Snowmass2021 - Letter of Interest

IceCube and IceCube-Gen2 Long Term Preservation

Thematic Areas: (check all that apply □/■)
□ (CompF1) Experimental Algorithm Parallelization
□ (CompF2) Theoretical Calculations and Simulation
□ (CompF3) Machine Learning
■ (CompF4) Storage and processing resource access (Facility and Infrastructure R&D)
■ (CompF5) End user analysis
□ (CompF6) Quantum computing
■ (CompF7) Reinterpretation and long-term preservation of data and code

Contact Information:
Paolo Desiati (University of Wisconsin–Madison) [desiati@icecube.wisc.edu],

Authors (alphabetical):
Paolo Desiati (University of Wisconsin–Madison) [desiati@icecube.wisc.edu],
Alex Olivas (University of Maryland) [aolivas@umd.edu],
Benedikt Riedel (University of Wisconsin–Madison) [briedel@icecube.wisc.edu],
David Schultz (University of Wisconsin–Madison) [dschultz@icecube.wisc.edu],
on behalf of the IceCube[1] and IceCube-Gen2[2] Collaboration [analysis@icecube.wisc.edu]

Abstract: (must fit on this page)
The IceCube Neutrino Observatory is a cubic kilometer neutrino detector deployed at the South Pole, focused on detecting GeV-EeV energy neutrinos. IceCube measures neutrinos by detecting the optical Cherenkov photons produced in neutrino-nucleon interactions. While data preservation has received community attention and is fairly well served in IceCube, software and analysis preservation are newer topics that we are only starting to acknowledge. IceCube is also working through the challenges of releasing the entire software chain as open source. This presents issues for an international organization, with contributions funded by agencies from different countries with differing licensing policies that are likely common to other large scientific collaborations.

[1] Full author list available at https://icecube.wisc.edu/collaboration/authors/snowmass21_icecube
[2] Full author list available at https://icecube.wisc.edu/collaboration/authors/snowmass21_icecube-gen2

The IceCube Neutrino Observatory [1], located at the South Pole, instruments a cubic kilometer of Antarctic ice. IceCube uses 5160 digital optical modules (DOMs) arranged on 86 strings in a hexagonal array to detect Cherenkov radiation from relativistic charged particles emitted during neutrino interactions. This configuration can detect neutrinos as high as EeV energies while the central densely instrumented region of DOMs, called DeepCore, is optimized to extend the detection energy down to a few GeV. The IceCube Upgrade plans to improve on the resolution of GeV neutrinos by adding an additional seven strings concentrated around DeepCore with varying DOM designs and additional calibration devices [2]. To improve resolution of TeV to EeV neutrino detection and increase the detection rate, IceCube-Gen2 will add 120 strings to instrument a total volume of 7.9 km^3 [3].

IceCube data analyses span a wide variety of scientific topics, from 10 GeV to EeV energy scales. Experimental data volumes are not insignificant, and even larger amounts of Monte Carlo simulation are essential for developing analysis methods for the identification of signal from background, for testing the performance of reconstruction algorithms, and for determining the background contamination of data analysis samples. Future extensions to lower and higher energies will significantly increase the data volume need for simulations, making it imperative to develop a scalable long term data, software, and analysis preservation plan.

Data Preservation

IceCube currently produces over 300 TB/year of raw experimental data, which are stored on hard disk and shipped once a year to the data center at the University of Wisconsin–Madison. Raw data are also filtered in real time at the South Pole and reduced to a 36 TB/year stream that is transferred north daily via bandwidth- and time-limited satellite links. Filtered data are then processed through subsequent levels of increasingly resource-intensive reconstruction algorithms down to science-level data.

IceCube has developed software to handle data movement from the South Pole to the central data warehouse at the University of Wisconsin–Madison and archival sites at NERSC and DESY-Zeuthen (Germany). For the satellite data transfers, the service makes use of the Iridium satellite systems for high-priority, low-volume data, e.g. realtime neutrino alerts, and the dedicated high-capacity TDRSS satellite system for the bulk of the filtered data. The unfiltered data stream is stored on two different physical media at the South Pole and shipped to the UW–Madison data center once a year during the austral summer.

To ensure the integrity of all data, the software maintains checksums of all files. If any files have not been transferred successfully from the South Pole, the software will re-transmit it. The data will not be removed from the South Pole until data integrity has been assured, i.e. the checksums of data arriving in the data warehouse or being stored on disk at the South Pole match the initial checksum. This system runs on several servers to achieve higher reliability and scalability.

More recently, we have developed software that automatically replicates the raw and filtered data from the UW–Madison deta center to their respective archives. Raw data is read off the hard disks shipped from pole, bundled into large archives, and transferred to the magnetic tape archival system at NERSC. The filtered data is archived on magnetic tape at DESY-Zeuthen, and has a mirror on hard disk to allow "local" access in Europe. Challenges involved with archiving data include differing requirements and access patterns at sites in Europe and the US, especially at NERSC where the tape system is not externally accessible for high-bandwidth transfers.

Software Preservation

IceCube's data processing and simulation software is preserved in version control. Historically this is `subversion`, with a server run by the Univesity of Maryland IceCube group. In the near

future, IceCube will be switching to `git` and using a cloud-hosting option, such as GitHub or GitLab. Analysis software is also usually in version control, though until recently there was no strict requirement. In recent years there has been more effort put into this area, and new analyses are required to be in version control, with a tagged or frozen version. Because of this, analysers have been early adopters of `git` for their software needs. While IceCube has decided on git as a near-future version control system, the makeup of the collaboration makes the standard GitFlow development model sub-optimal. The IceCube software group will be adopting a modified GitFlow, where every developer will retain commit privileges to the main development branch. Whether to branch and create a pull request (PR) will be left at the discretion of the developer. It was deemed that a PR and subsequent code review would place an undue burden on the core software group, leading to delays and friction between multiple development tracks.

Because IceCube relies almost entirely on open source or internally created software, proprietary licenses are not an issue, though releasing core code as open source remains a challenge due to licensing issues. However, keeping old software running can be a challenge if the project has been abandoned. This is an increasingly important and unrecognised issue, as IceCube has now entered its second decade of operation. Several strategies have been proposed, including building and running old software in virtual machines or container environments. When upgrading computing systems in the past, we have typically kept a physical system and/or virtual machine running the old system configuration available for a year or more, to complete certain analyses. The next upgrade cycle looks poised to use containers for this purpose, allowing a smoother transition of computing systems.

While it is technically feasible to preserve software inside a container, this does not appear to us as a sustainable option. If any updates need to be applied, the container will need to be rebuilt—a challenging prospect for a very old container with an OS and dependencies that are no longer supported. Additionally, the long-term viability of container image file formats is still unclear. We feel it is better to try to keep the code building on newer developer environments, fixing things as needed when upgrading the OS, compiler, and dependencies.

Analysis Preservation

Multi-messenger astronomy is a rapidly growing extension of astronomy involving the coordinated collection and interpretation of observations from a variety of experiments on electromagnetic emissions, gravitational waves, neutrinos and cosmic rays. The observation of cosmic objects through different messenger signals provides the unique opportunity to unveil the mystery of energetic emissions in the Universe, including the origin of the cosmic rays. Efficient access to data, fast processing, and the capability of pivoting analysis strategies based on rapidly growing multi-messenger network evidences, are now basic requirements for any observatory. It is necessary to provide a scalable infrastructure for reproducing analysis workflows on new data and for re-analyzing historical data.

IceCube has developed an Internal Data Repository which comprises the full documentation of final-level event data files, of release-controlled software used to perform event reconstructions and selections, and of the analysis workflow. The goal is to guarantee that published analyses are internally and exactly reproducible as long as they are not superseded by completely new strategies. The documentation includes experimental and simulation data, along with key quantities such as effective areas (i.e. response function) and likelihood functions. High level data analyses tend to evolve quickly while pursuing more efficient event ID classification (i.e. for neutrino flavor and rare background events) with state-of-the art reconstruction and machine-learning algorithms. On the other hand, background rejection has stabilized around a common strategy.

References

[1] M. G. Aartsen et al. The IceCube Neutrino Observatory: Instrumentation and Online Systems. *JINST*, 12(03):P03012, 2017.

[2] Aya Ishihara. The IceCube Upgrade – Design and Science Goals. *PoS*, ICRC2019:1031, 2020.

[3] M.G. Aartsen et al. IceCube-Gen2: The Window to the Extreme Universe. 8 2020.