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 3 km 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

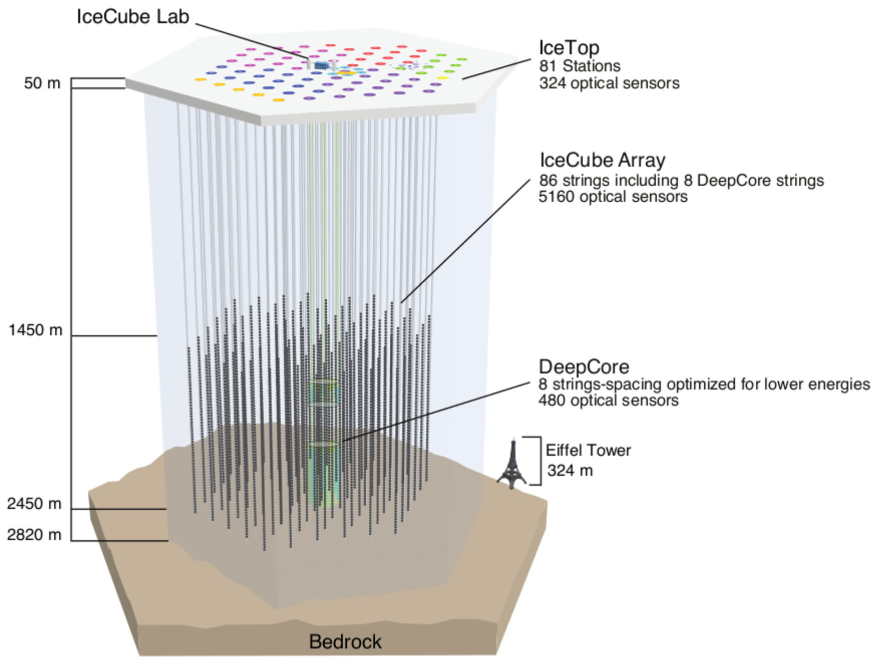

Figure 1: The IceCube Neutrino Observatory with its sub-array DeepCore and the air shower array IceTop.

1.2.1 IceCube

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

1.2.2 DeepCore

The remaining subset of in-ice DOMs is deployed in the deep ice below a depth of 1750 m forming a denser instrumented volume. This sub-array, the DeepCore [5] consists of eight specialized and closely spaced strings of sensors located around the central IceCube string. Its inter-string spacings of 75 m and inter-DOM spacings of 7 m are optimized for the detection of atmospheric neutrinos with energies typically in the range from 100 GeV to 400 TeV [4].

1.2.3 IceTop

The air shower array IceTop [6] consists of 81 Cherenkov tanks filled with clear ice that are arranged in pairs on the same, approximately 125 m, triangular grid onw hich the in-ice array is deployed. The two tanks at each surface station are separated from each other by 10 m. Each tank contains two standard IceCube DOMs. Air showers initiated in the atmosphere by cosmic rays are typically spread over a number of stations. The light generated in the tanks by the shower particles (electrons, photons, muons and hadrons) is 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 **??**) [2]. Separate circuit boards generate PMT high voltage, interface to the PMT pins [3], 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 [2]. 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 μsec 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 μsec 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 [2].

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 [2] 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.

2.2 Components

2.2.1 Glass Sphere and Harness

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

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

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

2.2.2 Cable Penetrator, Cable and Connector

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

The penetrator itself is a customized stainless steel feedthrough, with a plastic shell that is also molded onto the umbilical cable jacket. The penetrator has an o-ring seal facing the glass outside and is secured by a nut inside the sphere. External mechanical features like the penetrator are subject to large stresses during deployment and the refreezing process; the right angle bend was chosen for robustness, based on previous experience deploying AMANDA modules. The complete assembly, including umbilical cable and type XSJJ 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 [3]. 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°C to 45°C.

To reduce effects of the ambient magnetic field (550 mG, 17° 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 [2]. 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.

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

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

2.2.5 HV Supply and Divider

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

The generator output is carried to the PMT Base Board via a high voltage coaxial cable. This board is described in [3]. Its voltage divider presents a total resistive load of 130 MΩ. The PMT is operated with cathode at ground potential, so the anode signal output is AC coupled using a 1:1 bifilar wound toroid transformer mounted on the Base Board. The transformer secondary is then wired to the Main Board analog input with a coaxial cable.

2.2.6 Flasher Board

Each IceCube DOM contains a flasher board. The standard IceCube flasher board, which is included in every DOM except the color DOMs described below, is a circular board fitted with 12 LEDs (ETG-5UV405-30) with a peak wavelength of 399 nm; the FWHM of the LED spectrum is 14 nm. The LEDs are arranged in pairs, evenly spaced around the board with a 60° separation between each pair. One LED in each pair points outward horizontally, the other LED is tilted upward at an angle of 48°, which is close to the Cherenkov angle in ice (n = 1.36). The angular emission profile of each LED has a FWHM of 30° in air, which is modeled as a Gaussian emission profile with $\sigma = 13°$. After refraction through the DOM glass and into the ice, the value of σ is 9.7° in the polar direction and 9.8° in the azimuthal direction for the tilted LEDs, and 9.2° in the polar direction and 10.1° 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:

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

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 λ	measured λ	σ air	σ DOM, polar	σ 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 coefficent 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.

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

2.5.2 Optical Sensitivity Stability

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

The detector response stability is also measured with the *in situ* light sources in IceCube. Both the in-ice calibration laser [7] and the flasher LEDs show less than 1% difference in the total charge collected between 2012 and 2015.

3 The Cable Systems

3.1 Surface Cables

3.2 Surface-to-DOM Cables

3.3 Surface Junction Boxes

4 Drilling and Deployment

4.1 Geometry Calibration

The geometry of the detector is determined using drill and survey data during deployment (stage 1), and then corrected and refined using the LED flashers in ice (stage 2).

4.1.1 Stage 1 Geometry Calibration

The (x,y) coordinates of the string are calculated using the position of the drill tower. Before deployment, when the drill tower is in position, at least three of the tower corners are surveyed from at least one control point. The coordinates for the center of the hole in the tower floor are calculated from the corner coordinates. If x-y drift vs. depth has been calculated for the hole from drill data, the drifts are added to the (x,y) coordinates of individual DOMs, assuming that the string follows the hole center.

The depth of the lowest DOM on the string is calculated using pressure readings from the Paro/Keller pressure sensor, converted to depth by correcting for the compressibility of the water in the hole and the ambient (air) pressure measured before the pressure sensor hits the water. The distance from the tower floor to the water surface is measured with a laser ranger. The vertical DOM spacings are also measured with a laser ranger.

4.1.2 Stage 2 Geometry Calibration

The LED flashers are used to correct the relative depths of the strings. This correction is typically less than 1 m relative to the stage 1 data, but can be as large as 20 m (larger than the 17 m DOM spacing) in cases where the pressure sensor fails during string deployment before it takes the final depth reading. The correction is calculated by finding the leading edge of the time distribution of the light recorded by the receiving DOM, denoted t_0. The distance corresponding to the leading edge time is $d = c_{ice} \cdot t_0$, and the distances for all receiving DOMs are plotted as a function of the vertical distance between the flasher and the receiver, $z' = z_{receiver} - z_{flasher}$. The resulting plot described a hyperbola, $d = \sqrt{D^2 + (z' - \Delta z)^2}$, where D is the horizontal distance between the flasher string and the receiver string, calculated from stage 1 data, and Δz is the relative offset between the depths of the flashing and receiving string. The hyperbola fit is done simultaneously for each flashing string and all surrounding receiving strings in order to calculate the relative offsets, which are then applied to the z coordinate of all DOMs on the string.

Trilateration

Flasher corrections to the (x,y) coordinates of some DOMs in the center of the DeepCore subarray were calculated using the trilateration method. In this analysis, the 5 DOMs closest to the flasher on each of the three closest strings surrounding the flasher are selected, and a circle of radius $r = \sqrt{(d)^2 - (\Delta z)^2)}$ is drawn around each receiving DOM, where d is the distance between the DOM and the flasher calculated from the leading edge time of the received light, and Δz is the relative depth of the flashing and receiving DOMs calculated from the method described above. The intersection points of all the circles are calculated, and the (x,y) position of the flashing DOM is taken to be the average of the centroid of the intersection points. The error bars on the positions are 1σ from a Gaussian fit to all centroid values; the x and y corrdinates are fitted independently. The shifts relative to the deployment data are found to be less than 1 m, and agree with the drill head coordinates within the error bars.

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 $\pm 1 \mu s$ 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 atmopheric 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 occuring 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 Hemisphere (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.

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

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

Along with physics data, DOMs produce three additional streams of 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 occuring 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.

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

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

5.4.2 Hitspool Request System

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

5.4.3 Supernova System

SN secondary stream from DOMs and SNDAQ reference.

5.4.4 Triggers

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

Table: standard settings for triggers

Figure: trigger windows and readout windows.

Figure: example bright multi-trigger event.

Figure: SLOP triplet geometry?

5.4.5 Event Building

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

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

Events are written to a temporary file. When the temporary file reaches a configurable size, it is renamed to a standard unique name (made up of a "physics_" prefix, the run number, a sequentially increasing file number, and the starting and ending times contained within the file). When the PnF system sees a file with the agreed-upon name, it accepts it for processing (and filtering).

5.4.6 Configuration

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

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

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

The trigger configuration file specifies configuration parameters for all trigger components (in-ice, icetop, and global). These include the list of algorithms run by each component, along with readout window sizes and any other variable parameters (frequency, threshold, etc.)

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

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

5.4.7 Distributed Network Control

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

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

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

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

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

5.5 Online Filtering

(Erik B.; 3-4 pages)

5.5.1 Overview

The online processing and filtering system is charged with the immediate handling of all triggered events collected by the data acquisition system. This treatment includes application of calibration constants, application of event characterization and filtering software, extracting data quality monitoring information, generation of realtime alerts for events of astrophysical interest and creation of data files and metadata information for long term cataloging. The online processing and filtering system is a custom software suite that utilizes a computer cluster of ~20 standard servers located in SPS computing cluster at the experimental site at South Pole, Antarctica.The online processing and filtering system has been in operation since the start of operation of the 22 string configuration of IceCube in 2007.

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

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

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

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

- *Filtered data files* These files contain only events selected by the online filter selections. These events generally only include the SuperDST version of the DOM information and results from the online event reconstructions. These files are queued for transmission to the IceCube data warehouse by the data handling system using the TDRS satellites.

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

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

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

5.5.2 System Design

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

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

5.5.3 Components

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

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

- *PFServers* are central data flow managers within the online processing and filtering system. These servers receive data from the I3DAQDispatch event source, distribute events to and record results returning from the PFClient farm, and send filtered events to file writer, online monitoring and alert components. Typically there are 4 servers used in operation.

- *PFClient* is the core calibration, reconstruction and filtering process that is applied to each triggered event. In normal operation, ~ 400 of these clients operate in parallel to filter events in real time.

- *GCDDispatch* is a DB caching system to prevent the 400 PFClient processes from overwhelming the DB system when requesting calibration information. This system aggregates DB requests, makes a single database query and shares the results with all PFClients.

- *PFWriters* are responsible for creation of files and meta-data for the IceCube data catalog. These files are written in standard IceTray file format. There is one writer component for each file type created.

- *PFOnlineWriter* is responsible for extracting alert information from the data and forwarding these alerts in real time to the IceCube Live system.

- *PFMoniWriter* is responsible for aggregating per-event monitoring information, creating histograms and forwarding them to the IceCube Live monitoring system.

- *PFFiltDispatch* and FollowUp clients are responsible for looking for bursts of neutrino events on timescales from 100 seconds up to 3 weeks in duration. Any significant burst of neutrinos found generates alerts sent to partner observatories worldwide.

(Q: remove discussion of GFU, OFU, and PFFiltDispatch since this is being moved north?)

5.5.4 Performance

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

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

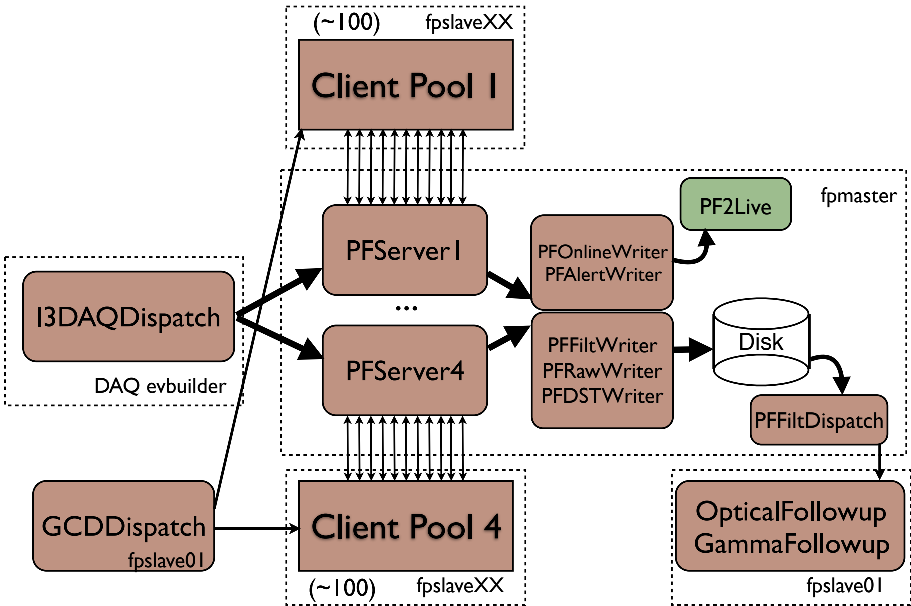

Figure 2: Internal components of the Online Processing and Filtering System. Arrows highlight the flow of data within the system.

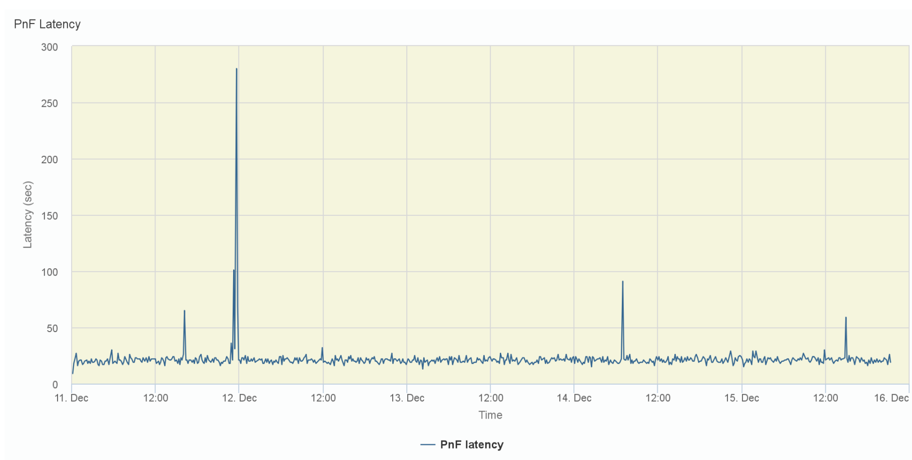

Figure 3: Typical Online Processing and Filtering System latency for a several day period. The latency defined as the time between DAQ event time and time when the online filtering processing is complete. The spikes in latency correspond to DAQ run transitions.

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 intendend 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 reimplementation 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.

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

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

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

5.7 IceCube-Live and Remote Monitoring

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

5.7.1 LiveControl

LiveControl executes in the background as a daemon and accepts user input via XML-RPC. Operators typically enter commands and check the basic detector status using a command-line interface. LiveControl is responsible for controlling the state of DAQ and online-processing, starting and stopping data-taking runs, and recording the parameters of these runs. Stardard operation is to request a run start, supplying a configuration file specifying the DOMs to include in data taking. LiveControl then records the run 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-definied 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 that the monitoring message be published using ZeroMQ PUB-SUB, allowing the message to be passed to an external process. The IceCube-Live website itself uses the Django framework and contains pages that create sophisticated views of monitoring data stored in the database. These pages include a front page displaying active alerts and plots of event rates and processing latencies from the previous few hours, and a page for each run that displays start time, stop time, and other essential data. The run page contains low-level diagnostic data that includes e.g. charge histograms, digitizer baselines, and occupancy for each DOM, and is used to diagnose problems with detector components that occurred during the run and to determine if the run can be used in physics analysis.

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

5.8 Operational Performance

(John K.; 1 page)

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

Figure: DAQ full uptime and clean uptime percentage.

Figure (optional): Downtime histogram.

6 Outlook

Discuss Gen2 here.

References

[1] Aartsen et al. Energy Reconstruction Methods in the IceCube Neutrino Telescope. *JINST*, 9:P03009, 2014.

[2] Abbasi et al. The IceCube data acquisition system: Signal capture, digitization, and timestamping. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 601(3):294–316, 2009.

[3] Abbasi et al. Calibration and characterization of the IceCube photomultiplier tube. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 618(1):139–152, 2010.

[4] Abbasi et al. Measurement of the atmospheric neutrino energy spectrum from 100 GeV to 400 TeV with IceCube. *Physical Review D*, 83(1):012001, 2011.

[5] Abbasi et al. The design and performance of IceCube DeepCore. *Astroparticle Physics*, 35(10):615–624, 2012.

[6] Abbasi 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.

[7] Kiryluk et al. IceCube standard candle refernce: ICRC proceedings 2007.