EWMS in Action

A User’s Guide to Adaptive, Extreme-Scale Workflows

Ric Evans-Jacquez
Research Software Engineer
UW-Madison / IceCube / WIPAC

HTC25 – June 2025

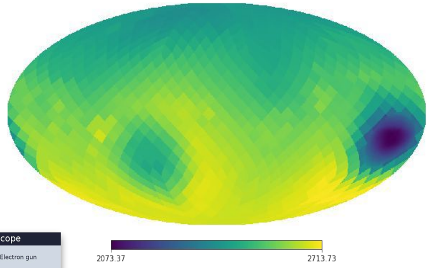
How can the Event Workflow Management System benefit me?

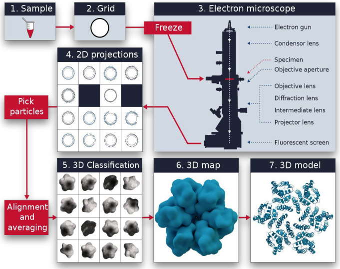
EWMS uses Condor to run workflows containing
many, many short-lived tasks

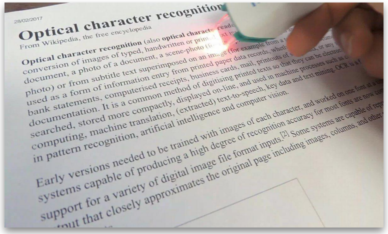
Event-Granular HTC Workflows

To be most efficient, we want to subdivide a workflow into **“smallest” unit of work**, **the event**

- Multi-Messenger Astrophysics alerts (IceCube and LIGO triggers)
- Astronomical observations (images)
- Cryogenic electron microscopy (cryo-EM) data
- Optical Character Recognition on pages in a book
- …







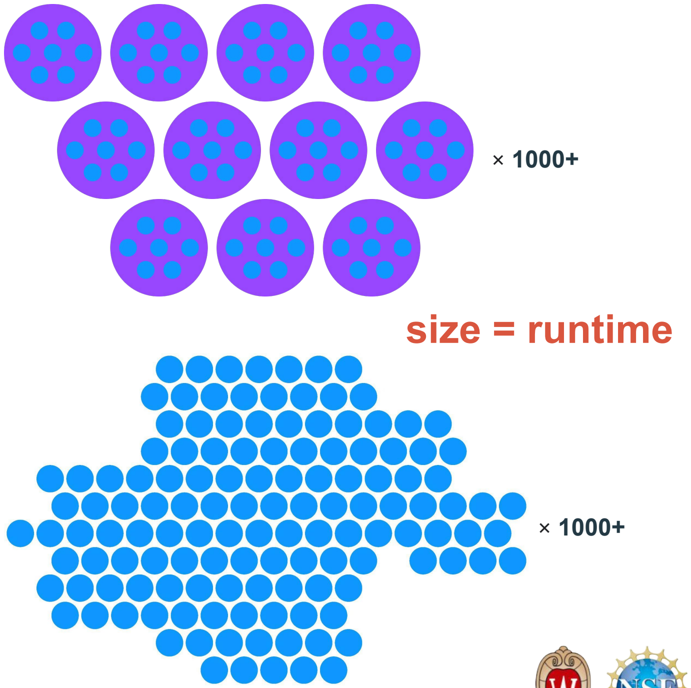
HTCondor’s Traditional Use

HTCondor is great at aggregating distributed resources and orchestrating workflows, but…

- ➢ Imposes **1:1 job-task** mapping
- ➢ Needs O(>30 min) jobs to be most efficient
    - ○ **Task lifetime >> Startup+Scheduling time**

In contrast, *events*…

- ➢ Much **shorter runtime per task**
- ➢ **1:N job-task** mapping
- ➢ **Dynamic allocation** of inputs and outputs



Condor is a massive bird, but we have Starlings

Andean Condor

Condor-shaped flock of Starlings

(not AI)

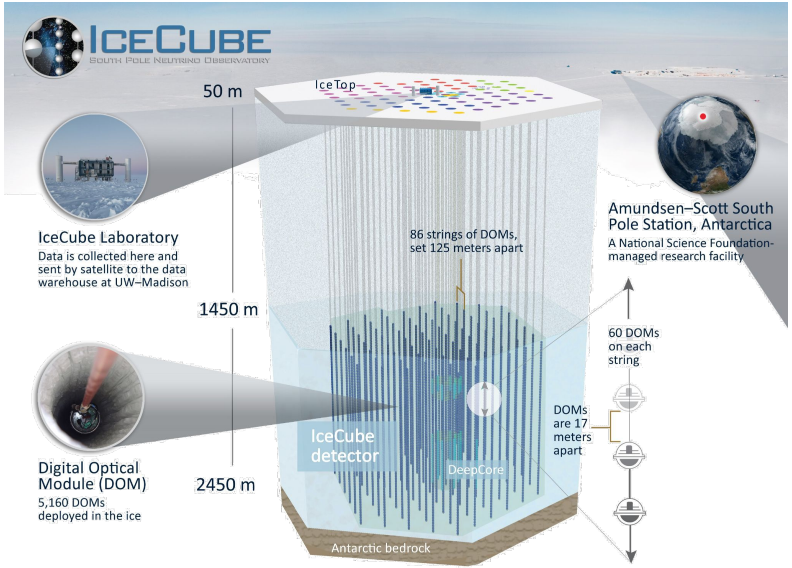
The First EWMS Application

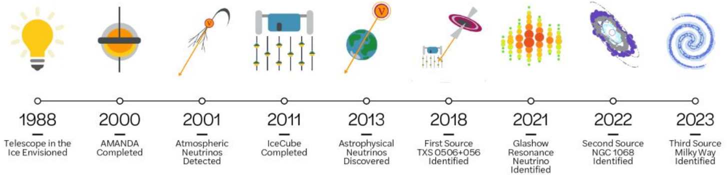
IceCube Neutrino Observatory's Realtime Alert System

Background

IceCube Neutrino Observatory

The IceCube Neutrino Observatory is a **cubic kilometer detector** located at the geographic South Pole, and the **premier facility** for identifying neutrinos > 10 GeV, particularly > 1 TeV **astrophysical neutrinos.**



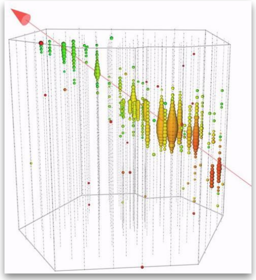


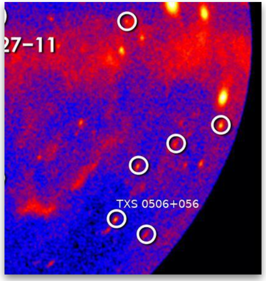
Background

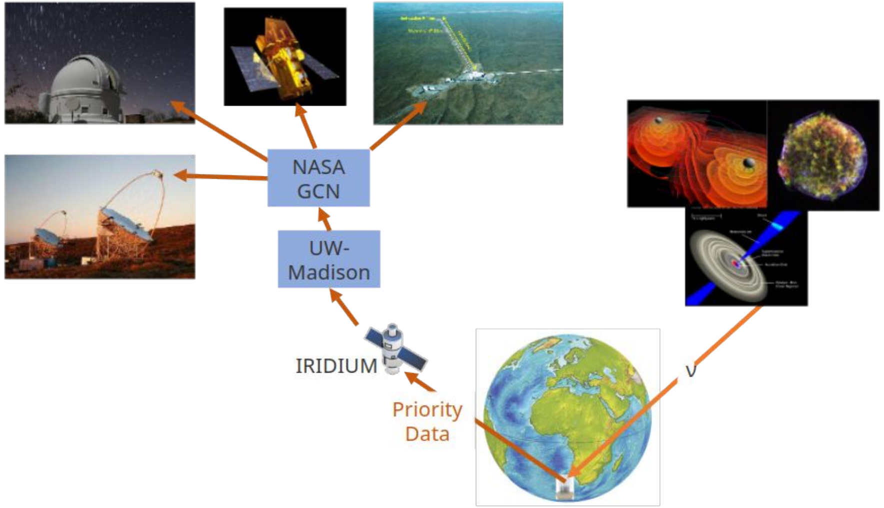
A neutrino is detected by IceCube!

Where did it come from?

Need to know where to point other telescopes for **immediate** follow-up observations.

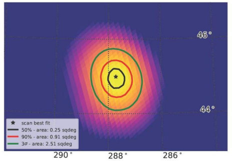




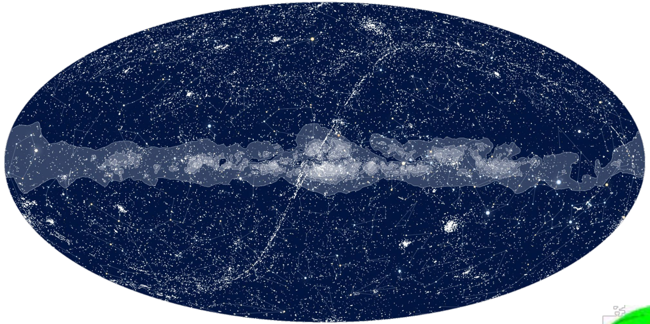


IceCube's Problem

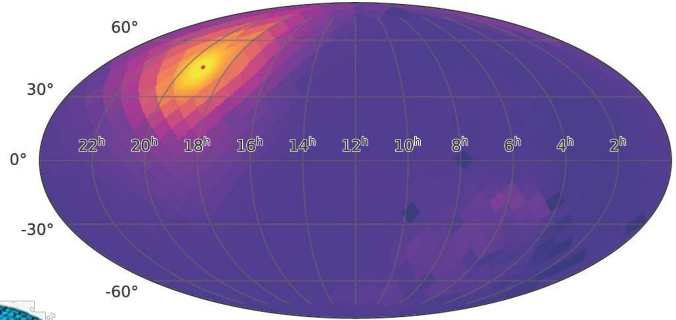
Need to reconstruct a *Sky Map*

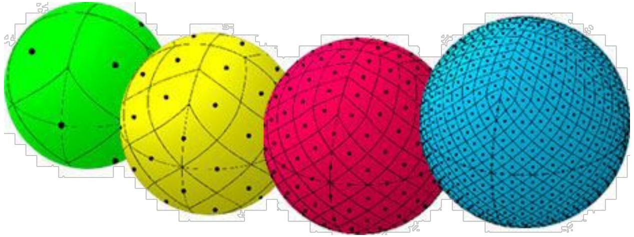


Most accurate and detailed directional reconstruction comes by scanning across the sky in *varying* granularity: O(100k) total pixels



"night sky"

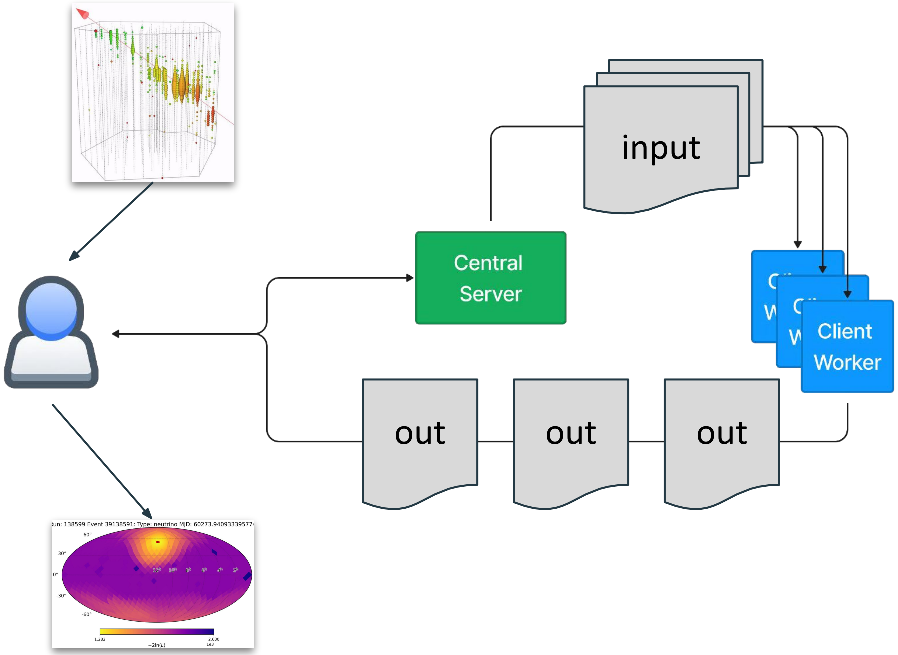




HEALPix algorithm

The Original Solution

1. **Preempt** *N* HTCondor nodes for **immediate availability**
2. **Generate** O(100k) **events** (5-tuples)
3. **Group** O(1k) events into ***N* "input" object**
   - ➢ 1 job gets 1 object, O(1k) events
4. **Submit** to HTCondor for ***N* jobs**
5. **Wait** for every job to finish while collecting *N* transferred output objects
6. **Assemble** resulting skymap
   - ➢ Produce the most **probable direction** and error



Are you using something similar?

The Three Realities of HTC

1. **We have a heterogeneous and finite compute pool – you cannot group input events efficiently ahead of time.**

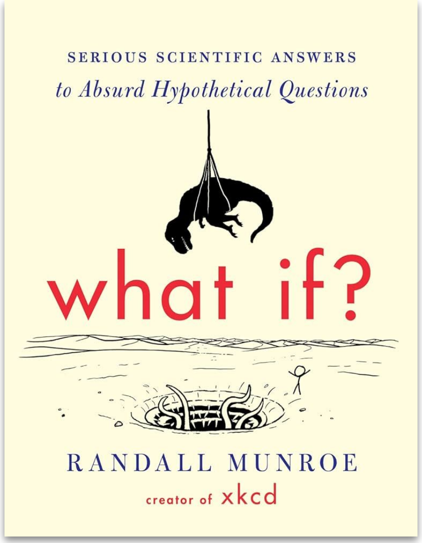
    What if you didn't have to?

2. **Task processes / CPUs will fail – even tested software.**

    What if you didn't have to rerun an entire job if the very last event fails?

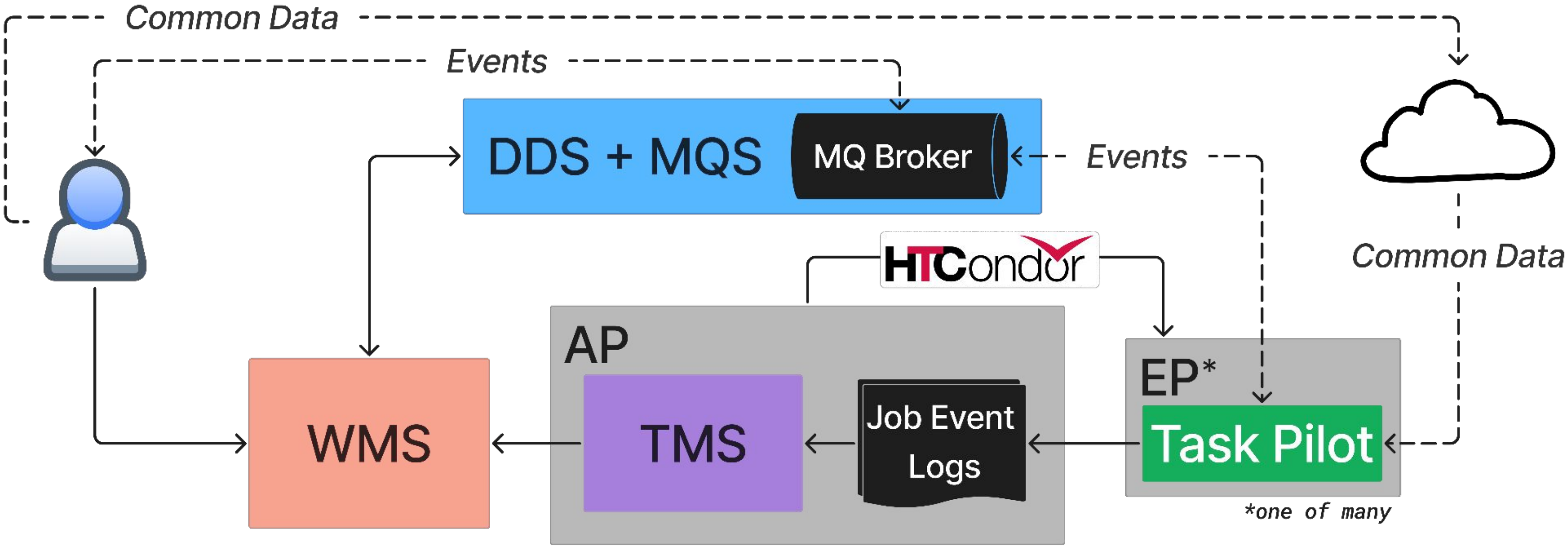
3. **Less-than-ideal job availability is unavoidable.**

    What if you didn't have to wait for your last job?



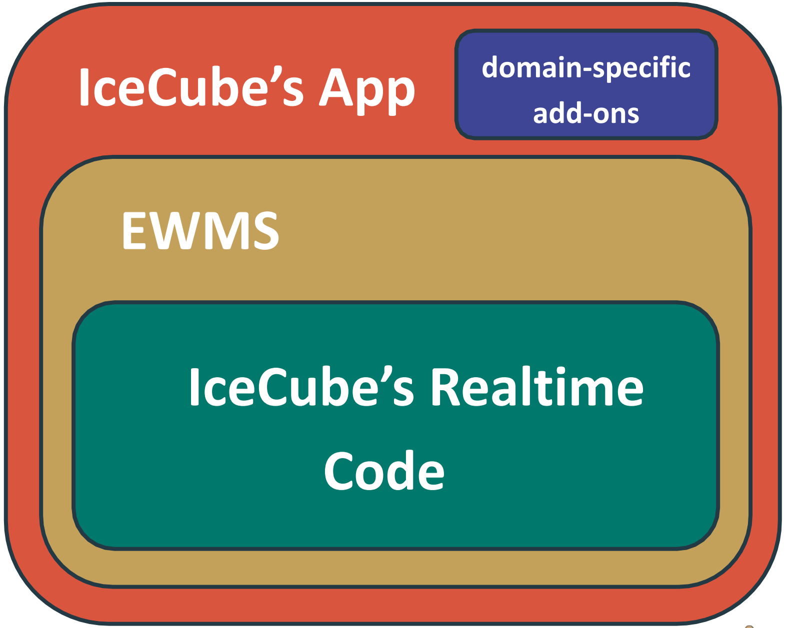
fun book… not an ad…

EWMS Design


Common Data
Events
DDS + MQS
MQ Broker
Events
HTCondor
Common Data
AP
TMS
Job Event Logs
EP*
Task Pilot
WMS
*one of many

IceCube Outsources HTCondor to EWMS

- **EWMS is a domain-agnostic system.**
- IceCube uses EWMS with an **external domain-specific server,** called “SkyDriver”.
- SkyDriver tells EWMS to tell HTCondor to **run physics code.**



Today

IceCube Outsources HTCondor to EWMS

(also not AI)

Penguin-shaped flock of Starlings

NATIONAL GEOGRAPHIC

Emperor penguins march along the ice on Cape Washington on Antarctica's Ross Sea. Contrary to popular belief, these charismatic birds do not live in Antarctica's vast interior, including at the South Pole—they stick to the coast.

PHOTOGRAPH BY PAUL NICKLEN, NAT GEO IMAGE COLLECTION

ANIMALS | EXPLAINER

Penguins don't live at the South Pole, and more polar myths debunked

We're setting the record straight on which cold-dwelling creature lives where.

By Brian Handwerk

February 3, 2020 • 11 min read

¯_(ツ)_/¯

Sounds neat… what do I need to do?

It’s all about event I/O

Event I/O

Dynamically-allocated inputs, outputs, and workforce

EWMS does not use HTCondor's file-transfer system (for events)

- ➢ 1:N tasks are complex
- ➢ No dynamic scaling task per job

Instead, EWMS uses message passing (HTTP + RabbitMQ)

- ➢ **Separates event I/O from job mechanics**
  - Additional **input(s)** are given when needed
  - **Outputs are immediately** relayed in real-time
- ➢ **Doesn't care about fluxuations in job count**
  - ***Can we increase/decrease number of jobs?***

(still not AI)

Starlings are not fed, they forage

Pilot-Based Workers

Ack-last & fail-fast paradigm

- ❖ Acknowledge input event only when task is done
- ❖ MQ will redeliver to another worker when no ack
- ❖ "Dead Letter" queue for problem events

The Pilot's built-in failover mechanism makes the workflow natively resilient to CPU crashes

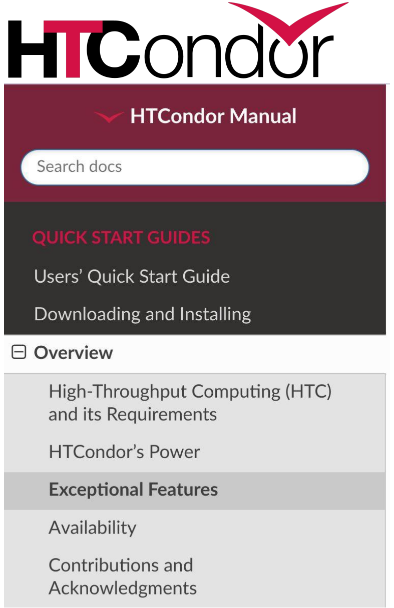
Backward compatible – invisible from user's POV

- ❖ User code interfaces with files

EWMS complements HTCondor

A few of HTCondor's *Exceptional Features*:

- ➢ Guaranteed execution
- ➢ Extreme scalability
- ➢ Parallelization without reimplementation
- ➢ Success in heterogeneous environments
- ➢ Adaptable to user requirements



Paraphrased from the HTCondor Manual

Whether User or Bot, it's the same

If our system is not flexible to adopt, then it won't be used!

HTTP / REST user interface

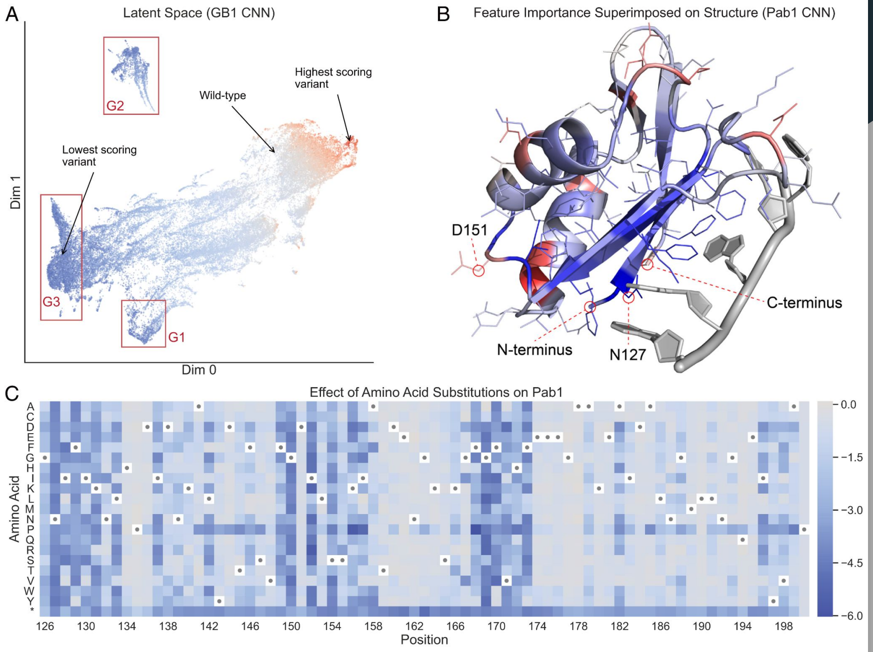
- ➢ Standardized **JSON input** – auto-documented
  - Not unlike HTCondor submit syntax
  - Validation by **JSON Schema & OpenAPI**
- ➢ **Multiple image versions** available
  - Allows users to **test customizations** (also scientific reproducibility)
  - Apptainer (optionally, Docker)

```
post_body = {
    "public_queue_aliases": ["input-queue", "output-queue"],
    "tasks": [
        {
            "cluster_locations": ["sub-2"],
            "input_queue_aliases": ["input-queue"],
            "output_queue_aliases": ["output-queue"],
            "task_image": "icecube/skymap_scanner:3.20.3",
            "task_args": "client --in {{INFILE}} --out {{OUTFILE}}",
            "environment": {},
            "n_workers": n_workers,
            "worker_config": {
                "max_worker_runtime": 60 * 10,
                "worker_disk": "512M",
                "worker_memory": "512M",
            },
        }
    ],
}
resp = await rc.request( method: "POST", path: "/v0/workflows", post_body)
```

Future CHTC Use Case #1

Using protein language model to drive engineering

- Enormous space of options - millions of potential variants
- Very small individual inference task
- Additional tasks can be added in discovered regions of interest



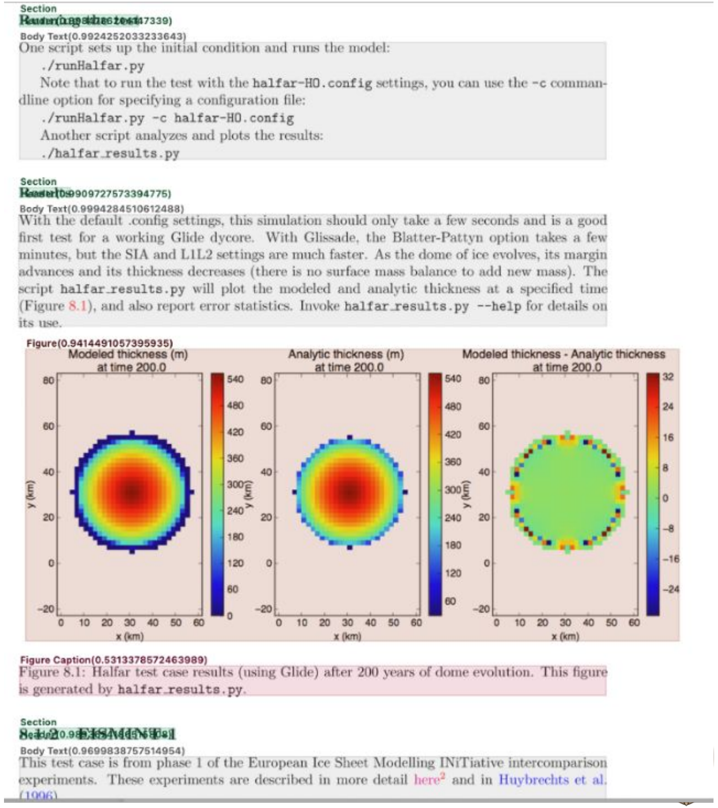
"Neural network interpretation. (A) A UMAP projection of the latent space of the GB1 sequence convolutional network (CNN), as captured at the last internal layer of the network. In this latent space, similar variants are grouped together based on the transformation applied by the network to predict the functional score." - from 10.1073/pnas.2104878118

- Ian Ross

Future CHTC Use Case #2

Scientific literature processing and data-mining (xDD)

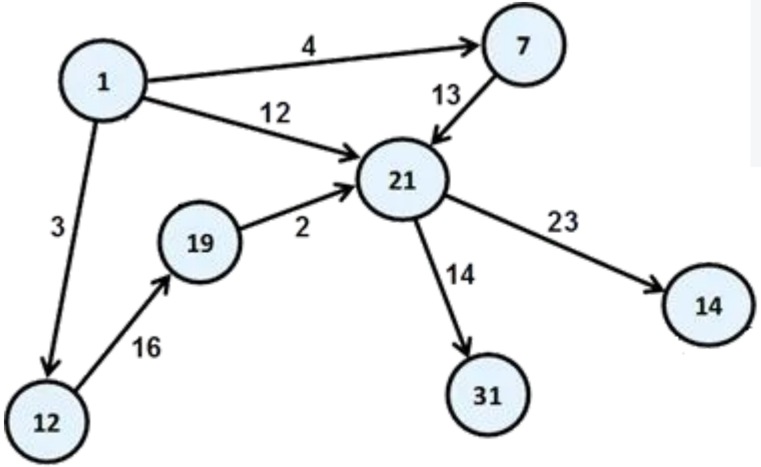
- **Stream of new articles,** each of which goes through processing:
  - Visual parsing (extract tables, figures, equations, captions, …)
  - Paragraph chunking and text embedding (for retrieval augmented generation "ask-xDD" agent)
- Occasionally deploy new processing workflows or applications across **~18M documents**
- "Standard" approach: **batch documents,** submit regularly
  - Individual documents are **O(min)** for each processing pipeline
  - Some docs will stop progress on the batch, resulting in need for re-bundling and re-submission

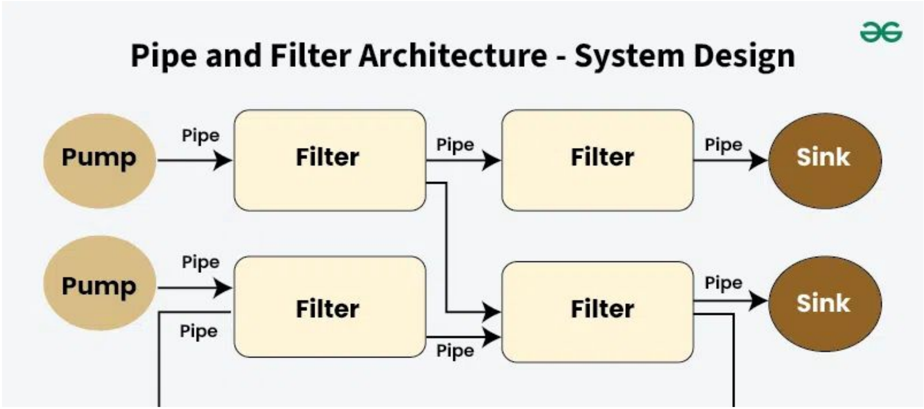


- Ian Ross

Future

Other Workflow Geometries





Node ➡ Task Type

Edge ➡ Event Message Queue

Node Weight ➡ Task Priority (workforce size)

Edge Weight ➡ Event Frequency

Acknowledgements

Thank You!

PIs

- Miron Livny
- Brian Bockelman
- Benedikt Riedel

Software Engineering

- Ric Evans (me)
- Benedikt Riedel
- David Schultz

CHTC

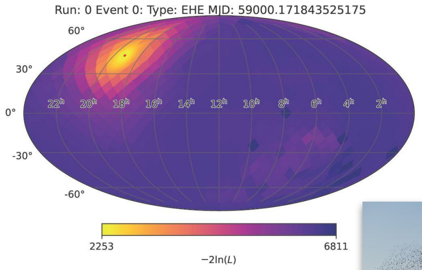
- Brian Aydemir
- Ian Ross

IceCube

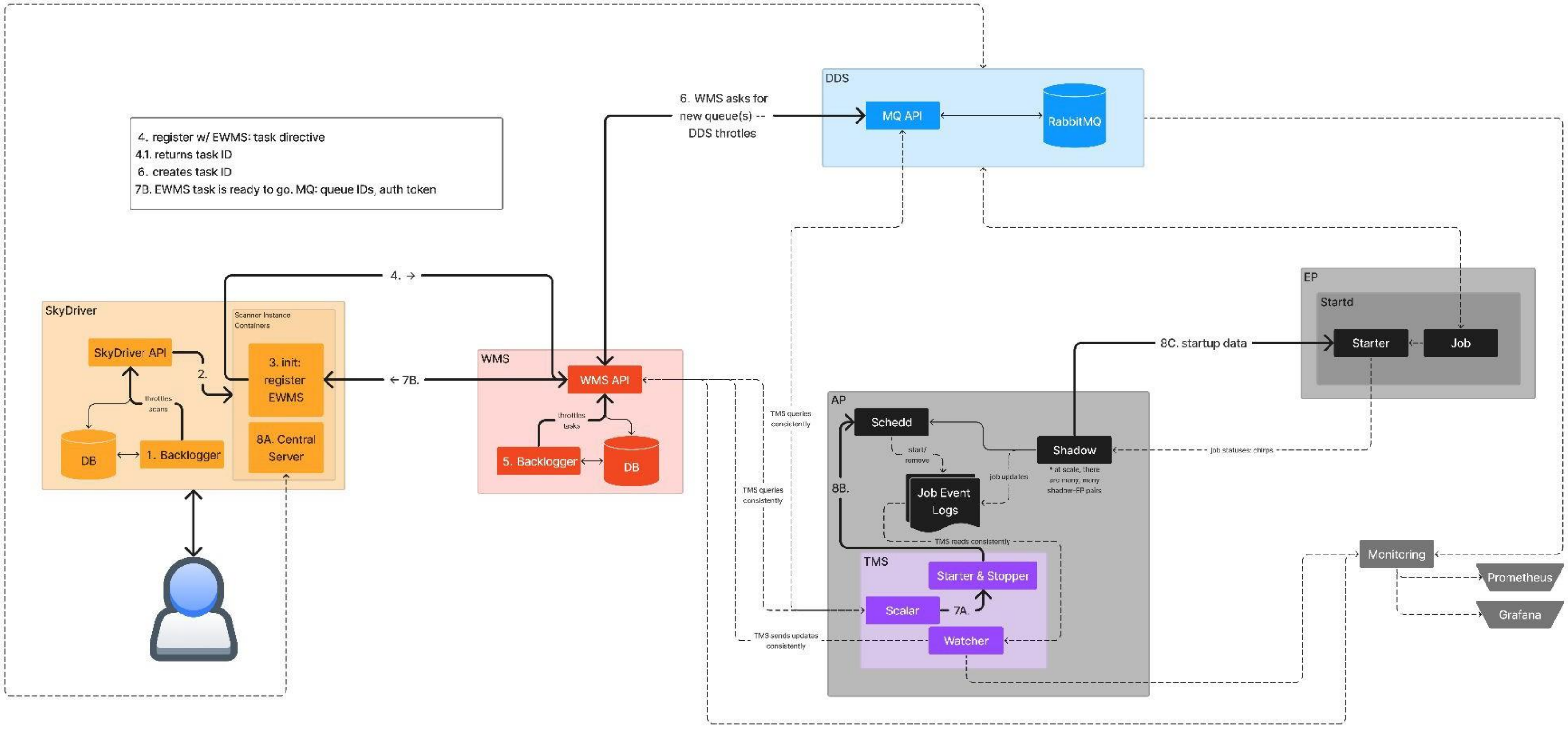
- Tianlu Yuan
- Massimiliano Lincetto
- Claudio Kopper
- Erik Blaufuss
- Christina Lagunas
- Robert Stein
- Giacomo Sommani
- Angela Zegarelli

National Science Foundation

- OAC #2103963
- OPP #2042807

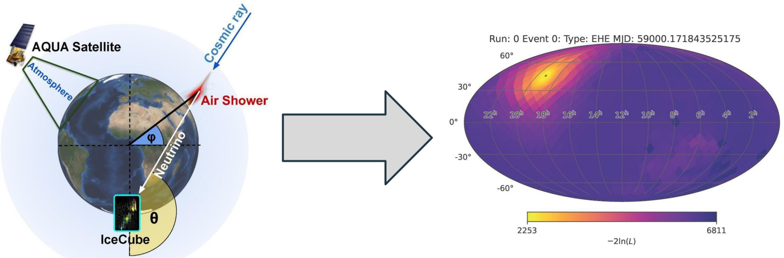


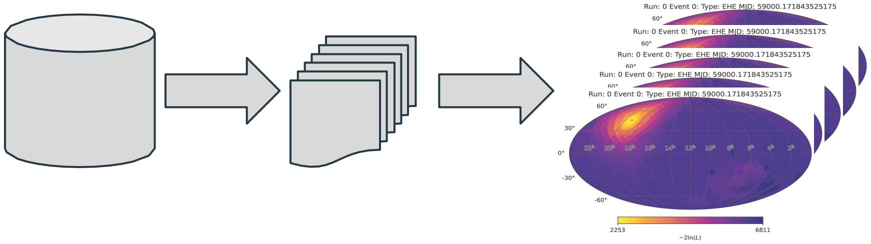
SkyDriver-EWMS Architecture



Two IceCube Use Cases

CASE 1: Massive Scale	**CASE 2: Moderate Scale**
Real-time Scans	*Historical Catalog & Simulation*
Fast & Resource Intensive -> High Priority	Steady/Predictable -> Lower Priority
➔ O(10k+) CPUs, spun up ASAP	➔ Varying # of CPUs, subject to availability





SkyDriver – Worker / Scanner Client POV