Snowmass2021 - Letter of Interest

IceCube and IceCube-Gen2 Storage and Processing Resources

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:
Benedikt Riedel (University of Wisconsin–Madison) [briedel@icecube.wisc.edu],

Authors (alphabetical):
Steve Barnet (University of Wisconsin–Madison) [barnet@icecube.wisc.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:
The IceCube Neutrino Observatory is a km^3 neutrino detector deployed at the South Pole. IceCube measures neutrinos by detecting the optical Cherenkov photons produced in neutrino-nucleon interactions. IceCube computing has evolved significantly from a collaboration-owned and operated computing model to heterogeneous and globally distributed computing resources, including Graphical Processing Units (GPUs). We believe that this shift towards a highly distributed computing pool will continue. One of the largest changes we foresee is the integration of high-performance computing facilities with limited network connectivity and the transition of collaboration-owned and operated resources to interactive Machine Learning/Artificial Intelligence-focused resources. The data analysis workflow itself will also continue to evolve from SSH-based access to resources to web-based workloads. The future data management and storage systems will review the use of POSIX-focused distributed filesystems vs. other storage systems, reduce the ”active” datasets to allow for more efficient storage, and explore novel ways to distribute the data to both data processing and users. Computing at the South Pole will undergo an overall simplification to allow for a more accessible workflow and reduce the load on operators.

[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] is designed to measure neutrinos from GeV to EeV energies. IceCube does this by deploying a hexagonal grid of Digital Optical Modules (DOMs) along 86 vertical strings deep in the glacier at the South Pole. Each DOM contains a photomultiplier tube and detects the Cherenkov radiation from relativistic charged particles emitted during neutrino interactions.

Compute resources

Over the past decade, IceCube has transitioned from using predominately collaboration-owned and operated computing resources to distributed computing resources across the collaboration and at national scales, using resources such as the Open Science Grid (OSG) and NSF's Extreme Science and Engineering Discovery Environment (XSEDE). This was driven by the need to accommodate the broadening of IceCube's science capabilities, including the addition of GPUs to the Monte Carlo workflow to model the optical properties of the ice.

Over the the next five to ten years, we forsee a continuation of this trend. A single institution can no longer host the computing needs of IceCube, in particular the GPU computing needs. Large-scale batch computing needs, e.g. Monte Carlo simulation or data processing, will be satisfied using distributed resources. The collaboration-owned computing resources will become interactive resources used for data exploration and Machine Learning/Artificial Intelligence (ML/AI) applications. This split is needed because of the iterative process needed to train ML/AI and the increasing demand for GPUs from Monte Carlo production, data processing, and analysers. External resources are much better suited for IceCube's GPU-based mass processing than interactive ML/AI training workloads. Additionally, mass processing lowers the load on external providers' networks compared to ML/AI, as training requires vast amounts of Monte Carlo simulation.

To support the increasingly distributed nature of the resources, IceCube will continue to rely on `HTCondor` [2] and our overlay `pyglidein` [3] to aggregate resources in the short-term. Over the long-term, the increasing adoption of containerized applications within scientific computing may alter how resources are aggregated, accessed, and utilized. This is particularly useful for (ML\AI) applications as they are designed to be deployed via containers. Adoption of container orchestration frameworks for resource aggregation depends on the support for distributed resource management, containers, and external access at high-performance computing (HPC) facilities.

Workflow

Interactive workflows will continue to evolve from "traditional" SSH-based access to `HTTP`-based access to resources, such as `JupyterLab`. Users in IceCube are already using self-managed `JupyterLab` instances on traditional computing clusters. Centralizing this effort is needed to allow for better support of containerized workloads, quality of service, and to facilitate dynamically scaling the interactive resource pool with demand. The last point in particular is important for delivering better support for ML/AI workloads that may take significant resources, e.g. several GPUs.

Access to new GPU-focused HPC resources will be of particular interest to IceCube in the coming years, particularly when considering the need for resources to simulate IceCube Upgrade and IceCube-Gen2. The HPC resources that IceCube currently utilizes predominately allow users to access the resources remotely. We have been able to utilize a handful of resources that do not allow remote access to resources through IceCube's workflow management system. Deploying IceCube's workflow management at an individual site is a significant effort.

In the past year, we have been exploring the use of cloud resources for IceCube, in particular for "bursty" applications, i.e. workflows that need a significant amount of compute time for relatively short periods [4, 5]. An application that has benefited is the the directional and energy reconstruction for our multi- messenger astrophysics alerts. While IceCube issues a preliminary

alert with a directional and energy reconstruction result at the time of detection, we perform an additional, more precise reconstruction once the data arrives in the Northern Hemisphere. This second reconstruction readily scales with the number of available cores. The on-demand availability of O(10,000) cores in the cloud allows us to reduce the time to result significantly. To distribute data to the individual instances, we employed `Apache Pulsar`, a pub-sub message system. The success of `Apache Pulsar` in this analysis has lead us to explore message passing systems as an alternative to the current workflow and data management systems.

The continually growing IceCube data set is becoming unwieldy for individual researchers to understand in depth. We saw this after a reprocessing campaign that ran the same processing over several years of data. This made it significantly easier for researchers to use several years of data for their results. We believe that having a central metadata catalog will similarly allow users to group and explore data for easily and reduce strain on the workflow and storage system.

Storage

Data management and storage systems will undergo significant changes over the coming years as we deploy new storage systems for detector data and Monte Carlo simulation to keep pace with growth. We are currently assessing whether to transition to an object store-based model, continue using a "traditional" distributed filesystem, such as Lustre or dCache, or adopt a hybrid model.

A majority of IceCube's data is write-once-read-many times and will require access by an increasing proportion of remote computing resources. The built-in remote data access and self-healing ability of currently available object storage systems make them significantly more attractive than current distributed filesystems. The main obstacle we face is that users are accustomed to the POSIX-like interface that current filesystems offer. We are discussing and planning possible software frameworks that will employ metadata to let researchers "query" for the data that they desire rather than search through the data set directly.

We are also exploring methods to reduce the size of the "active" dataset, i.e. available for interactive use. A large fraction of IceCube's data is needed only occasionally by a few data analysis projects. Keeping the majority of that data in a tape archive or other cold storage to be retrieved as needed, will allow us to improve the overall quality of service for data that is frequently used.

Major expansions of IceCube such as IceCube Upgrade and Gen2 will put additional demands on the data system. While the overall data rate is estimated to increase at a modest 10-20% for each addition to IceCube, the complexity of the data will increase significantly. The additional information gained from the extensions, in particular from the IceCube Upgrade's calibration devices, will necessitate a number of data reprocessing campaigns. These will require not just a massive amount of computing resources ($\sim$30M CPU-hours per reprocessing campaign), but also retrieving 3-4 PB of data from tape archive and distributing it across IceCube resources for processing.

South Pole Computing

Processing power and network connectivity to the instrument will remain limited. We are in the process of determining how much more data we can transfer via satellite to be able to change the data filtering and processing at the South Pole. The current plan is to simplify the South Pole processing step significantly by shifting expensive processing steps to facilities in the northern hemisphere. This is to ensure sufficient processing overhead for the IceCube Upgrade and allow for a more maintainable online system. We are also exploring how edge computing resources could be utilized for data processing and filtering at the South Pole.

An increased demand for scientific computing at the South Pole is expected in the coming years with the construction and operation of CMB-S4. IceCube and IceCube-Gen2 will explore how to collaborate with CMB-S4 on cyberinfrastructure at the South Pole.

References

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

[2] HTCondor Team. HTCondor 8.6.12, August 2018.

[3] D. Schultz et al. Wipacrepo/pyglidein: 1.1.9, October 2018.

[4] Igor Sfiligoi, Frank Würthwein, Benedikt Riedel, and David Schultz. Running a pre-exascale, geographically distributed, multi-cloud scientific simulation. In Ponnuswamy Sadayappan, Bradford L. Chamberlain, Guido Juckeland, and Hatem Ltaief, editors, *High Performance Computing*, pages 23–40, Cham, 2020. Springer International Publishing.

[5] Igor Sfiligoi, David Schultz, Benedikt Riedel, Frank Wuerthwein, Steve Barnet, and Vladimir Brik. Demonstrating a pre-exascale, cost-effective multi-cloud environment for scientific computing: Producing a fp32 exaflop hour worth of icecube simulation data in a single workday. In *Practice and Experience in Advanced Research Computing*, PEARC '20, page 85–90, New York, NY, USA, 2020. Association for Computing Machinery.