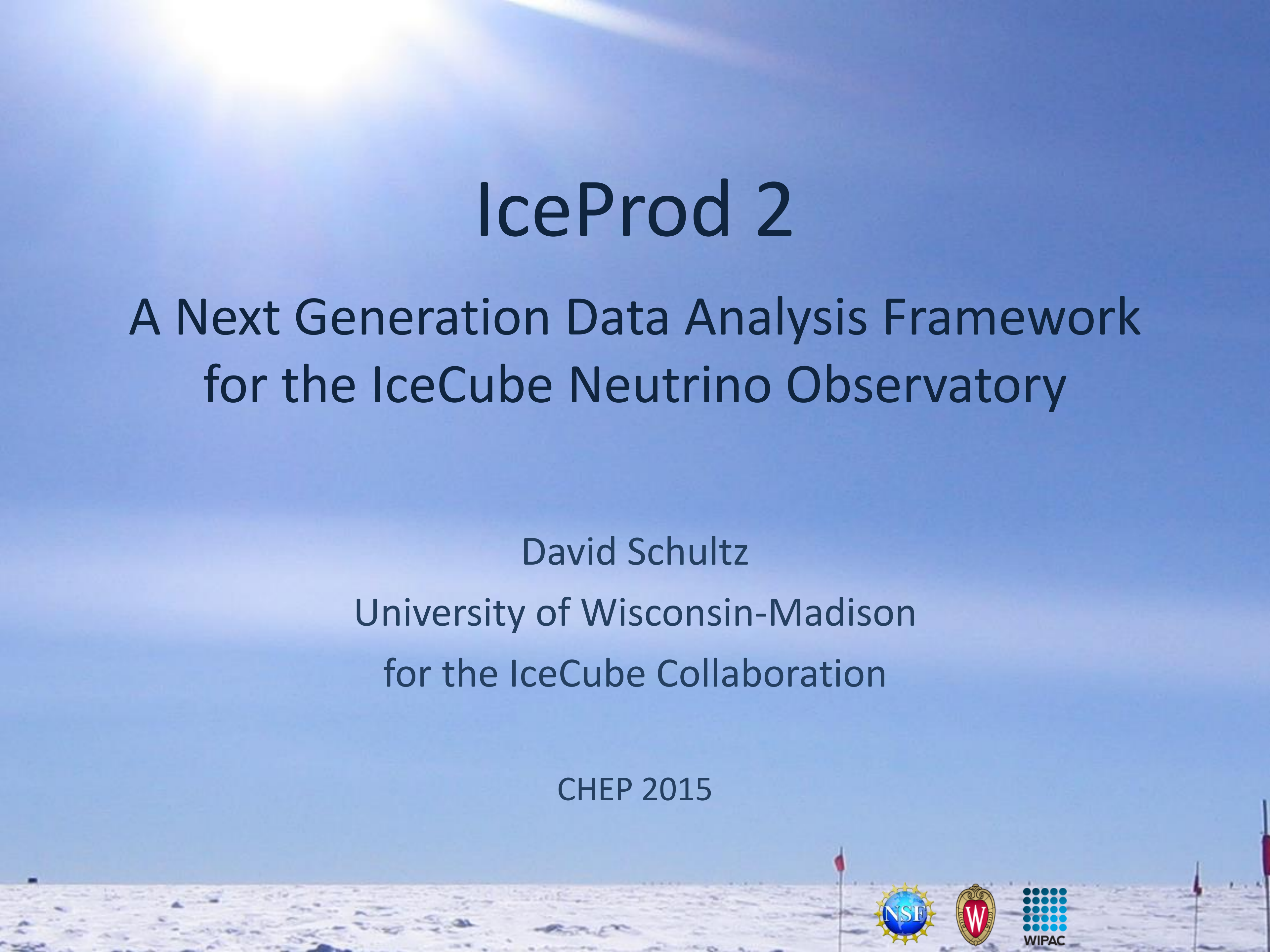


IceProd 2

A Next Generation Data Analysis Framework for the IceCube Neutrino Observatory

David Schultz

University of Wisconsin-Madison

for the IceCube Collaboration

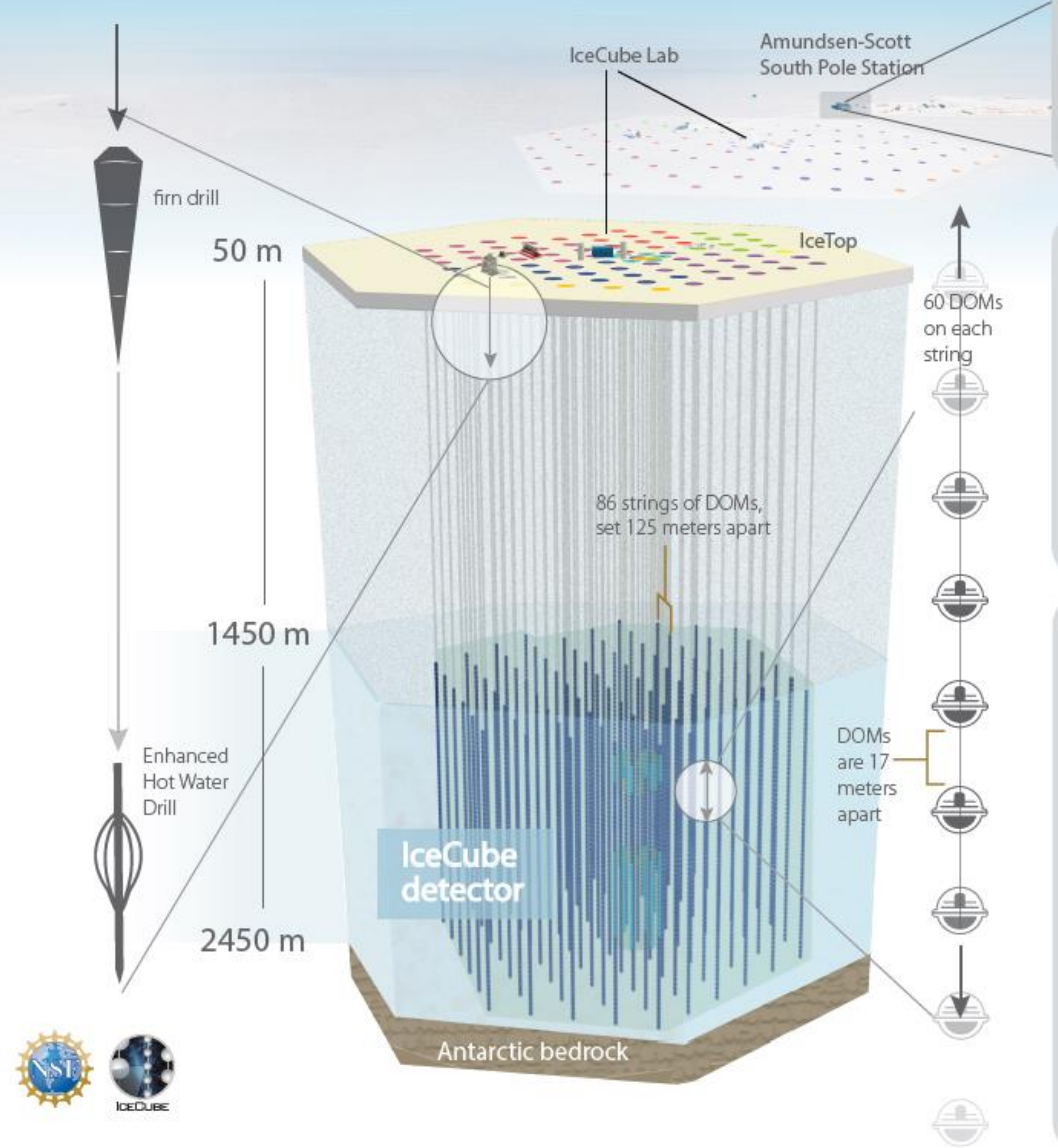
CHEP 2015

Outline

- IceCube and Computing
  - Requirements for a production system
- IceProd's Evolution
  - The long way from v1 to v2
- Status and Future Plans
  - Where do we go from here?

The IceCube Neutrino Observatory

Design and construction

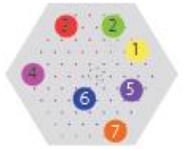


Detector Design

- 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

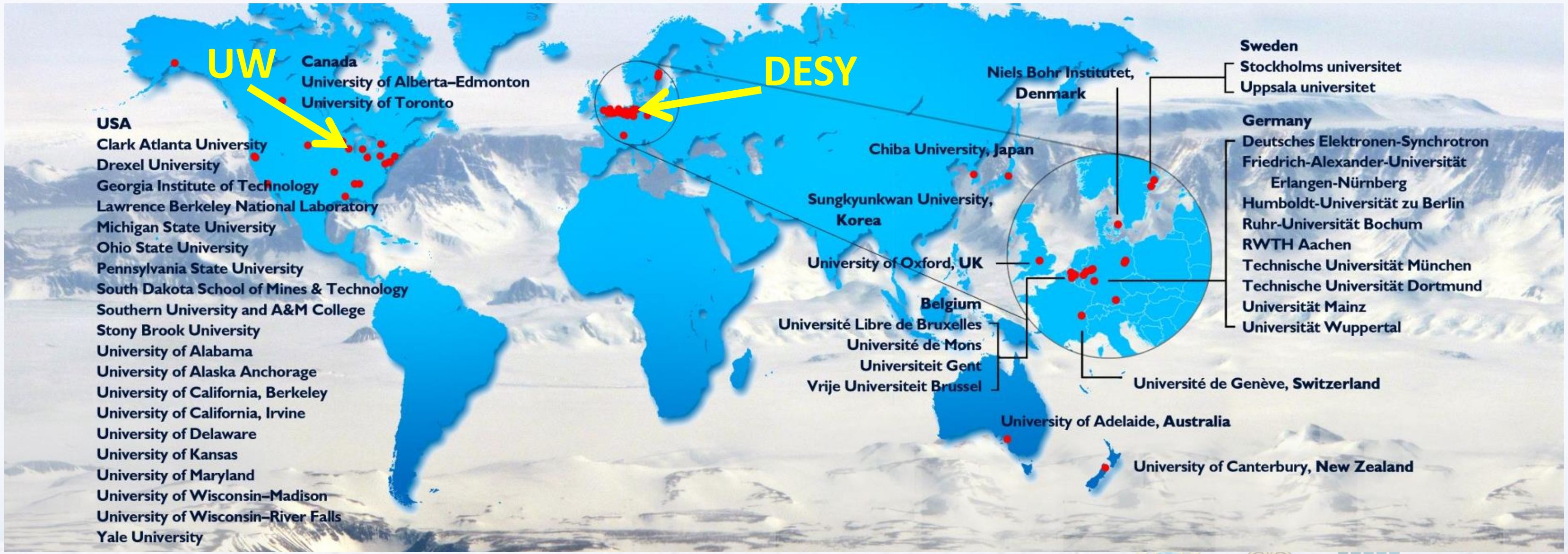
7 seasons of construction, 2004-2011



- 28,000 person-days to complete construction, or 77 years of continuous work
- 2.1 million kilograms of cargo was shipped, 0.5 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 760 liters of water per minute delivered at 88 °C and 7,600 kilopascals

IceCube Computing

- Medium size collaboration
  - 2 data centers and several smaller clusters
  - Most CPU compute is opportunistic
  - We can’t mandate that package X be installed


UW
DESY
Canada
University of Alberta–Edmonton
University of Toronto
USA
Clark Atlanta University
Drexel University
Georgia Institute of Technology
Lawrence Berkeley National Laboratory
Michigan State University
Ohio State University
Pennsylvania State University
South Dakota School of Mines & Technology
Southern University and A&M College
Stony Brook University
University of Alabama
University of Alaska Anchorage
University of California, Berkeley
University of California, Irvine
University of Delaware
University of Kansas
University of Maryland
University of Wisconsin–Madison
University of Wisconsin–River Falls
Yale University
Niels Bohr Institutet, Denmark
Chiba University, Japan
Sungkyunkwan University, Korea
University of Oxford, UK
Belgium
Université Libre de Bruxelles
Université de Mons
Universiteit Gent
Vrije Universiteit Brussel
Sweden
Stockholms universitet
Uppsala universitet
Germany
Deutsches Elektronen-Synchrotron
Friedrich-Alexander-Universität Erlangen-Nürnberg
Humboldt-Universität zu Berlin
Ruhr-Universität Bochum
RWTH Aachen
Technische Universität München
Technische Universität Dortmund
Universität Mainz
Universität Wuppertal
Université de Genève, Switzerland
University of Adelaide, Australia
University of Canterbury, New Zealand

IceCube Computing

- Diverse set of resources
  - Heavily invested in GPU accelerators
  - Some job types need:
    - 6+ GB of memory
    - 100 GB of disk
    - Long walltime

Some Requirements

1. Anything that requires root permission at a site is unlikely to work
   - We're not big enough to enforce policy, we only get an account on the submit node
2. Need more than just glideins
   - Many smaller sites don't have the setup for this
   - We do use it where we can

Some Requirements

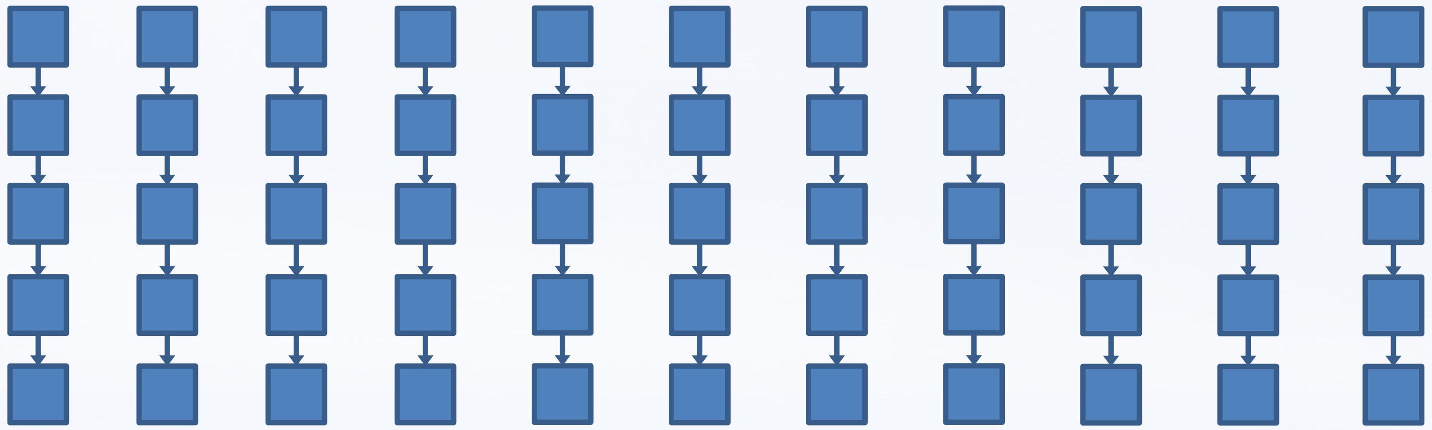
3. Must support multiple OS types, schedulers
   - SL, Ubuntu, SUSE; HTCondor, PBS, SGE, SLURM, …
4. Site-local disk mostly isn't available
5. Need to record all details about a job, forever
   - What is the configuration, where did it run, resource usage, efficiency, …

Other Goals

- A tool for single analysis as well as production
  - Individual users often submit 1M+ jobs for an analysis now
- Production debugging
  - Get the same environment as production, but offline

A Note about Workflow

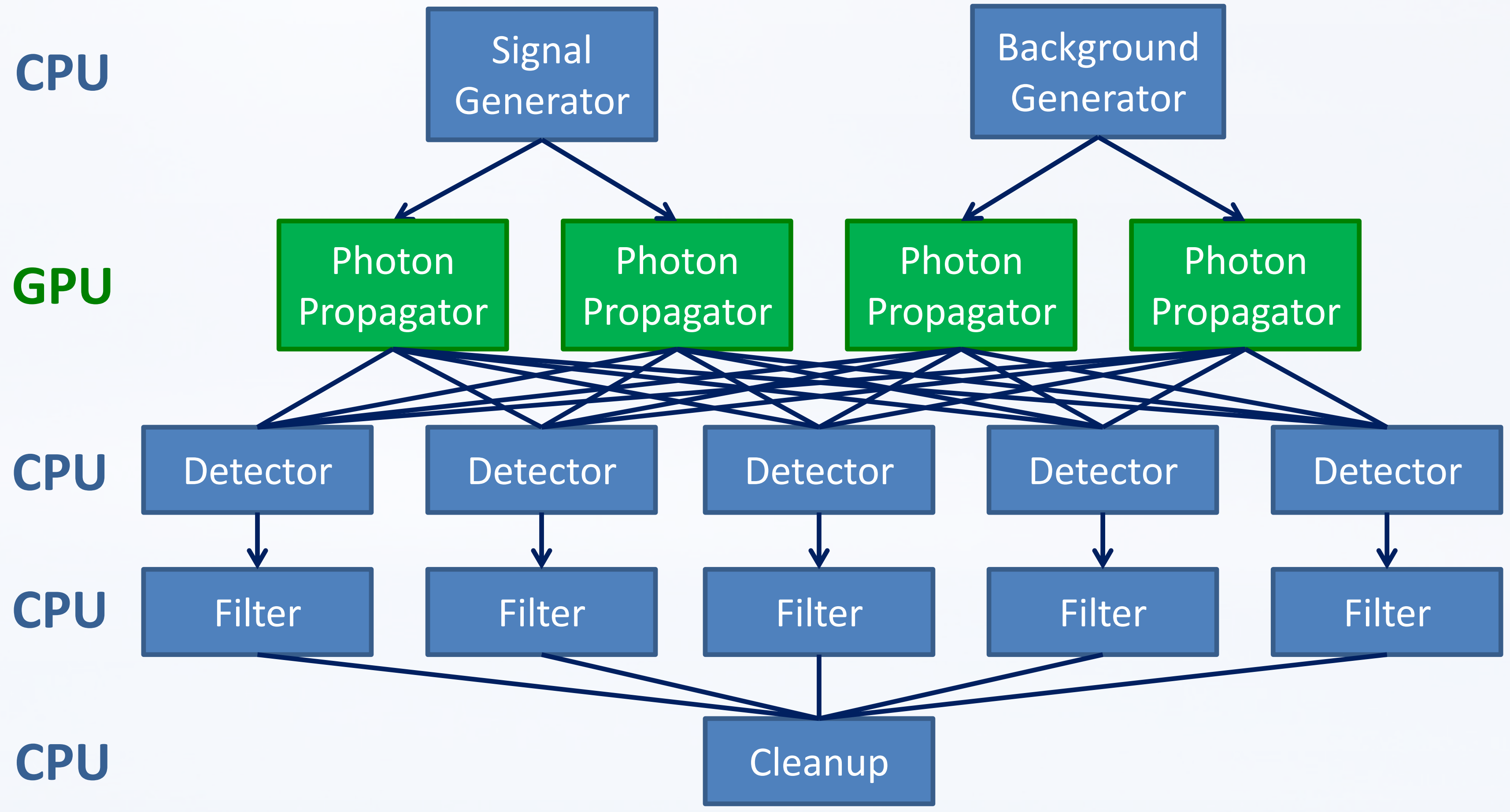
- Job sets are typically 1k - 100k parallel streams
  - Each stream has multiple jobs



- Typical to have 10s to 100s of job sets running at the same time

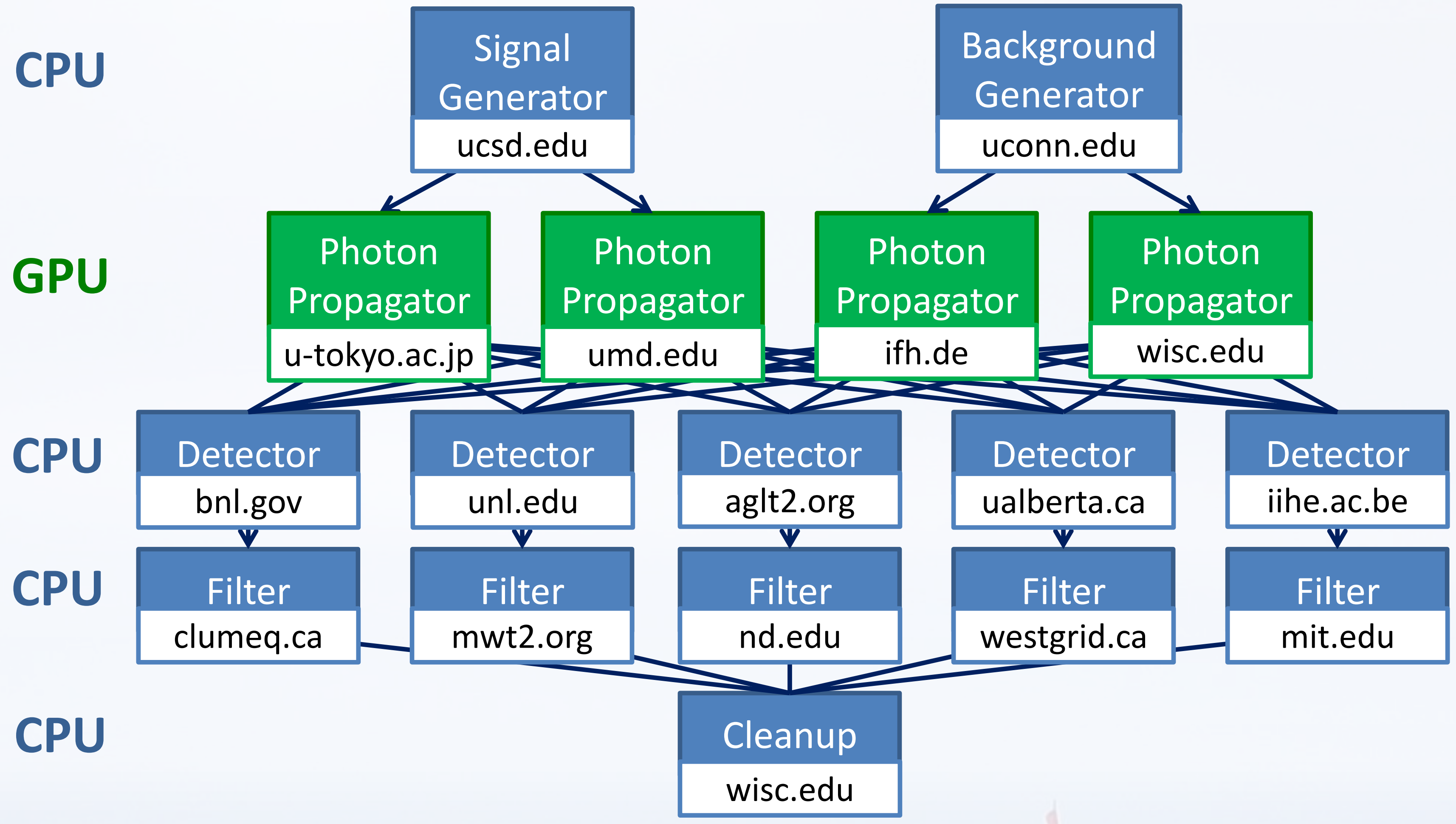
Job DAG

(directed acyclic graph)



Job DAG

(directed acyclic graph)



IceProd v1

- Site-local python daemons backed by a central database
- Plugins for each batch system
- gridftp/http directly to/from main data center
- XMLRPC to communicate with jobs
- Store configuration and statistics for every job

IceProd v1 Additions

- Initially, no concept of job inter-dependencies
  - Added about 3 years ago
  - Can do cross-site linking as of 1.5 years ago
- GPU-specific resources tacked on
  - Does not translate well to other resource types
- CPU/GPU normalization and reporting as of ~1 year ago

IceProd v1 Problems

- A long series of patches to increase functionality
- Early design decisions hampering new ideas
- Multiple bottlenecks appearing:

  DB - locking issues, queries per second

  Queue - designed for 100 jobs per hour, not 10k

IceProd v2

- Complete rewrite of codebase
  - Current usage is standard instead of extra feature
  - Focus on unit tests, documentation
  - Support for Python 3

Database

- Local SQLite database at each site
  - Can run completely standalone
  - Outage of master not a problem
  - Spread load out

Single-site

- Single-site installation in minutes
  - For individual users, or testing
  - Only requires Python
  - Installation as simple as:

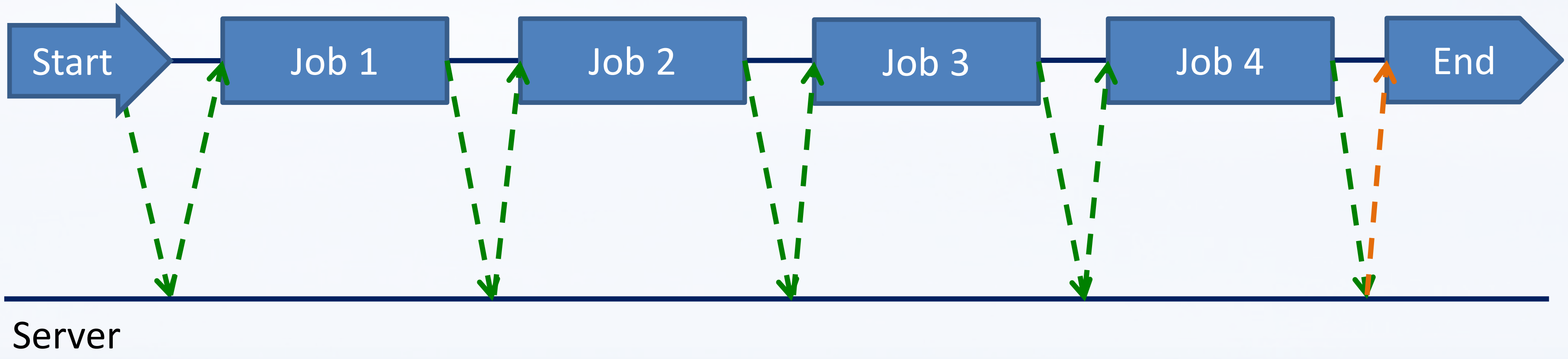
```
svn co iceprod; cd iceprod; bin/iceprod_server start
```

Software Distribution

- Better production software distribution
  - CernVM FileSystem (CVMFS) as primary method
    - Natively if possible (thanks to ATLAS/CMS)
    - Parrot user-level remote I/O otherwise
  - Software tarballs still accepted, but deprecated

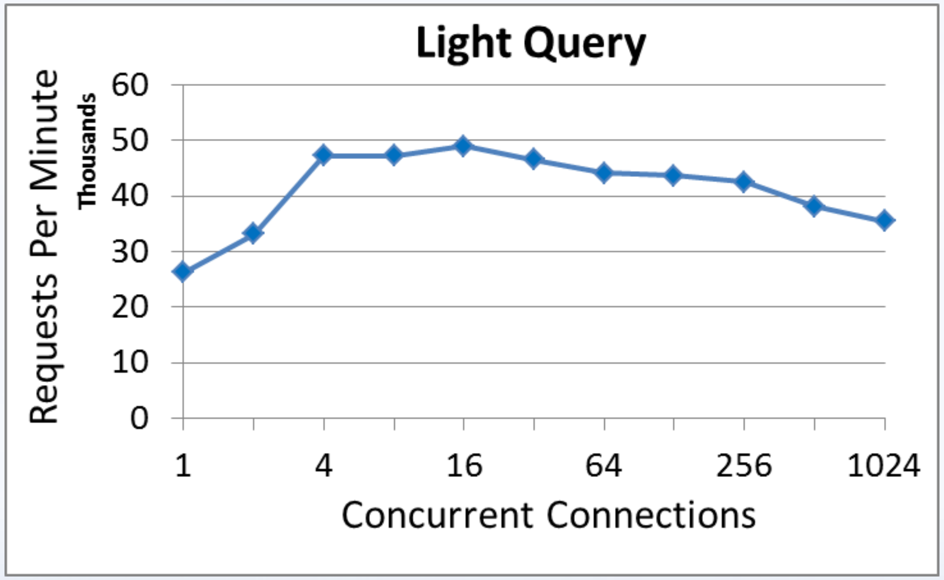
Job Optimization

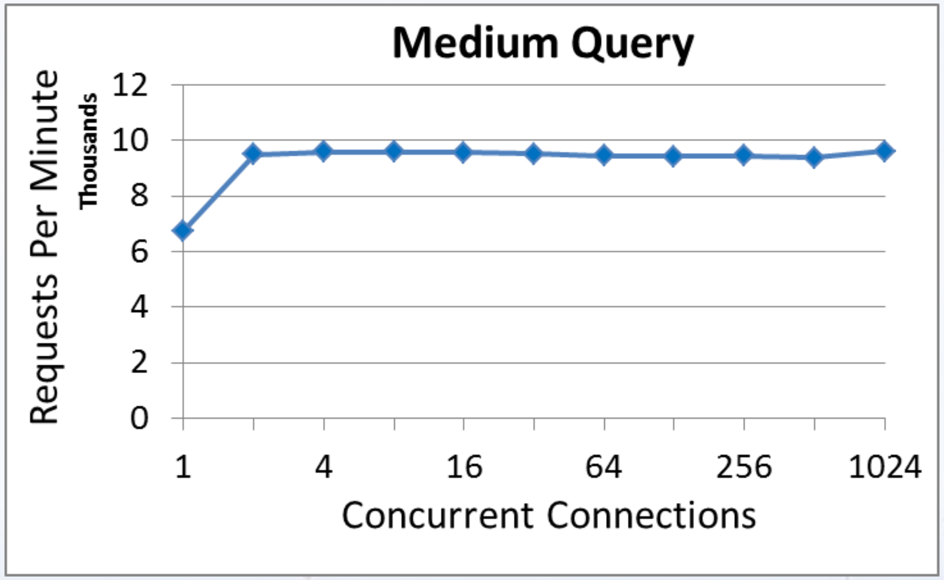
- Pilots
  - Run multiple shorter jobs serially without taxing the batch system
  - Better match between jobs and node resources



Scalability

- High-volume communications
  - Web server able to handle more than 10k connections per minute in a single thread





Users and Priority

- Multiple users
  - IceProd v1 doesn't have a concept of users, if you can log in you can do anything
  - IceProd v2 needs to support analyzers who shouldn't be able to modify other users' jobs
- Fair-share Priority
  - One set of jobs shouldn't completely starve all other jobs of resources

Web API

- JSON API
  - Many operations available either read-only or with authentication
    - Job set submission, monitoring, editing, removal
    - Job statistics (summary or for each job)
- Web / Mobile Focus
  - Integrated support for phones and tablets for management tasks

Status and Plans

- Currently in beta, initial deployment in the next few months

- Future:
  - Get analyzers in the collaboration to join
  - Work with other projects to adopt IceProd
    - Currently existing MoU with High Altitude Water Cherenkov (HAWC) gamma-ray observatory
    - We welcome others

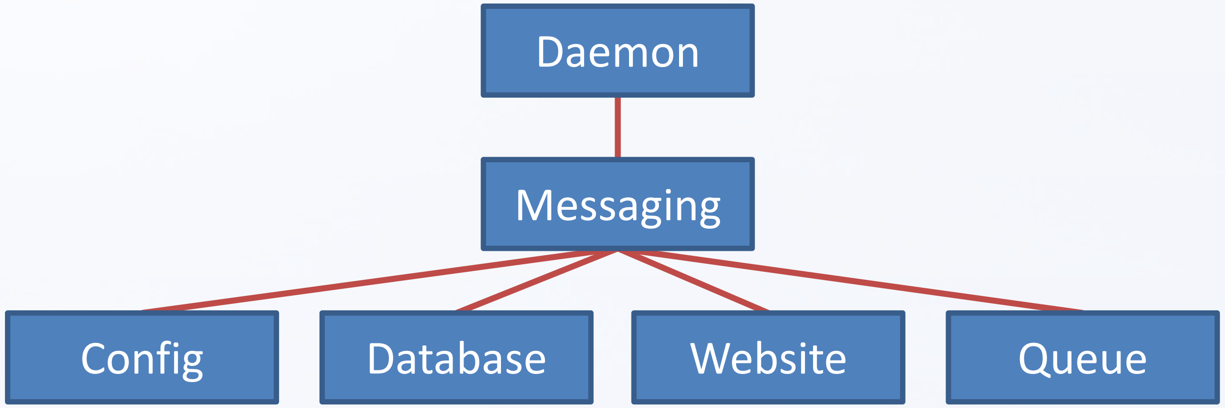
Summary

- IceCube has heterogeneous, uncontrolled resources and diverse job types
- IceProd2 can work on all of these
  - Easy to install user-level middleware
  - Simple to manage and monitor large sets of distributed jobs
  - Designed to scale from tens to millions of jobs

Questions?

Backup

Server Layout



- Each module in its own process
- ZeroMQ messaging between modules
- Hot config changes

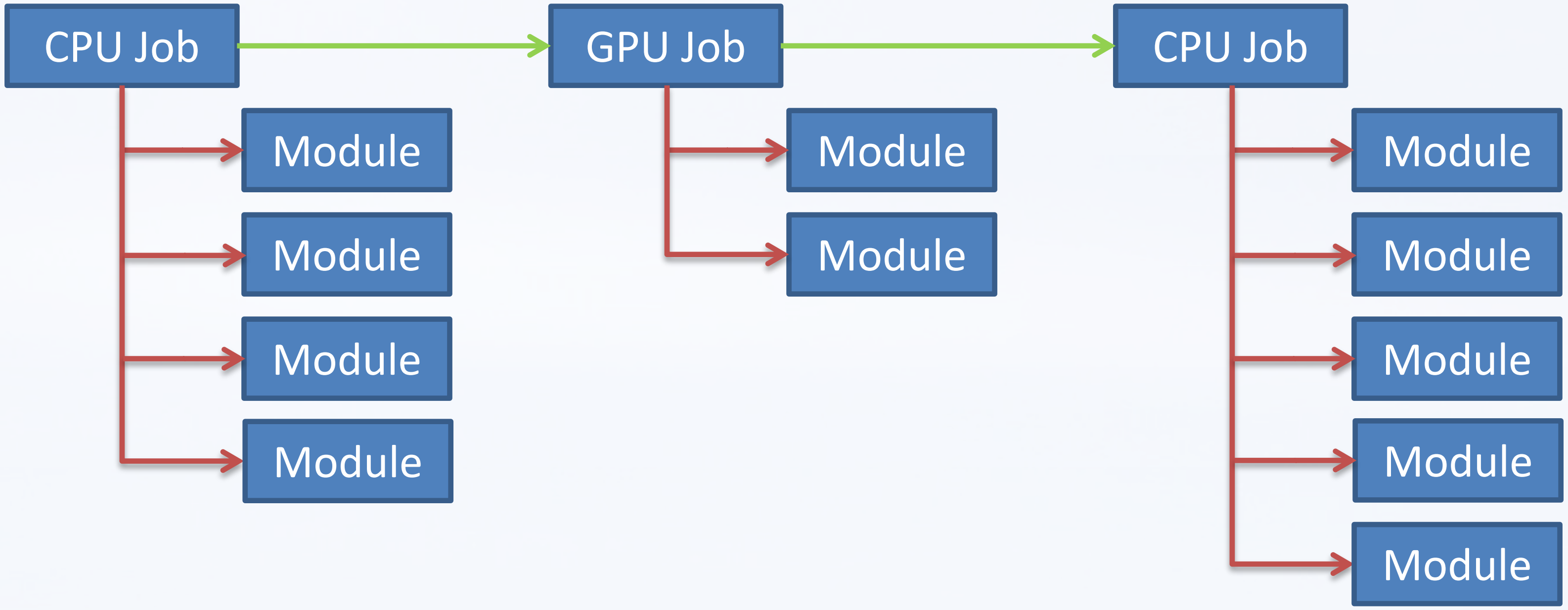
Pilot

- Can run a job config, or ask the server for jobs

```
if config
    run config
else
    check node resources
    loop forever
        ask for job matching resources
        run job or exit if no job
```

Modules

- Each job can be broken up into modules
  - Modules easily reordered, copied to other jobs


CPU Job
GPU Job
CPU Job
Module
Module
Module
Module
Module
Module
Module
Module
Module
Module
Module