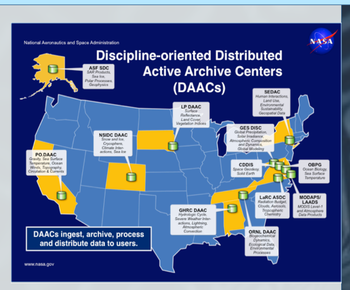
Building A Cloud Based Distributed Active Data Archive Center

Rahul Ramachandran[1], Katie Baynes[2], Kevin Murphy[3]

1. NASA MSFC, 2. NASA GSFC, 3. NASA HQ

Background:

NASA's Earth Science Data System (**ESDS**) Program facilitates the implementation of NASA's Earth Science strategic plan, which *is committed to the full and open sharing of Earth science data obtained from NASA instruments to all users.* The Earth Science Data information System (**ESDIS**) project manages the Earth Observing System Data and Information System (**EOSDIS**). Data within EOSDIS are held at Distributed Active Archive Centers (**DAACs**). One of the key responsibilities of the ESDS Program is to *continuously evolve the entire data and information system to maximize returns on the collected NASA data.*



Drivers for moving to the Cloud

- 2015 EOSDIS review to identify gaps recommended to investigate two areas related to commercial cloud computing and storage:
  - Do cloud providers offer potential for storage, processing, and operational efficiencies?
  - Will it lead to potential development of new data access and analysis paradigms?
- In response, ESDS initiated prototypes investigating advantages and risks of leveraging cloud computing: "Cumulus" project

Cumulus Project

- *Goal:* To design and develop a functional "light weight" data ingest, archive and distribution "cloud native" framework

- *Objectives:*
  - Demonstrate core DAAC (ingest/processing/ archive/distribution) functions can be performed on a commercial cloud
  - Demonstrate Cumulus can operate within NASA's security compliance policies
  - Provide cost estimates from running a few data streams
  - Provide options for future operational strategy
  - Provide planning tool for DAACs to enable transition to Cumulus
  - Provide guide for future Cumulus operations by DAACs

Functional Requirements

- *Data Acquisition:* identify new data products and retrieve data for ingest
- *Data Ingest:* product validation for data integrity, provide common preprocessing and allow DAACs to add custom ingest preprocessing
- *Metadata Publication:* extract or generate granule-level metadata and link with collection-level metadata
- *Data Archive:* optimize data-files archives based on input parameters
- *Data Distribution:* support several distribution methods for controlling egress charges to users and provide hooks to build new distribution/access/analysis applications that can link to data store

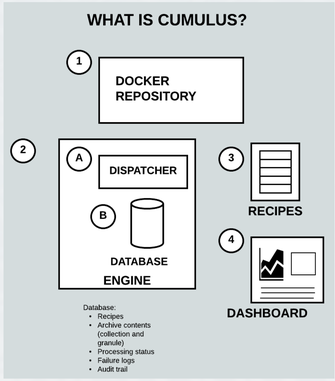
System Requirements

- Be extensible to allow addition of new functionality in phases and comply with required standards
- Allow DACC operators to add/configure data streams and define rules-driven data ingest, including preprocessing to meet DAAC Levels of Services needs
- Optimize performance and overall cloud costs
- Interoperate with existing core EOSDIS components

Cumulus:

What is Cumulus?

- Lightweight framework consist of:
  - **Repository of Docker Images** (stand alone functions) that can constitute steps in a recipe
  - **Orchestration Engine** (Service Endpoint) that controls invocation of docker images
    - Database to store recipes
  - **Recipe(s)** or Configuration file(s) that define ingest processing, publication, and archive operations
  - **Dashboard** to allow operators to create and execute recipes, to check status of execution, and to track errors



Cumulus Trade Offs

Advantages

- Maintains thematic data stewardship expertise to support specific community needs
- Collocates compute and data to enable new science and application of large scale analysis
  - Opens innovation around data by other communities
  - Collocation of data supports cross thematic interdisciplinary science
- Common framework reduces redundant tools/services and enables sharing
- Brings to bear economies of scale
- Enables consistent performance

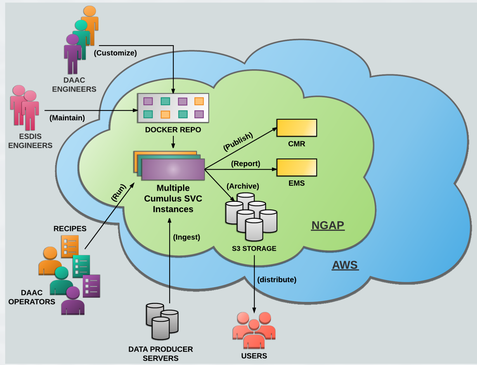
Disadvantages

- Perceived lack of control over data (Golden copy issues)
- Will require a systematic transition
- Requires modifications in existing operations
  - New governance policies and procedures are needed
- Retraining existing staff with different skill set
- Possible vendor lock-in

Cumulus Deployment Model

- DAACS will have access to the Cumulus code base and will use it to deploy and run Cumulus instances on secure cloud (NGAP). Both engineering and operation responsibilities fall on the DAAC.

- *Operational Model*
  - DAACS are responsible for most of the engineering and all of the operations
  - ESDIS controls, monitors, and pays for cloud resources
  - ESDIS also contributes to Cumulus core code base



This model may change to a **single instance** as it is likely the Orchestration Engine will be replaced by a native AWS infrastructure service (Step Functions)

Development Status

- Designed and developed in 4 incremental phases with each phase typically consisting of 6 development springs
  - Each phase focuses on 1 or 2 data streams from a specific DAAC to test ingest streams based on their representativeness for certain instrument platform
- For Phase 1, a set of airborne datasets from the
- Global Hydrology Resource Center DAAC was selected
  - Use case tests Cumulus ability to handle a wide variety of datasets typical for data ingest requirements of an airborne field campaign
- Prototype activity is currently almost through Phase 2 of the 4 phases

Lessons Learned:

- *Cost savings potential for future airborne data streams*
  - Hardware procurement for HS3 airborne field campaign was based on best case estimates of data storage but the total data storage was far less resulting in sunk costs
  - Cost savings on hardware alone would justify managing data streams for future field campaigns using the cloud –based Cumulus data framework
- *Detailed on-boarding process will be required*
  - Current operational code used at DAACs predates existence of Docker so transition to cloud require time as well as training in order to convert existing DAAC processing software