AGU American Geophysical Union · AGU24 Washington, D.C. | 9–13 December 2024 · IN13B

Open-Source Data Engineering at NASA: CCMC's Approach to Managing Petabyte-Scale Heliophysics Data

Matthew Lesko[1], Damian Barrous-Dume[1], Masha Kuznetsova[2], Polymnia Manessis[3], M. Leila Mays[2], Phil Poole[3], Karen Scheiber[3], Edgar Russell[4], Tina Tsui[2] and Chinwe C. Didigu[3]

(1) Community Coordinated Modeling Center, NASA GSFC, Navteca, Greenbelt MD, United States, (2) NASA GSFC, Community Coordinated Modeling Center, Greenbelt MD, United States, (3) Community Coordinated Modeling Center, NASA GSFC, Adnet, Greenbelt MD, United States, (4) Community Coordinated Modeling Center, NASA GSFC, Eclipse Technical Systems, Greenbelt MD, United States

Open Science

The Four Pillars of the Open Source Science Initiative (OSSI)

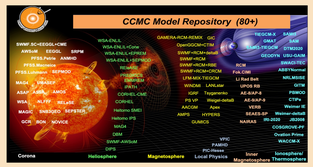
https://nasa-impact.github.io/ossi-website/

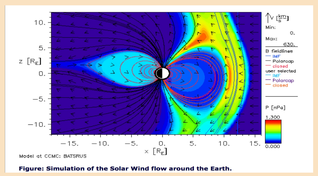
- OPEN (**TRANSPARENT**) SCIENCE: scientific process and results should be visible, accessible, and understandable
- OPEN (**ACCESSIBLE**) SCIENCE: data, tools, software, documentation, and publications should be accessible to all (FAIR)
- OPEN (**INCLUSIVE**) SCIENCE: process and participants should welcome participation by and collaboration with diverse people and organizations
- OPEN (**REPRODUCIBLE**) SCIENCE: scientific process and results should be open such that they are reproducible by members of the community

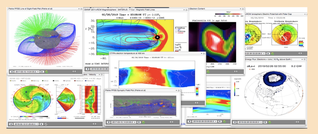
Models at CCMC

The CCMC hosts over 80 space weather models covering the solar, heliosphere, magnetosphere, ionosphere, and thermosphere.

Models are run both on-demand ("run-on-request" or "instance runs") and Continuous/Real-Time with the latest instrument data and observed phenomena.







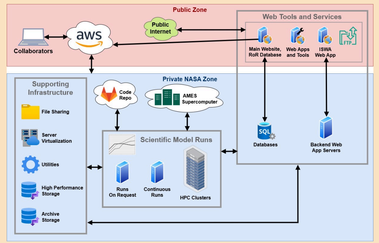
Thank you to all the CCMC staff!
https://ccmc.gsfc.nasa.gov/staff/

About the CCMC

The Community Coordinated Modeling Center (CCMC) is a pioneer in Open Science. The CCMC began as a multi-agency partnership in 2000 with the core vision to provide open and equal access of the latest modeling capabilities to all to advance space weather and space science research. All model runs and information stored at the CCMC are freely searchable and accessible throughout the world. The CCMC believes the path to Open Science must be fostered with both internal and inter-agency collaborations. With this in mind at the CCMC we offer environments for researchers to build collaborative in AWS.

CCMC Hybrid Cloud Infrastructure

The CCMC has significant resources available to the public (light pink) and internally (light blue). The commercial cloud (AWS) and NASA high-performance computing (HPC) clusters outside of CCMC are used frequently. The CCMC infrastructure is fundamentally hybrid across multiple datacenters and the cloud.

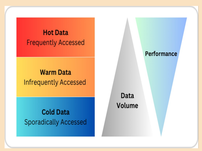


Data Complexity

Modern scientific modeling depends on a continuous stream of observations. These observations are canonically "Big Data": incredible amounts of "Volume, Velocity, and Variety". While the size of datasets and the transfer speeds are difficult enough, it is the "Variety" that quickly overwhelms.

Our work at the CCMC requires (HPC) clusters and data ingestion, transformation, assimilation and distribution across multiple computing environments, including AWS. Not only must CCMC collects observational data from many locations, but we must often transform and move the data across computing environments.

Building on the CCMC's experience with "DevOps" and "infrastructure as code", the CCMC has built a platform to support our multitudinous and ever-growing data movement tasks. The tools, processes, and methods are often referred to data engineering or data orchestration.



Our data orchestration platform enables data tiering: categorizing it by access frequency. We metaphorically refer to the urgency of access as the data "temperature": hot, warm, or cold.

The CCMC develops the platform in GitLab, an open-source collaborative software development environment. Our data orchestration projects use the full spectrum of GitLab features that have already improved our software delivery process. We require Merge Requests (also known as Pull Requests) for code review, attribution, and documentation. We use GitLab's continuous integration/delivery (CI/CD) feature, automating testing of our code and then moving that code to production once demonstrated sufficiently correct.

Our data orchestration tool is open-source Apache Airflow. Apache Airflow is used to create, track, alert, and remediate our data workflow. We also use and contribute to the open-source tools `s3watcher` and `fswatcher`: they watch for data arrival in AWS S3 or a Linux filesystem.

Our observability tools of choice are Grafana and Prometheus. These two open-source tools collect, aggregate, transform, and present time-series metrics in interactive dashboard, and can alert operators when anomalous events occur. Enhancing our observed metrics across all domains has provided surprising improvements, such as when an increase in network costs was introduced from a bug in our data transfer process, accidentally sending the same data multiple times.

Data Orchestration with Airflow

- Fig 1: Overview of the entire Airflow environment, such as number of workflows, number that are healthy/unhealthy, and so on.
- Fig 2: Workflows ("DAG") are visible from a "single pane of glass". Scientists and software developers can quickly see the name, project owner, schedule, including next and last run, and importantly the success or failure status.
- Fig 3: Clicking on a specific DAG brings up greater detail, such as historical run metrics. Besides the binary "success" or "failed", can help quickly spot trends and outliers, such as if a process is taking significantly longer than usual.
- Fig 4: A DAG refers to a **d**irected **a**cyclic **g**raph – a computer science abstraction that provides mathematical guarantees for task ordering. A workflow with a task error can be re-run from the point of error, not always from the beginning.
- Fig 5: The DAG is written in Python. An inline code editor can be used for understanding and debugging (although we store our DAGS in GitLab source control).
- Fig 6: An audit (event) log to aid debugging.
- Fig 7: Calendar View of a DAG, covering success/failures over days and weeks.

Fig1

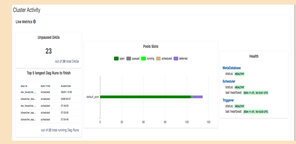


Fig2

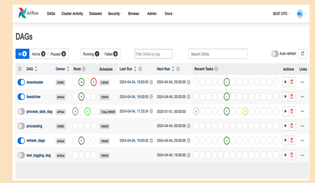


Fig3

Fig4

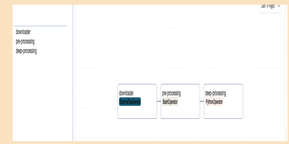


Fig5

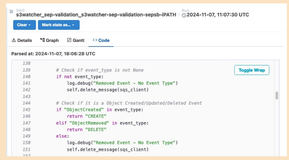
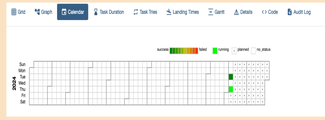


Fig6

Fig7



Monitoring and Observability with Grafana and Prometheus

Our observability platform is composed of Grafana, the visible user interface, and Prometheus, the backend service. They are used to collect, integrate, and search for time series metrics. They are separate open source projects but often combined together, as CCMC as has done.

Originally designed for computer system metrics, Grafana/Prometheus are flexible enough to work with any metric, broadly defined, leading to easy customization to an organization's critical assets.

Examples:

- Fig 1: Overview of storage and transfer of an AWS S3 bucket.
- Fig 2: Operating System metrics for an AWS server.
- Fig 3: Query Editor to interactively view your queried data (top) as query is constructed. Supports PromQL, similar to SQL, with the "Code" button. Note the Transformations Tab for modifying the queried data before presentation.
- Fig 4: Presentation of queried metrics in a table form.
- Fig 5: Custom dashboard retrieving AWS Billing data. Dashboards have wide support for visualization beyond a table or time series graph.
- Fig 6: A creative use of metrics - this dashboard summarizes our GitLab projects and level of access by our software developers.

Fig1

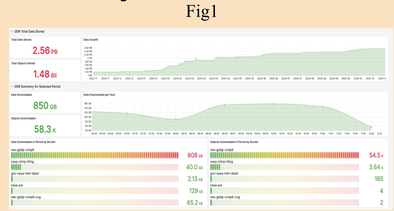


Fig2

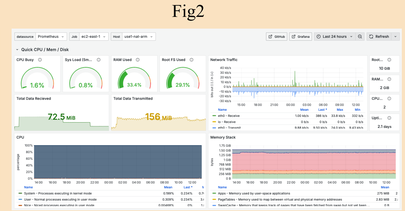


Fig3

Fig4

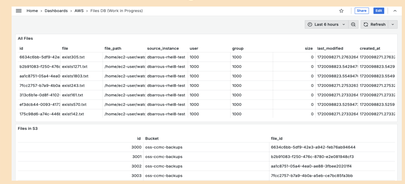


Fig5

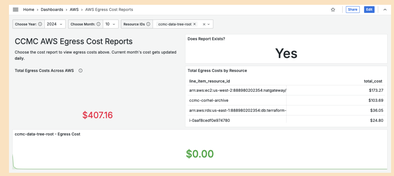
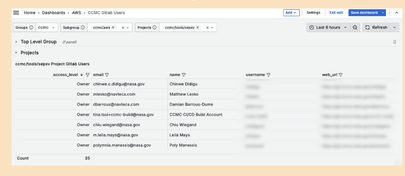


Fig6



Software & Platforms

- https://github.com/apache/airflow
- https://github.com/HERMES-SOC/fswatcher
- https://github.com/HERMES-SOC/s3watcher
- https://github.com/puppetlabs/puppet
- https://github.com/hashicorp/terraform

Visit Us!

https://ccmc.gsfc.nasa.gov