

# IceCube Instrumentation and Online Systems

The IceCube Collaboration

## 1 Introduction

### 1.1 IceCube Science

(1 page)

### 1.2 A Functional Description of the IceCube Instrument

In order to observe astrophysical neutrinos, the primary science goal of the experiment, IceCube exploits the fact that charged particles moving through the ice at super-luminal speed emit Cherenkov photons. An enormous detection volume is required since the cross-sections of neutrinos are small for producing secondary charged particles in interactions with ordinary matter. The glacial ice cap at the South Pole is about 3km thick and therefore predestined as operation site since it is not only offering aplenty interaction material but also a medium with unmatched high quality. Cherenkov light is produced in cascades of neutrino-induced muons penetrating the deep, other high-energy particles as well as generated by by cosmic-ray muons that result from cosmic-ray interactions in the atmosphere above Antarctica. Due to a Cherenkov photon yield of  $\mathcal{O}(10^5)$  visible photons per GeV of shower energy, the long optical attenuation length in South Pole ice and large-area photomultipliers (PMTs) it is possible to instrument cubic kilometers of ice with a rather wide spacing of detectors. The basic detection unit in IceCube in order to capture the Cherenkov light is the digital optical module or *DOM* which is covered in great detail in Sec. ???. Encapsulated in a 1/2" thick glass pressure sphere to withstand the extreme pressure in the deep ice, the main components of a DOM are a 10" PMT, embedded high-voltage generation, a flasher calibration board, and a mainboard containing the analog and digital processing circuitry for PMT pulses. The digitized data is fed to a central computing facility at the surface via a unique cable system, see Sec. ??. Aspects of detector deployment and ice drilling are covered in Sec. ??. An overview of the data flow as well as its readout, processing and filtering are subjects of Sec. ?? where we also cover the data handling, monitoring and operational performance of the observatory. The IceCube instrument consists of three sub-detectors – IceCube, DeepCore and IceTop – using the same instrumentation design of embedded digital optical modules and associated surface readout. A schematic layout of the array is shown in Fig. 1

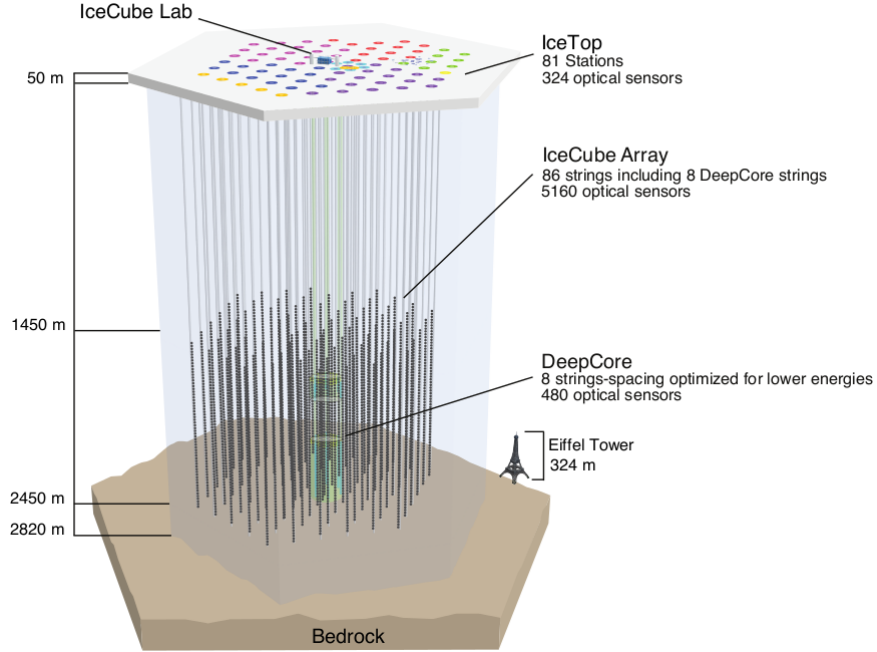


Figure 1: The IceCube Neutrino Observatory with its sub-array DeepCore and the air shower array IceTop.

### 32 1.2.1 IceCube

33 In order to detect the Cherenkov photons emitted by charged particles traversing the ice,  
 34 5160 DOMs are deployed between 1450 m and 2450 m below the glacial surface on 86  
 35 vertical strings each holding 60 DOMs deployed along a copper cable. The main *deep-*  
 36 *ice* array consists of 78 strings with a vertical separation of the DOMs on each string of  
 37 17 m. These strings are deployed on a hexagonal grid with 125 m horizontal spacing and  
 38 spanning a volume of one cubic kilometer of ice. *Talk about energy range covered by this*  
 39 *instrumentation and primary science goal*

### 40 1.2.2 DeepCore

41 The remaining subset of in-ice DOMs is deployed in the deep ice below a depth of 1750 m  
 42 forming a denser instrumented volume. This sub-array, the DeepCore [1] consists of eight  
 43 specialized and closely spaced strings of sensors located around the central IceCube string.  
 44 Its inter-string spacings of 75 m and inter-DOM spacings of 7 m are optimized for the  
 45 detection of atmospheric neutrinos with energies typically in the range from 100 GeV to  
 46 400 TeV [2].

### 47 **1.2.3 IceTop**

48 The air shower array IceTop [3] consists of 81 Cherenkov tanks filled with clear ice that  
49 are arranged in pairs on the same, approximately 125 m, triangular grid on which the  
50 in-ice array is deployed. The two tanks at each surface station are separated from each  
51 other by 10 m. Each tank contains two standard IceCube DOMs. Air showers initiated in  
52 the atmosphere by cosmic rays are typically spread over a number of stations. The light  
53 generated in the tanks by the shower particles (electrons, photons, muons and hadrons) is  
54 a measure of the energy deposit of these particles in the tanks.

## 2 The Digital Optical Module

(Chris Wendt; 10 pages)

### 2.1 A Functional Description of the DOM

The DOM (Figure ??) is the fundamental data acquisition unit for IceCube, containing a downward-facing 10" diameter photomultiplier tube (PMT) and associated circuit boards that allow near-autonomous operation. Data acquisition, control, calibration, communication and low voltage power conversion are integrated in one annular circuit board that fits around the neck of the PMT (Main Board, Section ??) [?]. Separate circuit boards generate PMT high voltage, interface to the PMT pins [?], delay PMT signals, and generate calibration light flashes that can reach other DOMs. Key requirements for the DOM include the precise recording of a wide variety of PMT pulse widths and amplitudes, robustness in a challenging deployment environment, and long term reliability.

The PMT detects signals from neutrino events ranging over energies 10 GeV–10 PeV and distances from a few meters to 500 m away. Corresponding PMT waveforms can have amplitudes from 1 mV up to and beyond the linearity limit of the PMT ( $\sim 2$  volts) and widths from 12 nsec to 1500 nsec. In order to accommodate such a variety of signals, the DOM includes multiple digitizers with overlapping dynamic range and different sampling speeds [?]. Each DOM is triggered independently by detection of individual photons, starting a recording of the PMT waveform that includes further photons arriving up to 6.4  $\mu$ sec later. The trigger time is saved along with the waveform shape, which reveals the times of arriving photons relative to this reference. The DOM typically accumulates such triggered data for a period of ? to ? sec before sending as a block.

DOMs transmit their data to computers in the IceCube Laboratory building using a twisted wire pair that also provides power. Wire pairs are bundled together to form the vertical down-hole cables and the horizontal surface cables. Each wire pair is shared between two DOMs, with data transfers initiated by a surface computer. Separately, dedicated wiring to neighbor DOMs above and below allows quick recognition of local coincidences where nearest or next-to-nearest neighbors trigger within a common 1  $\mu$ sec time window. Local coincidence triggers often have complex PMT waveforms reflecting multiple photons detected in each DOM, which are therefore saved in full detail; otherwise the DOM saves abbreviated information appropriate to single photon detection [?].

The DOM is capable of interpreting commands from the surface that specify tasks for configuration, data taking and transmission, monitoring or self-calibration. Self-calibration functions establish PMT and amplifier gains as well as sampling speed. The RAPCal system [?] is implemented for tracking each local DOM clock's offset from universal time, allowing PMT pulses that were independently recorded in many DOMs to be recombined into events by surface computers.

## 92 2.2 Components

### 93 2.2.1 Glass Sphere and Harness

94 The glass sphere housing has diameter 13" and thickness 0.5". Spheres are specified to  
95 protect the inside electronics and PMT against long term applied pressure of 250 bar (2.6 km  
96 water depth) as well as temporary overpressure up to 690 bar during refreezing of melted  
97 ice in the drill hole. They were produced by Benthos, Inc., based on a design for deep sea  
98 environments but using glass with very low potassium or other radioactive trace elements  
99 that would contribute to the dark noise count rate (Section ??). Optical transmission was  
100 measured for representative samples as 93% at 400 nm, decreasing to 50% at 340 nm and  
101 10% at 315 nm (normal incidence, excluding Fresnel reflection).

102 Each sphere is assembled from two hemispheres that mate precisely at the equator; after  
103 evacuating and backfilling with dry nitrogen, a butyl rubber sealant is applied around the  
104 seam, and covered with wide plastic tape. The interior gas pressure is set to 0.5 bar so the  
105 seal remains tight even at ambient south pole air pressure 0.6 bar.

106 The DOM is held by an aluminum band with rubber gaskets against the glass above  
107 and below the equator seam. Figure ?? shows how it is attached to the main down-hole  
108 cable via a system of steel rope and chain that carries the weight load around the DOM.  
109 The main cable bends around the DOM, and the DOM axis stays vertically aligned with  
110 the string.

### 111 2.2.2 Cable Penetrator, Cable and Connector

112 A customized penetrator assembly brings three wire pairs out through a 16.3 mm hole in  
113 the DOM glass. They are routed inside the umbilical cable, visible in Figure ??, and  
114 terminate at a pressure-tight, waterproof connector that mates with a similar connector  
115 and thus continues each pair into the main cable. One wire pair connects ultimately with a  
116 computer in the IceCube Laboratory building, carrying power and the bidirectional digital  
117 communications stream. The other two pairs lead to neighbor DOMs directly above and  
118 below (Figure ??), carrying local coincidence digital pulses that signify time correlated hits  
119 in nearby DOMs (Section ??).

120 The penetrator itself is a customized stainless steel feedthrough, with a plastic shell that  
121 is also molded onto the umbilical cable jacket. The penetrator has an o-ring seal facing the  
122 glass outside and is secured by a nut inside the sphere. External mechanical features like  
123 the penetrator are subject to large stresses during deployment and the refreezing process;  
124 the right angle bend was chosen for robustness, based on previous experience deploying  
125 AMANDA modules. The complete assembly, including umbilical cable and type XSJJ  
126 connector, was produced by SEACON (California).

### 2.2.3 PMT, Gel and Magnetic Shield

DOMs use the 10" diameter Hamamatsu R7081-02 PMT, or the corresponding high-quantum-efficiency (HQE) version for Deep Core strings. Its properties have been measured and described in [?]. It is specified by Hamamatsu for the wavelength range 300 nm–650 nm, with peak quantum efficiency around 25% (34% for HQE) near 390 nm. It features a box-and-line dynode chain with 10 stages, and is operated at gain  $10^7$  (Section 2.4.1).

The PMT bulb faces downwards in the bottom glass hemisphere, secured in high-strength silicone gel to a depth surrounding the photocathode area. The gel provides mechanical support for the whole assembly of PMT and circuit boards, as well as good optical coupling. Gel thickness between PMT envelope and glass sphere is approximately 1 cm. Originally the gel was supplied as General Electric RTV6136-D1, and later as a similar formulation from Quantum Silicones (Virginia, USA). It is optically clear with transmission 97% at 400 nm, 91% at 340 nm, and 65% at 300 nm (normal incidence). The refractive index is 1.41, yielding less than 0.1% Fresnel reflection as light passes from the sphere glass into the gel and then into the PMT envelope. The characteristics of the cured gel are specified to remain stable in the temperature range  $-70^{\circ}\text{C}$  to  $45^{\circ}\text{C}$ .

To reduce effects of the ambient magnetic field (550 mG,  $17^{\circ}$  from vertical), a mu-metal cage surrounds the PMT bulb up to the neck join. It was constructed as a wire mesh with typical wire spacing 66 mm and wire diameter 1 mm, blocking about 4% of the incident light. The resulting interior magnetic field is a factor 2.8 below the external field, pointing mostly along the axis and therefore reducing efficiency by less than 2% for this type of PMT [?].

Other interior DOM components are held in place by attachment to the PMT, mostly via screws into a molded plastic collar glued around the neck. The PMT base board is soldered directly to the pins.

### 2.2.4 Main Board and Delay Board

The Main Board and its operation has been described in detail in [?]. It interfaces to other boards as shown in Figure ?? and itself provides many key functions of the DOM:

- Control all the devices inside the DOM, including the high voltage power supply for the PMT, the flasher board, and various sensors (pressure, temperature, power supply voltage monitor). Also supply necessary DC power to the subsystems.
- Digitize the PMT waveforms, using the custom ASIC (ATWD: analog transient waveform digitizer) and a continuous sampling ADC.
- Carry out computing functions. This includes executing PMT gain calibration, compressing the digitized waveform, temporarily storing the data, creating data packets and time stamping them.

- 163 • Communicate with the data acquisition system on the surface.
- 164 • Exchange timing pulses with the surface DAQ to calibrate the internal DOM clock.
- 165 • Exchange "local coincidence" pulses with the adjacent DOMs.

166 PMT waveforms are captured by the ATWD chips with sampling period 3.3 nsec, start-  
 167 ing after a discriminator trigger is recognized by the sequencing logic. The total sampling  
 168 interval is 427 nsec, encompassing the time spread of most photons arriving from up to  
 169 100 m away. The Delay Board is connected between the discriminator and the ATWD  
 170 preamplifiers, adding a pretrigger window in this recorded waveform that allows precise  
 171 fitting of the leading edge time. It contains a  $\sim 10$  m long, 0.25 mm wide, serpentine cop-  
 172 per trace embedded in the dielectric and sandwiched between ground planes, giving a total  
 173 delay of about 75 nsec. After accounting for triggering delays, this causes the waveform  
 174 recording to start at least 10 nsec before the leading edge time.

### 175 2.2.5 HV Supply and Divider

176 The PMT high voltage subsystem consists of a resistive voltage divider circuit (PMT  
 177 Base) directly solder-mounted on the PMT and a separate high voltage control board. The  
 178 high voltage control board includes a DAC and an ADC for setting and reading out the  
 179 PMT high voltage, connected to the Main Board with a digital interface. It also holds  
 180 the high voltage generator, which is a custom encapsulated module designed by EMCO  
 181 (California). The maximum high voltage is 2047 volts, specified for up to  $30 \mu A$  current.  
 182 The set voltage is proportional to the DAC output, and the actual voltage is monitored via  
 183 a high-impedance divider and the ADC. The output ripple is less than 1 mV, and stability  
 184 is better than 200 ppm over 8 hours. Power consumption is 300 mW at full load.

185 The generator output is carried to the PMT Base Board via a high voltage coaxial  
 186 cable. This board is described in [?]. Its voltage divider presents a total resistive load  
 187 of  $130 M\Omega$ . The PMT is operated with cathode at ground potential, so the anode signal  
 188 output is AC coupled using a 1:1 bifilar wound toroid transformer mounted on the Base  
 189 Board. The transformer secondary is then wired to the Main Board analog input with a  
 190 coaxial cable.

### 191 2.2.6 Flasher Board

192 Each IceCube DOM contains a flasher board. The standard IceCube flasher board, which  
 193 is included in every DOM except the color DOMs described below, is a circular board fitted  
 194 with 12 LEDs (ETG-5UV405-30) with a peak wavelength of 399 nm; the FWHM of the  
 195 LED spectrum is 14 nm. The LEDs are arranged in pairs, evenly spaced around the board  
 196 with a  $60^\circ$  separation between each pair. One LED in each pair points outward horizontally,  
 197 the other LED is tilted upward at an angle of  $48^\circ$ , which is close to the Cherenkov angle

in ice ( $n = 1.36$ ). The angular emission profile of each LED has a FWHM of  $30^\circ$  in air, which is modeled as a Gaussian emission profile with  $\sigma = 13^\circ$ . After refraction through the DOM glass and into the ice, the value of  $\sigma$  is  $9.7^\circ$  in the polar direction and  $9.8^\circ$  in the azimuthal direction for the tilted LEDs, and  $9.2^\circ$  in the polar direction and  $10.1^\circ$  in the azimuthal direction for the horizontal LEDs.

The LEDs are controlled by a current pulse applied to each LED through a high speed MOSFET driver with a series resistor. The LEDs can be turned on individually or in any combination of the 12, using a bitmask in the flasher DAQ configuration. The photon output of the LED depends on the width and brightness, or amplitude of the driving current pulse. The current amplitude is controlled by the brightness setting in the flasher DAQ, which may have a value between 0 and 127; the maximum LED current is 240 mA. The current pulse width is controlled by the width setting in the flasher DAQ, which is twice the current pulse width in nanoseconds and may have a value between 0 and 127; the maximum current pulse width is 70 ns. The minimum stable pulse width that we can achieve is about 6 ns. The photon output is related to the brightness and width by the following empirically derived relationship:

$$N = 1.17 \times 10^{10} (0.0006753 + 0.00005593B) \left( W + 13.9 - \frac{57.5}{1 + B/34.4} \right) \quad (1)$$

where  $N$  is the number of photons,  $B$  is the brightness setting (maximum value 127) and  $W$  is the width setting (maximum value 127). The photon output per LED ranges from  $10^6$  to  $10^{10}$  photons per flash, which is equivalent to the energy output of GeV - 100 TeV cascades. The maximum LED pulse rate is 610 Hz. The current waveform is read out by ATWD digitizer channel 3 during flasher operation in order to measure the precise rise time of the flasher pulse. The flasher light output begins 8.3 ns after the current pulse is recorded. Although flashers can be operated in multiple DOMs in the same run, the DAQ does not support time-synced flashing of LEDs on different DOMs, so coincident flasher events happen only by chance.

The flasher LEDs are used for a variety of calibration purposes:

- verifying the timing response of the DOMs throughout the software
- measuring the position of the deployed DOMs in ice
- measuring the optical properties of the ice
- verifying the performance of cascade reconstruction algorithms in measuring position, direction and energy

#### *Color DOMs*

There are 16 DOMs (8 on string 79 in the center of IceCube, 8 on string 14 on the edge of IceCube) fitted with multiwavelength flasher boards, called color DOMs or cDOMs.



Table 1: Properties of the cDOM LEDs

LED	nominal $\lambda$	measured $\lambda$	$\sigma$ air	$\sigma$ DOM, polar	$\sigma$ DOM, azimuthal
UVTOP335-FW-TO39	340 nm	338 nm	51.0°	36.1°	42.9°
NS370L_5RFS	370 nm	371 nm	55.2°	39.1°	42.9°
LED450-01	450 nm	447 nm	6.8°	4.8°	5.3°
B5-433-B505	505 nm	494 nm	6.4°	4.5°	4.9°

Each cDOM includes 3 LEDs with a nominal wavelength of 505 nm, 3 LEDs with a nominal wavelength of 450 nm, 3 LEDs with a nominal wavelength of 370 nm and 3 LEDs with a nominal wavelength of 340 nm. The LEDs are arranged in pairs as on the standard flasher board, but all LEDs point outward horizontally. The arrangement of the pairs is shown in Figure XXcdom sketch from cdom wikiXX. The properties of the LEDs are given in Table 1.

## 2.3 Production and Testing

## 2.4 Calibration

### 2.4.1 DOMCal

### 2.4.2 Flasher Calibrations

## 2.5 Performance and Reliability

We have over  $N$  DOM years in ice. What can be said about the reliability? This section could be quite important.

### 2.5.1 Baseline Stability

The digitizer pedestals are set to about 10% of the maximum value in order to capture signals that go below the baseline. The baseline refers to the average value of this offset, whereas the pedestal pattern refers to the bin-to-bin variation of the pedestal setting. The pedestal pattern is subtracted off inside the DOM, but the baseline is subtracted during data processing at the surface. Since 2012, the average baseline value is set by the DAQ configuration in order to ensure stability. The average value differs for each digitizer channel in each DOM, ranging from XX to XX ADC counts. The pedestal pattern is computed at the beginning of each run by comparing 25 averaged pedestals. The autocorrelation coefficient between the pairs of averaged pedestals is computed to detect light contamination in the pedestals, and the shift between the baseline of the pairs is calculated to determine that the baseline is stable. This procedure ensures that fewer than 1 DOM in 1000 runs will contain a contaminated baseline.

258 The baselines for each digitizer channel in each DOM are monitored with beacon hits,  
 259 forced triggers which are collected at a rate of XX Hz in each DOM. The average value of  
 260 the beacon baselines are very stable, with an average stability of XX counts year to year.  
 261 The baselines are sensitive to radio frequency interference (RFI). In 2009, RF signals from  
 262 a radar transmitter broadcasting at 46.3 MHz appeared as sinusoidal or spiky signatures  
 263 in the waveform baselines. Also in 2009, a DOM 68-42 “Krabba” was damaged by a too-  
 264 high voltage setting during DOM calibration, and appeared to begin sparking, causing  
 265 sinusoidal waveforms to appear in the baselines of neighboring DOMs.

## 266 2.5.2 Optical Sensitivity Stability

267 The detector response in IceCube is calibrated with low energy muons as described in  
 268 [?]. The detector response is monitored in each run using the track detection probability  
 269 (TDP) calculated from high multiplicity muon tracks with more than 30 hits in IceCube.  
 270 The muon tracks are reconstructed using the likelihood methods described in [?]; charge  
 271 and time information from the DOM under study excluded from the reconstruction. The  
 272 TDP is defined for each DOM as the ratio of the number of detected tracks within 100 m  
 273 of the DOM to the total number of tracks within 100 m of the DOM. This ratio depends  
 274 both on the optical properties of the ice near the DOM and the optical efficiency of the  
 275 DOM. We do not attempt to separate these effects in the TDP, but rather use the TDP  
 276 to monitor the overall stability of the detector response. Figure XX shows the TDP on  
 277 string 80, which includes both standard and HQE DOMs; the TDP is 20% - 25% higher for  
 278 HQE DOMs than for neighboring standard DOMs. The TDP is stable to within 1% since  
 279 2012, when the baselines were stabilized by being set in the DAQ configuration. Figure  
 280 XX shows the difference in the TDP for all DOMs between a run in 2012 and a run in  
 281 2015.

282 The detector response stability is also measured with the *in situ* light sources in IceCube.  
 283 Both the in-ice calibration laser [?] and the flasher LEDs show less than 1% difference in  
 284 the total charge collected between 2012 and 2015.

285	<b>3</b>	<b>The Cable Systems</b>
286	<b>3.1</b>	<b>Surface Cables</b>
287	<b>3.2</b>	<b>Surface-to-DOM Cables</b>
288	<b>3.3</b>	<b>Surface Junction Boxes</b>



## 5 Online Systems

(John K; 12-15 pages)

The IceCube online systems comprise both the software and hardware at the detector site responsible for data acquisition, event selection, monitoring, and data storage and movement. As one of the goals of IceCube operations is to maximize the fraction of time the detector is sensitive to neutrino interactions (“uptime”), the online systems are modular so that failures in one particular component do not necessarily prevent the continuation of basic data acquisition. Additionally, all systems are monitored with a combination of custom-designed and industry-standard tools so that detector operators can be alerted in case of abnormal conditions.

### 5.1 Data Flow Overview

The online data flow consists of a number of steps of data reduction and selection in the progression from photon detection in the glacial ice to candidate neutrino event selection, along with associated secondary data streams and monitoring. An overview of the data flow is shown in Fig. ??.

Since the majority of photons detected by the DOMs are dark noise, a first-level *local coincidence* (LC) is formed between neighboring DOMs deployed along the same cable, using dedicated wire pairs within the in-ice cable. DOM-level triggers, or *hits*, with corresponding neighbor hits are flagged with the LC condition, while hits without the condition are compressed more aggressively. The LC time window as well as the span of neighbor DOMs up and down the cable can both be configured, with standard settings of a  $\pm 1\mu\text{s}$  coincidence window and neighbor span of 2 DOMs.

All DOM hits are read out to dedicated computers on the surface by the data acquisition system (DAQ). The next level of data selection is the formation of *triggers* by the DAQ system. LC-flagged hits across the detector are examined for temporal and in some cases spatial patterns that suggest a common causal relationship. A number of different trigger algorithms run in parallel, described in Sect. ?. All hits (both LC-flagged and non-LC hits) within a window around the trigger are combined into *events*, the fundamental output of the DAQ, and written to disk. The event rate is approximately 2.5 kHz but varies with the seasonal atmospheric muon flux, and the total DAQ data rate is approximately 1TB/day.

The DAQ also produces *secondary streams* that include time calibration, monitoring, and DOM scaler data. The scaler data, which is monitoring the noise rate of each DOM in 1.6 ms bins, is used in the supernova data acquisition system [?] to detect a global rise from many  $O(10)$  MeV neutrino interactions occurring in the ice from a Galactic core-collapse supernova. The time calibration and monitoring streams are used to monitor the health and quality of the data-taking runs.

The raw DAQ event data is then processed further with a number of *filters* in order to select a subset of events (less than 10%) to transfer over satellite to the Northern Hemi-

sphere (see Sect. ??). Each filter, typically designed to select events useful for a particular physics analysis, is run over all events using a computing cluster in the IceCube Lab. Because of limitations both on total computing power and bounds on the processing time of each event, only fast directional and energy reconstructions are used. The processing and filtering system is also responsible for applying up-to-date calibrations to the DAQ data; processed events, even those not selected by the online filters, are stored locally for archival.

A dedicated system for data movement handles the local archival storage to tape or disk, as well as the handoff of satellite data (see Sect. ??). This includes not only primary data streams but also monitoring data, calibration runs, and other data streams.

[Add experiment control paragraph]

Figure: data flow diagram indicating major subsystems to be described in this section: DAQ (incl. SNDAQ), PnF, I3Live, and SPADE/JADE.

## 5.2 SPS and SPTS

South Pole System: breakdown of computing hardware used at the pole between hubs, DAQ, PnF, other machines, and infrastructure. Internal network bandwidth. Redundancy, system monitoring (Nagios), and paging system.

Brief mention of SPTS as northern test and validation system. Replay capabilities.

## 5.3 Data Readout and Timing

### 5.3.1 Communications and Cable Bandwidth

Description of communications protocol and messaging strategy. Reference to RAPCal and how it fits in. Event compression.

### 5.3.2 Master Clock System

The GPS clock and time string fanout tree, from master clock (and hot spare), to Tier I and Tier II fanouts, the DSB card, and into the DOR card.

### 5.3.3 DOR Card and Driver

DOR card description: comms / readout, power control and measurement, RAPCal initiation, and clock string readout. Clock modes (internal / external). DOMs per card and cards per hub.

Brief description of driver. Proc file interface. Data transfer over PCI bus via DMA.

Figure: Combined clock fanout tree hierarchy and hub diagram (DSB and DOR cards, power distribution).

## 5.4 Processing at the Surface

(Dave G; 2-3 pages)

IceCube’s data acquisition system (DAQ) is a set of components running on dedicated servers in the IceCube Lab. As described in ??, physics data is read from the DOMs by the StringHub component and a minimal representation of each HLC hit is forwarded to the “local” Trigger components (either in-ice or icetop.) The “local” Trigger components run these minimal hits through a configurable set of algorithms and form windows around interesting temporal and/or spatial patterns. These time windows are collected by the “global” Trigger and used to form non-overlapping trigger requests. These trigger requests are used by the Event Builder component as templates to gather the complete hit data from each StringHub and assemble the final events.

### 5.4.1 DOMHub and Hit Spooling

The *StringHub* is responsible for periodically reading all available data from each of its connected DOMs and passing that data onto the downstream consumers. It also saves all hits to a local “hit spool” disk cache, as well as queuing them in an in-memory cache to service future requests from the *Event Builder* for full waveform data.

The StringHub component is divided into two logical pieces. The front-end *Omicron* controls all of the connected DOMs, forwarding any non-physics data to its downstream consumers and sorting the hits from all DOMs into a single time-ordered stream before passing them to the back-end *Sender*. The Sender caches SLC and HLC hits in memory, then forwards a few fields from each HLC hit to the appropriate local Trigger.

In order to avoid saturating the 2006-era network, the StringHub sends only simplified HLC hit data to the local Triggers. Fields extracted from the full HLC are the string and DOM, the time of the hit, and a few DOM trigger fields (type, mode, configuration ID), for a total of 38 bytes per hit.

After the Trigger components have determined interesting time intervals, the Event Builder will send each interval to the Sender, and it will return a list of all hits within the interval, pruning all older hits from the in-memory hit cache after each interval.

*Dave G doesn’t know the details of this*

One core assumption of the DAQ is that each component operates on a time-ordered stream of data. IceCube’s DAQ uses its *Splicer* to accomplish this. The Splicer is a common object which gathers all input streams during the setup phase; no inputs can be added once it’s started. Each stream pushes new data onto a “tail” and the Splicer collates the data from all streams into a single output stream. When a stream is closed, it pushes an end-of-stream marker onto its tail which causes the Splicer to ignore all further data.

The HKN1 (Hansen-Krasberg-1) algorithm at the core of the Splicer uses a tree of *Nodes* to sort a constant number of time-ordered data streams. Each Node contains a queue of pending data, a handle for a peer Node, and another handle for a sink Node (shared with the peer) to which sorted data is sent.

399 The tree is assembled by associating each input stream with a separate Node and adding  
400 the Node to a list. This list is then run through a loop which pulls off the first two Nodes,  
401 links them together as peers and creates a new Node which will act as the sink for both  
402 Nodes. The new Node is then added to the list. The loop exits when there is only one  
403 Node left.

404 When a new piece of data arrives, it is added to the associated Node's queue. That  
405 Node then goes into a loop where it compares the top item from its queue with the top  
406 item from the peer's queue and the older of the two items is passed onto the sink Node  
407 (which adds the item to its queue and starts its own comparison loop.) This loop stops as  
408 soon as the original Node or its peer has no more data.

409 Along with physics data, DOMs produce three additional streams of data. The scaler  
410 data, which is monitoring the noise rate of each DOM in 1.6 ms bins, is used in the su-  
411 pernova data acquisition system [?] to detect a global rise from many  $O(10)$  MeV neutrino  
412 interactions occurring in the ice from a Galactic core-collapse supernova. The time calibra-  
413 tion and monitoring streams are used to monitor the health and quality of the data-taking  
414 runs.

415 As hits move from the front end to the back end, they are written to the Hit Spool, a  
416 disk-based cache of files containing hits. These files are written in a circular order so that  
417 the newest hits overwrite the oldest data. The first hit time for each file is stored in an  
418 SQLite database to aid in fast retrieval of raw hit data.

419 One problem with the current DAQ design is that it only reads data when the DAQ  
420 is running, so the detector is essentially "off" during hardware failures or the periodic full  
421 restarts of the system. There is a plan to split the StringHub into several independent  
422 pieces to eliminate these blind spots. The front end (Omicron) piece will be moved to  
423 an always-on daemon which continuously writes data (including secondary, non-physics  
424 data and other metadata) to the disk cache. Part of the back end (Sender) piece will  
425 become a simple Hit Spool client which reads data from the disk cache and sends it to the  
426 downstream consumers, while another simple component will listen for time intervals from  
427 the Event Builder and send back lists of hits taken from the Hit Spool.

#### 428 5.4.2 Hitspool Request System

429 Subsystems such as SnDAQ and HESE can send time interval requests to a HitSpool  
430 Request daemon. This daemon passed the request onto every hub, where another daemon  
431 gathers the hits in that interval from each hub and forwards them to the final "sender"  
432 daemon which bundles them up to be transferred to the North where the full waveforms  
433 can be used for further analysis.

#### 434 5.4.3 Supernova System

435 SN secondary stream from DOMs and SNDAQ reference.



#### 436 5.4.4 Triggers

437 General description of trigger architecture. Separation of trigger window and readout  
438 window. How trigger windows depend on geometry. Thorough description of all different  
439 trigger algorithms. Trigger and readout window merging.

440 Table: standard settings for triggers

441 Figure: trigger windows and readout windows.

442 Figure: example bright multi-trigger event.

443 Figure: SLOP triplet geometry?

#### 444 5.4.5 Event Building

445 When the Event Builder receives a request from the Global Trigger, it extracts the indi-  
446 vidual readout windows and sends them to the appropriate subset of the hubs. The hubs  
447 each send back a list of all hits within the window. When all hubs have returned a list of  
448 hits, the event is assembled.

449 Extraneous data is stripped from the trigger and hit payloads and bundled into an  
450 event payload.

451 Events are written to a temporary file. When the temporary file reaches a configurable  
452 size, it is renamed to a standard unique name (made up of a “physics\_” prefix, the run  
453 number, a sequentially increasing file number, and the starting and ending times contained  
454 within the file). When the PnF system sees a file with the agreed-upon name, it accepts it  
455 for processing (and filtering).

#### 456 5.4.6 Configuration

457 Configuration of the DAQ is managed by two sets of files; a cluster configuration file and  
458 a hierarchical tree of run configuration files.

459 The cluster configuration file is used to launch the detector components. It specifies  
460 component hosts, startup paths, and other options needed to start each component on  
461 the appropriate host in the cluster. This allows components (other than StringHub) to be  
462 easily moved to different hosts for troubleshooting, load balancing, maintenance, etc.

463 The run configuration file specifies settings which may vary between runs. It includes  
464 the name of a trigger configuration file as well as the names of DOM configuration files for  
465 all hubs included in a run.

466 The trigger configuration file specifies configuration parameters for all trigger compo-  
467 nents (in-ice, icetop, and global). These include the list of algorithms run by each com-  
468 ponent, along with readout window sizes and any other variable parameters (frequency,  
469 threshold, etc.)

470 The DOM configuration file holds a list of all DOMs which a hub should read from for  
471 the run. Every configuration parameter for each DOM is specified.

472 The cluster configuration file may be modified but a note should be sent to the “log-  
473 book” email list. Run configuration files are treated as static once they’ve been used for  
474 data taking. If a configuration file needs to be altered, the modified file is given a new name  
475 (often with an incremented “-V###” suffix indicating the version number) to ensure that  
476 the exact detector configuration is always discoverable.

#### 477 5.4.7 Distributed Network Control

478 The central coordination point for the DAQ is a “command-and-control” daemon named  
479 CnCServer. It manages and monitors components, and acts as the main external interface  
480 for the DAQ.

481 CnCServer uses a standard component interface to query and control the components,  
482 and a separate interface for components to expose internal data used for monitoring the  
483 health of the detector and tracking down bugs and/or performance problems.

484 CnCServer starts out knowing nothing about the detector components. During the  
485 DAQ’s launch phase, components are given the host and port used by CnCServer. When  
486 components start up, each one sends CnCServer its name, host/port pairs, and the types  
487 of inputs and outputs it expects, and CnCServer adds them to its internal list.

488 To start a run, CnCServer is given the name of the run configuration. That run  
489 configuration may include all components or only a subset. Using that file, CnCServer  
490 builds a list of components required for the run. Each of the included components is told  
491 to connect to their downstream neighbor, then to use the run configuration to initialize  
492 as appropriate (configure DOM hardware, initialize trigger algorithms, etc.). Once all  
493 components are successfully configured CnCServer instructs them to start, working its way  
494 from the Event Builder back to the String Hubs.

495 When a run is in progress, CnCServer regularly checks that components are still active  
496 and that data is flowing between components. If it detects a problem, it stops the run and  
497 relaunches all the components. CnCServer also periodically collects all components’ mon-  
498 itoring data and writes it to monitoring files which can be used for post-mortem diagnosis  
499 of detector failures.

### 500 5.5 Online Filtering

501 (*Erik B.; 3-4 pages*)

#### 502 5.5.1 Overview

503 The online processing and filtering system is charged with the immediate handling of all  
504 triggered events collected by the data acquisition system. This treatment includes appli-  
505 cation of calibration constants, application of event characterization and filtering software,  
506 extracting data quality monitoring information, generation of realtime alerts for events of  
507 astrophysical interest and creation of data files and metadata information for long term

508 cataloging. The online processing and filtering system is a custom software suite that uti-  
509 lizes a computer cluster of  $\sim 20$  standard servers located in SPS computing cluster at the  
510 experimental site at South Pole, Antarctica. The online processing and filtering system has  
511 been in operation since the start of operation of the 22 string configuration of IceCube in  
512 2007.

513 In IceCube, each triggered event consists of a collection of digitized waveforms recorded  
514 by the digital optical modules (ref daq-dom paper). To be useful for physics, each of these  
515 waveforms requires application of calibration constants that allow the waveform units to  
516 be converted from the raw units (ADC counts per sample bin) to more physical units (mV  
517 measured in each fixed time,  $\sim$ ns bin). These calibration constants are independently  
518 measured (domcal ref?) and stored in an online database for use by the online processing  
519 and filtering system. Next, each DOM's waveform is deconvolved using the known DOM  
520 response to photons to extract the light arrival time and amplitude information. This series  
521 of time and amplitude light arrival information for each DOM is the base until for event  
522 reconstruction and characterization. The online processing and filtering system encodes  
523 this information for each DOM in a compact data format known as the SuperDST, and  
524 occupies  $\sim 5\%$  of the file size of the full waveform information. Any DOM readout whose  
525 SuperDST information is found not to a good representation of the original waveform, or  
526 sees very high amounts of light also have the full waveform readout saved in addition to  
527 the SuperDST.

528 Each event is then characterized with a series of event reconstruction algorithms that  
529 attempt to match the observed patterns of recorded light in the SuperDST with known pat-  
530 terns of light from track and showering event hypotheses (ref reco papers? e-reco paper?).  
531 These characterizations (location, direction, and energy) and their overall goodness-of-fit  
532 produced by these reconstructions are used to select interesting events by a filter selection.  
533 The filter criteria are set by the IceCube collaboration for each season and are tuned to  
534 select events of interest to specific analyses. Each season there are about 2 dozen unique  
535 filter selections in operation. Some of these filters trigger are designed to search for events  
536 of wide astrophysical interest to the scientific community and trigger alerts that are dis-  
537 tributed to followup observatories worldwide.

538 The online processing and filtering system also extracts and aggregates data quality  
539 and monitoring information from the data as is it processed. This information includes  
540 stability and quality information from the DOM waveform and calibration process, rates  
541 of DOM readouts, and rates and stability information for all detector triggers and filters.  
542 This information is aggregated for each data segment and reported to the IceCube Live  
543 monitoring system.

544 Finally the online processing and filtering system writes several data files that make up  
545 the long-term data catalog of the IceCube experiment. These include:

- 546 • *Filtered data files* These files contain only events selected by the online filter se-  
547 lections. These events generally only include the SuperDST version of the DOM

548 information and results from the online event reconstructions. These files are queued  
549 for transmission to the IceCube data warehouse by the data handling system using  
550 the TDRS satellites.

551 • *SuperDST data files* These files contain the SuperDST version of DOM readout in-  
552 formation for all triggered events as well as summary information from the online  
553 filtering process. This file set is intended as the long-term archive version of IceCube  
554 data.

555 • *Raw data files* These files contain all uncalibrated waveforms from all DOMs for every  
556 event. This large data set is saved until final data quality assurance on the SuperDST  
557 sample can be completed.

558 (Q: include some information on file catalog sizes per day?)

## 559 5.5.2 System Design

560 The online processing and filtering system uses a modular design, where each module is  
561 responsible for a portion of data processing built around a central master server node  
562 and a scalable number of processing client nodes. The central master server focuses on  
563 data distribution and aggregation tasks (requesting data blocks from the DAQ, collating  
564 event monitoring information and writing data files), while the client process focus on the  
565 per-event processing tasks (event calibration, reconstruction, analysis, and filtering).

566 The system is built upon the IceCube analysis software framework, IceTray (ref?),  
567 allowing standard IceCube algorithms to be used in the online processing and filtering  
568 system without modifications. Additionally, the system uses the Common Object Request  
569 Broker Architecture (CORBA) system as means for interconnecting the modular portions  
570 of the system. Specialized classes are used to provide CORBA interconnections within the  
571 IceTray system, allowing file-like interfaces that let data to stream from one component  
572 to another using native IceTray formats. Use of a CORBA name server and a dynamic  
573 architecture allow for the addition and remove of filtering clients as needed to meet the  
574 processing load from annual filtering changes and overall rate variations from seasonal  
575 variations in the detector trigger rate.

## 576 5.5.3 Components

577 The flow of triggered event data in the online processing and filtering system is shown in  
578 Figure 2 highlighting the flow of data from the DAQ system, through the master server  
579 and clients, to files on disk and online alerts. Several standard components in the online  
580 processing and filtering system include:

581 • *I3DAQDispatch* is a process to pickup event data from the data acquisition system  
582 data cache and forward to the PFServer components.

- 583 • *PFServers* are central data flow managers within the online processing and filtering  
584 system. These servers receive data from the I3DAQDispatch event source, distribute  
585 events to and record results returning from the PFClient farm, and send filtered  
586 events to file writer, online monitoring and alert components. Typically there are 4  
587 servers used in operation.
  - 588 • *PFClient* is the core calibration, reconstruction and filtering process that is applied  
589 to each triggered event. In normal operation,  $\sim 400$  of these clients operate in parallel  
590 to filter events in real time.
  - 591 • *GCDDispatch* is a DB caching system to prevent the 400 PFClient processes from  
592 overwhelming the DB system when requesting calibration information. This system  
593 aggregates DB requests, makes a single database query and shares the results with  
594 all PFClients.
  - 595 • *PFWriters* are responsible for creation of files and meta-data for the IceCube data  
596 catalog. These files are written in standard IceTray file format. There is one writer  
597 component for each file type created.
  - 598 • *PFOnlineWriter* is responsible for extracting alert information from the data and  
599 forwarding these alerts in real time to the IceCube Live system.
  - 600 • *PFMoniWriter* is responsible for aggregating per-event monitoring information, cre-  
601 ating histograms and forwarding them to the IceCube Live monitoring system.
  - 602 • *PFFiltDispatch* and FollowUp clients are responsible for looking for bursts of neutrino  
603 events on timescales from 100 seconds up to 3 weeks in duration. Any significant burst  
604 of neutrinos found generates alerts sent to partner observatories worldwide.
- 605 (Q: remove discussion of GFU, OFU, and PFFiltDispatch since this is being moved  
606 north?)

#### 607 5.5.4 Performance

608 The online processing and filtering system is designed to filter triggered events as quickly  
609 as possible after collection by the data acquisition system. A key metric is processing  
610 system latency, defined as the duration of time between the data acquisition trigger and  
611 the completion of event processing and filtering. A typical latency history for the system  
612 is shown in Figure 3, showing typical system latencies of  $\sim 20$  seconds.

613 (Q: include other system performance information? Performance bottlenecks?)

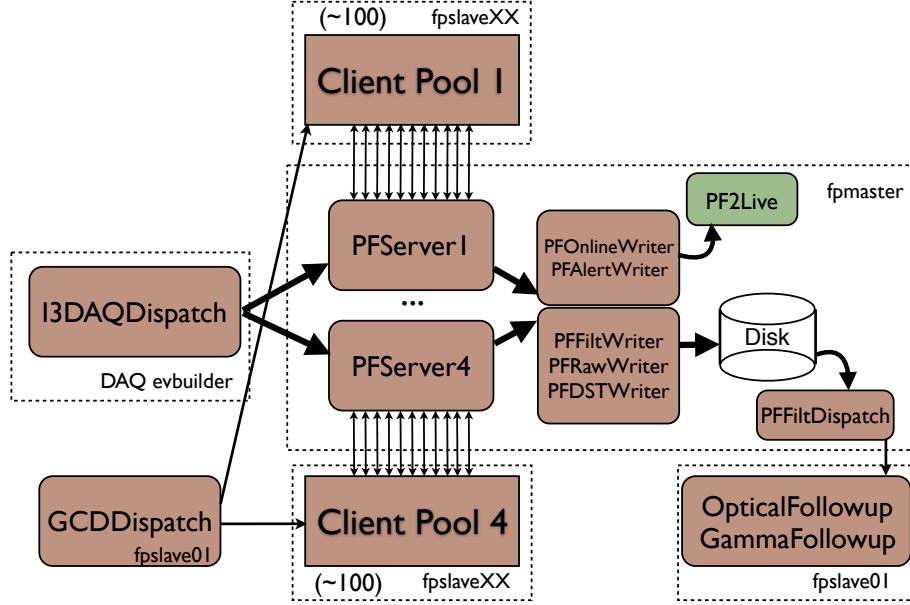


Figure 2: Internal components of the Online Processing and Filtering System. Arrows highlight the flow of data within the system.

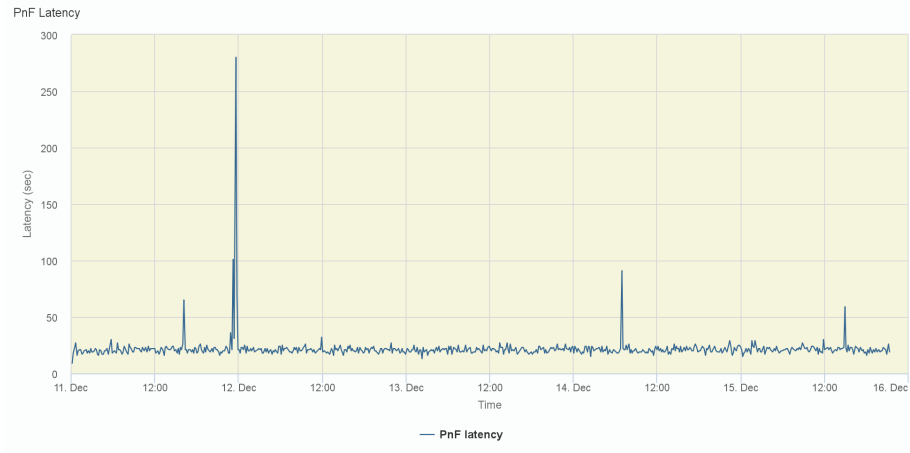


Figure 3: Typical Online Processing and Filtering System latency for a several day period. The latency defined as the time between DAQ event time and time when the online filtering processing is complete. The spikes in latency correspond to DAQ run transitions.

## 5.6 Data Handling

(P. Meade; 1 page)

(J. Braun, background and additional description of communications modes)

The bulk of South Pole Station data traffic is handled by geosynchronous satellite links. Due to the unfavorable location at the South Pole, only geosynchronous satellites with steeply inclined orbits reach far enough above the horizon to establish a link. For a given satellite, this link provides four to six hours of communications once per sidereal day. Multiple geosynchronous satellites are currently utilized by USAP, providing a  $\sim 12$ -hour window of connectivity with bandwidth of 1 Mbps or higher. For the remainder of the day, Iridium satellites allow limited voice and data connectivity and provide up to 2.4 kbit/s of bandwidth per connection.

IceCube incorporates Iridium modems into two separate systems. The IceCube Teleport System (ITS) uses the Iridium short burst data mode to send short messages of 1.8 kB or smaller with a typical latency of 30 seconds. Messages may both originate or terminate at the ITS Iridium modem at the South Pole. Messages also contain a recipient ID indicating the intended host to receive the message, allowing a many-to-many communications infrastructure between systems running at the South Pole and systems in the Northern Hemisphere. The IceCube Messaging System (I3MS) incorporates multiple Iridium modems and uses the Iridium RUDICS data mode, providing a 2.4 kbit/s bidirectional serial stream per modem and a minimum latency of  $\sim 1.5$  seconds. I3MS runs as a daemon on both ends of the link, accepts messages via ZeroMQ, and transports those messages across the link based on message priority and fair sharing of bandwidth among all users. I3MS message recipients listen for messages using ZeroMQ PUB-SUB, allowing a given message to be sent to multiple recipients.

*Generation description of system architecture. Stream definitions, dropboxes, and data pickup. Archival vs. transfer to TDRSS system.*

Data handling is provided by three servers named jade02, jade03, and jade04. The jade servers operate independently of one another and each of them are capable of handling the nominal data volume by itself. Having three servers allows for data handling to continue seamlessly in case of hardware failure or maintenance.

Each server runs a copy of the Java Archival and Data Exchange (JADE) software (stylized “jade”). As its name implies, the jade software is written in the Java programming language. It is a reimplement and expansion of earlier prototype software called South Pole Archival and Data Exchange (SPADE), written by Cindy Mackenzie. The jade software has four primary tasks: consumption, archival, satellite transmission, and real-time transmission.

The jade software is configured with a number of data streams, which consist of a data server, a dropbox directory, and a filename pattern. The data stream dropbox directories are checked on a regular basis for new files. A file pairing scheme (binary and semaphore) prevents files from being consumed before they are finished being produced. For each file, a checksum calculated on the data server is compared to a checksum calculated on the jade server. This method ensures that the file was copied without error. After this, the original data file is removed from the data host.

657 After consumption, files are routed according to the configuration of their data stream.  
658 Files that are too large to send via the satellite link are archived to a configurable number  
659 of archival media copies. The prototype SPADE software archived to LTO tapes, while  
660 the later jade software archives to large (2+ TB) hard disk drives. All of the archival data  
661 is buffered on the jade server until the archival media is complete. In case of failure while  
662 creating the archival media, all of the files can be immediately written to fresh archival  
663 media with a single command.

664 Files that are too large to send via the real-time link, but small enough to send via the  
665 satellite link are queued for satellite transmission. The jade software attempts to bundle  
666 multiple files together into 1 GB bundle archives to allow satellite link operators to manage  
667 the daily data transmission. Very large files ( $>1$  GB) are split apart into multiple 1 GB  
668 bundles for the same reason. The jade software will only transfer a configurable number  
669 of bundles to the satellite relay server. If satellite transmission is not possible, the jade  
670 software will buffer the excess bundles on the jade server, to avoid flooding the relay server  
671 unnecessarily.

672 Small files ( $<50$  KB) with high priority status information are sent via the real-time link.  
673 The real-time link is provided by the IceCube Messaging Service (I3MS). The jade software  
674 uses JeroMQ, a pure Java implementation of the ZeroMQ (ZMQ) protocol, to connect to  
675 I3MS. In cases where the real-time link is not available, I3MS will queue the messages to  
676 be sent when the link becomes available. All I3MS messages are also sent to jade to send  
677 via the satellite link to ensure delivery if the real-time link should be unavailable for an  
678 extended period of time.

## 679 5.7 IceCube-Live and Remote Monitoring

680 IceCube operations are controlled and monitored centrally by IceCube-Live, a suite of  
681 high-level software implemented mostly in the Python language. IceCube-Live consists  
682 of two major components: LiveControl, responsible for controlling data-taking operations  
683 and collecting monitoring data, and the IceCube-Live website, responsible for processing  
684 and storing monitoring data as well as presenting this data in webpages and plots that  
685 characterize the state of the IceCube detector.

### 686 5.7.1 LiveControl

687 LiveControl executes in the background as a daemon and accepts user input via XML-  
688 RPC. Operators typically enter commands and check the basic detector status using a  
689 command-line interface. LiveControl is responsible for controlling the state of DAQ and  
690 online-processing, starting and stopping data-taking runs, and recording the parameters  
691 of these runs. Standard operation is to request a run start, supplying a configuration  
692 file specifying the DOMs to include in data taking. LiveControl then records the run  
693 number, configuration, start time, etc. and sends a request for DAQ to begin data taking.



After data taking commences successfully, LiveControl waits a specified amount of time, generally eight hours, then stops the current run and automatically starts a new run using the same configuration. This cycle continues until stopped by a user request or a run fails. In case of failure, LiveControl attempts to restart data taking by starting a new run. Occasionally a hardware failure occurs, and it is impossible to start a run with the supplied configuration because requested DOMs are unpowered or temporarily unable to communicate with the IceCube DAQ. In this case, LiveControl cycles through predefined partial-detector configurations in an attempt to exclude problematic DOMs. This results in taking data with half of the IceCube strings or fewer, but it greatly reduces the chance of a prolonged complete outage where no IceCube data is recorded.

A secondary function of LiveControl is the collection, processing, and forwarding of monitoring data from DAQ, online-processing, and other components. Monitoring data consists of a JSON dictionary with a well-defined format including a creation time, sender name, priority, data name, and either JSON data or a single integer or floating-point value. This data is forwarded to LiveControl using ZeroMQ and queued internally for processing. A few monitoring quantities indicate serious problems with the detector, e.g. the online-processing latency is too high. LiveControl provides a database of checked monitoring values indexed by service and data name and raises an alert if the value is out of the specified range or hasn't been received in a specified amount of time. The alert usually includes an email to parties responsible for the affected subsystem and, for serious problems, triggers an automated page to winterover operators. Several other types of monitoring data trigger a response by LiveControl. These include alerts generated internally by subsystems, and such alerts may trigger emails and pages from LiveControl. All monitoring data are forwarded to the IceCube-Live website for further processing and display.

### 5.7.2 IceCube-Live Website

Priority	Transport System	Daily Messages	Daily Data	Typical Latency
1	ITS (Iridium)	10,000	1 MB	1 minute
2	I3MS (Iridium)	150,000	5 MB	1–5 minutes
3	JADE (Geosynchronous)	300,000	100 MB	1 day

Table 2: Statistics for IceCube monitoring messages

Two operational copies of the IceCube-Live website exist: one inside the IceCube network at the South Pole, and one in the Northern Hemisphere. Monitoring data reaches the northern website based on priority and using both geosynchronous and Iridium data transport, summarized in table 2.

Messages reaching the website are processed by the DBServer daemon and inserted into one of several database tables depending on content. Messages also may contain directives requesting DBServer to send email, by specifying email recipients and content, or requesting

726 that the monitoring message be published using ZeroMQ PUB-SUB, allowing the message  
727 to be passed to an external process. The IceCube-Live website itself uses the Django  
728 framework and contains pages that create sophisticated views of monitoring data stored in  
729 the database. These pages include a front page displaying active alerts and plots of event  
730 rates and processing latencies from the previous few hours, and a page for each run that  
731 displays start time, stop time, and other essential data. The run page contains low-level  
732 diagnostic data that includes e.g. charge histograms, digitizer baselines, and occupancy  
733 for each DOM, and is used to diagnose problems with detector components that occurred  
734 during the run and to determine if the run can be used in physics analysis.

735 Finally, the IceCube-Live website in the Northern Hemisphere transmits messages to  
736 LiveControl using ITS and I3MS. This capability is used to retransmit messages sent using  
737 the popular Slack chat service to the South Pole, allowing the IceCube winterover operators  
738 to chat with experts in the Northern Hemisphere during periods with no geosynchronous  
739 satellite connectivity. This connection also provides a limited capability to control the  
740 detector, allowing operators in the north to e.g. remotely issue HitSpool requests.

## 741 5.8 Operational Performance

742 (*John K.; 1 page*)

743 Explanation of how design choices, system monitoring, and winterovers result in high  
744 uptime. Discussion of median downtime and various causes of downtime. Possible basic  
745 failure analysis of hardware components.

746 Figure: DAQ full uptime and clean uptime percentage.

747 Figure (optional): Downtime histogram.

## 6 Outlook

Discuss Gen2 here.

## References

- [1] R. Abbasi, Y. Abdou, T. Abu-Zayyad, M. Ackermann, J. Adams, J. Aguilar, M. Ahlers, M. Allen, D. Altmann, K. Andeen, et al. The design and performance of IceCube DeepCore. *Astroparticle physics*, 35(10):615–624, 2012.
- [2] R. Abbasi, Y. Abdou, T. Abu-Zayyad, J. Adams, J. Aguilar, M. Ahlers, K. Andeen, J. Auffenberg, X. Bai, M. Baker, et al. Measurement of the atmospheric neutrino energy spectrum from 100 GeV to 400 TeV with IceCube. *Physical Review D*, 83(1):012001, 2011.
- [3] R. Abbasi, Y. Abdou, M. Ackermann, J. Adams, J. Aguilar, M. Ahlers, D. Altmann, K. Andeen, J. Auffenberg, X. Bai, et al. IceTop: The surface component of IceCube. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 700:188–220, 2013.