IceCube SkyDriver – A SaaS Solution for Event Reconstruction using the Skymap Scanner

Eric Evans-Jacquez[1,*], *David* Schultz[1], *Brian* Bockelman[2], *Massimiliano* Lincetto[3], *Miron* Livney[2,4], *Benedikt* Riedel[1], and *Tianlu* Yuan[1]

[1]Dept. of Physics and Wisconsin IceCube Particle Astrophysics Center, University of Wisconsin–Madison, Madison, WI 53706, USA
[2]Morgridge Institute for Research, Madison, WI 53715, USA
[3]Fakultät für Physik & Astronomie, Ruhr-Universität Bochum, D-44780 Bochum, Germany
[4]Dept. of Computer Science, University of Wisconsin–Madison, Madison, WI 53706, USA

Abstract.
The IceCube Neutrino Observatory is a cubic kilometer neutrino telescope located at the geographic South Pole. To accurately and promptly reconstruct the arrival direction of candidate neutrino events for Multi-Messenger Astrophysics use cases, IceCube employs Skymap Scanner workflows managed by the SkyDriver service. The Skymap Scanner performs maximum-likelihood tests on individual pixels generated from the Hierarchical Equal Area isoLatitude Pixelation (HEALPix) algorithm. Each test is computationally independent, which allows for massive parallelization. This workload is distributed using the Event Workflow Management System (EWMS)—a message-based workflow management system designed to scale to trillions of pixels per day. SkyDriver orchestrates multiple distinct Skymap Scanner workflows behind a REST interface, providing an easy-to-use reconstruction service for real-time candidate, cataloged, and simulated events. Here, we outline the SkyDriver service technique and the initial development of EWMS.

1 Introduction

The IceCube Neutrino Observatory [1] is the world's preeminent neutrino telescope, located at the geographic South Pole. Its primary mission is to find and observe high-energy astrophysical neutrinos from extra-terrestrial sources, such as active galactic nuclei or transient astrophysical phenomena. IceCube has created a new method to observe our universe at the highest energies and longest distances. Beyond high-energy neutrino astrophysics, IceCube's research spans various fields including particle physics, cosmic ray physics, and geosciences.

IceCube is a key facility within the field of Multi-Messenger Astrophysics (MMA), which focuses on observing astrophysical phenomena using various messengers, such as gravitational waves, neutrinos, and electromagnetic emissions. Combining observations from various observatories provides a much deeper understanding of astrophysical processes at their origins. Most MMA discoveries so far are transient in nature—only observable across multiple messengers for a relatively short period. The arrival direction of the messenger is key for allowing other astronomical observatories to perform realtime follow-up analyses.

*e-mail: eevans@icecube.wisc.edu

Unlike a traditional telescope, IceCube observes the entire sky continuously and does not require dedicated observations of a given source. Since the origin of the event is unknown until further analysis, it is particularly important to deliver its coordinates as fast and precise as possible. Once a potential high-energy astrophysical neutrino is observed at the South Pole, IceCube's Realtime Alert System [2]—an automated data analysis chain—is initiated. The alert is distributed to the global science community through the General Coordinates Network (GCN) [3] and, for exceptional alerts, Astronomer's Telegram (ATel) [4]. The current MMA pipeline transmission was a key part in identifying the flaring blazar TXS 0506+056 as the first high-energy neutrino point source [5].

These events demand a thorough reconstruction to estimate the neutrino arrival direction and its uncertainty. Within the MMA pipeline, IceCube employs a likelihood scan, analyzing regions across the entire sky, to perform this directional reconstruction. Managing this highly computationally intensive task entails orchestrating calculations for up to 3 million distinct sections of the sky. This is the driving motivation for recent upgrades to the Skymap Scanner and the prototype SkyDriver service.

This paper outlines major enhancements made to Skymap Scanner [6] and the creation of SkyDriver [7]. By leveraging distributed computing resources, such as cloud, high-performance (HPC), and high-throughput (HTC) computing, we expedite realtime processing. This boost in scalability not only enhances usability but also enables broadening the range of scanned events to include IceCube's historical catalog. Researchers can re-analyze events and experiment with new reconstruction algorithms. The advancements in scalability are seamlessly managed by the new SkyDriver service. Moreover, a representational state transfer (REST) interface empowers users with remote control, enabling them to initiate scans and retrieve the ultimate result—the skymap.

2 Background

The Skymap Scanner v1 is the existing system to determine the arrival direction through a likelihood scan used by IceCube's MMA pipeline and has been in production for several years. During development there was no deliberate effort to establish proper software versioning for Skymap Scanner v1, thus this name is retrospectively assigned.

The main objective of the Skymap Scanner system is to derive the likelihood, energy, and associated error of the probable direction of the neutrino event by testing thousands of potential directions, to the granularity of equal area regions created using the Hierarchical Equal Area isoLatitude Pixelation (HEALPix) algorithm [8, 9]. This likelihood calculation method can be distributed among thousands of CPUs, each processing a single pixel at a time.

Skymap Scanner v1 faces challenges with scalability and user-friendliness. The system's deployment is limited to IceCube's dedicated resources in Madison, Wisconsin. To expedite the time required for results, other jobs on the IceCube cluster are evicted, causing significant disruption to ongoing data analyses by other users, particularly those requiring lengthier run times. The system is tightly coupled with IceCube's MMA pipeline, making it difficult for non-expert users to operate and utilize the scanning likelihood technique for other analyses. The system is also tied to a single reconstruction algorithm, making software improvements difficult and developing new methods unfeasible.

The components of the Skymap Scanner v1 specific to the MMA pipeline are currently being migrated into SkyMist [10], a next-generation realtime alert system for IceCube. This work is occurring concurrently with the development of Skymap Scanner v3 and SkyDriver (Skymap Scanner v2 is an abandoned prototype and is not mentioned further in this paper).

2.1 Workflow Managers in the High Energy Physics Community

In an optimal high-throughput workflow, a one-to-one mapping of jobs to individual work units is ideal for maximizing efficiency. However, in practical terms, especially in High Energy Physics (HEP) experiments processing vast numbers of events, a strict one-to-one mapping becomes unmanageable and risks overloading system components, including the batch scheduler, data access system, workload tracking system, and databases. Addressing this issue is where SkyDriver's workflow management draws its largest distinction with traditional Large Hadron Collider (LHC) workflow managers. By leveraging message-queue (pub-sub) technology to efficiently distribute 'small' data units, combined with utilizing HTCondor's platform, the system offers a robust solution for managing such granularity and scale.

3 Skymap Scanner – Methodology and Implementation

The Skymap Scanner's method for analyzing a neutrino's arrival direction makes it an excellent candidate for integrating distributed computing advancements. The Skymap Scanner uses the HEALPix algorithm to divide the sky into equal area regions, referred to as "pixels". The Skymap Scanner analyzes and refines pixels recursively, selecting sub-regions for further refinement and subdividing them into finer pixels at the next resolution level. This iterative procedure continues until the final resolution is achieved, identifying the pixel with the highest global likelihood as the arrival direction of the neutrino. A skymap's resolution can be expressed using HEALPix's N_{side} parameter, which determines the number and size of pixels. The current N_{side} progression, 8-64-512, results in at minimum 3072 total pixels—the exact count may vary slightly due to nuances in the refinement process.

Analyzing each of these pixels is statistically and computationally independent. This means that each pixel is treated as a potential directional source where the observed neutrino originated independently of any other pixel. This allows each statistical test to occur on a different process/core. This problem area has made it possible to design a 1:N manager-worker model—also known as server-client or work queue model—where the computationally intensive pixel analyses are off-loaded to many distributed resources, or "client workers". The following subsections provide a comprehensive overview of the Skymap Scanner v3 model.

3.1 Skymap Scanner Central Server

The central server undertakes various non-computationally intensive tasks, such as creating a common input file, creating pixels, etc. To start a scan, the user provides an IceCube event and associated detector configuration data. With this information, the server generates a "startup" file containing the common input data for the individual workers. The startup file is stored in a globally accessible location and fetched at startup, usually through HTTP downloads from an S3-compatible [11] bucket or a shared filesystem across workers.

Once the startup file is generated and accessible, pixel creation commences. The server generates pixels at the user's specified resolution, each equipped with the name of the scan's reconstruction algorithm and event-specific details, as described in Table 1. These pixels are then distributed to clients using a remote message broker—this keeps the server independent from the client workers. This strategic design facilitates efficient concurrency.

Transitioning to the collection and reporting phase, the server waits for server-bound messages containing likelihood calculations. It regularly reports its progress and partial skymap to the user. These updates are saved to disk, logged to the console, and, if required, sent to SkyDriver for post hoc analysis. Upon accumulating a sufficient amount of likelihood statistics, the server dispatches pixels for the next iteration (determined by the configurable N_{side}

progression)—the exact number of server-bound messages needed may be less than 100% of the current iteration and is optimized to accelerate refinement. If no further pixel generation is needed, the scan concludes, offering the final skymap and computing statistics.

3.1.1 MQClient – Message Passing Interface

To distribute the workload across the workers, we employ RabbitMQ [12] in combination with EWMS's MQClient Python package [13]. The MQClient package provides a common interface across different message broker protocols including Apache Pulsar [14], RabbitMQ, and NATS.io [15]. Message delivery is independent of the recipient processes, enabling CPUs with higher computational power to handle extra pixels as required.

3.2 Skymap Scanner Client Worker

Each CPU worker hosts a Skymap Scanner client responsible for analyzing pixels transferred by the MQClient. Each incoming client-bound message specifies the reconstruction algorithm to utilize. The outgoing server-bound message contains a pixel's likelihood statistics—a handful of floating point numbers. Reducing the size of in-flight data is crucial for a performant scan. A Skymap Scanner instance may consist of thousands of worker CPUs.

The Skymap Scanner client has a flexible, platform-independent design, requiring only an HTTP connection and adequate processing capacity (~8 GB of memory and ~2.5 GB of disk space). Adhering to this design philosophy, the client software is contained in publicly available images on Docker Hub [16]. In production, the worker CPUs are orchestrated by HTCondor [17], Kubernetes [18], or a combination of these. For HTCondor clusters, such as the Open Science Pool [19], Apptainer [20] images are made available on the CernVM File System (CVMFS) [21].

3.2.1 Pixel Analysis on the EWMS Pilot

Each pixel's analysis is performed using IceCube's IceTray software framework [22, 23] and reconstruction algorithms [24]—all are included in the client software. An EWMS pilot [25] supervises two main tasks: (1) data transfer utilizing the distributed message passing interface and (2) execution for the science payload, which involves the likelihood calculation for the respective reconstruction algorithm. Given the operational scale of the Skymap Scanner, transient errors among workers are not uncommon. The pilot automatically handles these errors, terminating the worker if needed. Any pixel that is disrupted by a transient error is requeued and redelivered to a functioning worker (pilot). The terminated worker is quarantined to prevent future disruptions during the event scan.

Table 1. An explanation of the terms used to refer to data within the Skymap Scanner

Term	Description	Other Names
Pixel (Message)	A set of two-dimensional coordinates, minimally precomputed datapoints, and reconstruction algorithm	Client-bound message
Pixel's Likelihood Statistic	A set of data points representing the likelihood that the pixel is the event's arrival direction	Pixel-reco, server-bound message
Skymap	A collection of likelihood statistics for a given neutrino event	Result

4 SkyDriver – Methodology and Implementation

4.1 Reconstruction-as-a-Service

Empowering physicists with easy-to-use scans is a key component of SkyDriver. To help achieve this, we added a REST interface and manager for the Skymap Scanner system. This allows physicists to focus on their main objective of event analysis, instead of learning various configuration settings and computing setups, as well as how to optimize them for efficient and performant usage. This decreases onboarding and contributes to expanding the user base.

SkyDriver features robust systems-level error handling, both automated and manual. This supports science by offloading technical expertise to technical experts. In a large integrated system, there can be many points of failure. Designing with this in mind, we have built custom software and leveraged third-party solutions to minimize the propagation of transient errors while reporting critical errors with detail. This distinction is highlighted through a comparison of a single faulty worker CPU versus corrupted input data—the former can be remedied, but the latter cannot. This concentrated focus on fault tolerance is a driving factor in the Event Workflow Management System (EWMS), mentioned in section 6.

Version management and containerization play a crucial role in the development of SkyDriver and the Skymap Scanner v3. Each update to any system dependency is recorded and updates to the core functionality trigger a new release for the Skymap Scanner, facilitated through our Continuous Integration / Continuous Deployment (CI/CD) pipelines. SkyDriver users can select specific Skymap Scanner versions, promoting scientific reproducibility and enabling system-level debugging.

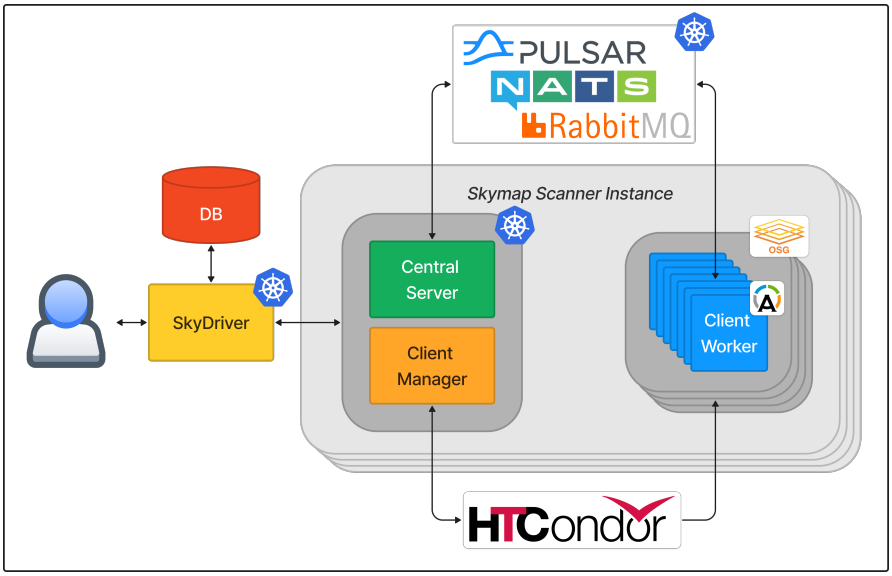

Figure 1. SkyDriver technical diagram including several Skymap Scanner instances. The SkyDriver REST interface and database are on the left, hosted in a dedicated Kubernetes cluster. For each scan, SkyDriver provisions a Skymap Scanner instance consisting of a Central Server and Client Manager, also in a dedicated Kubernetes cluster. The Client Manager then starts Client Workers via a cluster system (HTCondor pictured here, running on the Open Science Pool via Apptainer containers).

4.2 Scalability and Automated Orchestration

To address the challenge of fast and reliable neutrino event reconstruction, our service places a significant focus on scalability and automation. Scalability is defined among three independent vectors: instance scalability, workforce scalability, and pixel resolution scalability.

Instance scalability is accomplished by the management of multiple concurrent scans. This is a non-trivial problem since each scan can occupy thousands of CPU cores. SkyDriver effectively regulates the concurrency load in order to successfully provide its scanning services. This is especially important for scanning historical cataloged events, which do not necessitate the fast processing time of a realtime alert within the MMA pipeline.

Workforce scalability is supported by utilizing a remote message broker for communication and an orchestrated cluster for computing power. Message passing is inherently friendly to asynchronous and concurrent processing—a message is received when it is asked for.

Pixel resolution scalability, the most domain-specific vector, refers to the ability to adjust how the pixels on the Scanner server are generated. This is particularly useful in continuous integration testing. Whereas a production-sized scan involves thousands of pixels, a test scan uses a dozen. This option also provides the ability for future scans to use different resolutions. Optimizations to the system can be done by changing the N_{side} progression series.

These scalability possibilities add additional complexity to an already complex system. While defining the pixel resolution is the user's responsibility, the scalability of the Skymap Scanner instances and each Scanner's workforce is controlled by SkyDriver. Kubernetes is used to automate the orchestration of the Skymap Scanner's server and the middle-man component, the Client Manager, as seen in Figure 1.

In addition, SkyDriver's services, its database, and its REST server are contained in a Kubernetes cluster. The message-passing broker server is similarly handled in a dedicated remote cluster. Separating this service from the SkyDriver system reinforces the emphasis on loose coupling promoted by the EWMS technique. The broker can be scaled to handle an arbitrary increase in message volume without affecting other components.

4.2.1 Google Cloud Platform Testing

In June 2023, we tested the service's architecture on Google's Cloud Platform (GCP) [26] to simulate large-scale scanning of cataloged events. These tests involved running concurrent scans with numerous IceCube events, utilizing tens of thousands of CPUs. During this testing, we identified areas that experienced slowdowns, such as message queue broker scalability and transient errors that emerged with high concurrency. These issues have since been fixed. A notable finding was confirming that worker CPUs started more quickly in the cloud compared to HTCondor clusters. To make the most of this, we intend to combine both types of orchestrators in our production approach for the MMA pipeline. We will initiate scans on a cloud platform to take advantage of its fast startup speed and then smoothly transition to utilizing HTCondor clusters for their cost-effectiveness as they become available.

5 Results and Measuring Performance

The upgraded Skymap Scanner provides a standard data format for the computed skymap. Storing the entire skymap is made possible by greatly reducing the data created during the reconstruction. This data reduction is (1) necessary for data transfer within the message-passing system and (2) allows millions of skymaps to be persisted in the SkyDriver database. Data is stored and exchanged in the JSON format, ensuring compatibility across systems and human-readable representation. This empowers researchers to utilize the entire skymap to

analyze and develop new reconstruction methods. Third-party automated services, such as SkyMist, can tap into this stream for additional instantaneous functionality.

The scan creates performance metrics throughout the scan and persists them in SkyDriver's database after completion. Users can manually monitor live progress using this data. This has been useful in testing and development. Researchers can also use these metrics to form additional analyses, such as evaluating new reconstructions, benchmarking the availability of certain CPU clusters, assessing scaling techniques, etc. Experts and users can tune SkyDriver to incorporate such findings and optimize performance for future scans. Furthering efficiencies is central to SkyDriver and the EWMS system.

6 Future Work – Event Workflow Management System (EWMS)

Ultimately, the Skymap Scanner will be a client of the Event Workflow Management System. The system, like SkyDriver, will be capable of managing many clustered workflows. However, EWMS will do much more. Beyond pixel reconstruction, EWMS workflows will support a broad range of tasks. It will support workflows of various topologies; whereas the Skymap Scanner is a 1:N client-server model, other workflows may be N:M, 1:N:M, N:M:O, etc. This is made possible by leveraging multiple message-passing queues and EWMS pilot configurations.

This expansion brings the cluster orchestration intelligence and ease of use to other potential use cases and domains, such as non-neutrino astronomical observations (images), cryogenic electron microscopy (cryo-EM) data, and optical character recognition on pages in a book. Creating a centralized system for managing massively parallelizable workflows with computationally independent workers can reduce duplicated efforts among scientific domains. This will allow engineers and researchers to focus on their unique applications.

In this future design, SkyDriver will be one facade to EWMS. It will retain the public-facing components specific to the Skymap Scanner—input validation, the REST routes, and its database—while cluster orchestration will be outsourced. Backward compatibility will be a priority during this transition. The microservice enhancements in the Skymap Scanner and SkyDriver, previously outlined in this paper, make this pending upgrade possible.

7 Conclusion

The Skymap Scanner's evolution from its initial version to the current v3, coupled with the development of SkyDriver, introduces a user-friendly, scalable, and efficient solution for neutrino event reconstruction. This integration efficiently manages massive parallelization while ensuring fault tolerance and reliability. The system's scalability is exemplified through instance, workforce, and pixel resolution scalability, ultimately orchestrated using a combination of Kubernetes and HTCondor.

SkyDriver's Software-as-a-Service solution streamlines event reconstruction. Physicists and researchers can now perform analyses without the overhead of technical configuration settings and computing intricacies. This democratization of access accelerates research, broadens user adoption, and fosters collaboration. In support of scientific reproducibility, SkyDriver incorporates version management and containerization, enabling researchers to select specific publicly available versions of the Skymap Scanner for their analyses. SkyDriver's data schema and performance metrics provide insights for continual enhancement.

IceCube's SkyDriver marks a significant step forward within the IceCube Neutrino Observatory. By integrating the Skymap Scanner with the power of distributed computing and

orchestration, SkyDriver offers an innovative approach to addressing the challenges of accurately and rapidly determining the arrival direction of candidate neutrino events. It presents a model for managing complex scientific workflows across various scientific domains.

Acknowledgments

This work was partially funded by the U.S. National Science Foundation (NSF) under grants OPP-2042807 and OAC-2103963.

References

[1] M.G. Aartsen et al. (IceCube), JINST **12**, P03012 (2017), 1612.05093

[2] M.G. Aartsen et al. (IceCube), Astropart. Phys. **92**, 30 (2017), 1612.06028

[3] L. Singer, J. Racusin, Bulletin of the AAS **55** (2023), https://baas.aas.org/pub/2023n2i108p02

[4] R.E. Rutledge, Publications of the Astronomical Society of the Pacific **110**, 754 (1998)

[5] M. Aartsen et al. (IceCube), Science **361**, 147 (2018)

[6] *Skymap Scanner*, https://github.com/icecube/skymap_scanner/, accessed: 2023-08-09

[7] *SkyDriver*, https://github.com/WIPACrepo/SkyDriver/, accessed: 2023-08-09

[8] K.M. Górski, E. Hivon, A.J. Banday, B.D. Wandelt, F.K. Hansen, M. Reinecke, M. Bartelmann, The Astrophysical Journal **622**, 759 (2005)

[9] *HEALPix*, https://healpix.sourceforge.io, accessed: 2023-08-09

[10] M. Lincetto, E. Evans-Jacquez, B. Riedel, D. Schultz, T. Yuan et al., PoS **ICRC2023**, 1106 (2023)

[11] *Amazon Web Services - S3*, https://aws.amazon.com/s3/, accessed: 2023-08-10

[12] *RabbitMQ*, https://www.rabbitmq.com/, accessed: 2023-08-09

[13] *OMS-MQClient*, https://github.com/Observation-Management-Service/MQClient/, accessed: 2023-08-09

[14] *Apache Pulsar*, https://pulsar.apache.org/, accessed: 2023-08-09

[15] *NATS.io*, https://nats.io/, accessed: 2023-08-09

[16] *Docker Hub*, https://hub.docker.com/, accessed: 2023-12-08

[17] D. Thain, T. Tannenbaum, M. Livny, Concurrency and Computation: Practice and Experience **17**, 323 (2005)

[18] *Kubernetes*, https://kubernetes.io/, accessed: 2023-08-10

[19] R. Pordes, D. Petravick, B. Kramer, D. Olson, M. Livny, A. Roy, P. Avery, K. Blackburn, T. Wenaus, F. Würthwein et al., *The open science grid*, in *J. Phys. Conf. Ser.* (2007), Vol. 78 of *78*, p. 012057

[20] *Apptainer*, https://apptainer.org/, accessed: 2023-08-10

[21] J. Blomer, B. Bockelman, P. Buncic, B. Couturier, D.F. Dosaru, D. Dykstra, G. Ganis, M. Giffels, H. Nikola, N. Hazekamp et al., *The cernvm file system: v2.7.5* (2020), https://doi.org/10.5281/zenodo.4114078

[22] T. DeYoung, *IceTray: A software framework for IceCube*, in *14th International Conference on Computing in High-Energy and Nuclear Physics* (2005), pp. 463–466

[23] *IceTray*, https://github.com/icecube/icetray-public/, accessed: 2023-08-09

[24] M. Aartsen et al. (IceCube Collaboration), JINST **9**, P03009 (2014), 1311.4767

[25] *ewms-pilot*, https://github.com/Observation-Management-Service/ewms-pilot/, accessed: 2023-08-09

[26] *Google cloud platform*, https://cloud.google.com/, accessed: 2019-31-01