

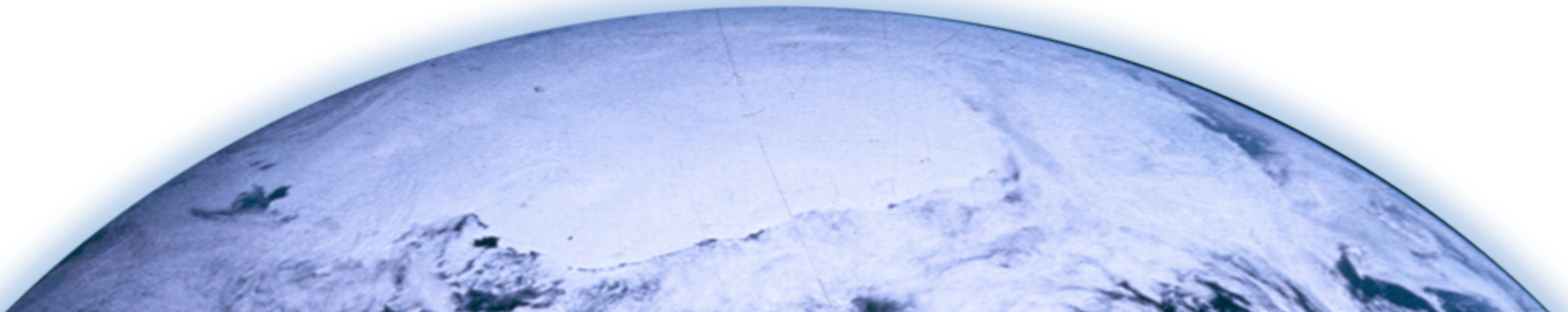
IceCube and Condor

Ian Rae

University of Wisconsin–Madison

IceCube Collaboration

- 14 Universities in the US
- 15 Universities in Europe
- 2 Universities in Japan and New Zealand



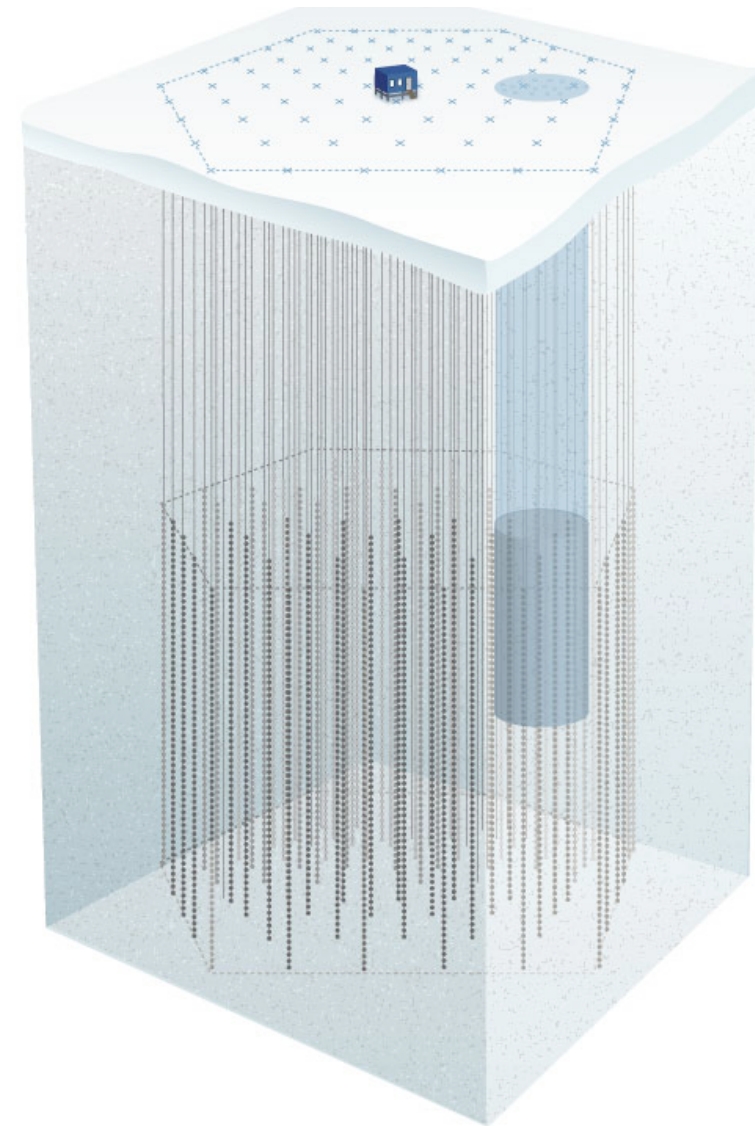
What is IceCube?

- Antarctic neutrino observatory
- Constructed in ice with a volume of 1 km^3
- Digital optical modules (DOMs) do the actual detection



In-Ice Arrangement

- DOMs arranged in 2450m long strings
- 60 DOMs per string
- Currently 59 strings deployed, final goal is 80+ strings
- Completion expected in 2011



Simulation Production

- Simulate detector in various states
 - 40 strings, 59 strings, 80 strings, etc.
- Monte Carlo method
- 1 neutrino per 1 million background particles

CPU Requirements

- One CPU hour needed to generate one detector second
- Three months livetime is 7,776,000 CPU hours
- Need Condor!

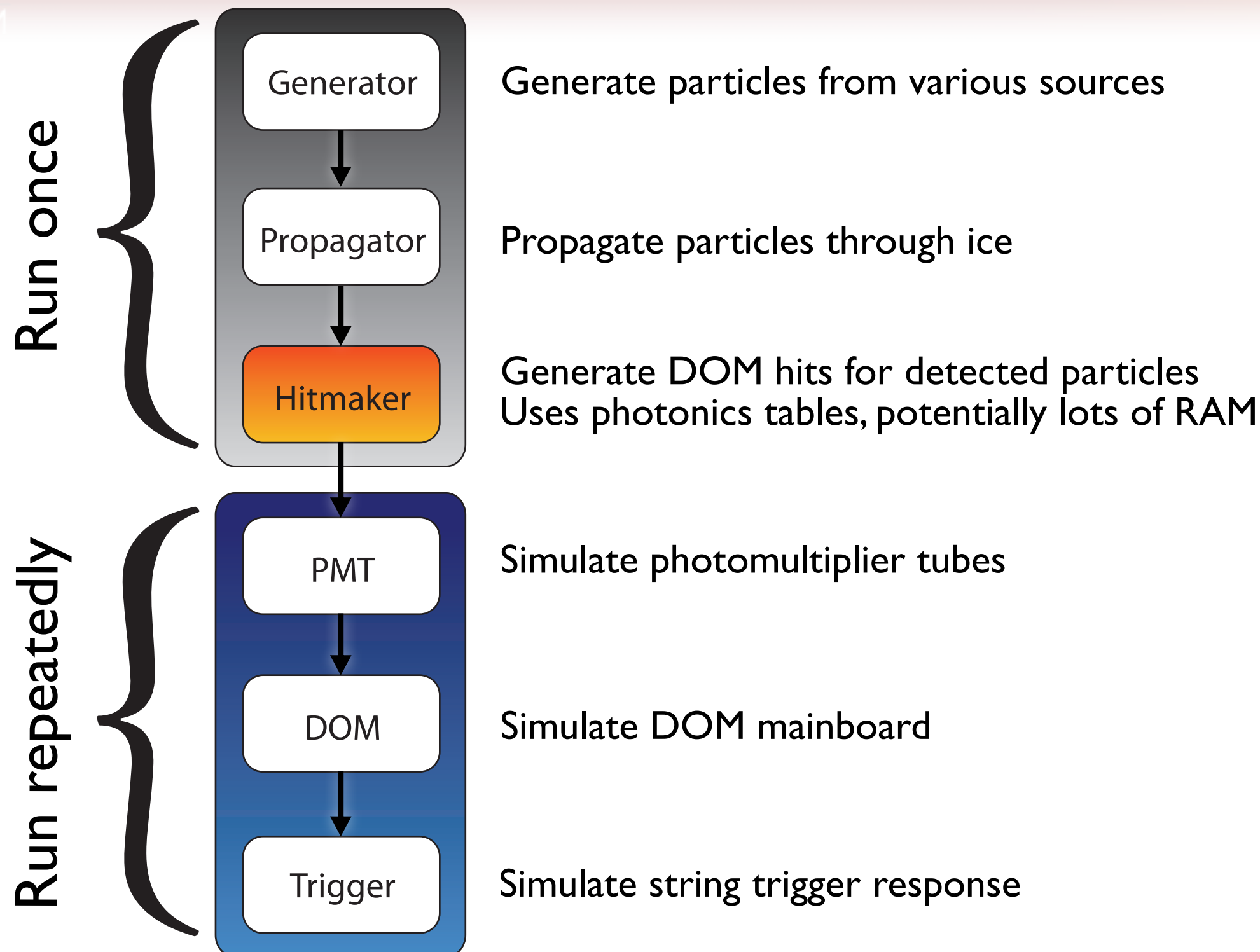
Dataset Workflow

- XML configuration files
 - Divided into sections called “trays”
 - Trays correspond to different parts of a simulation
 - Each tray executes in its own process
- Typical dataset has 10,000 simulation jobs

Photonics Tables

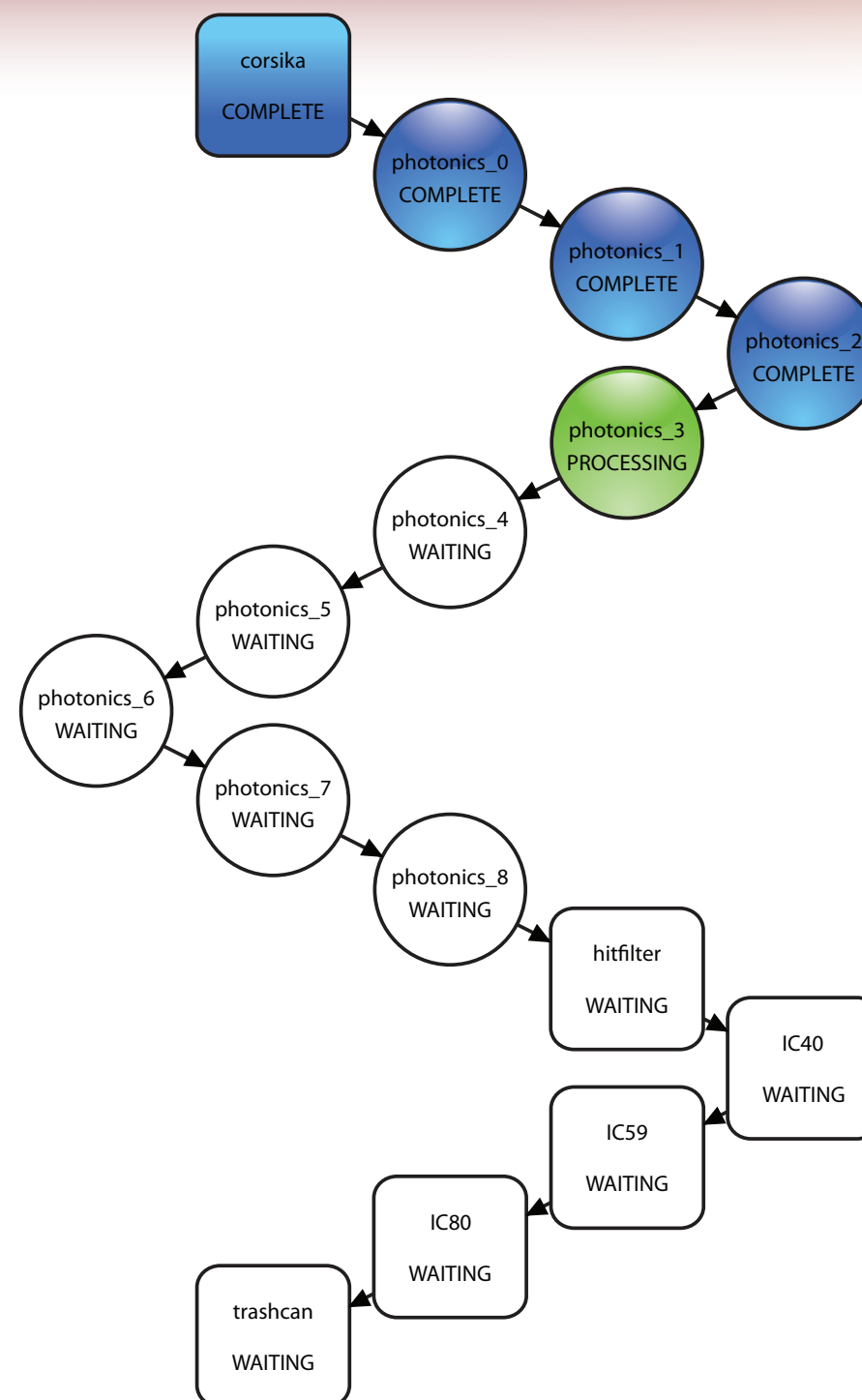
- Pre-computed data about the ice
- 14GB per set of tables
- Two sets
 - IceCube
 - AMANDA (IceCube's predecessor)

Job Workflow



Old Implementation

- Daemons generate Condor submit files
- One simulation job is one Condor job
- Job execution is entirely linear (a “big snake”)

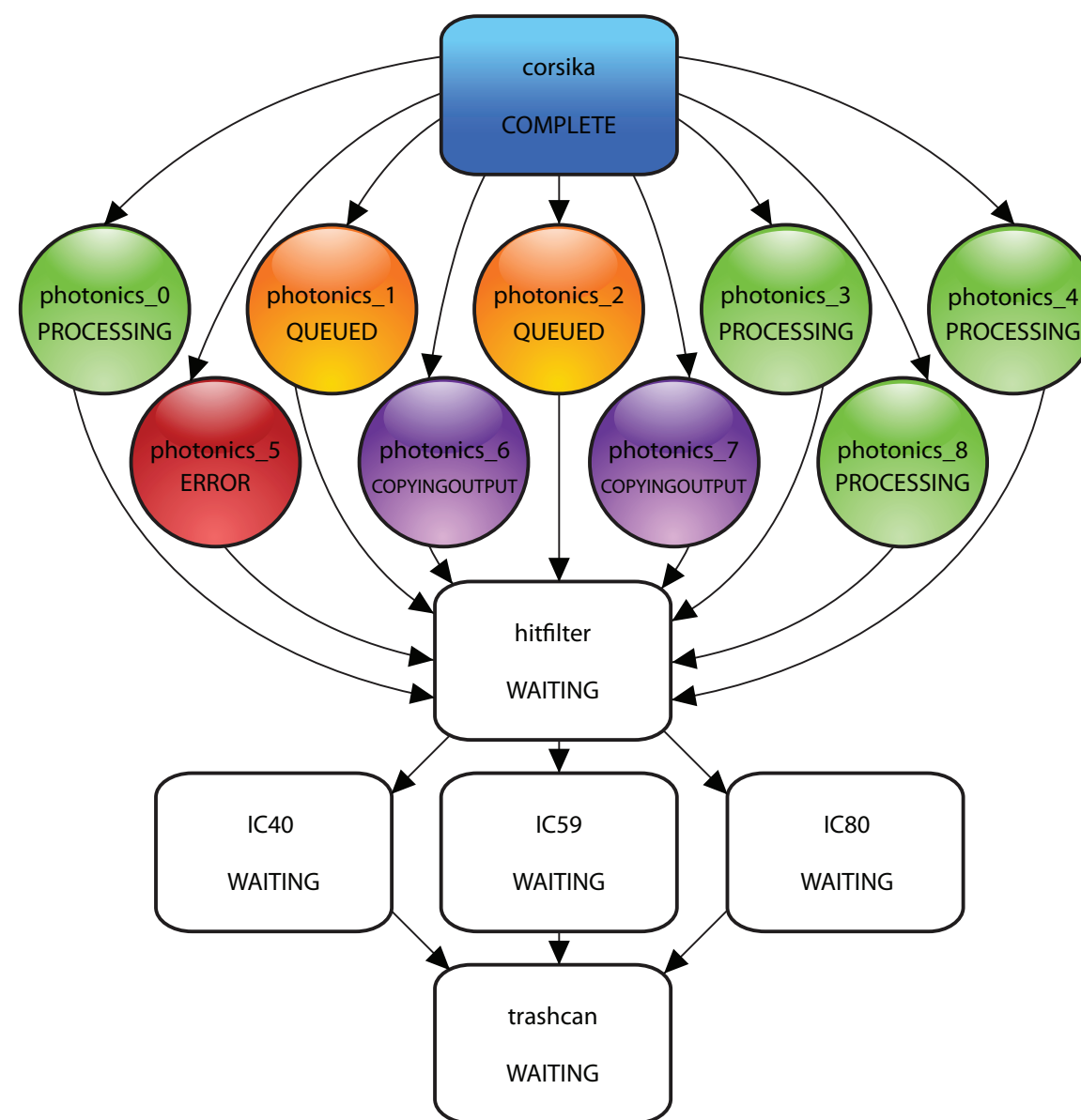


Disadvantages

- Photonics tables required for all trays
 - Assumed pre-installed on grid site
- Hardware requirements based on needs of greediest tray
- No checkpointing within jobs
- Not as parallel as it could be

New Implementation

- Break jobs into “tasks”
- In general, each task executes a single tray
- Use DAGMan to manage dependencies
- One simulation job is 15-24 Condor jobs



Advantages

- Photonics tables can be binned
 - 1 GB per bin
 - Small enough to stage on-demand
- Hardware requirements on per-task basis
- Checkpoints between every task
- Increased parallelism

Implementation Details

- Really simple
- Great fit for IceCube's existing code
- Daemons generate DAGs and submit files
- Intermediate files generated by tasks stored at IceCube and fetched using GridFTP

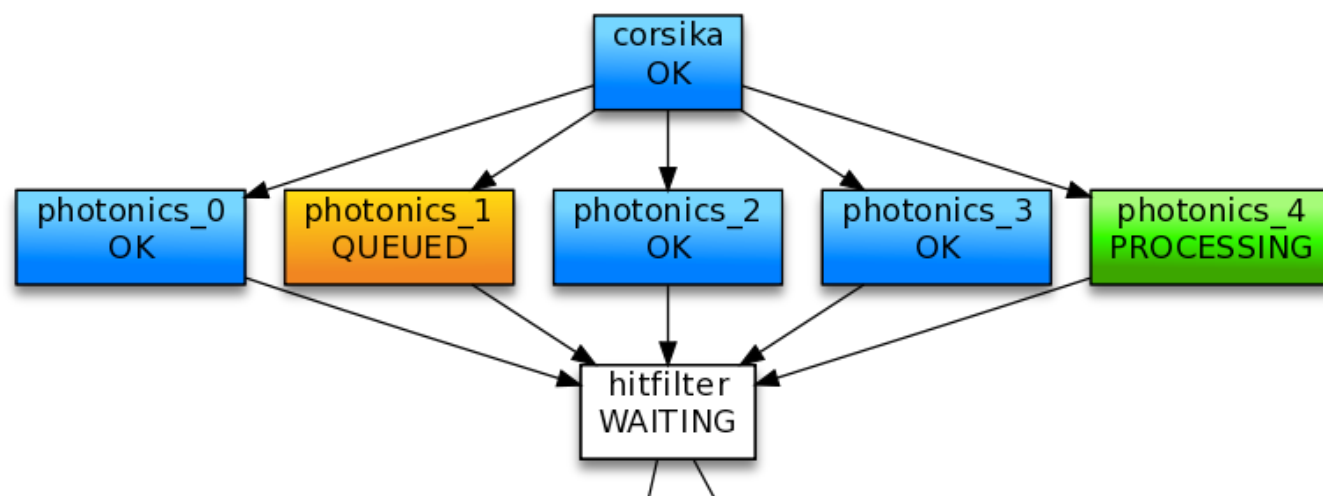
Job Performance

- Running only one simulation job, no gain from DAGs
- Increased parallelism cancelled out by network overhead
- Running 1,000 simulation jobs, DAGs give 25-30% improvement in runtimes
- Reuse data already transferred

Web Interface

Tabular display

NAME	TRAY	ITERATION	HOST	STATUS	START	FINISH
corsika	0	externs	glow-c003.icecube.wisc.edu	OK	2009-04-19 14:41:39	2009-04-19 16:20:21
corsika	0	0	glow-c003.icecube.wisc.edu	OK	2009-04-19 16:20:21	2009-04-19 16:44:26
phonics	1	0	glow-c003.icecube.wisc.edu	OK	2009-04-19 16:49:36	2009-04-19 16:58:16
phonics	1	1		QUEUED		
phonics	1	2	glow-c020.icecube.wisc.edu	OK	2009-04-19 16:49:08	2009-04-19 17:11:24
phonics	1	3	glow-c041.icecube.wisc.edu	OK	2009-04-19 16:48:40	2009-04-19 17:07:56
phonics	1	4	glow-c042.icecube.wisc.edu	PROCESSING	2009-04-19 16:49:21	
hitfilter	2	0		WAITING		



Graphical display

Future Work

- Task clustering
- Condor job router
- Glide-ins for photonics tables

Summary

- Large increase in flexibility
- Improvement in aggregate performance
- Access to more resources (e.g. OSG)
- Didn't require huge changes

Questions?

Presenter: Ian Rae

E-mail: ian@cs.wisc.edu

<http://www.icecube.wisc.edu/>