “resumable”

High-level architecture



PBS Job: *N* nodes, *H* hours

Service Node
(PBS job payload script)
1. Start CM, schedd, etc.
2. Submit jobs (optional)
3. Start *N-1* startds
4. Wait for shutdown signal (and terminate PBS job early)

HTCondor Pool

Node 1
- Central Manager
- Submitter
- Pool monitoring (shutdown if necessary)

Node 2..N
- startd

Shared File System

HTCondor job queue

Singularity image

Simulation scripts

Simulation input and output files

Results

- Expended 16.5K node-hours of our allocation to process 84K simulations
  - nVidia K20X ~5x slower than GTX 1080 for our workload
- This test showed that it is feasible for us to do useful work on Titan
  - Could be a pretty significant GPU resource for us, which is what we need the most
  - Good preparation for similar environments, like Summit

Results

Main downside: only practical for self-contained projects

- Integrating Titan’s resources into IceCube’s systems would be challenging
    - Networking and authentication restrictions
    - Various policy restrictions (e.g. no cron, low ulimits)
    - HTCondor’s upcoming file-based job submission feature looks promising for some cases

Thoughts

- Persistent central manager would simplify things a lot
  - Already possible to do, but seems to go against the spirit of Titan’s User Guide

- Native CVMFS support would be great
  - IceCube’s full CVMFS repo is 600GB and containerizing it would be a pain

Thank you

Backup Slides

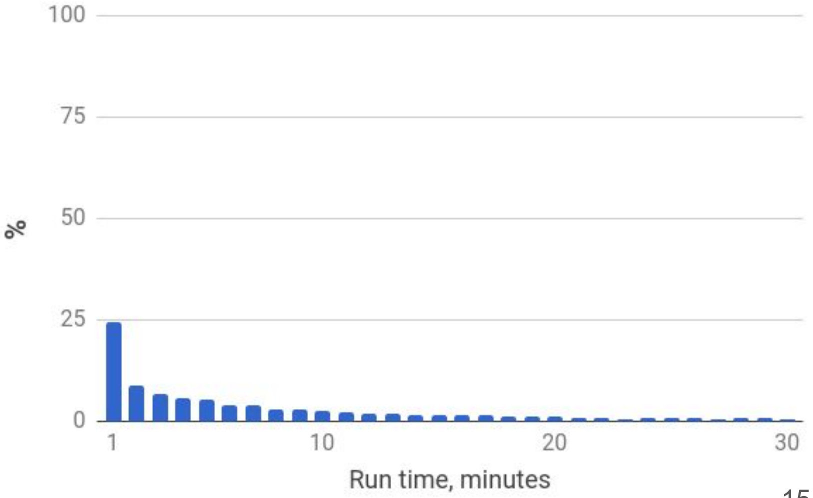
Why we need HTCondor

PBS scheduling policy on Titan			
Min Nodes	**Max Nodes**	**Max Walltime**	**Aging Boost**
11,250	-	24 hours	15 days
3,750	11,249	24 hours	5 days
313	3,749	12 hours	0 days
126	312	6 hours	0 days
1	125	2 hours	0 days

- Only 2 jobs that request less than 126 nodes can run simultaneously
- Job service node restricted to 200 processes, 1024 open files
- Task management tools unfriendly for HTC workloads like ours

Selected simulation project

- 84,000 simulations of photons propagating through the detector
- Simulations are independent and each requires a single GPU
- Run times indeterminate *a priori*
- Inconvenient run time distribution
  - Range: 0 to 90 minutes
  - Median: 5 minutes
  - 90th percentile: 30 minutes



Status of Singularity on Titan

Singularity has been disabled on Titan since late April/early May.

I am guessing it’s because the Cray microkernel used on Titan does not support the prctl option PR_SET_NO_NEW_PRIVS, which is required for secure operation.

According to Titan support, bringing Singularity back is *“a high priority”*, and *“good progress is being made on a solution”*, but no ETA.

https://www.sylabs.io/2018/05/whatsnew-singularity-2-5-why-affects-everyone-using-containers/

Resources

- Instructions for building containers compatible with Titan’s GPUs
  - https://github.com/olcf/container-recipes

- IceCube’s Titan scripts. Use with care :)
  - https://github.com/WIPACrepo/titan-singularity
  - https://github.com/WIPACrepo/titan-htcondor