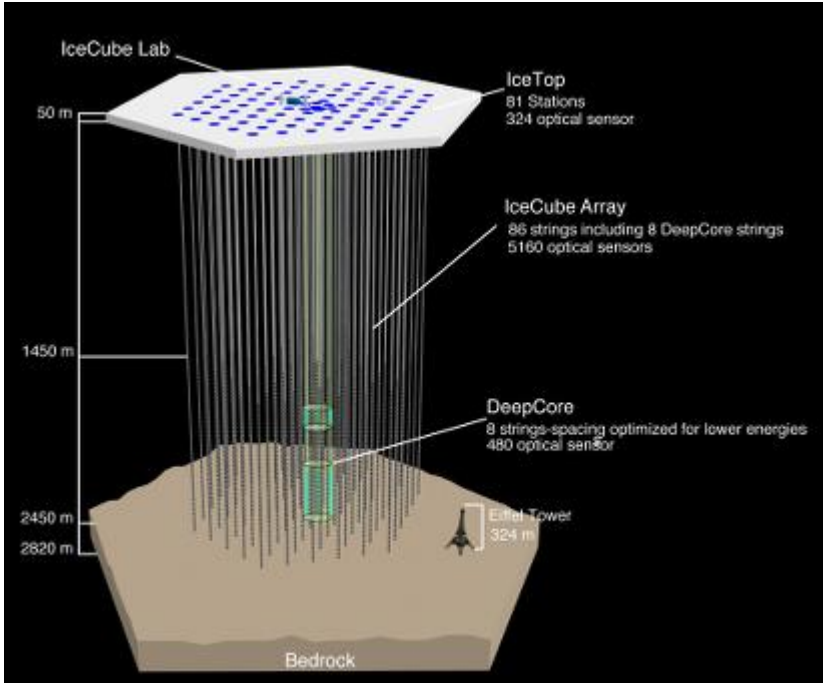
jade

An End-To-End Data Transfer and Catalog Tool

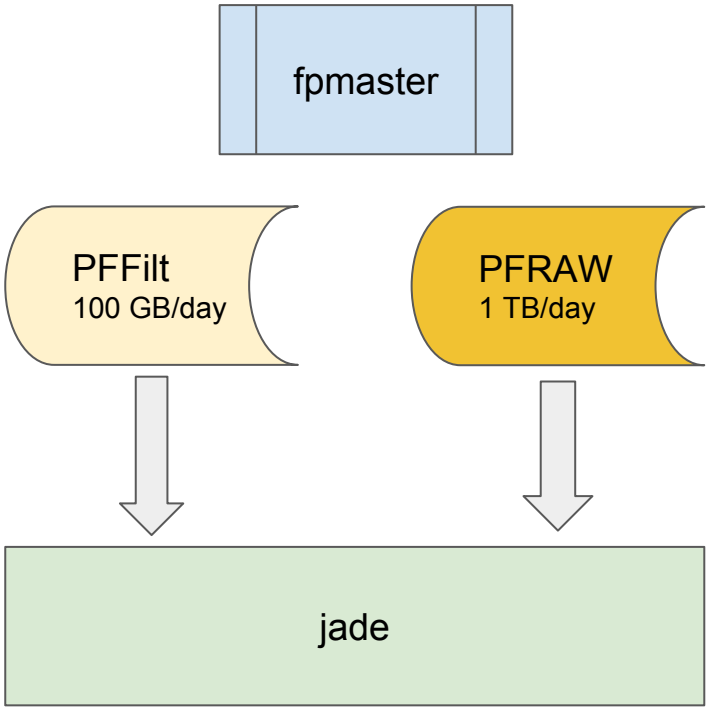
Patrick Meade (UW-Madison)
CHEP 2016 - 12 Oct 2016, 12:00 PM

The IceCube Neutrino Observatory

- Located at the South Pole
- Antarctic ice is huge and very clear
- ~5000 optical sensors (DOM)
  - DOM ~= Basketball
- Particles collide in ice leaving light trails behind (Cherenkov Radiation)
- DOMs report incident light
- Software in the IceCube Lab reconstructs events
- ~1 TB/day → Processing/Filtering



Processing and Filtering (PnF)



- ~1 TB/day → PnF

- Event processing and filtering
- ~1 TB/day Raw Data

- 10% selected for satellite
- ~100 GB/day Filtered Data

- ~1.1 TB/day → Data Transfer

jade Origins 1

- South Pole Archive and Data Exchange (SPADE) was the original data archive and transfer system
- Cindy Mackenzie created SPADE in 2005
- SPADE ran on JBoss 4 as part of a larger Java Management Extensions (JMX) ecosystem
- By 2010, JMX ecosystem never materialized

jade Origins 2

- 2012: Java Archiver and Data Exchange (jade) is born

- The core of SPADE: components running in a JBoss application
- The core of jade: a “simple” job engine

- Don’t Ever Write A Job Engine

- 2015: jade v2 is born
- jade v2 uses the SPADE component model, as individual processes

jade Family

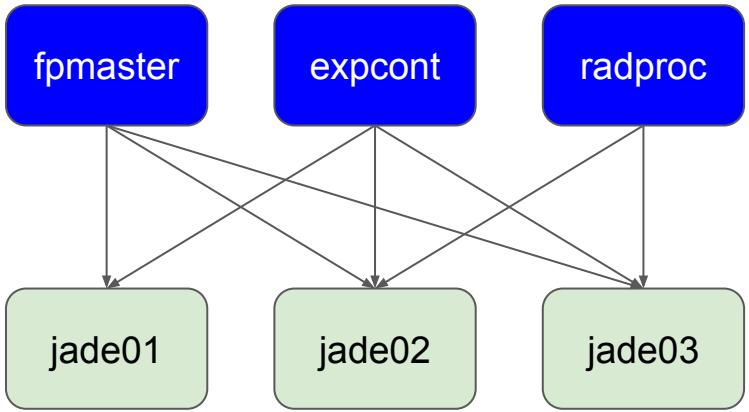
- jadeite
  jade at the South Pole Systems (SPS)

- nephrite
  jade at UW-Madison (jade-in-the-north)

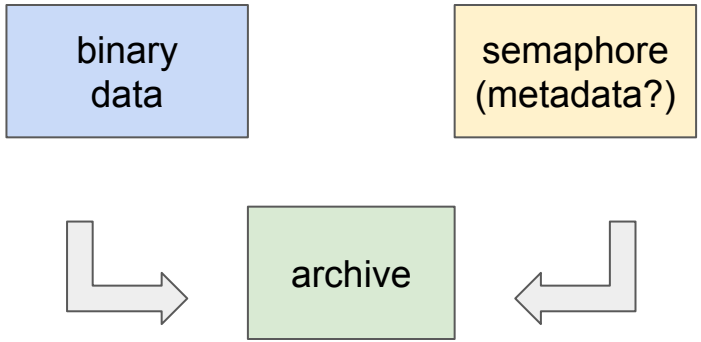
- kanoite
  jade Long Term Archive (archive to DESY, NERSC)

fetcher - (South Pole jade component)

- Scan data hosts for new files
- Semaphore files to signal readiness
- jade host claims file in DB
- Checksum at data host
- Copy to jade host
- Checksum at jade host
- Delete from data host



processor - (South Pole jade component)

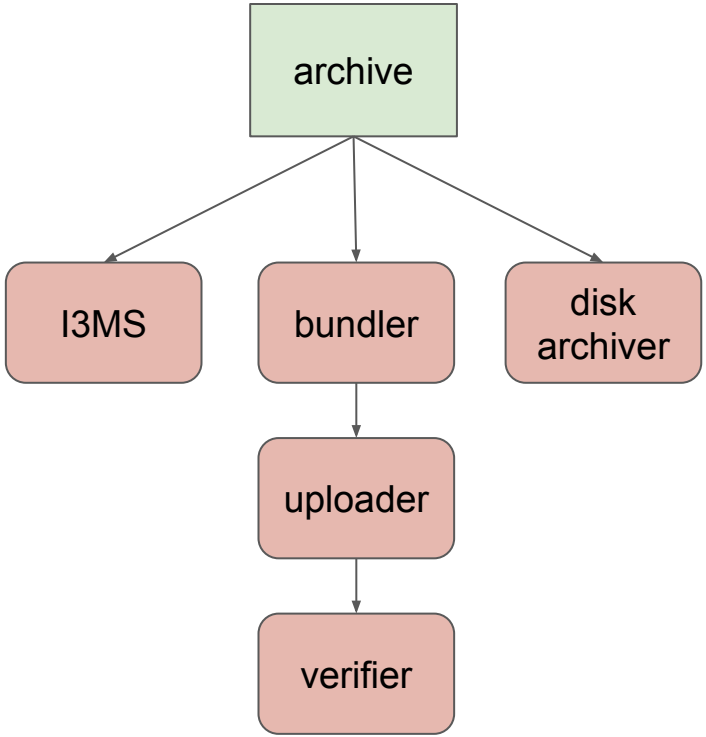


- Generate or validate metadata
- Compress to archive
  - .tar.gz for RAW
  - .tar.bz for Filtered
- Checksum archive

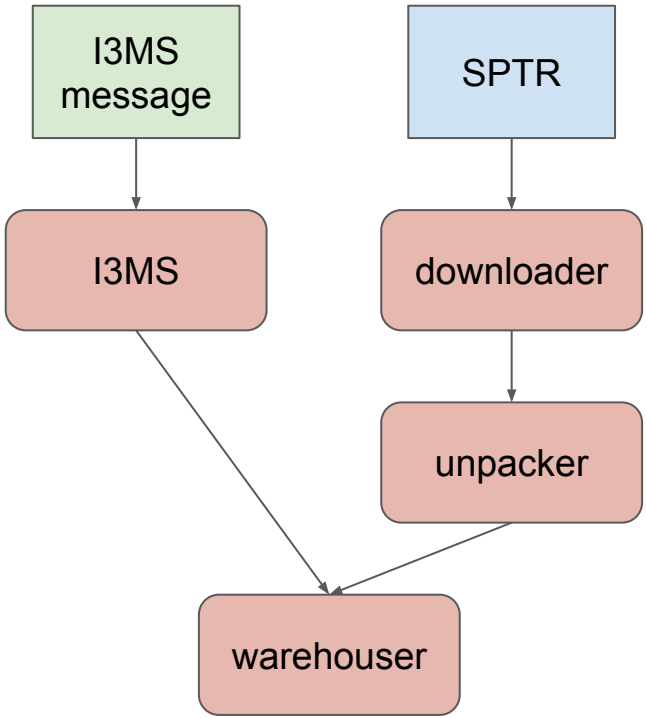
archiver - (South Pole jade component)

- Copy archive files to multiple archival destinations

- IceCube Message System (I3MS)
- South Pole Transfer Relay (SPTR)
- Archival Disk



nephrite (jade at UW-Madison)



- Receive files from IceCube Message System (I3MS)
- Archive files to Data Warehouse

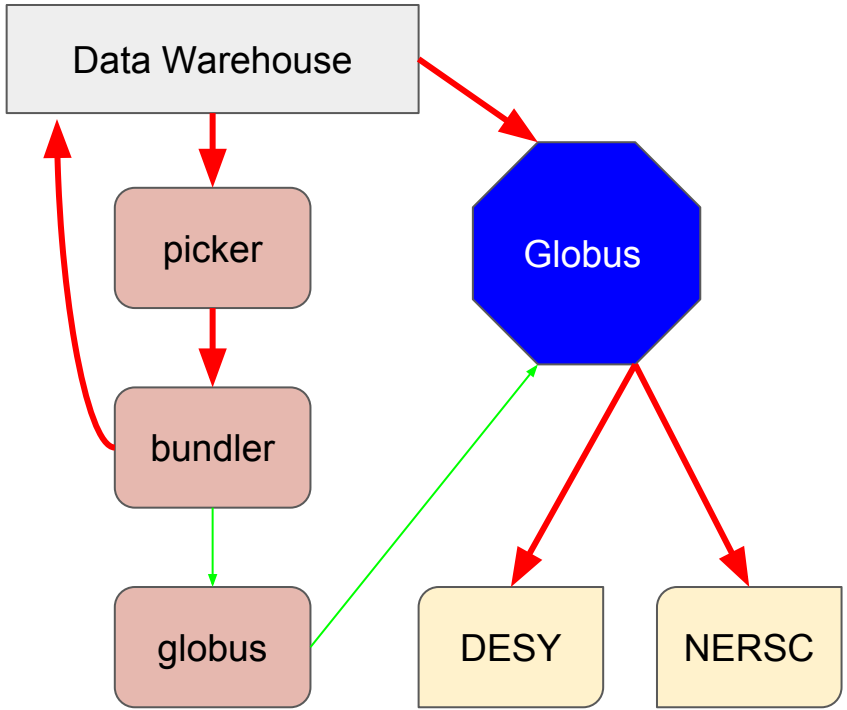
- Receive files from South Pole Transfer Relay (SPTR)
- Unpack ~1 GB satellite bundles
- Archive files to Data Warehouse

kanoite (jade Long Term Archive)

- picker selects files to be bundled
- bundler creates archive bundles
- globus schedules transfers

- Globus has a REST API to move data between sites

- IceCube collaborators DESY and NERSC provide long term storage



File Catalog (alpha)

- Each jade family has its own database

- Need a one-stop queryable catalog for all files

- If a supernova alert comes, which files cover that time-span?
- Which collaborators have File X in long term archive?
- Which files belong to Run 128577?

- Combines data archive and transfer databases,
  with science processing databases.

And Then...

Lessons Learned: Architecture

- DON'T: Write a Job Engine to handle everything
- DO: Do one thing and do it well (UNIX philosophy), rinse, repeat

- DON'T: Wait for later to add monitoring
- DO: Start with monitoring; even a No-OP service should be monitorable

- DON'T: Put configuration in code, databases, or external services
- DO: Put configuration in a text file that the operator can edit

Lessons Learned: Components

- DON'T: Look up your work tasks in a Job Engine database
- DO: Have a defined inbox and outboxes

- DON'T: Let a component get stuck
- DO: Have a quarantine directory to hold problem files

- DON'T: Have special signal methods between components
- DO: Atomically move work to downstream components

- DON'T: Be afraid to shut things down
- DO: Put components into a FULL STOP mode if they cannot work

Lessons Learned: Operation

- DON'T: Make your operators guess if things are working OK
- DO: Signal monitoring software (Nagios) if anything goes wrong

- DON'T: Make it difficult to find or read the logs
- DO: Log human readable information about every error

- DON'T: Require the operators to be co-developers
- DO: "Have You Tried Turning It Off And On Again?"

Lessons Learned: Software

- DON'T: Expose database keys as public identifiers
- DO: Use UUIDs as public identifiers; (DB keys still OK for internal use!)

- DON'T: Use ZeroMQ; hiding errors makes it harder to handle them
- DO: Use RESTful APIs (few, large) or Websockets (many, small)
  - Bonus: REST APIs are usually easy to monitor!

- DON'T: Pick the wrong database
- DO: Understand document and relational databases; transactions

Lessons Learned: Developer

- DON’T: Be afraid to say “My Software Sucks”
- DO: Consider what problem you are solving

Questions?

Thank You For Your Kind Attention! ^_^

Patrick Meade (UW-Madison)
patrick.meade@icecube.wisc.edu

jade: An End-To-End Data Transfer and Catalog Tool

The IceCube Neutrino Observatory is a cubic kilometer neutrino telescope located at the Geographic South Pole. IceCube collects 1 TB of data every day. An online filtering farm processes this data in real time and selects 10% to be sent via satellite to the main data center at the University of Wisconsin-Madison. IceCube has two year-round on-site operators. New operators are hired every year, due to the hard conditions of wintering at the South Pole. These operators are tasked with the daily operations of running a complex detector in serious isolation conditions. One of the systems they operate is the data archiving and transfer system. Due to these challenging operational conditions, the data archive and transfer system must above all be simple and robust. It must also share the limited resource of satellite bandwidth, and collect and preserve useful metadata. The original data archive and transfer software for IceCube was written in 2005. After running in production for several years, the decision was taken to fully rewrite it, in order to address a number of structural drawbacks. The new data archive and transfer software (JADE2) has been in production for several months providing improved performance and resiliency. One of the main goals for JADE2 is to provide a unified system that handles the IceCube data end-to-end: from collection at the South Pole, all the way to long-term archive and preservation in dedicated repositories at the North. In this contribution, we describe our experiences and lessons learned from developing and operating the data archive and transfer software for a particle physics experiment in extreme operational conditions like IceCube.

jade Origins 1.5

- JBoss was a heavy application container
- SPADE code (Java 1.4) was starting to age
  - Limited Collections API usage
  - No Generics
  - `synchronized` blocks to handle concurrency
- Management by web interface ... over a satellite link
- Java Archiver and Data Exchange (jade) is born