IceCube's Long Term Archive Software

Patrick Meade
Wisconsin IceCube Particle Astrophysics Center
Madison, WI
pmeade@icecube.wisc.edu

Benedikt Riedel
Wisconsin IceCube Particle Astrophysics Center
Madison, WI
briedel@icecube.wisc.edu

David Schultz
Wisconsin IceCube Particle Astrophysics Center
Madison, WI
dschultz@icecube.wisc.edu

ABSTRACT

IceCube is a cubic kilometer neutrino detector located at the South Pole. It generates 1 TiB of raw data per day, which must be archived for possible retrieval years or decades later. Other low-level data products are also archived for easy retrieval in the event of a catastrophic data center failure. The Long Term Archive software is IceCube's answer to archiving this data across several computing sites.

CCS CONCEPTS

• **Applied computing** → Physics; • **Computer systems organization** → **Grid computing**; Client-server architectures.

ACM Reference Format:
Patrick Meade, Benedikt Riedel, and David Schultz. 2019. IceCube's Long Term Archive Software. In *Practice and Experience in Advanced Research Computing (PEARC '19), July 28-August 1, 2019, Chicago, IL, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3332186.3332196

1 INTRODUCTION

The IceCube Neutrino Observatory [5] is located at the geographic South Pole and was completed at the end of 2010. It is designed to detect interactions of neutrinos by instrumenting over a gigaton of South Polar ice with 5160 optical sensors. Its primary scientific objective has been the discovery of astrophysical neutrinos, which was achieved in 2013 [1], and the identification and characterization of their sources, one of which was found in 2018 [2, 6]. Other science objectives include indirect detection of dark matter, searches for other exotic particles, studies of neutrino oscillation physics, and detection of the neutrino burst from a Galactic core-collapse supernova [10]. A multi-messenger collaboration with optical, X-ray, gamma-ray, radio, and gravitational wave observatories provides multiple windows onto neutrino sources.

2 DATA

During nominal operation, the detector generates 1 TiB of raw data every day. An online processing and filtering system is located in the IceCube Laboratory (ICL) building above the detector on the surface of the ice. This system selects 80 GiB of filtered data to be relayed via satellite each day. Both raw and filtered data are also archived to two copies written to hard drives. These copies are shipped back to the University of Wisconsin-Madison (UW-Madison) during polar summer. The Wisconsin IceCube Particle Astrophysics Center (WIPAC) at UW-Madison ingests the 365 TiB of yearly raw data into the Data Warehouse when it arrives in the north.

WIPAC runs more advanced reconstructions on the filtered data that arrives daily via satellite, generating data products called Level 2 (L2), Level 3 (L3), etc. The filtered data are generally not used directly for analysis, but retained in case of old bugs or new features in IceCube's analysis software. These additional reconstructions consume about 1200 CPU hours per day of data.

Since completion at the end of 2010, IceCube has generated 3 PiB of raw data, 200 TiB of filtered data, and 2 PiB of L2 and L3 data. The primary priority is to archive the raw data, as other data can be regenerated from it. Given the size and computational resources to run this regeneration, it is advantageous to archive all common data for IceCube in the event of a catastrophic data center failure.

3 ARCHIVAL HISTORY

The IceCube Collaboration has two partner institutions that store an archival copy of the data generated by WIPAC: German Deutsches Elektronen Synchrotron-Zeuthen (DESY) and U.S. Department of Energy-National Energy Research Scientific Computing Center (NERSC). DESY stores a copy of all data except the raw data in its dCache [9] tape archive. NERSC stores all data in its High Performance Storage System (HPSS) [14] tape archive.

The initial implementation of the Long Term Archive software was based on the Java Archiver and Data Exchange (JADE) [11] software. It is used by IceCube to transfer data over the satellite between the South Pole and WIPAC, as well as handling the yearly raw disk shipment. For long term archival, JADE was extended with a new mode to transfer from WIPAC to NERSC and called JADE LTA.

JADE LTA has successfully archived 3 PiB of raw data from WIPAC to NERSC over the last two years. However, there are serious shortcomings that prompted efforts to develop new archive software. First, JADE was not designed to run on cluster machines for parallel operations, and making JADE LTA do this required significant hacking and reconfiguration. Second, JADE LTA required manual operation to initiate transfer operations. Third, the reporting tools for JADE LTA were ad-hoc and made extracting insights difficult. Finally, JADE LTA relied upon the Globus Online service [8] to transfer data between GridFTP [3] endpoints. This service is regrettably transitioning to a closed-source commercial model, so WIPAC needs to find another data transfer service.

PEARC '19, July 28-August 1, 2019, Chicago, IL, USA

ACM ISBN 978-1-4503-7227-5/19/07…$15.00
https://doi.org/10.1145/3332186.3332196

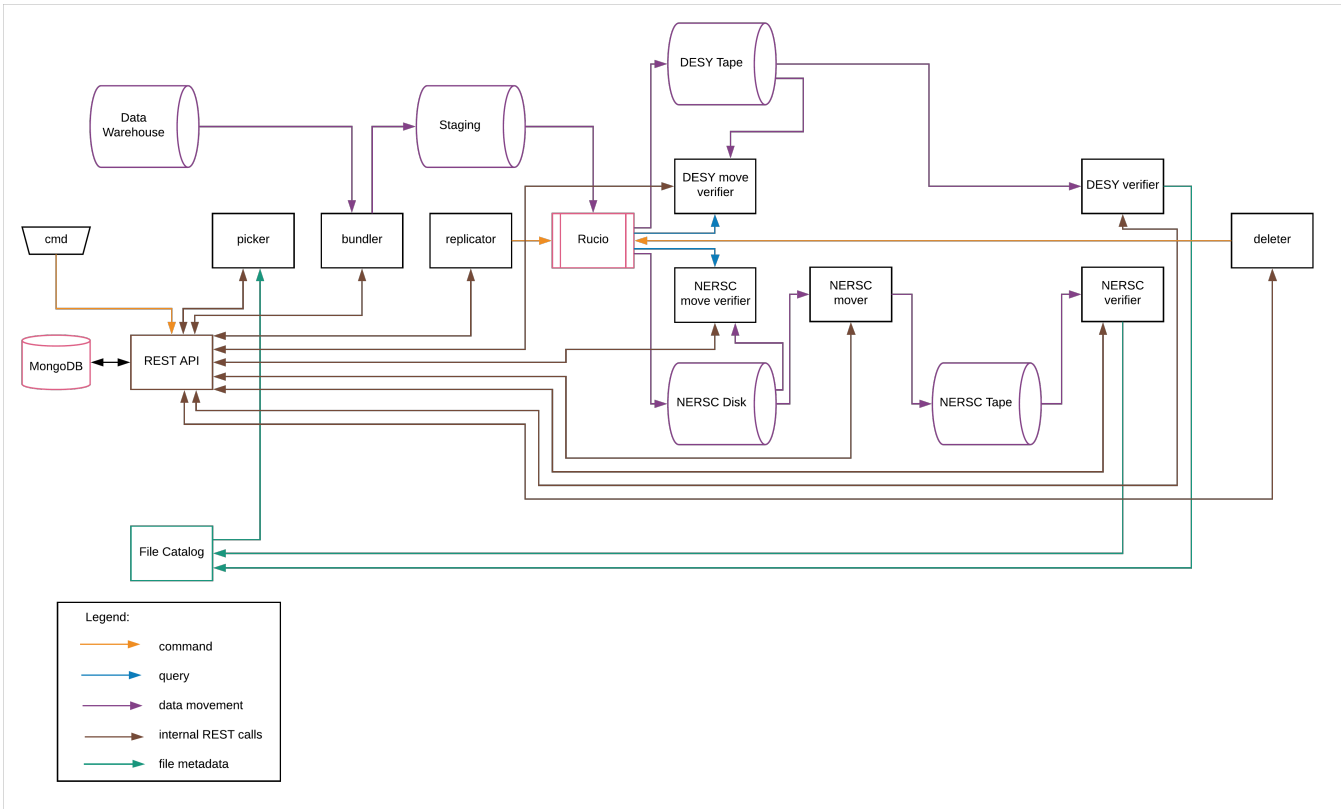

Figure 1: Component diagram showing a transfer from WIPAC to both NERSC and DESY, archiving files.

4 NEW SOFTWARE DESIGN

The new archive software drops the JADE name because it is written in Python. It is simply called Long Term Archive (LTA). The LTA software draws on a successful model used in IceProd - IceCube's production workflow software [12]. A central database service with a REST interface [7] retains most of the system state. Worker modules request and process work, and then submit results back to the central REST database. This approach allows workers to be added to the system very easily and scales well for the use case.

The goals of LTA are more ambitious than JADE LTA. JADE previously provided tools that allow the operator to batch individual transfer operations. LTA takes a transfer request between a source and a destination, and ideally the operator is not involved until LTA informs them that the data has reached its destination. LTA also leverages existing commercial tools like ElasticSearch, Grafana, Kibana, and Prometheus for monitoring and reporting.

Instead of relying on Globus Online to perform transfer operations, LTA relies upon the Rucio software [4]. Rucio is designed to replicate large volumes of data between storage sites. It has been used in production by ATLAS for several years, and has recently opened up to other scientific experiments as part of a push towards common, collaborative software in high energy physics.

4.1 File Catalog

WIPAC produces a range of data products with varying lifecycle characteristics. Some are sent via satellite, while others are carried from the South Pole on hard drives. Some of the data products live continuously in the Data Warehouse at WIPAC. Others are transferred to archival sites for indefinite storage. This diversity of transfer methods and storage locations means there is no single place that can answer the question, "Where can WIPAC find file X?"

The WIPAC File Catalog [13] is the service that provides the answer to this question. All of the data products produced by WIPAC are recorded in the File Catalog. Services that transfer or store the data products are expected to update the File Catalog.

LTA depends heavily on the File Catalog service, to know which files to archive, update file metadata with archive locations, and locate files to be restored. It is a central component of LTA, as shown in figure 1.

4.2 LTA Database

LTA performs its work in a widely distributed way. A single machine might be tasked with orchestrating transfers between the Data Warehouse and archival sites. However, no single machine is sufficient to create 3 PiB of archive files. Some LTA modules run on cluster machines allocated to WIPAC, and still others run on machines at partner sites like DESY and NERSC. These modules

scattered across dozens or hundreds of machines do not and cannot share a common file system. The modules require a service to coordinate their actions.

The LTA Database (LTA DB) defines a REST interface over four different kinds of entities: TransferRequests, Files, Bundles, and Status. TransferRequests capture the operator's intent to move data from one location to another: from the Data Warehouse to an archive site, or vice versa. Files identify the individual data products that require movement. Bundles record how the data products are grouped together into large archive files. Status allows a module to communicate its health and work, for monitoring and reporting purposes.

4.3 Workflow

There are two directions for archival data flow. One data flow is bundling and sending WIPAC data to an archival site at a partner institution, as seen in figure 1. LTA can send bundles to multiple archival sites at the same time, so each site has an identical copy. The other data flow is retrieving archival bundles from the partner institution and restoring the files to the Data Warehouse at WIPAC, as seen in figure 2. In both flows, multiple modules work together in order to create a data pipeline between the source and destination sites.

Both flows begin with the operator using a command line tool to add a TransferRequest to the LTA DB. The TransferRequest indicates the source and destination sites, along with the files that are to be moved. After the operator has added the TransferRequest to the LTA DB, the downstream modules query the LTA DB and begin to process the request.

4.3.1 Archival Workflow. The *Picker* consumes a TransferRequest from the LTA DB to determine which files need to be bundled for transfer to an archival site. Using the source directory provided in the TransferRequest, it queries the WIPAC File Catalog to determine which files have been picked for transfer to the archival site. It then adds File entities to the LTA DB for further processing.

The *Bundler* consumes File entities from the LTA DB. Each destination site has a preferred archive file size, and the Bundler creates archival bundles that are appropriately sized for their intended destinations. The archival bundle is in ZIP64 format. This format was chosen for its ability to survive corruption and data loss, as well as the ability to access an arbitrary file within a bundle without sequentially reading the entire bundle. After creation of the ZIP archive, it is transferred to a staging disk that Rucio can access. A Bundle entity is then registered with the LTA DB for further processing.

The *Replicator* consumes Bundle entities from the LTA DB and registers them with Rucio. A replication rule is added to Rucio that instructs Rucio to copy the ZIP archive from the staging disk at WIPAC to the staging disk(s) at the destination site(s). Rucio checks its own database and performs this transfer automatically. Finally, the Replicator updates the TransferRequest entity LTA DB, to indicate the archival bundles are transferring to the destination site(s).

The *MoveVerifier* queries Rucio to determine if the bundle has been successfully replicated to the destination site(s) and updates the LTA DB. The Bundle entity is updated to record that it exists at a new location. The TransferRequest entity is updated to indicate the archival bundle for that destination is now accessible. After these entities have been updated, verification of the bundle at the destination site can begin.

The *NerscMover* is a special worker module that facilitates work at NERSC. NERSC has Data Transfer Nodes (DTNs) that are used by third parties to transfer data in and out of NERSC disk. The *NerscMover* is responsible for issuing commands to the HPSS system to stage the data from NERSC disk into the tape archive. The *NerscMover* updates the TransferRequest in the LTA DB. The archival bundle is marked inaccessible, to indicate that it now exists in a tape system that is not directly accessible to Rucio.

The *ArchiveVerifier* reads the archival bundle at the destination site(s) and ensures that the SHA512 checksum matches the checksum generated by the *Bundler* when the archival bundle was first created. It updates the LTA DB to indicate that the archival bundle has been verified at the destination. The *ArchiveVerifier* also updates the File Catalog with new location information for each file in the archival bundle, as well as the bundle itself.

The *Deleter* checks TransferRequest entities in the LTA DB to determine if all of the required Bundle entities have been verified at all of their destinations. It sends commands to Rucio to remove the archival bundle files from staging disks. The *Deleter* also updates the TransferRequest entity in LTA DB to indicate that the transfer request is now complete.

4.3.2 Restore Workflow. The *Locator* consumes a TransferRequest from the LTA DB to determine which files need to be brought back from archive to the Data Warehouse at WIPAC. The *Locator* uses the destination directory provided in the TransferRequest to query the File Catalog. The *Locator* updates the TransferRequest in the LTA DB with information about the archival bundles that need to be copied back to WIPAC.

The *NerscRetriever* is a special worker module that facilitates work at NERSC. As a compliment to the *NerscMover* module in the archival workflow, the *NerscRetriever* issues commands to the HPSS tape archive to copy files back to NERSC disk. After the archival bundle is copied to disk, it is now accessible by Rucio.

The *Replicator* queries the LTA DB for TransferRequests with accessible bundles that need to be copied back to staging disk at WIPAC. It adds a replication rule to Rucio to prompt Rucio to copy the file to WIPAC.

As in the archival workflow, the *MoveVerifier* queries Rucio to determine if the bundle has been successfully replicated to the destination site(s) and updates the LTA DB. In the context of the restore workflow, this destination site is the staging disk at WIPAC.

The *Unpacker* consumes archival bundles that have been transferred back to staging disk at WIPAC. It verifies the checksum of the archival bundle, unpacks the files from the archival bundle, places the file into the Data Warehouse, verifies the checksums of the files in the Data Warehouse, and updates the File Catalog to indicate that a copy of the file now exists locally at WIPAC.

The *Deleter* sends commands to Rucio to remove the archival bundle files from staging disks. It updates the TransferRequest in the LTA DB to indicates that the transfer request is complete.

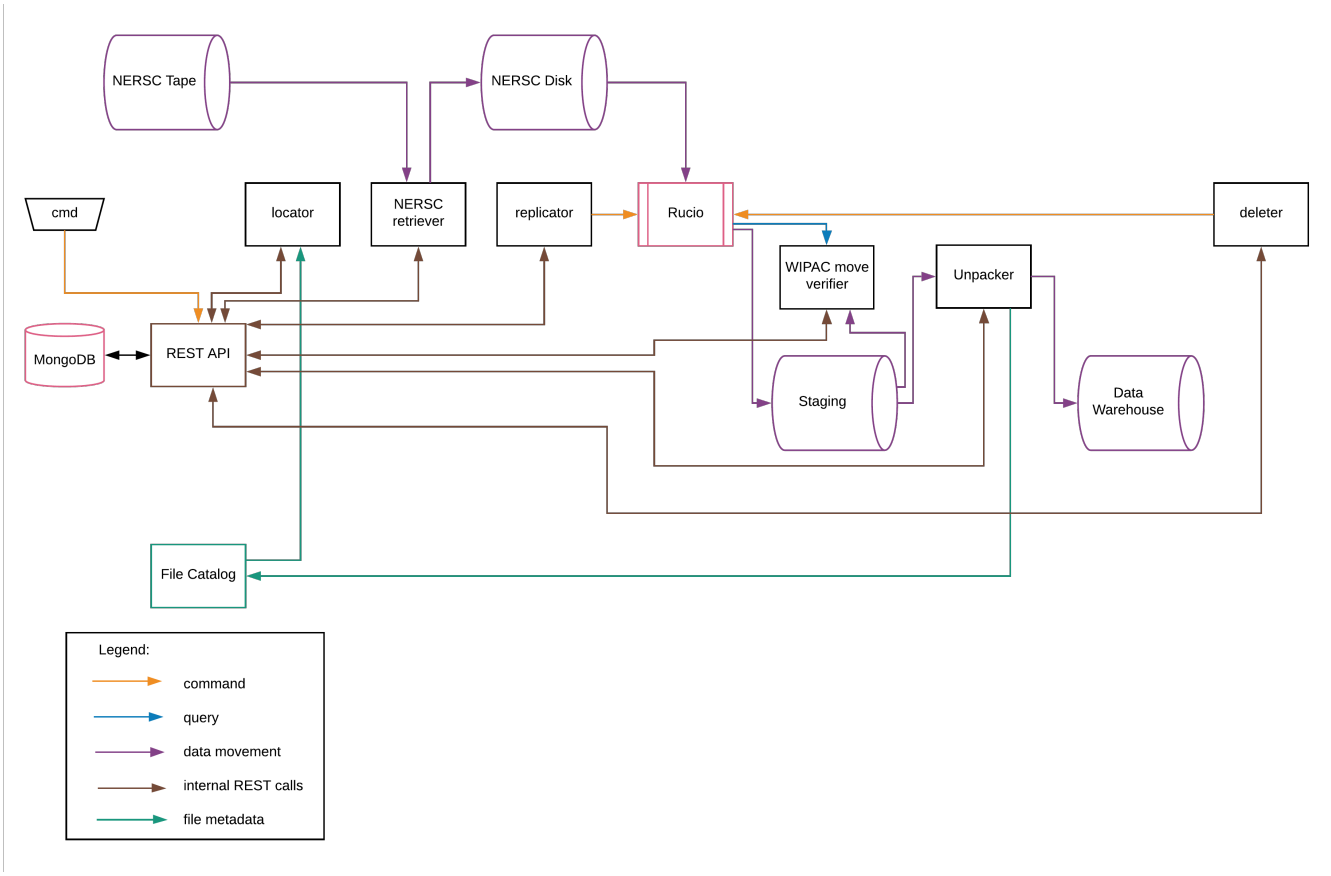

Figure 2: Component diagram showing a transfer from NERSC to WIPAC, restoring files.

4.4 Monitoring

LTA is a highly automated system intended to move PiB-scale data across diverse sites with minimal operator intervention. Ideally, the operator specifies the data to be archived from and/or restored to the Data Warehouse, and everything happens with no further intervention. However, there will be a need to respond to error and failure conditions that cannot be resolved by retrying the operation at later time, where a human may need to take action.

One of the challenges of LTA is that not all of the diverse sites where LTA modules run are owned or controlled by WIPAC. Partner sites are often willing to provide shell access to systems where the LTA module may run. The module can often reach out to external sites, but access directly to it through a firewall is often forbidden. This is a typical configuration for cluster machines where LTA components directly handling PiB-scale data would run.

LTA modules use the LTA DB as a channel for status information. On a regular basis, the LTA module contacts the LTA DB to report on its current status. This "heartbeat" allows WIPAC to monitor the status of LTA modules regardless of where they may be running. If a module stops running, it will also stop sending heartbeats. After a reasonable time, the monitoring system can inform the system administrators that intervention is required.

Querying the LTA DB through the REST interface, it is possible to integrate status monitoring and reporting with a diverse set of tools. To provide flexibility, monitoring is treated as just another module accessing the REST interface, with one output type per module. This allows using multiple tools at the same time, or migrating from one tool to another. WIPAC has historically used Nagios to monitor its systems, but LTA is exploring integration with Prometheus as a modern real-time monitoring solution. For long-term monitoring and trend analysis, ElasticSearch with either Grafana or Kibana as a frontend is available.

5 USAGE

Archival of raw data was the initial priority, and JADE LTA has archived 3 PiB of raw data from WIPAC to NERSC over the last two years. Starting with the new LTA software, non-raw data will be transferred to both NERSC and DESY. This should begin within the next few months. Additional raw data for current and future years will also be archived using LTA.

Restoration is not just a theoretical concept. WIPAC has restored individual runs for specialized analysis several times over the last two years. Additionally, WIPAC will do a full raw data retrieval sometime in 2019 for a reprocessing campaign, to apply corrections

and new reconstructions. While full retrievals are fairly rare, they do happen from time to time.

6 CONCLUSIONS

JADE LTA was a critical piece of software in archiving 3 PiB of data from WIPAC to NERSC. WIPAC's data replication requirements have outgrown JADE LTA and prompted the creation of better long term archival software. The new LTA software is more powerful, has better performance, and reinforces WIPAC's commitment to data archival and reproducibility.

The value added by the LTA system is packaging files into bundles that are appropriately sized for replication to partner sites. Better integration with Rucio may enable the software to meet the needs of other diverse institutions. WIPAC continues to investigate that and other possibilities for LTA.

ACKNOWLEDGMENTS

USA – U.S. National Science Foundation-Office of Polar Programs, U.S. National Science Foundation-Physics Division, Wisconsin Alumni Research Foundation, Center for High Throughput Computing (CHTC) at the University of Wisconsin-Madison, Open Science Grid (OSG), Extreme Science and Engineering Discovery Environment (XSEDE), U.S. Department of Energy-National Energy Research Scientific Computing Center, Particle astrophysics research computing center at the University of Maryland, Institute for Cyber-Enabled Research at Michigan State University, and Astroparticle physics computational facility at Marquette University; Belgium – Funds for Scientific Research (FRS-FNRS and FWO), FWO Odysseus and Big Science programmes, and Belgian Federal Science Policy Office (Belspo); Germany – Bundesministerium für Bildung und Forschung (BMBF), Deutsche Forschungsgemeinschaft (DFG), Helmholtz Alliance for Astroparticle Physics (HAP), Initiative and Networking Fund of the Helmholtz Association, Deutsches Elektronen Synchrotron (DESY), and High Performance Computing cluster of the RWTH Aachen; Sweden – Swedish Research Council, Swedish Polar Research Secretariat, Swedish National Infrastructure for Computing (SNIC), and Knut and Alice Wallenberg Foundation; Australia – Australian Research Council; Canada – Natural Sciences and Engineering Research Council of Canada, Calcul Québec, Compute Ontario, Canada Foundation for Innovation, WestGrid, and Compute Canada; Denmark – Villum Fonden, Danish National Research Foundation (DNRF), Carlsberg Foundation; New Zealand – Marsden Fund; Japan – Japan Society for Promotion of Science (JSPS) and Institute for Global Prominent Research (IGPR) of Chiba University; Korea – National Research Foundation of Korea (NRF); Switzerland – Swiss National Science Foundation (SNSF).

REFERENCES

[1] M. G. Aartsen et al. 2013. Evidence for High-Energy Extraterrestrial Neutrinos at the IceCube Detector. *Science* 342 (2013), 1242856. https://doi.org/10.1126/science.1242856 arXiv:astro-ph.HE/1311.5238

[2] M. G. Aartsen et al. 2018. Multimessenger observations of a flaring blazar coincident with high-energy neutrino IceCube-170922A. *Science* 361, 6398 (2018), eaat1378. https://doi.org/10.1126/science.aat1378 arXiv:astro-ph.HE/1807.08816

[3] William Allcock, John Bresnahan, Rajkumar Kettimuthu, Michael Link, Catalin Dumitrescu, Ioan Raicu, and Ian Foster. 2005. The Globus striped GridFTP framework and server. In *Proceedings of the 2005 ACM/IEEE conference on Supercomputing*. IEEE Computer Society, 54.

[4] M Barisits, Thomas Beermann, V Garonne, T Javurek, M Lassnig, C Serfon, ATLAS collaboration, et al. 2018. The ATLAS Data Management System Rucio: Supporting LHC Run-2 and beyond. In *Journal of Physics: Conference Series*, Vol. 1085. IOP Publishing, 032030.

[5] M.G. Aartsen et al. 2017. The IceCube Neutrino Observatory: instrumentation and online systems. *Journal of Instrumentation* 12, 03 (2017), P03012. https://doi.org/10.1088/1748-0221/12/03/P03012

[6] M. G. Aartsen et al. 2018. Neutrino emission from the direction of the blazar TXS 0506+056 prior to the IceCube-170922A alert. *Science* 361, 6398 (2018), 147–151. https://doi.org/10.1126/science.aat2890 arXiv:http://science.sciencemag.org/content/361/6398/147.full.pdf

[7] Roy T Fielding and Richard N Taylor. 2000. Architectural styles and the design of network-based software architectures.

[8] Ian Foster. 2011. Globus Online: Accelerating and democratizing science through cloud-based services. *IEEE Internet Computing* 15, 3 (2011), 70–73.

[9] Patrick Fuhrmann and Volker Gülzow. 2006. dCache, storage system for the future. In *European Conference on Parallel Processing*. Springer, 1106–1113.

[10] Francis Halzen and Spencer R. Klein. 2010. IceCube: An Instrument for Neutrino Astronomy. *Rev. Sci. Instrum.* 81 (2010), 081101. https://doi.org/10.1063/1.3480478 arXiv:astro-ph.HE/1007.1247

[11] P. Meade. 2017. jade: An End-To-End Data Transfer and Catalog Tool. *J. Phys. Conf. Ser.* 898, 6 (2017), 062050. https://doi.org/10.1088/1742-6596/898/6/062050

[12] David Schultz, Vladimir Brik, Jakob van Santen, Heath Skarlupka, Carl Witt, and Alex Olivas. 2019. WIPACrepo/iceprod: v2.5.1. https://doi.org/10.5281/zenodo.2561158

[13] David Schultz, Jan Oertlin, Vladimir Brik, Patrick Meade, and Heath Skarlupka. 2019. WIPACrepo/file_catalog: 1.1.1. https://doi.org/10.5281/zenodo.2561164

[14] Richard W Watson and Robert A Coyne. 1995. The parallel I/O architecture of the high-performance storage system (HPSS). In *mss*. IEEE, 27.