# Present and Future of the IceCube DAQ and Online Systems

Kael Hanson

University of Wisconsin Physics Department and

Wisconsin IceCube Particle Astrophysics Center (WIPAC)

VLVnT 2015 – Rome, Italy

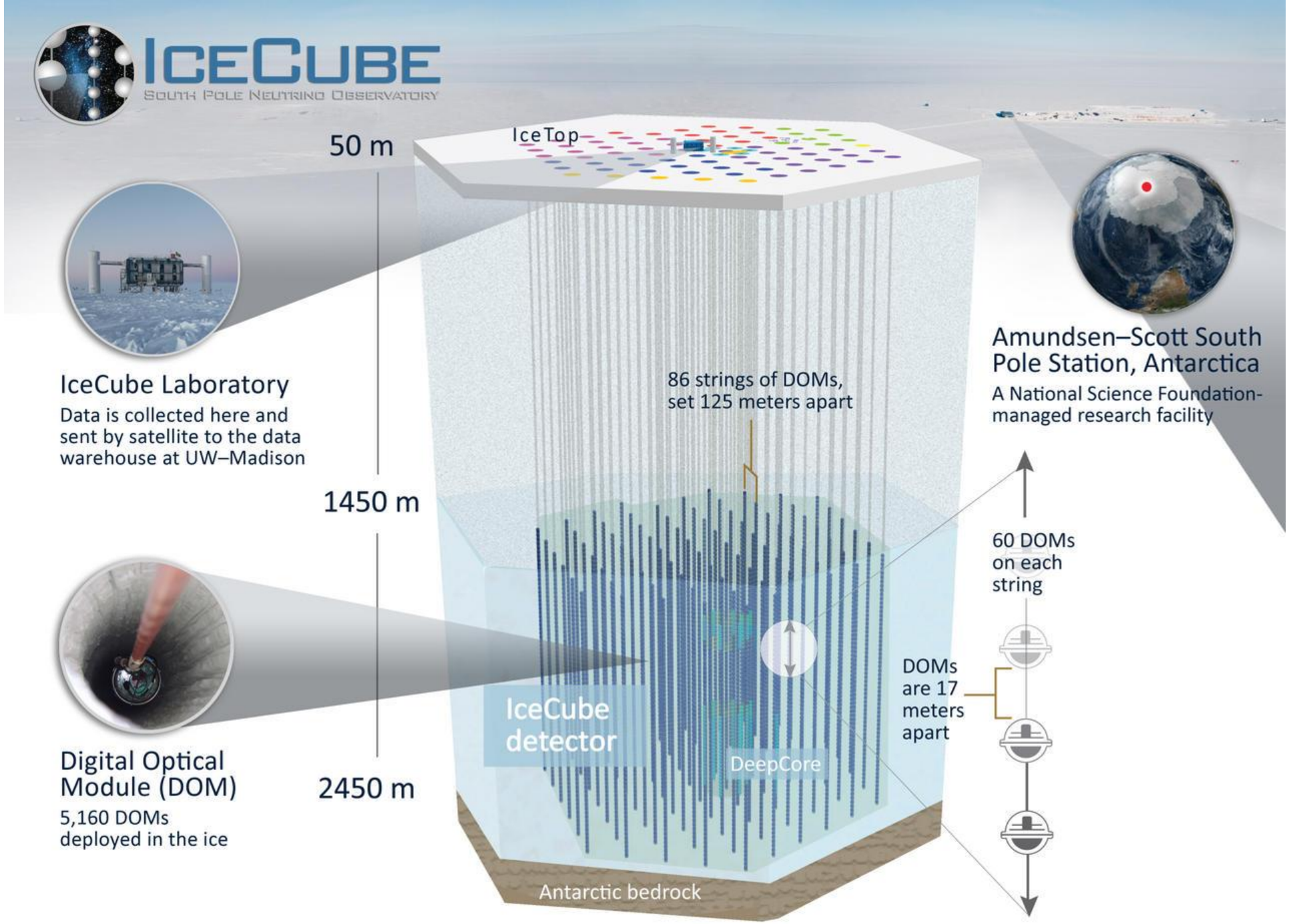
ICECUBE GEN2
COLLABORATION

*Many figures from David Heereman PhD thesis – thanks David!*

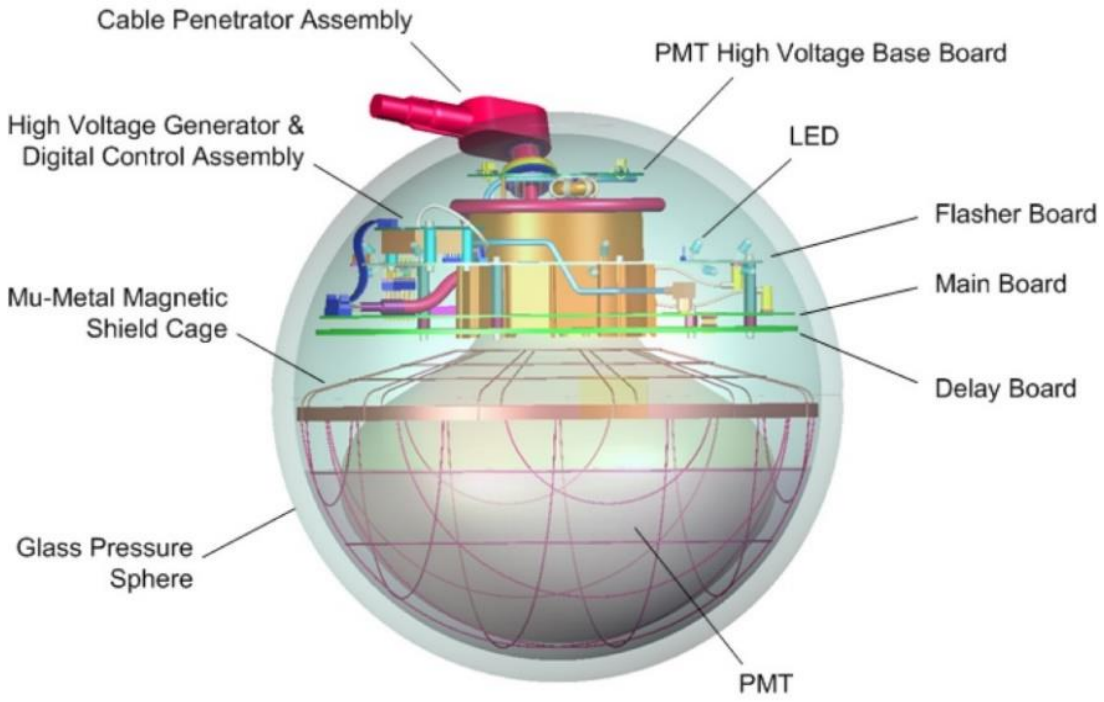
WIPAC
WISCONSIN ICECUBE PARTICLE ASTROPHYSICS CENTER

# Motivation & Other Places to Look

- IceCube DAQ covered recently:
  - John Kelley has a nice DAQ summary talk from 2013 VLVnT. I try not to repeat too much of what's in there since it covers the basics;
  - Also Volker Baum's talk from that conference.
- I want to focus on recent enhancements to the IceCube DAQ and one topic in particular: a technique called *hitspooling* which allows storage of the full non-triggered raw data coming from IceCube DOMs for, in principle, weeks.
- Then we explore where we are going with this;
- I hope some of the lessons learned from this may be beneficial to other super-large neutrino telescope DAQ fans.

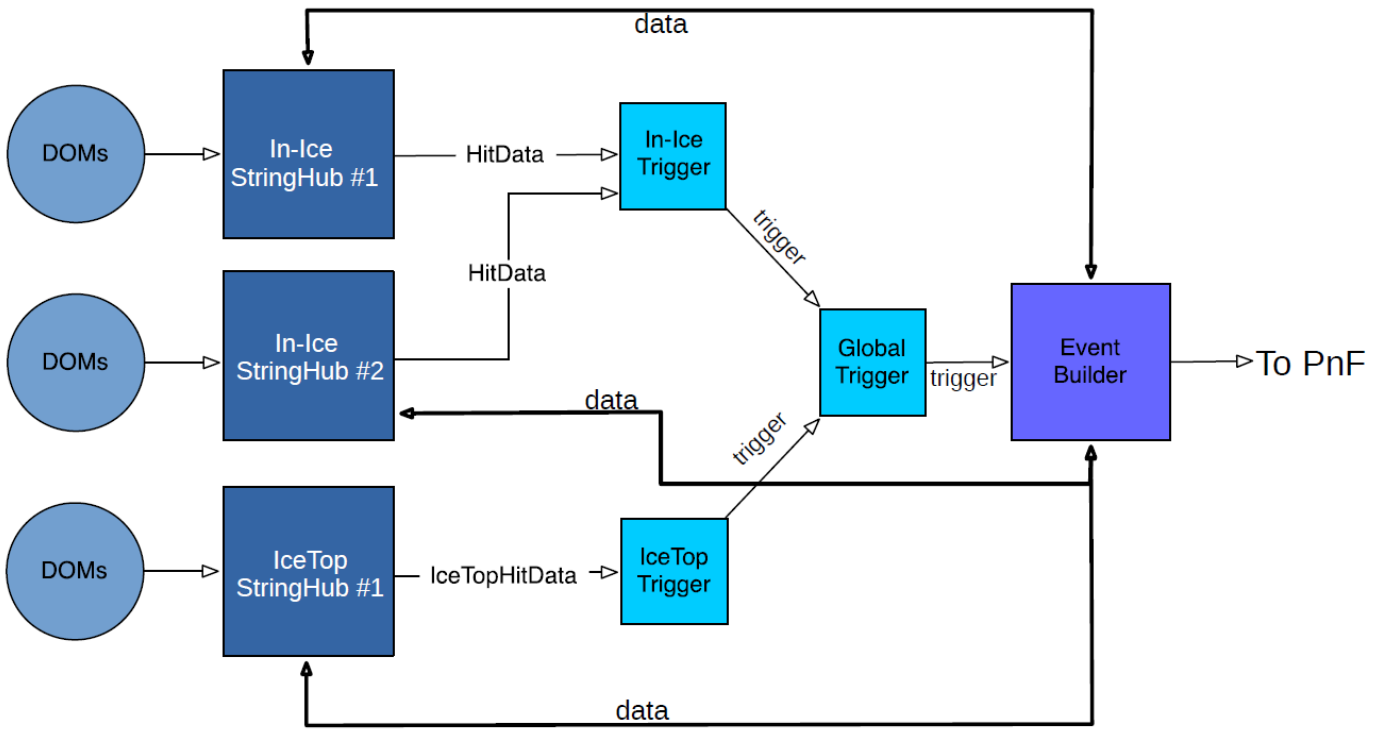

IceCube
South Pole Neutrino Observatory
50 m
IceTop
1450 m
2450 m
IceCube Laboratory
Data is collected here and sent by satellite to the data warehouse at UW–Madison
Digital Optical Module (DOM)
5,160 DOMs deployed in the ice
86 strings of DOMs, set 125 meters apart
IceCube detector
DeepCore
Antarctic bedrock
Amundsen–Scott South Pole Station, Antarctica
A National Science Foundation-managed research facility
60 DOMs on each string
DOMs are 17 meters apart

# DOM




- From surface DAQ perspective the DOM hides most of the details of low level signal capture and provides a binary stream of hit packets upon request.
- It has a relatively reasonable buffer depth – 8k hits – so given 500 Hz as rough rate of hits, system can tolerate > 10 seconds latency;
- It happened that GC and JIT in the JVMs of the StringHubs conspired to blow through this latency and cause data gap, 1-2s, at run start. This has now been remedied.
- As long as you can keep reading and the DOM does not produce more than 45 kB/s of hit data all runs smoothly and it now normally does.
- DOM emits time calibration packets so that StringHubs can transform from DOM LO time to UTC derived time as hits roll in.
- DOM also accumulates noise scalers. Discuss later.

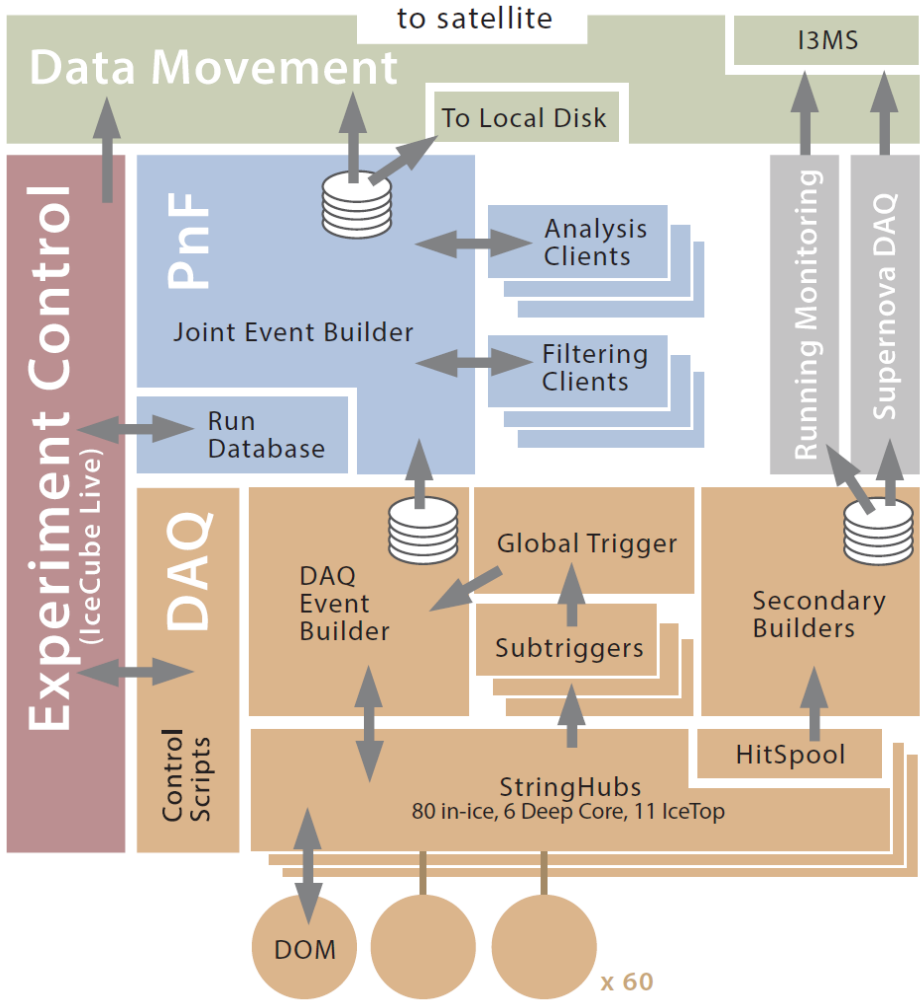
# The IceCube Data Acquisition System (DAQ)



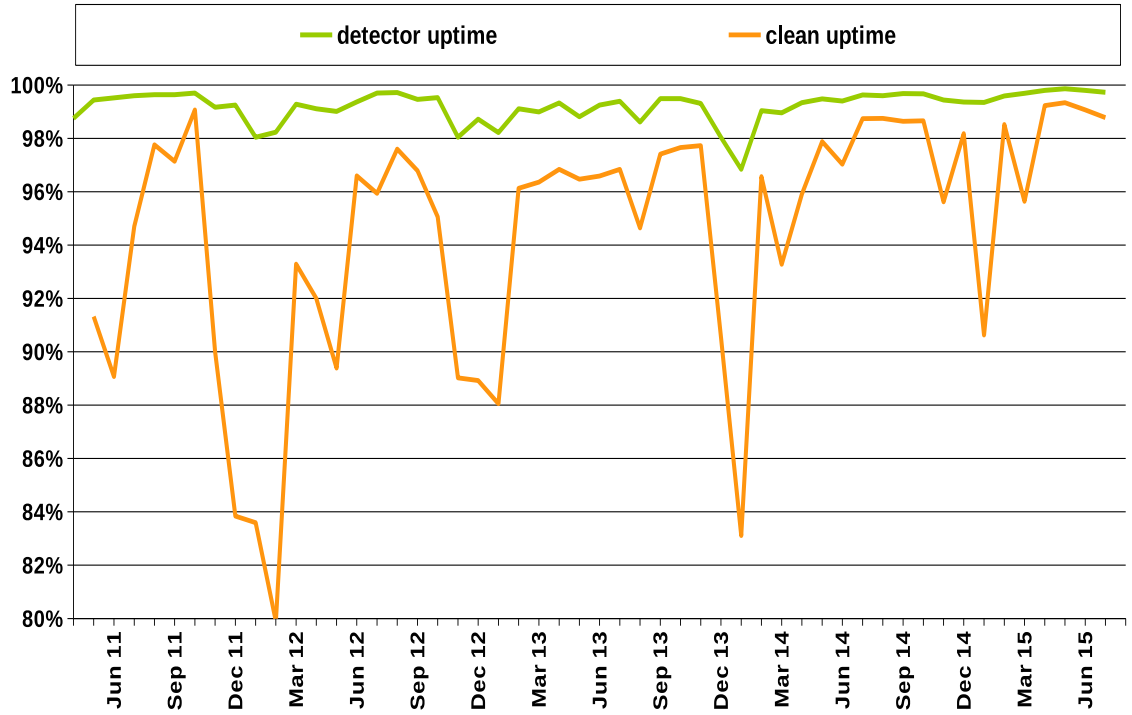

- Surface DAQ is network distributed: 1 host per string plus trigger and event builder nodes
- Kernel code in Hubs to deal with PCI hardware which does on-the-wire comms.
- Rest is Java (160 kLoC) for speedy stuff and Python (70 kLoC) for glue stuff.
- Network configuration via XML-RPC soon to become ZeroMQ
- Hubs buffer hits which get requested by Event Builder on trigger signal. Buffers are shallow – 30 – 50 seconds. More on this later.
- Total latency is O(3-4) seconds.
- Input rate ~ 150 MB/sec
- Output rate ~ 10 MB/sec – 140 MB/sec 'falls on the floor' and is lost forever. Sort of. More on this later.
- Trigger rate approx. 2,700 Hz – mostly CR muons but atmospheric neutrinos every few minutes and HE cosmics every week or so.

# Online System

- Once 10 MB/sec events reach disk they are sent to processing clients in “largest DataCenter on the Antarctic Continent”
- ICL Dell PowerEdge R720 servers:
  - 11 infrastructure
  - 24 filtering
  - 4 monitoring and verification
  - 4 DAQ
  - 2 spares, DM-Ice, radio
- Data from filter farm reduced to fit over daily satellite transfer (100 GB/day).
- All raw data written to disk – LTO tapes RIP Oct 2014
- Decision of filters, satellite B/W allocation handled by TFT Board of IceCube – typical yearly change in detector and processing configuration.



# Historical Uptime Performance




- Detector uptime is a key metric monitored and reported regularly to stakeholders (i.e. Collaboration and funding agencies).
- Define two classes:
  - **Detector Uptime:** at least some fraction (typically most, maybe 1 string missing) of IceCube is running.
  - **Clean Uptime:** nominal detector configuration (i.e. 86 strings and all of IceTop) running and analysers can assume standard simulation applies.
- Degradations to detector uptime are DAQ crashes or other unplanned downtime. Lately the largest contribution to this has been run cycling which takes O(60 sec).
- Degradations to clean uptime are calibration activities, maintenance, upgrades. As you can see from chart this is highly seasonal.

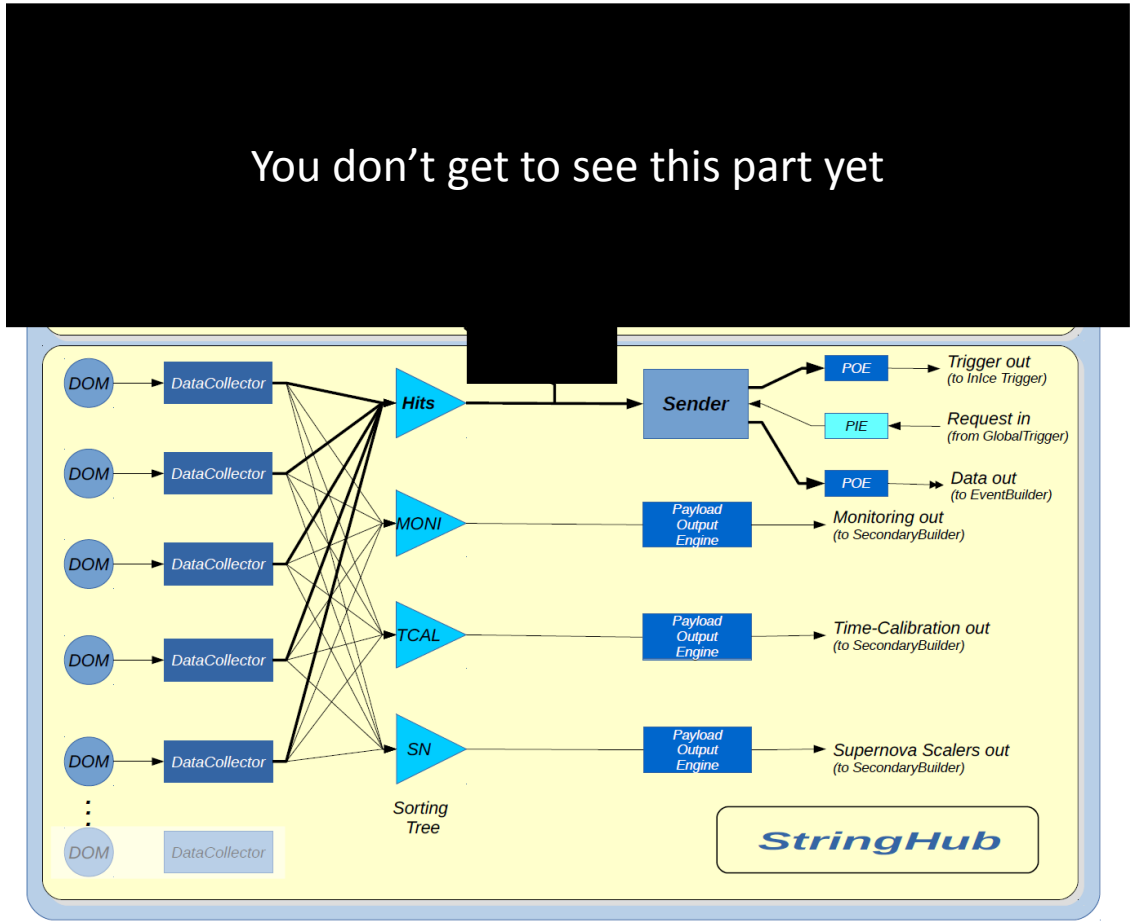
# DAQ Improvement #1: Stopless Runs

- Crazy as it sounds we were losing 200 sec per day to ‘run cycle’ transitions. IceCube software had grown up around the concept of a run lasting no more than 8 hours.
- In most cases the run configuration was not changed at all. A new run number was generated for administrative purposes.
- The first solution was to try to minimize the cycle time. However there is a ton of configuration information sent to the DOMs at run start, waveform pedestals are taken, … hard to get this below 60 sec.
- The longer-term solution was to rework the DAQ to sneak in run number transitions without actually stopping the data collection from DOMs in the case (the usual one) where configuration doesn’t change.
- This work started in 2012 and finished early 2015.
- We still cycle every 24 hours – there are some internal 32-bit counters which might overflow and DAQ experts are not ready to sign off on long-period runs. We are ultimately limited by 48-bit 40 MHz clock in DOM which overflows every 81 days.
- Nonetheless the typical uptime has gone from 99.7% to 99.9%.

# DAQ Hit Spooling

Deep buffers for DOM hits

# HitSpooling – Deep Buffering of IceCube Hits

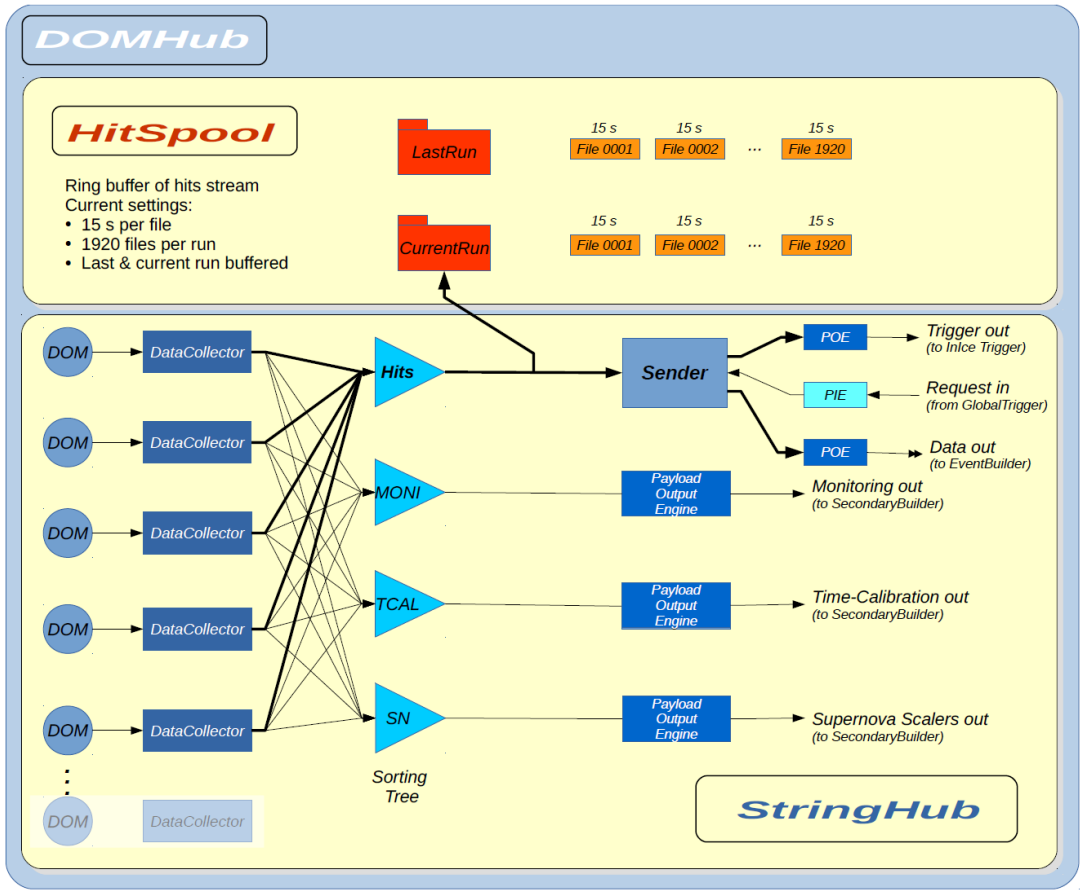



So, recall how I said that the StringHub buffer depths were maybe 1 minute deep? Well, that's fine for most well-behaving runs – the triggers and event readouts usually occur within 5 seconds.

There's another system – the Supernova DAQ (really online trigger but OK let's call it SNDAQ) – which renders triggers on a much slower time scale. This is mostly due to a desire to accurately measure noise in a time window around triggers.

- SNDAQ consumes scaler stream from DOMs: each DOM records its noise triggers in 1.6384 ms bins – SNDAQ aggregates this and looks for overall non-poissonian fluctuations as evidence for SN neutrino burst in ice.
- Until 2013 this was all you'd get from a SN event. The 'neutrino light curve' would itself be extremely useful but there is potentially so much more we want to milk out of these rare events.

# HitSpooling – Deep Buffering of IceCube Hits - 2




Each StringHub is a full blown Linux computing host with a large hard disk (was 40 GB when we started this project but now it's 2 TB). **An idea formed:**

*Why not use this disk as a temporary storage for raw hits coming from the IceCube PMTs and let requestors ask for time segments within this pool (across all hubs)*

At the time there were 2 use cases:

- Provide SNDAQ a back-door to request full detector snapshot in case of Milky Way GC supernova
- Allow safety buffer for runs which crashed because of too much data – we actually never exercised this.

A tee was inserted before the **Sender** back-half of StringHub to siphon off the raw hits and write them to disk file:

- The original HitSpool implementation used a double directory structure: LastRun/CurrentRun but that had a maximum history of 16 hours due to 8 hour run structure.
- New, not-yet-released hit spool code uses *full disk* – maximum buffer depth is **1 week+**.

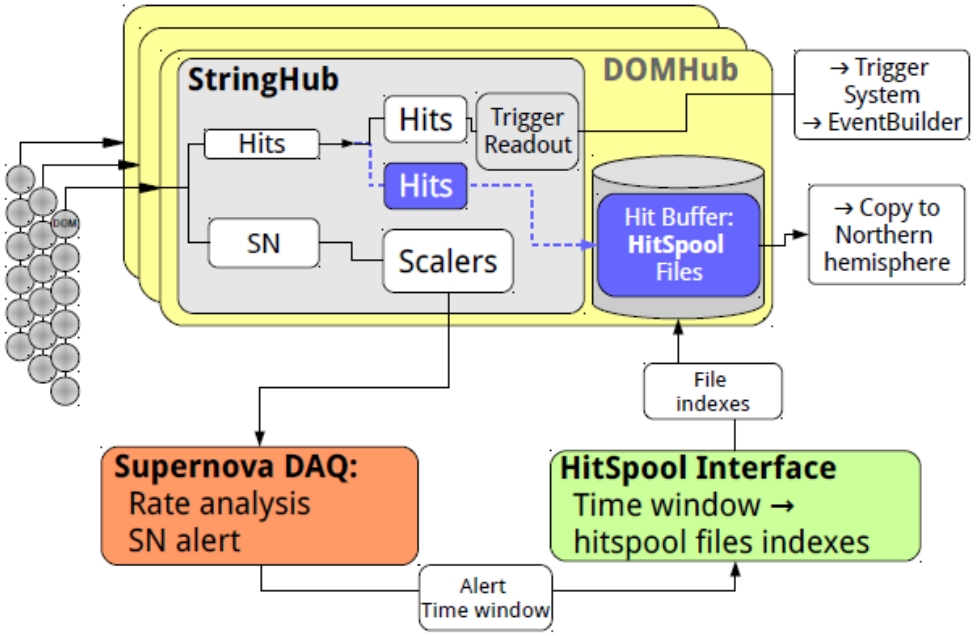
# Hit Spooling Interface Architecture

So how does this work? Say for example SNDAQ sees a high significance SN trigger candidate – now defined as 7.65 sigma which happens about 1 every other week – and wants to request hit spool data. It opens a ZeroMQ connection to a server running on the experimental control machine, passing:

- Time window [t0, t1]
- Where to put the data

The interface in turn fires off requests to daemons running on each hub – these daemons are responsible for locating the proper segment in time and scp'ing it to the server machine. N.B.: the granularity of time is currently limited to copying entire files which last 15 sec, somewhat arbitrarily.

The server then tars up the files from all hubs and delivers to the destination specified in the request.




It can take 30 mins to move files off of hubs in manner which doesn't load CPU and network – the services include logic to inhibit overlapping requests.

# Additional Use Cases

- Until 2015 the SNDAQ was the only user of this utility.
- Now we have 1 more client – the online HESE filter
  - Motivation: flavor physics from measurement of gammas from thermalized neutrons following a large hadronic cascade in ice.
  - Photons come at thermalization time → 100 us later.
  - Could use DAQ with large event window but people felt this was disruptive for filters – many long overlapping events which would have to be split.
  - Hit spooling runs ‘out of band’ and has little / no impact – additionally one gets very detailed noise rate measurement around the HESE event time.
- Status: implemented and under review by TFT.

# DAQ Future

# OmicronD: Data Collection and Triggering get a Divorce

- It has been very handy to have the entire DAQ change states together: all components go through configuration, running, stopping synchronously. Life is pretty easy like this.
- However, as stated before, the typical run configuration of IceCube is pretty static – changing on timescale of months to perform periodic calibrations of IceTop. And even then the In-Ice array of DOMs ideally would not need to know about this.
- Additionally, the triggers have historically been touchy (this is actually fixed now and they seem to rock hard stable since that fix). However, one can still imagine scenarios where stellar collapse nearby (100's of parsecs) sends off enough neutrinos to cause trouble for IceCube (but not kill off humanity).

# Benefits

- The safest, most efficient means of data acquisition would be:
  - Collect data pretty much all the time and spool to HUGE buffers like hitspooling
  - Have independent processes come in, look at the data, do the trigger and event building thing totally separate from collectors and they keep state variables like run information.
  - The collectors are always collecting – no run gaps.
- In fact, this actually makes a lot of the DAQ *easier*: currently we do a lot of bookkeeping in the buffers to know when it's OK to drop old data. For sufficiently deep buffers you can just punt.
- Data collection is fairly simple task and doesn't require networking – it could be rewritten in C for example while the rest of the network aware code is Python and Java.

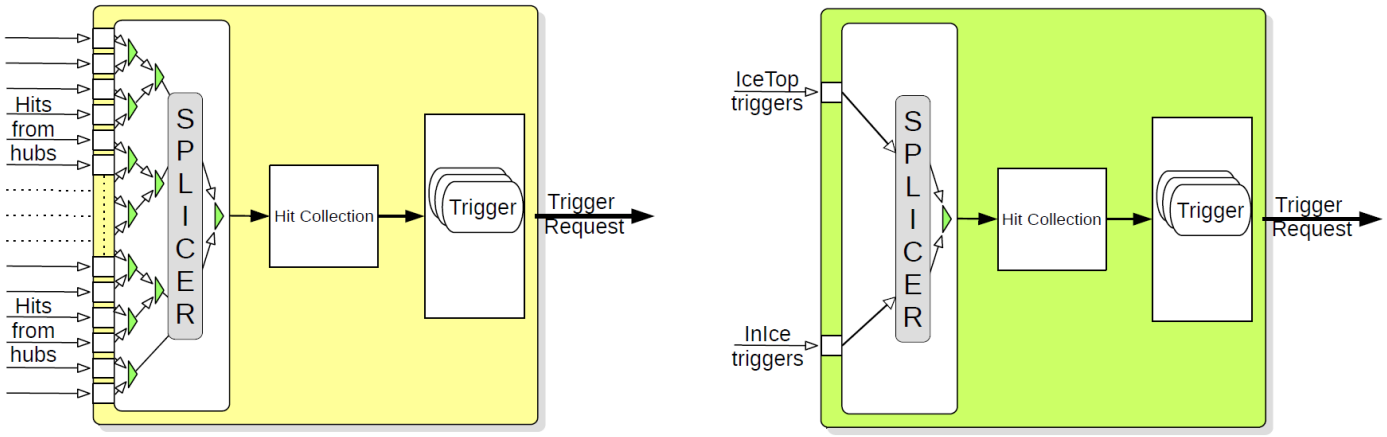
# Liabilities

- But how does one deal with configuration?  Now the concept of run is happening outside of the data collecting StringHubs.  Nevertheless they must at least infrequently be told how to configure the DOMs; a separate system must be erected.
- DOM configuration and trigger configuration no longer exist bundled – more bookkeeping here for the analyzers, in principle.

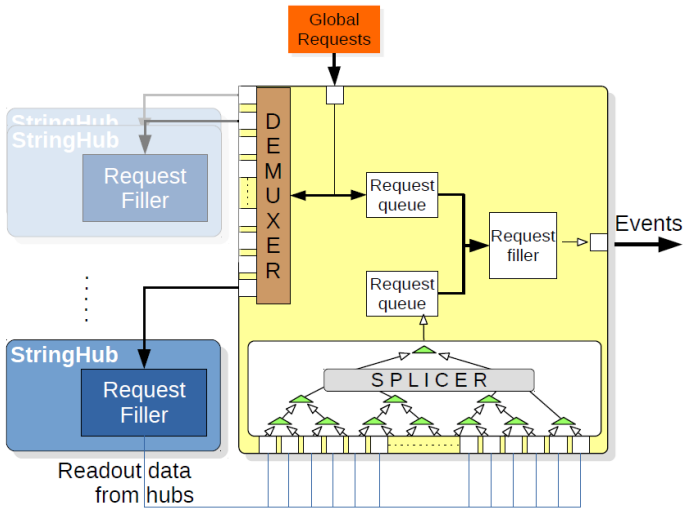
# Summary

- Described IceCube DAQ design and a bit of the online system;
- The system continues to evolve as IceCube science mission expands;
- Analyses such as HESE are being subsumed into near-real time Pole filters;
- IceCube DAQ team seeks to improve robustness of system and provide more information available to analysis.
- Major new feature, hit-spooling, is gaining traction for analyses outside of initial SN use case
- Looking to the future, the entire data collection process is foreseen to move in this direction with complete separation from triggering and event building.

# Trigger Detail



# Event Builder Detail



# Secondary Builder Detail