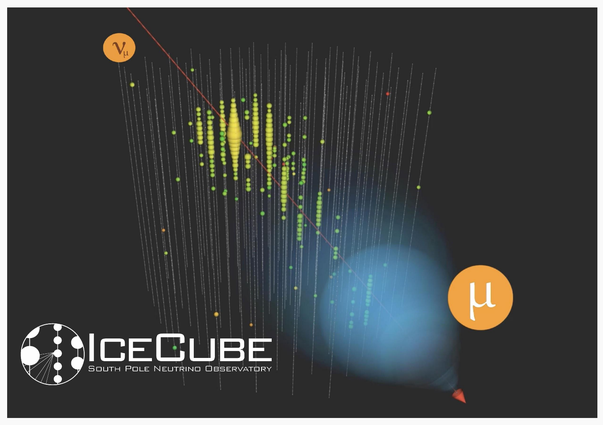
IceCube Experience using XRootD-based Origins with GPU workflows in PNRP

David Schultz[1], Igor Sfiligoi[2]

[1] Wisconsin IceCube Particle Astrophysics Center, University of Wisconsin–Madison

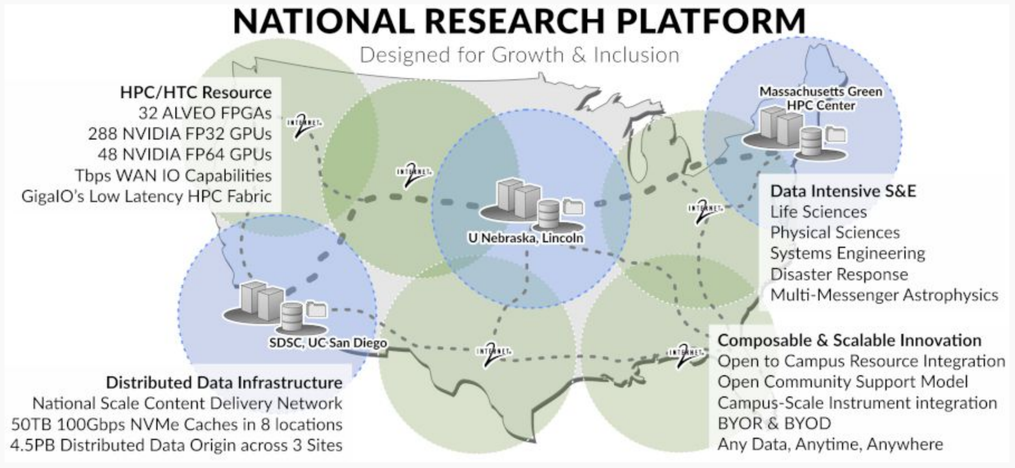
[2] San Diego Supercomputer Center, University of California San Diego

The IceCube Neutrino Observatory is a cubic kilometer neutrino telescope located at the geographic South Pole. Understanding detector systematic effects is a continuous process. This requires the Monte Carlo simulation to be updated periodically to quantify potential changes and improvements in science results with more detailed modeling of the systematic effects. IceCube's largest systematic effect comes from the optical properties of the ice the detector is embedded in. Over the last few years there have been considerable improvements in the understanding of the ice, which require a significant processing campaign to update the simulation.



IceCube normally stores the results in a central storage system at the University of Wisconsin–Madison, but it ran out of disk space in 2022. The Prototype National Research Platform (PNRP) project thus offered to provide both GPU compute and storage capacity to IceCube in support of this activity.

As part of using PNRP, the storage access was provided via XRootD-based origins, a first for IceCube computing. We report on the overall experience using PNRP resources, with both successes and pain points.



Setup of storage:

1. Create XRootD Origin, with a base path that accepts SciTokens from IceCube
2. Add entry in Topology for the XRootD origin

Using storage:

1. Generate a SciToken using HTCondor SciToken CredD
   - This will automatically update access tokens in jobs as needed
2. Add osdf:// urls to condor file transfer
3. Add to job requirements:
   regexp("osdf",HasFileTransferPluginMethods)

Job matching:

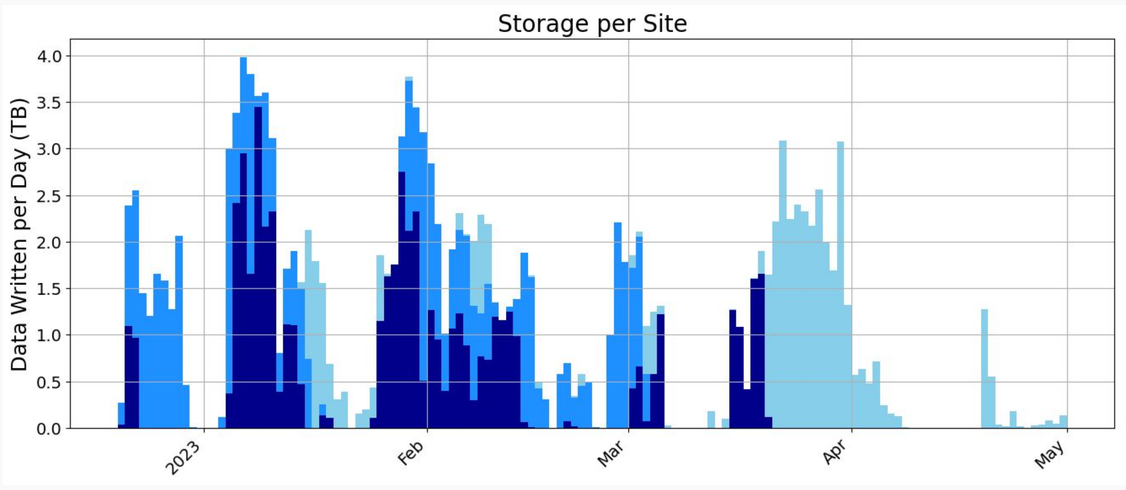
- Force use of site local Origin at the two large GPU sites (Nebraska and Massachusetts)
- All other sites write data to the SDSC Origin

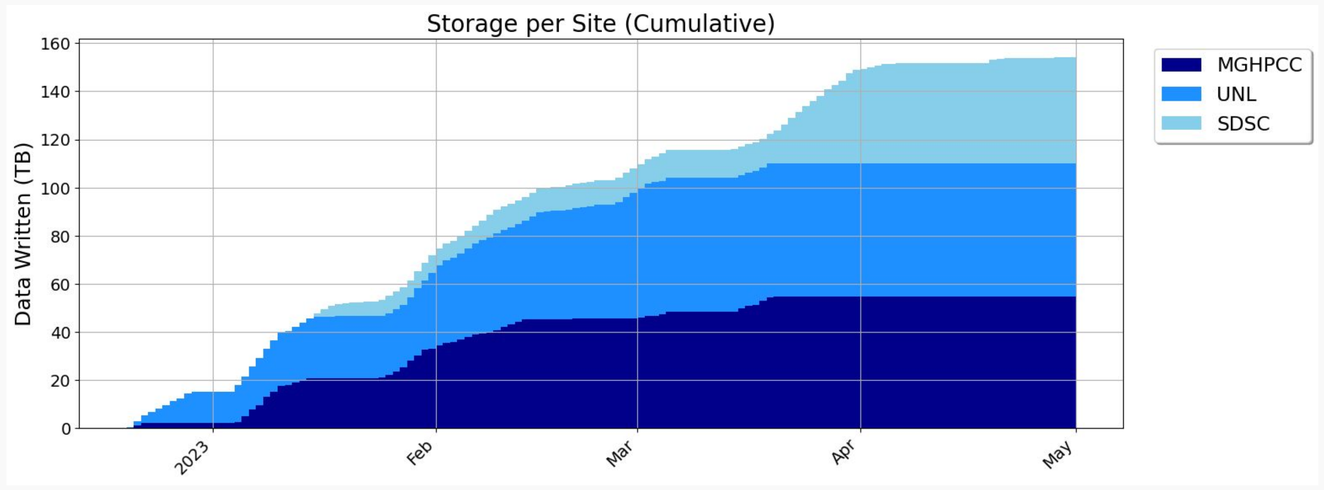
HTCondor SciTokens Config:

```
LOCAL_CREDMON_PROVIDER_NAME = scitokens
JOB_TRANSFORM_NAMES = $(JOB_TRANSFORM_NAMES) AddSciToken
JOB_TRANSFORM_AddSciToken @=end
[
Requirements = (JobUniverse =?= 5 && ifThenElse(isUndefined(NeedsOSDF), False,
NeedsOSDF));
Eval_Set_OAuthServicesNeeded = strcat( "scitokens ", OAuthServicesNeeded ?: "");
]
@end
# Change this to match the OSDF issuer name
LOCAL_CREDMON_ISSUER = https://chtc.cs.wisc.edu/icecube
LOCAL_CREDMON_TOKEN_AUDIENCE = ANY
# Change this for the paths your token should write into, relative to the IceCube
root
LOCAL_CREDMON_AUTHZ_TEMPLATE = read:/production write:/production
LOCAL_CREDMON_PRIVATE_KEY = /etc/condor/.secrets/scitokens_private.pem
# Default value
# LOCAL_CREDMON_TOKEN_LIFETIME = 1200
# Change this to match the key ID generated
LOCAL_CREDMON_KEY_ID = 7672
```

Usage Experience

Since this was one of the first workflows with PRNP storage, the setup took several weeks fiddling with the details until it worked consistently. It has been fairly stable, with some pain points all around the storage interface. When there were transfer errors, it wasn't clear what had failed or why. Additionally, there seems to be a delay between job submission and token generation that can result in a rejected job submission by HTCondor. The CPU and GPU computing side of PNRP was near perfect in usage, well above Open Science Grid standards.





Acknowledgements

This work was partially funded by the U.S. National Science Foundation (NSF) under grants OAC-1826967, OAC-1541349, OPP-2042807, OAC-2030508, OAC-2112167 and CNS-1730158.