CUMULUS: NASA'S CLOUD BASED DISTRIBUTED ACTIVE ARCHIVE CENTER PROTOTYPE

Rahul Ramachandran, Katie Baynes, Kevin Murphy
NASA MSFC, NASA GSFC, NASA HQ
Ali Reza, Ian Shuler
DevSeed
Dan Pilone
Element84

Abstract
NASA's Earth Science Data System (ESDS) Program serves as a central cog in order to facilitate the implementation of NASA's Earth Science strategic plan. Since 1994, the ESDS Program has committed to the full and open sharing of Earth science data obtained from NASA instruments to all users. 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. An independent review was conducted in 2015 to holistically review the EOSDIS in order to identify gaps. The review recommendations were to investigate two areas: one, whether commercial cloud providers offer potential for storage, processing, and operational efficiencies, and two, the potential development of new data access and analysis paradigms. In response, ESDS has initiated several prototypes investigating the advantages and risks of leveraging cloud computing. This paper describes one such prototyping activity named Cumulus. Cumulus is being designed and developed as a "native" cloud-based data ingest, archive and management system that can be used for all future NASA Earth science data streams.
Cumulus will foster design of new analysis/visualization tools that can leverage collocated data from all of the distributed DAACs as well as elastic cloud computing resources.

1. INTRODUCTION

One of NASA's key Earth science objectives, to "advance knowledge of Earth as a system to meet the challenges of environmental change, and to improve life on our planet" [1], addresses questions such as, "How is the global Earth system changing?," "What are the key causes of these changes?," and "What will the changes look like in the future?". NASA's Earth science program, known as the Earth Science Division, endeavors "to develop a scientific understanding of Earth's system and its response to natural or human-induced changes, and to improve prediction of climate, weather, and natural hazards" [1]. Within the Earth Science Division, the Earth Science Data System (ESDS) Program serves as a central cog in order to facilitate the implementation of NASA's Earth Science strategic plan [2]. The ESDS Program manages the lifecycle of Earth science data, providing system infrastructure and data services, which maximize the scientific returns for research and applied science from NASA's satellite missions and field campaign experiments. Since 1994, the ESDS Program has committed to the full and open sharing of Earth science data obtained from NASA instruments to all users as soon as the data becomes available for use.

Additionally, the ESDS Program manages the Earth Observing System Data and Information System (EOSDIS) in order to support science investigators seeking to understand Earth as a system. The EOSDIS consists of Science Investigator-Led Processing Systems (SIPS) focused on processing mission data from instruments, a common set of tools and services to support data discovery, accessibility, and usability, and twelve Distributed Active Archive Centers (DAACs) that provide science discipline specific knowledge for curation and support and are responsible for data stewardship [3]. DAACs [4] are located at different host organizations that are widely recognized by the science community for their expertise in the relevant disciplines. Each DAAC supports a data user community with their specific science needs for data discovery, access, and use.

One of the key responsibilities of the ESDS Program ("ESDS") is to continuously evolve the entire EOSDIS to maximize returns on the collected NASA data. An independent review was conducted in 2015 to holistically review the EOSDIS in order to identify gaps (or, areas for potential efficiencies and enhanced capabilities), specifically focusing on optimizing common data operations across DAACs. The recommendations from the review panel suggest that ESDS investigate two areas: one, whether commercial cloud providers offer potential for storage, processing, and operational efficiencies (including improved cross-DAAC collaboration), and two, the potential development of new data access and analysis paradigms. In response to these recommendations, ESDS has initiated short-lived (18 months) prototypes investigating the advantages and risks of leveraging cloud computing along three dimensions: data storage (i.e., *Does a commercial cloud platform provide access benefits and cost savings?*), data processing (i.e., *Would the use of elastic computing provide processing efficiency as well as cost savings?*), and data access (i.e., *By collocating data and compute, would it enable new data analytics capabilities including on-demand processing?*).

As a result, three key prototypes are currently being developed and tested. The first prototype, called NASA Compliant General Application Platform (NGAP), will provide a platform on which all the other cloud prototypes can operate. This platform will provide all the necessary security controls, tools, configuration management processes, and procurement mechanisms required in order to operate effectively in a commercial cloud environment. NGAP is being built on top of the Amazon Web Services (AWS) cloud infrastructure. The second prototype is focused on application hosting in the cloud. Several existing common software tools, including the metadata catalog (Common Metadata Repository (CMR)) and certain websites (Earthdata Web Infrastructure (EDWI)) are being migrated from on-premise servers to NGAP to improve availability, redundancy, and scalability, and performance and cost reductions. The third prototype, called Cumulus, is being developed as a native cloud-based data ingest, archive, distribution and management system. Cumulus will serve as a common data framework alongside the cloud-based architecture.

In this paper, we will describe the long term vision for Cumulus, its requirements, overall architecture, and implementation details, as well as lessons learned from the completion of the first phase of this prototype.

2. CUMULUS OVERVIEW

2.1 Vision
The Cumulus data framework, as well as the overall cloud vision, is visually presented in the figure below.
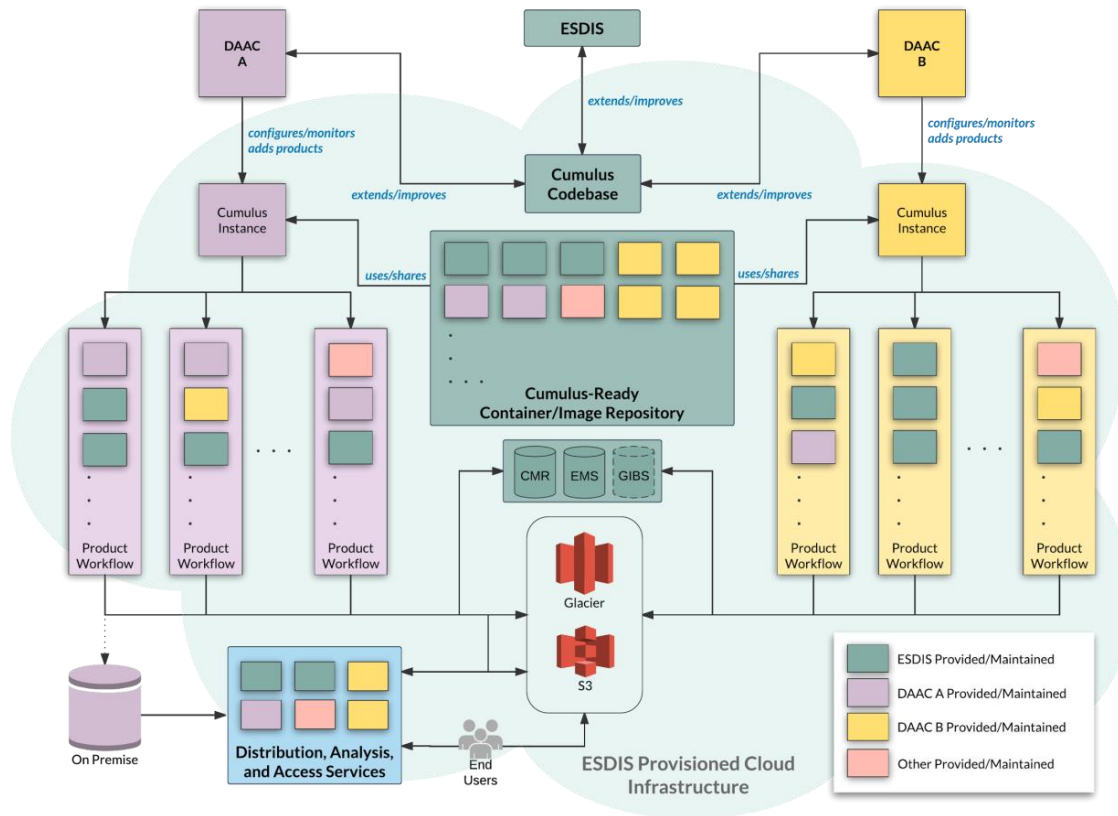

Figure 1: Overall Cumulus vision and its use by different DAACs

NGAP provides the secure cloud infrastructure based on NASA's IT security policies. The NGAP cloud infrastructure will be utilized not only to host apps but also to provide software frameworks for optimized science data processing, as well as for data ingest, archive, distribution, and management. Each DAAC, with its specific science discipline expertise, will be able to run customized instances of the Cumulus framework on the NGAP cloud infrastructure. A Cumulus instance can include multiple workflows to ingest and archive multiple data streams in order to generate different science data products. The core Cumulus code base will be maintained and shared across the DAACs. In addition, a repository of docker containers for performing specific data stewardship steps will be maintained and shared across DAACs. Some of these docker containers can be core DAAC operations, such as data acquisition from a data producer to be shared and used by all DAACs, while other containers can be DAAC discipline-specific operations, which are required to meet the needs of the specific science community served by an individual DAAC. Each DAAC will be able to construct data product workflows by

utilizing the Cumulus framework as well as leveraging the common container repository, and as such will be able to improve the Cumulus framework and contribute new containers to the repository. This approach will improve cross-DAAC collaborations and DAAC productivity, thus reducing redundant efforts across DAACs. Both S3 and Glacier will be used for storage-based on needs. Data lifecycle rules will be applied on different data streams based on past distribution patterns in order to determine when and how to store the data to minimize storage costs, while still providing a consistent level of service to the data user community. Finally, it is envisioned that a new set of data distribution, analysis, and access services and tools will be developed by the DAACs, the EOSDIS project, and the user community in order to effectively access and utilize both the cloud data stores as well as the elastic cloud computing capabilities. These services, applications, and tools will encourage new data-centric scientific investigations that previously were not possible because of the data movement overhead from storage to compute.

2.2 Prototype Objective

The main objective of this prototyping activity is to develop a "native" cloud-based data ingest, archive and management system (Cumulus) that can be used for all future NASA data streams. "Native" implies a system that leverages all the components of a cloud infrastructure provided by the vendor for efficiency (in terms of both processing time and cost). Future data streams cover both satellite missions (a continuous ingest stream of homogenous data) and aircraft missions (an ad hoc ingest of heterogenous data). As such, this activity will produce a working Minimum Viable Product for Cumulus to be tested and evaluated.

3. CUMULUS REQUIREMENTS AND DESIGN

3.1 Requirements

Functional Requirements
Cumulus must provide the following functions common to all DAACs, irrespective of their specific discipline:
1. *Data Acquisition:* Cumulus will provide data discovery by crawling remote data repositories to identify new data products as well as retrieve the data for ingest.
2. *Data Ingest*: Cumulus will perform product validation for data integrity as well as be able to reinsert broken data granules back into the collection. Cumulus will provide common product preprocessing capabilities such as product renaming and product reformatting capabilities. It will allow individual DAACs to add their custom ingest preprocessing, and thus can serve as a framework for a customizable science data processing system. Also, Cumulus will generate the metrics required for data monitoring and reporting at DAACs.
3. *Metadata Publication:* Cumulus will extract and/or generate granule-level metadata and link the granule-level metadata with collection-level metadata stored in the catalog. Cumulus will publish the complete granule-level metadata records, including properly identified data access links pointing to S3 storage locations, into metadata catalogs.

4. *Data Archive:* Cumulus will optimize data-files archives based on the input configuration parameters such as storage cost and distribution requirements. Cumulus will perform the required checks for data integrity and track archive metrics.
5. *Data Distribution:* Cumulus will support several distribution methods for controlling egress charges to users and will provide hooks to build new distribution/access/analysis applications that can link to the data store. These applications will include data subscription service to users. In addition, Cumulus will track distribution metrics.

Overall System Requirements
1. Cumulus will be extensible so as to allow the addition of new functionality in phases and will comply with the required ESDIS standards [5] for data and metadata.
2. Cumulus will allow DAAC operators to add/configure data streams and define rules-driven data ingest, including adding preprocessing to meet the Levels of Services needs at each DAAC. Furthermore, Cumulus will allow policy-driven archive configurations to define specifics (i.e., where to stage, where to store, type of store, restrictions, etc.).
3. Cumulus will optimize performance and overall cloud costs, and, as such, it needs to ensure that it leverages cloud elasticity in order to scale horizontally to increases in data ingest rates.
4. Cumulus will interoperate with existing core EOSDIS components such as the CMR metadata catalog, GIBS Browse Imagery service, Metrics Reporting system, and existing Science Teams (SIPS) data servers.

3.2 Data Framework Design on AWS Infrastructure

The guiding principles used in designing the Cumulus framework included asynchronous computing (i.e., allowing execution of data pipelines per-granule to run asynchronously) and transparency (i.e., ensuring that a data pipeline is transparently defined with a json file). Having an architecture enable fully asynchronous execution where a particular step does not block the processing of other granules ensures a parallel flow and increases the speed and efficiency of processing.

The core components of Cumulus (i.e., Cumulus-factory, Data Processing as a Service (DPS), Cumulus-API, and the distribution app) are described below.

Cumulus-factory
The Cumulus-factory is responsible for handling all data processing and is the heart of the platform. The cumulus-factory is an event-driven asynchronous task runner where each task is queued, then discovered by the dispatcher and sent to the appropriate service, either Lambda or ECS, for execution. The Cumulus-factory uses the DPS to ingest and processes granules. The major components of Cumulus-factory are Ingest, Processing Steps, CMR Updater, and Archiver.
- *Ingest* is responsible for bringing granules to the Cumulus space. It uses a combination of Lambda function and EC2 Container Service (ECS) to scan, transfer, and sync data. It supports various distribution mechanisms including HTTP and FTP protocols. Ingest can also read Product Delivery Records (PDRs) and download data files specified by a PDR.

- *Processing Steps* are steps that are shared among all collections, such as ingesting the data, pushing the metadata to CMR, or archiving the data, but there are also steps that are unique to each collection. A collection might require extracting thumbnails from an HDF file or generating Netcdf files from original files. These steps are scripted in any number of computer languages and frameworks.
- *CMR Updater* is a Lambda function that receives a granule and updates the CMR with the metadata of the granule, granted that the metadata information of the granule is available as a file on an S3 bucket. Operators can configure the CMR Updater via the Cumulus-API.
- *Archiver* is a Lambda function that moves the files of each granule to appropriate buckets depending on whether the files should be private, protected, or public. In Cumulus, each file can be private (not accessible from internet), protected (accessible with an Earthdata account), and public (no authentication is needed).

Data Processing as a Service (DPS)
Data Processing as a Service (DPS) is a data pipeline that uses AWS' Lambda, EC2, ECS, S3, and SQS services [6] to ingest, process, and archive data. The use of AWS Lambda allows one to code without provisioning or managing servers, thereby reducing the compute time. DPS enables running synchronous and asynchronous processing pipelines on AWS via a given workflow (or, recipe). DPS is primarily a dispatcher that reads messages from a queue and launches tasks in the cloud, given that messages follow a specific schema. The dispatcher holds together various parts of a processing service because every component of the processing pipeline communicates via the dispatcher. For example, dispatchers start all tasks; if a task must start after another task, the dispatcher handles the successive actions. The dispatcher expects to receive events with well-defined payloads in order to decide the next action. All pieces of the pipeline use the same schema to send messages to the dispatcher. When the dispatcher receives the messages, it looks at its payload and determines the next action. The next action could be adding a new message to another queue, invoking a lambda function, or starting an ECS task.

Cumulus-API
The Cumulus-API is the protected interface for the system operators to manage the platform and gain insight into its operations. To abstract the Cumulus-API, a dashboard GUI will be provided to the system operators.

Distribution App
The distribution app is the public interface of Cumulus that allows public users to access and download granules processed by Cumulus.
.

4. INITIAL LESSONS LEARNED

4.1 Development Status

The Cumulus prototype is currently being designed and developed in four incremental phases, with each phase typically consisting of six development sprints. Each phase focuses on one or two data streams from a specific DAAC selected to test ingest streams based on their representativeness for certain instrument platform (satellite versus airborne). For Phase 1, a set of airborne datasets from a recent Hurricane and Severe Storm Sentinel (HS3) field campaign from the Global Hydrology Resource Center DAAC was selected. The Cumulus prototype had to successfully demonstrate ingest and archive of these airborne datasets on the cloud infrastructure. Details for the airborne datasets are listed in the table below. While both the number and the size for these files are small, Cumulus has to deal with a wide variety of datasets typical for data ingest requirements of an airborne field campaign. The Cumulus prototype activity is currently half-way through Phase 2 of the four phases.

Hurricane and Severe Storm Sentinel (HS3) Global Hawk Advanced Vertical Atmospheric Profiling System (AVAPS) Dropsonde System	2,851 files	1.39 GB
Hurricane and Severe Storm Sentinel (HS3) Global Hawk Cloud Physics Lidar (CPL)	473 files	112.14 GB
Hurricane and Severe Storm Sentinel (HS3) Global Hawk High Altitude MMIC Sounding Radiometer (HAMSR)	15 files	2.31 GB
Hurricane and Severe Storm Sentinel (HS3) High-altitude Imaging Wind and Rain Airborne Profiler (HIWRAP)	40 files	24.85 GB
Hurricane and Severe Storm Sentinel (HS3) Hurricane Imaging Radiometer (HIRAD)	212 files	7.95 GB
Hurricane and Severe Storm Sentinel (HS3) World Wide Lightning Location Network (WWLLN) Storms	256 files	171.09 MB

Table 1: HS3 Datasets selected to test Cumulus for Phase 1

4.2 Lessons Learned
Even though, Cumulus prototyping effort has just started. There are two key lessons learned after the completion and evaluation of phase 1. These are:

Cost savings potential for future airborne data streams
Hardware procurement for HS3 airborne field campaign was based on best case estimates of data storage. Because of the nature of field campaign missions, there is always uncertainty in the amount of data to be collected. Due to variety of reasons, the total data collected during the HS3 field campaign was far less than anticipated. This shortage resulted in an overestimation of hardware and storage, which then resulted in subsequent sunk costs. This expense would not have occurred with the use of the Cumulus data framework because the capacity can be increased only as needed. Therefore, the use of Cumulus would have resulted in substantial

savings. 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
Cumulus represents a major shift in how data ingest, archive code is developed within DAACs. Consequently, an on-boarding process for DAACs to be able to use the Cumulus data framework must be developed, and the DAAC must be transitioned gradually into this new vision. Current operational code used at DAACs predates the existence of Docker sothe transition requires time as well as training in order to convert some of the existing DAAC processing software to be "Docker-ready". DAACs must be able to deploy Docker images used with a Cumulus workflow (recipe) outside of AWS in order to allow for testing, and as such all dependencies (e.g., specific versions of software stack outside of Docker) should be documented and specific dependencies on AWS resources should be removed.

5. SUMMARY

This paper describes NASA's ESDS Program's vision for evolving the existing data and information system by utilizing commercial cloud infrastructure. The paper focuses on one of the prototyping activities, namely the Cumulus data framework, currently underway to attain this vision. Cumulus is being designed and developed as a "native" cloud-based data ingest, archive and management system that can be used for all future NASA Earth science data streams. Cumulus provides a common data framework that can encompass both ingest/archive as well as data processing needs for any of NASA's Earth Science missions. Cumulus will ensure all data needs for future NASA missions are built using this common code base. Cumulus will provide the common framework to enable sharing and use of standalone functional components via Docker images. Cumulus will provide an optimized framework for data ingest, processing, storage, and access via the use of native cloud services (AWS), such as Lambda function. It will also enable a metrics-driven approach for optimization of data management, storage, and distribution. For example, a DAAC can assign a dataset to be moved from S3 to Glacier after a specified time period based on historical distribution patterns for a similar dataset, thereby minimizing storage costs. Finally, Cumulus will foster design of new analysis/visualization tools that can leverage collocated data from all of the distributed DAACs as well as elastic cloud computing resources.

6. REFERENCES

[1] NASA, "Science Plan," 2014.
[2] ESDS-Program, "NASA Earth Science Data Systems Program," NASA Earth Data Website, 2016. [Online]. Available: https://earthdata.nasa.gov/earth-science-data-systems-program. [Accessed: 27-Dec-2017].
[3] EOSDIS-Project, "About EOSDIS," NASA Earth Data Website, 2016. [Online]. Available: https://earthdata.nasa.gov/about.
[4] EOSDIS-Project, "EOSDIS Distributed Active Archive Centers (DAACs)," NASA Earth Data Website, 2016. [Online]. Available: https://earthdata.nasa.gov/about/daacs.

[5] ESDIS Standards Office, "Standards, Requirements and References," NASA Earth Data Website, 2016. [Online]. Available: https://earthdata.nasa.gov/about/esdis-project/esdis-standards-office-eso.

[6] AWS, "AWS Lambda," Amazon Web Services, 2016. [Online]. Available: https://aws.amazon.com/.