IceProd - A dataset management system for IceCube: Update

David Schultz, Juan Carlos Díaz Vélez

WIPAC, UW-Madison
What is IceProd

Data provenance
- Configuration - which software, what versions, when/where it ran, …

Dataset submission
- Monitor job status, resource usage
- Retry failed jobs - resubmit with different requirements
What is IceProd

Central Server
- MySQL
- Master

Batch Submitter
- SQLite
- Client
  - Queueing

HTCondor

Workers
- Job Pilot
  - Job

Website
- Sync job info
- Status updates, logs
- Job assigned to pilot

Job assigned to pilot
- Status updates, logs

Sync job info
- Website
Successes in the last 1.5 years

Switch from IceProd v1 to v2 in late 2016

Pilot job infrastructure
- Run multiple tasks sequentially and in parallel
  -- Reduces startup overhead, connection costs with server
- Resource monitoring in real-time
  -- cpu, gpu, memory, disk usage, time
Scaling bottlenecks

Scaling bottleneck at ~4k nodes
- Database is not responsive enough
  -- Queuing tasks is a complex operation

Scaling bottleneck with many datasets processing
- Split / distributed database partially at fault
  -- Design choice from years ago, no longer relevant
  -- We only run one central instance now
Near-term plans

IceProd 2.4 release next week

- Fix the scaling bottlenecks
  -- Unified, more performant database
- Simple queryable REST API
- Multi-user + authentication
  -- Allow non-production users to submit datasets
Near-term plans

Central Server
- Website
- REST API

Mongo DB

Batch Submitter
- Client
- Queueing
- HTCondor

Workers
- Job Pilot
- Job

Status updates, logs
Get waiting jobs
Job assigned to pilot
Future directions

- Distributed storage support
  -- Intermediary file storage at more than one location

- Supercomputer support
  -- Need a less connected way to still submit and monitor jobs, handle site firewalls

- Better monitoring

- Finding new bottlenecks
Summary

- Switched to IceProd 2 at end of 2016
- Scaling bottlenecks identified and being addressed
- Opening up to analyzers soon
  -- Tracking and catalog of private simulations, lower level analyses
- Plenty of future work to improve
Backup
Details on database issues

Currently have a split database
- MySQL master, SQLite clients
- These need to be synced, which can cause problems
  -- Updates can get lost if they timeout, or if race conditions occur
- SQLite client isn’t as performant
  -- Probably asking too much of it, as it’s a limited clone of the MySQL db

Client connection issues caused by SQLite slowness
- Connection floods asking for a new task -> some succeed, most fail and retry again
- Causes 100% cpu usage, delay other calls
- Can “lose” task completions / resets
IceProd Pilots

We run a pilot inside the HTCondor job:

- Aggregate communications with the IceProd server
  -- IceProd pilots are whole-node jobs: one communication link per node
- Resource monitoring in real-time
  -- cpu, gpu, memory, disk usage, time
- Future: Asynchronous file transfer
  -- stage in/out files for next/prev jobs while jobs execute
- Future: Dynamically resizable “jobs”
Dynamically resizable slots

Dynamically resizable slots give more resources to a long-running job to try and complete it before the end of the glidein lifetime.