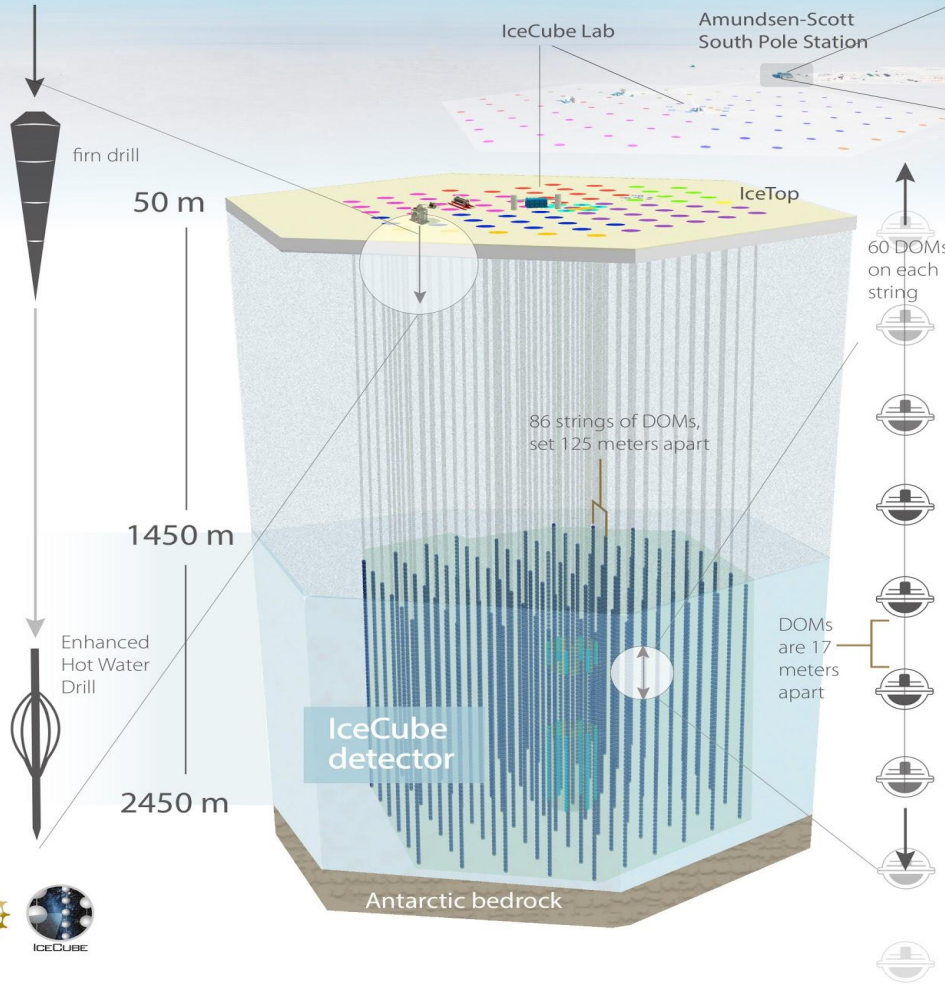
Consolidating Computational Resources at IceCube

David Schultz, Gonzalo Merino, Vladimir Brik
Wisconsin IceCube Particle Astrophysics Center

HTCondor Week 2017

The IceCube Neutrino Observatory

Design and construction

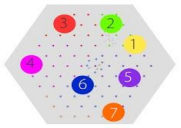


Detector Design

- 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
- 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

IceCube's Computing Ecosystem

1. IceCube's HTCondor cluster at UW-Madison
2. Flocking access to OSG, and various UW HTCondor pools
3. Clusters owned by institutions that are part of IceCube collaboration
4. XSEDE allocations on supercomputers (Comet, Bridges, XStream)

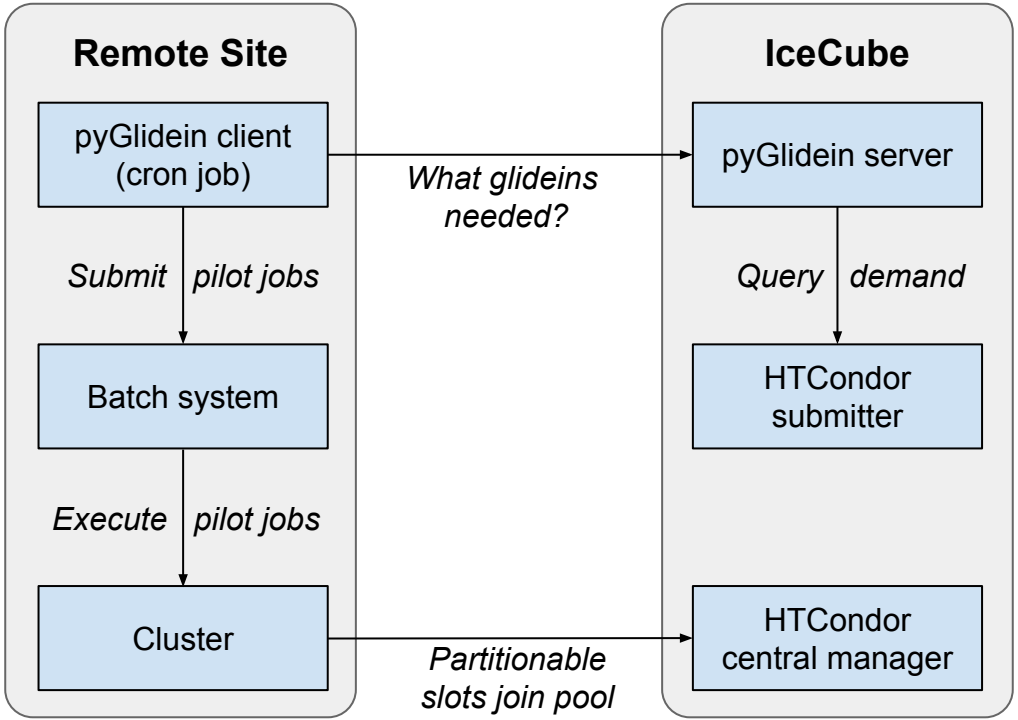
Goal: consolidate available resources into a unified system based on HTCondor.

This presentation focuses on IceCube software that uses HTCondor glideins to integrate collaborator and supercomputer sites into a single HTCondor pool.

IceCube's pyGlidein software

- https://github.com/WIPACrepo/pyglidein
  - Uses glideins to make many sites' resources accessible via a single HTCondor pool
  - In production since 2015, making difficult-to-access resources easy to use
  - People like it: simple, easy to set up by a non-expert, low maintenance
  - "Client" is a cron job that submits pilots to the local cluster
  - Can submit to HTCondor, PBS, SLURM, UGE, LSF
  - For technical details see David Schultz's Condor Week 2016 presentation: http://research.cs.wisc.edu/htcondor/HTCondorWeek2016/presentations/ThuSchultz_IceCubeGlideins.pdf

IceCube's pyGlidein software



IceCube’s pyGlidein software

- Originally geared for collaborator sites
  - Simplicity makes getting people on-board and adding new sites easier
  - Support for site admins provided via a Slack channel
  - Each site locally customizable
- Combines diverse resources into a single HTCondor pool
  - Centralized management of priorities, accounting, monitoring
  - Up to 6000 CPUs, 600 GPUs seen in pool so far
  - Pool is password-protected; ran into some scalability issues

Use case: CHTC resource allocation

- Before glideins: IceCube flocked to CHTC
  - Couldn’t prioritize important workflows or UW users from our side
- Now: all CHTC resources are glideins
  - Can implement resource allocation policies on per-site basis using RANK expressions
  - IceCube submitters tag jobs based on owner’s LDAP groups

```
include command : /usr/local/libexec/condor/set_affiliation.sh
SUBMIT_EXPRS = $(SUBMIT_EXPRS) Affiliation
```

  - At start-up glideins probabilistically pick a RANK to satisfy resource allocation policy

```
if [ "$rand" -lt 45 ]; then export _condor_RANK=...
elif [ "$rand" -lt 70 ]; then export _condor_RANK=...
...
```

Work-in-progress

- Collect glideins’ HTCondor logs
- Auto-update pyGlidein software
- Reconfigure HTCondor of running glideins