

AN OFFLINE TEST OF THE ICECUBE “FIRE DRILL” SYSTEM

Spencer Griswold

May 12, 2021

Department of Physics and Astronomy, University of Rochester, NY

Motivation

- ▶ The next galactic CCSN offers the opportunity to make a groundbreaking multimessenger measurement
- ▶ *Exercising operational readiness ensures we are ready to fully seize that opportunity should it arise*

The IceCube “Fire Drill” system

1. A simulated CCSN signal was injected on top of existing data
2. Combined data was added to IceCube’s data pipeline upstream of the main DAQ on IceCube’s offline test system
3. A supernova trigger was formed with significance $> 10\sigma$

THE OFFLINE “FIRE DRILL” METHOD

1. Simulation

- For each module in the detector, lists of hit times are generated

2. Injection

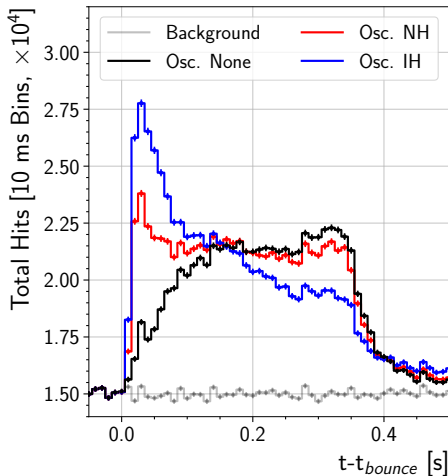
- Injected hits are re-formatted based on existing hits
- Existing hits are updated against injected hits

3. Processing

- Auxiliary files are generated
- Validation is performed

4. Evaluation

- A CCSN trigger is formed using the IceCube supernova data acquisition (SNDQA) system



Total DOM hits from CCSN, assuming different ν mass hierarchies. Model from Nakazato et al. 2013. Simulated using ASTERIA

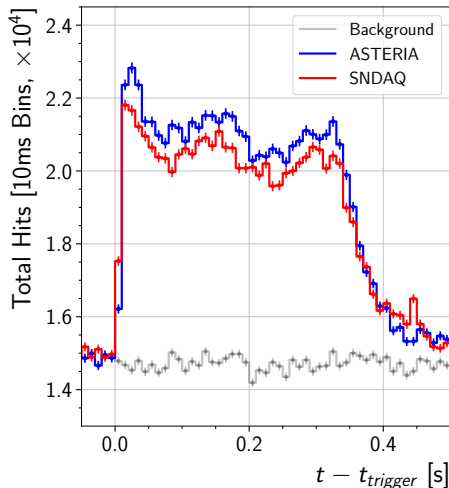
RESULTS OF THE OFFLINE TEST

Trigger Formation

- ▶ SNDAQ formed a $> 10\sigma$ trigger
 - The SNEWS alert threshold is 8.4σ

Lessons Learned

1. Errors with injection & processing
 - Some injected hits are missing
2. Data buffering request was poorly formed
3. Alerts from offline system were sent to defunct mailing list
4. We can use this to send alerts to SNEWS from the Offline system



Comparison between simulated hits (ASTERIA) and hits observed post-trigger formation (SNDAQ)

Current Goal: Perform an Online “Fire Drill” test

- ▶ Would exercise automated and manual elements of SN alert escalation
 - (Auto) Formation of trigger
 - (Auto) Distribution of notifications
 - (Auto) Formation of data buffering request
 - (Manual) Detector operators confirm alert and back-up data
 - (Manual) Working groups confirm alert

Current Work

- ▶ The offline “Fire Drill” method is obtrusive to regular data flow
 - Interrupting the regular data flow would be harmful to other IceCube efforts
- ▶ We are modifying SNDAQ to accommodate signal injection in a non-obtrusive way

The first “Fire Drill” serves as a proof of principle, and has already identified issues in our alert escalation.

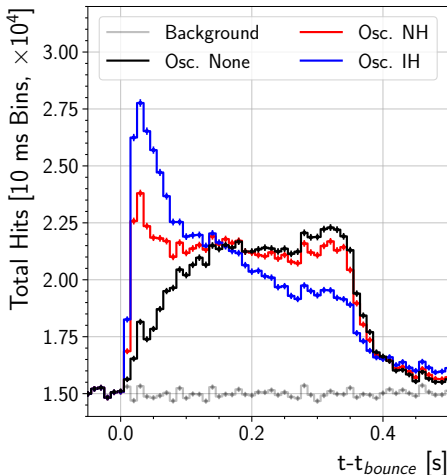
THANK YOU FOR YOUR ATTENTION

QUESTIONS?

CCSN DETECTION IN ICECUBE

SNDAQ: online system searching for correlated rate increases in the DOMs

- ▶ CCSN produce $\mathcal{O}(10 \text{ MeV})$ ν 's
 - Individually, too dim to trigger SMT8
 - Collectively, produces correlated rise in rate in all DOMs
- ▶ Numerous tasks follow formation of a SN trigger, categorized as...
 - Automated notifications
 - Manual checks
(Human makes a decision)
- ▶ *Each component of alert escalation must be tested to ensure readiness*
 - IE Super-K monthly data challenges



Total DOM hits from CCSN, assuming different ν mass hierarchies. Model from Nakazato et al. 2013

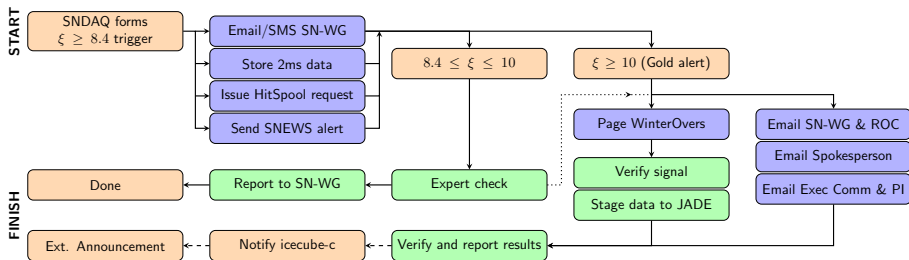
CCSN ALERT ESCALATION

▶ Automated Tasks

- Verify that correct notifications are sent to the correct places

▶ “Human” Tasks

- Ensure people know what to do in the case of high significance alerts
- **Expert check** might be a weak link
 - SN-WG is small and SNDAQ alert verbosity is high



Simplified escalation scheme – some elements have been omitted for clarity

OFFLINE SIGNAL INJECTION ON SPTS

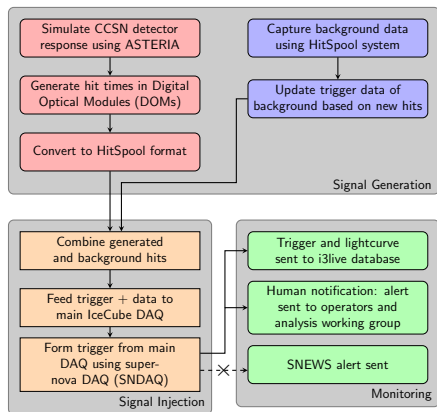
Goal: Replay a pDAQ run and test SNDAQ with an injected CCSN signal

Motivation:

- ▶ Tests SNDAQ's trigger formation
- ▶ Notifications are easy to control
- ▶ Provides a useful toolset for offline testing

Steps:

1. Simulate hits from CCSN
2. Inject new hits into HitSpool files
3. Validate and stage new files
4. Run SNDAQ during pDAQ run



Flowchart outlining method for offline Firedrill

PREPARING TO INJECT HITS

1. Generate SN signal using ASTERIA

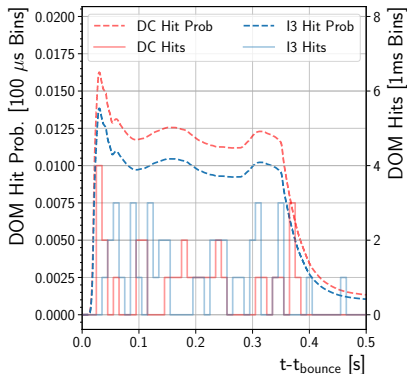
- DOM hits with sub-ns timestamps
- Later, hits are converted to SN scalars

2. Acquire DOM launch times from HitSpool files

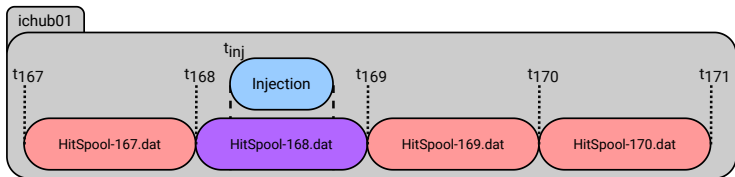
3. Choose an injection start time

4. Flag files which will contain injected hits

- HitSpool files cover ~ 15 s
- Injected signal lasts ~ 10 s

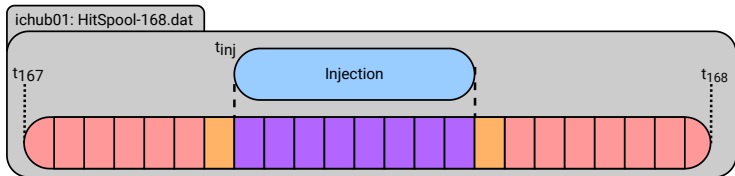


DOM Hits binned at 1ms for illustrative purposes



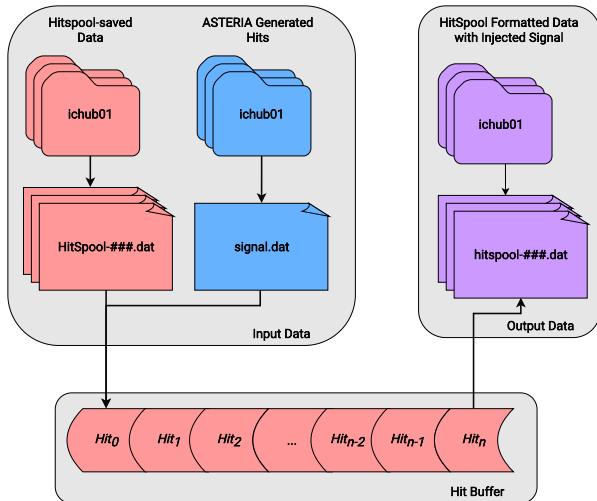
PROCESSING INJECTED HITS

- ▶ Adding new hits requires updating the old hits' LC flag
1. Identify parts of HitSpool files that will contain injected hits
 - ~ 10 s window plus $10\mu\text{s}$ borders
 - Local coincidence occurs within $1\mu\text{s}$
 2. Copy unmodified hits at front of file
 3. Load HitSpool and Injected hits into buffer
 4. Update HitSpool Hits' LC flags against injected hits
 5. Copy unmodified hits at end of file



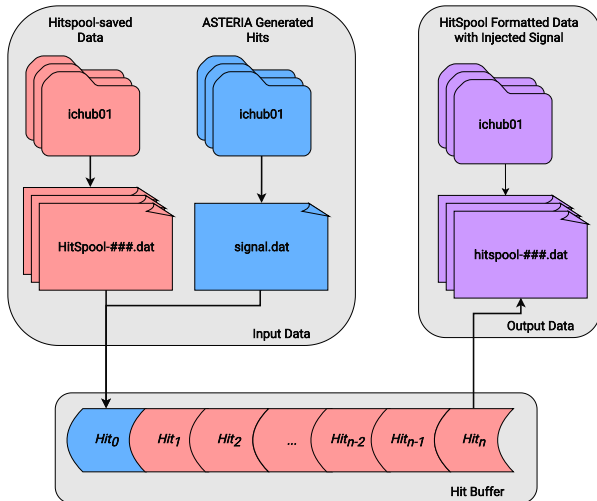
PROCESSING INJECTED HITS

- ▶ The raw HitSpool-saved data and the list of ASTERIA-generated hits are both loaded into a buffer according to UTC timestamp



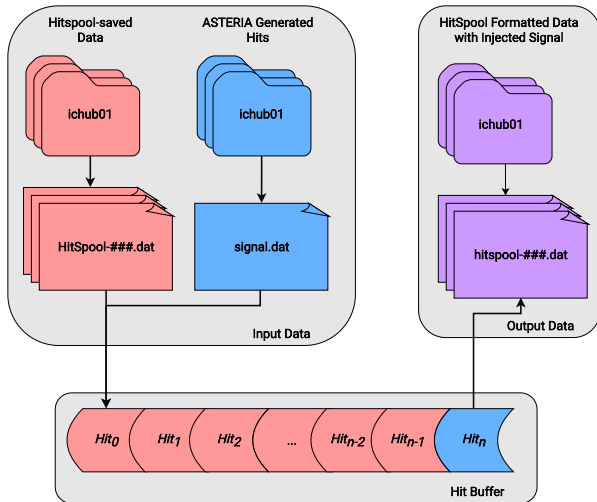
PROCESSING INJECTED HITS

- ▶ When ASTERIA-generated hits are added to the buffer, all other hits in the buffer are updated. New LC flags could be activated.



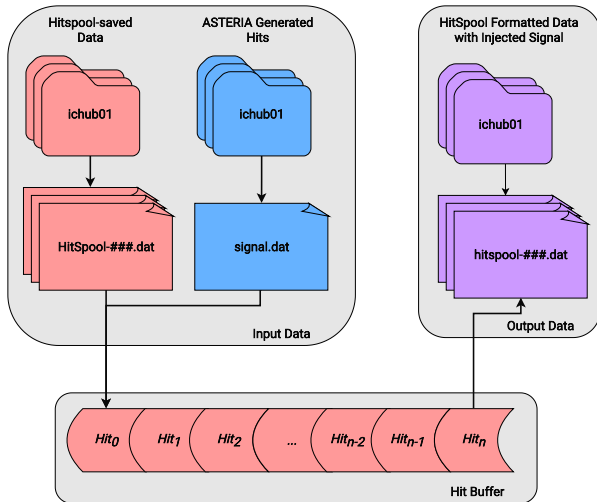
PROCESSING INJECTED HITS

- ▶ When an ASTERIA hit is removed from the back of the buffer, all other hits in the buffer are updated.



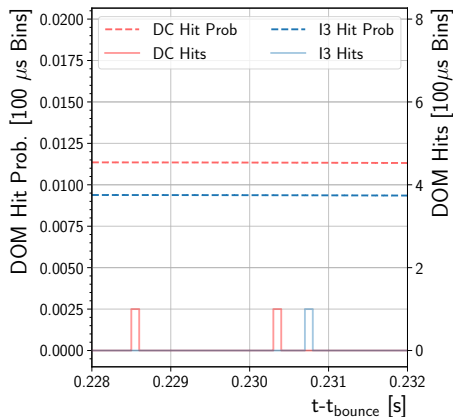
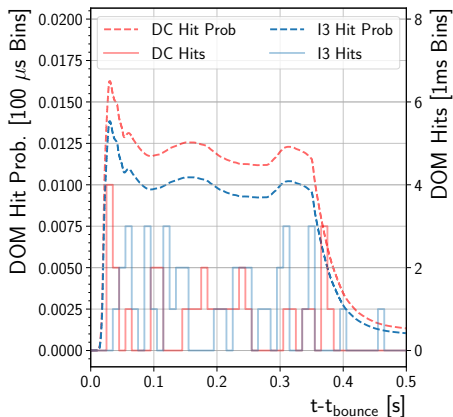
PROCESSING INJECTED HITS

- Once all ASTERIA hits have passed through the buffer, it is flushed and the remainder of the HitSpool file is copied



GENERATING THE INJECTION HITS

- ▶ ASTERIA computes photonic energy deposition per volume from CCSN ν
- ▶ This may be scaled against effective volume to produce the expected number of hits a DOM measures in a time bin.



GENERATING THE INJECTION HITS

- ▶ sub-ns precision hits are obtained by sampling a uniform distribution between bin edges. The bin's count determines the number of samples.
- ▶ Each hit is recorded as a timestamp in 10^{-10} s paired with the DOM ID

