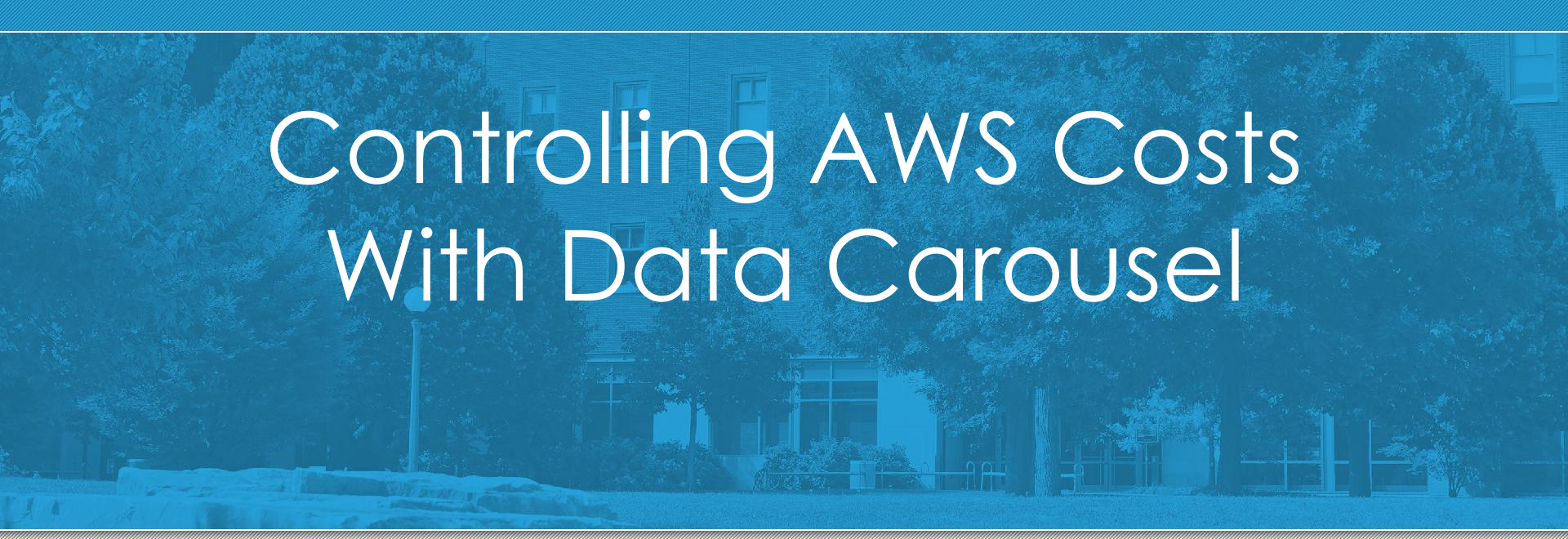


Controlling AWS Costs With Data Carousel

Ben Galewsky, Don Petravick, Greg Daues - NCSA, University of Illinois
John Readey - HDF Group
Ryan Kolak - Amazon Web Services

Overview

- Introduce Terra Fusion Dataset
- Describe the financial and operational challenges of hosting this in AWS
- Present a data carousel architecture that dramatically reduces the cost while make the data available for science
- Show our prototype implementation
- Next steps and community input

What are we trying to prove?

That we can create a process that runs on a fixed schedule to efficiently restore specifically required data from Glacier and make it available to user supplied batch jobs

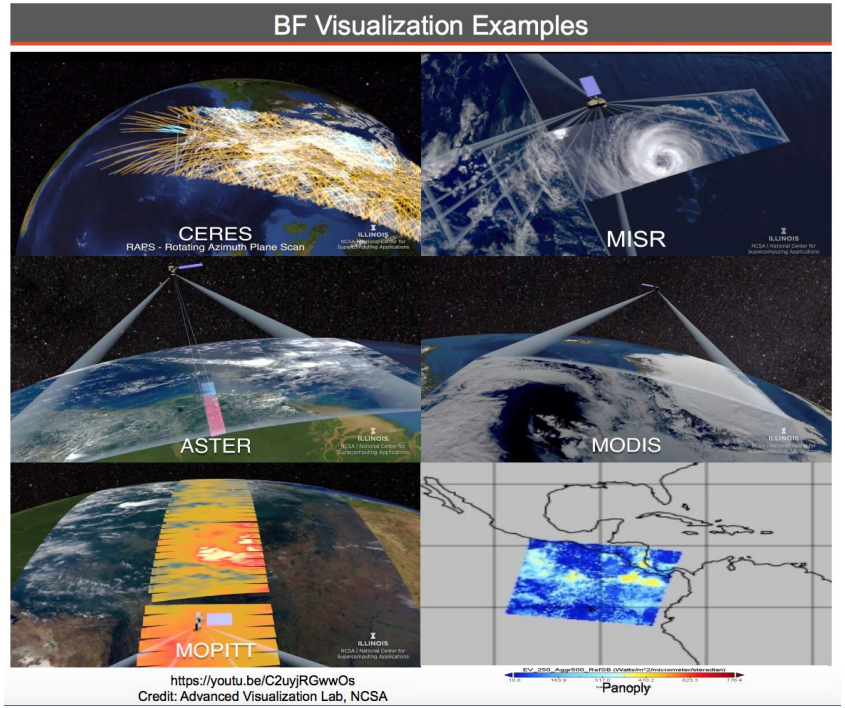
The Dataset used for Prototyping

Terra Fusion Dataset - NASA ACCESS program

- Fusion provides common format and structure for data of
  - MODIS, MISR, ASTER, CERES, MOPITT of Terra satellite
- 84303 Terra Fusion HDF5 files (years 2000 -- 2015)
- Dataset is 2.4 PB in size
- Each Fusion product file has granularity of one Terra orbit
  - Range in size [15 GB, 50 GB]
- Generation
  - Raw, original radiance (L1B Data) gathered from NASA DAACs
  - Fusion system executed on Blue Waters system at NCSA
  - Transfer: BW NearLine tape => NCSA GPFS Condo fs => AWS S3
- Provide the means for synergistic use of the data of the five instruments

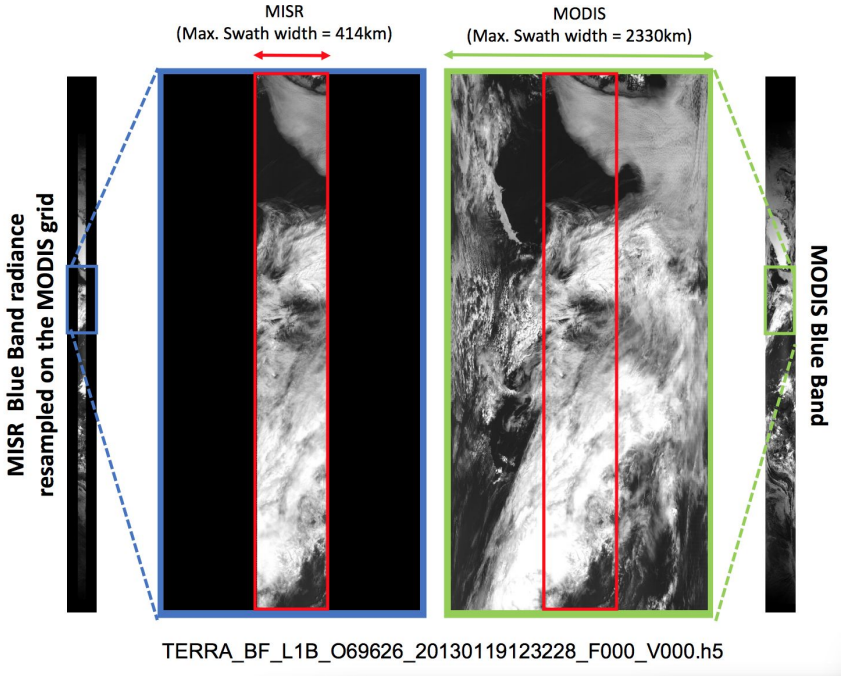
Visualizations Example

https://modis.gsfc.nasa.gov/sci_team/meetings/201810/posters/digirolamo.pdf



Resampling and Reprojection Example

https://modis.gsfc.nasa.gov/sci_team/meetings/201810/posters/digirolamo.pdf



Science Driver

Original motivation is to enable science via a science-ready fused dataset for the **entire** Terra Mission.

Allow students to start projects without data wrangling ~1M files.

Seen as a **sine qua non** of **mission-scale science**.

- See https://earthdata.nasa.gov/esds/competitive-programs/access/terra-data-fusion-products

In practice science would be foregone without data support like this.

Resources vs Science Value

Commercial hosting the resulting ~2.4 PB data set

- S3 Standard (List) ~$600,000/year
- An additional backup copy in Deep Glacier (~$28,000)

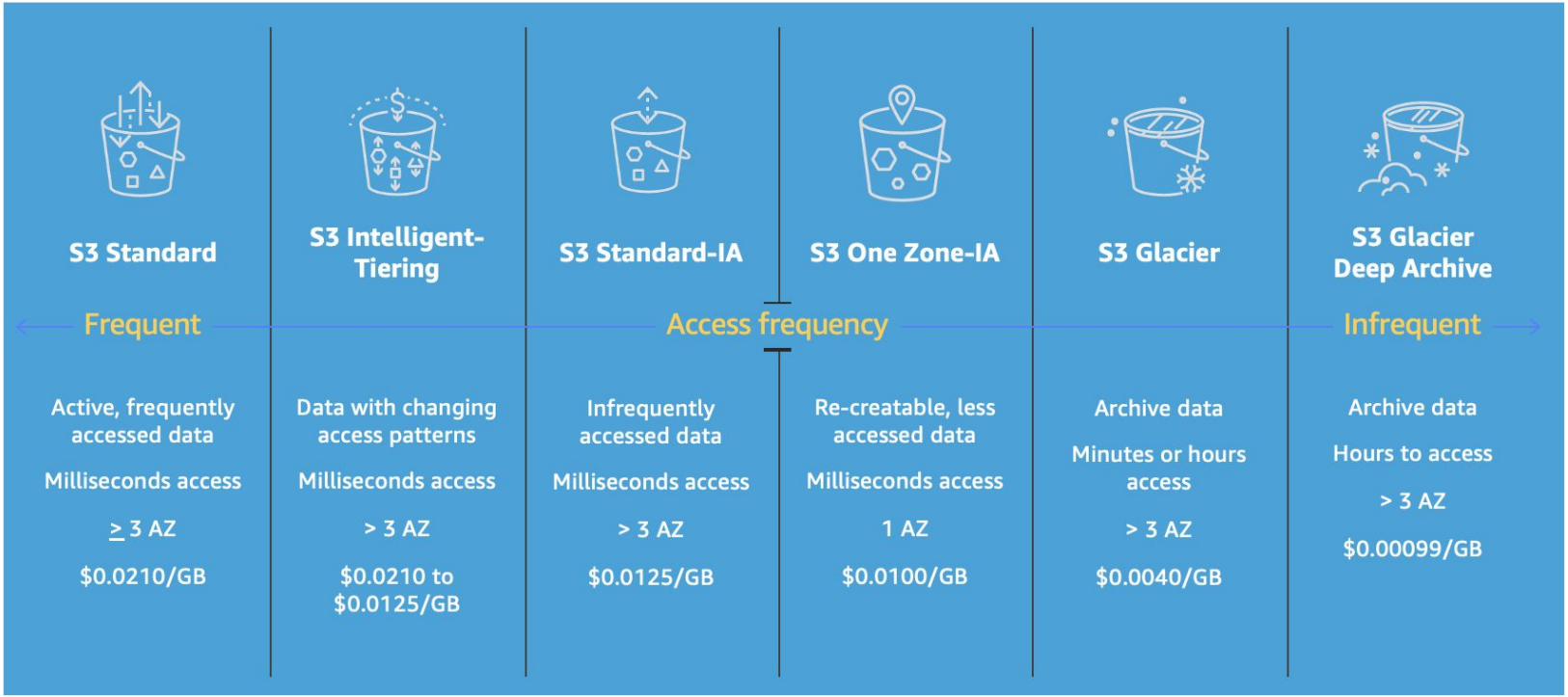
This level of cost implies:

- A substantial user community
- Requiring large data inputs
- Significant science programs.

One way to look at this

- A bit of a chicken and egg problem.
- Is there a more gradual way to start?

Storage on AWS



Deep Archive Storage Class

What are the constraints?

- Data needs to be restored before it can be accessed
  - Standard Retrieval - 12 hours - $0.01/GB
  - Bulk Retrieval - 48 hours - $0.0025/GB
- Standard Storage rate for restored copy ($0.023/GB)
- Minimum of 180 days of Storage

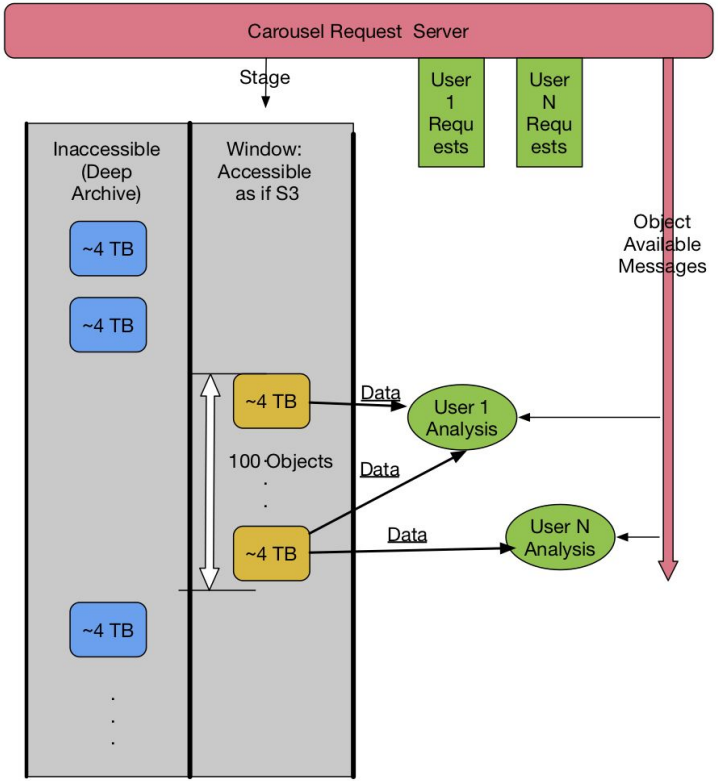
Glacier Constraints/Characterisics

Characteristic	AWS Specification	Consequences for 2.4 PB data set.
Restore requests/day	35 requests /PB	84/ requests/Day
Maximum Object Size	Max 5 TB object	Data must be packed in to a container format (e.g ZIP or similar) Speed limit ~14 days for a full tour
Retrieval Granularity	One Aws object	Pack to minimize retrieval of most frequent access pattern.
Duration of useful access to data once staged @ no additional cost	S3-like access for 24 hours upon restore	AWS elastic compute provides ample compute capacity for an analysis. LIkely Best to trigger compute on completion of every stage operation.

Large scale analysis requires planning.

- Analysis of 2.4 PB is more measure twice, cut once than spin over data many times.
- Illinois Scientists indicate two weeks for a computer-think cycle would serve their their needs for PB scale analysis.
  - Re-enforced by costs for analysis compute cycles and cost for string outputs.
- Packing the existing 32 GB files into a “container” format seem feasible.
  - Deep Glacier does not stage partial objects.
- Software to orchestrate common staging/job dispatch needed to keep to budget.

Data Carousel Functionality



- On ingest, composite files are built up to near the ~5TB limit.
- Multiple users request files from the Carousel Request Server.
- Based on queued up file requests
  - The carousel stages up to 50 files/day.
  - Files are available for 24 hours as S3 objects.
- As files are staged, the carousel signals that code may now be run against the staged file.
- After 24 hours, the file reverts to inaccessible.

Scientist's View

- There is key science requiring large data sets.
- My problem is file-wise parallelizable.
- This is the only way I can get the science done.
- I'm provided with this batch mode access, but it really is not so bad when I compare it to long queue delays and tape access at an HPC center.
- I need to plan carefully to ration my large-scale processing budget anyway

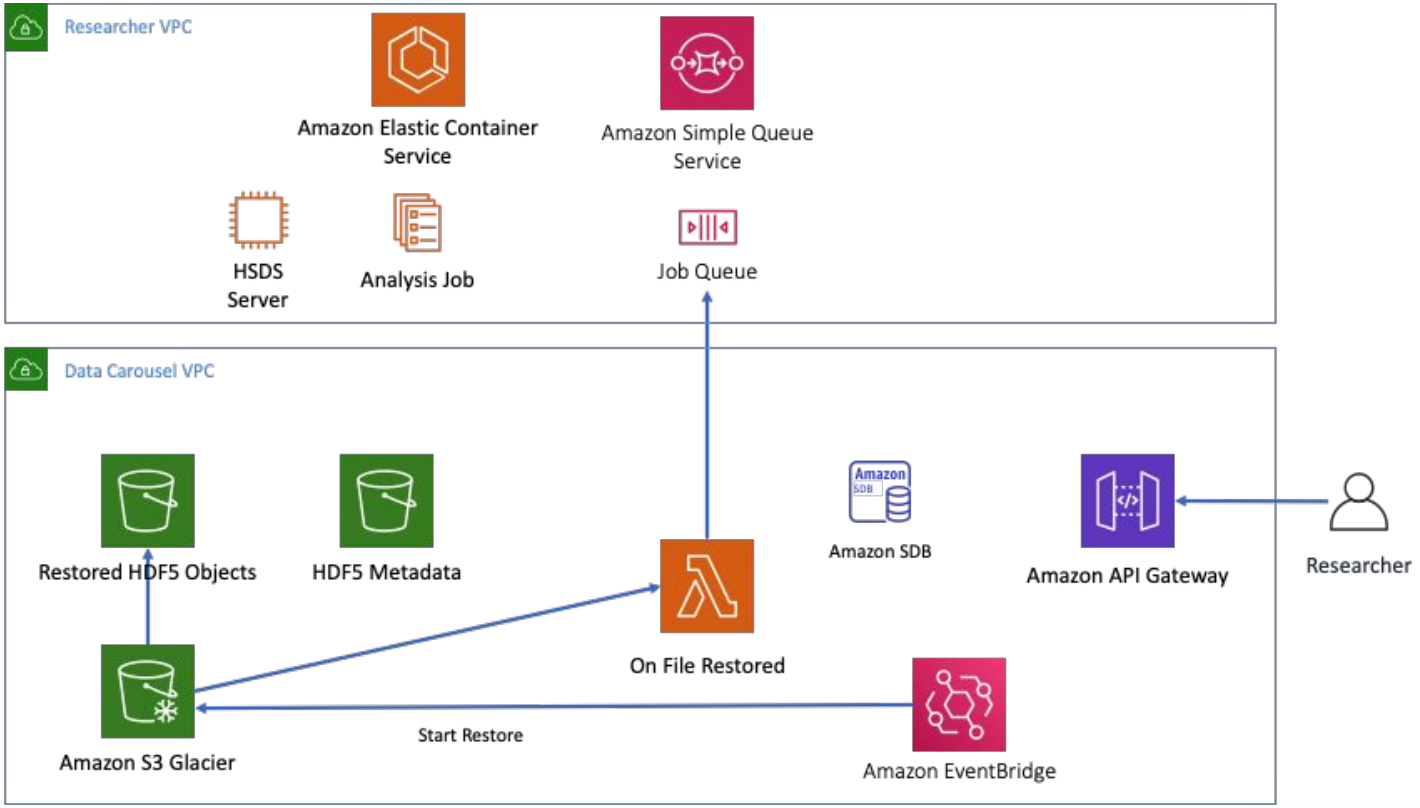
Resource Manager's View

- **I can budget!** Annual costs capped at sum of storage + maximum number of full turns permitted on on the data.
  - I pay staging costs, users pay other access costs.
- Maximum cost less that ⅓ cost of S3 Hosting.
- I have a per-file, per-user view of interest in the data set.
- The use community does not need to communicate between themselves to orchestrate staging.
- Savings from budget each time a file is not needed.
- Costs decrease to mere Deep Glacier storage if interest in the data set totally wanes.
- The carousel software is re-useable and applicable many data sets.

Annual Storage/Movement Costs (2.4 PB)

Storage Method	# full accesses/year	Annual Cost
S3	Unlimited	~$600.000
S3 Infrequent Access	26	~$984,000
Glacier	26	~$270,000
Deep Glacier	26	~$180,000
Deep Glacier (light interest)	13	~$100,000
Deep Glacier (no interest)	0	~$28,000

The Proof of Concept



HDF5 on S3

- The TerraFusion dataset uses the HDF5 file format.
- The Data Carousel provides a mechanism to rotate files from Glacier to S3 for users to access.
- Users can download these files from S3 to their computing infrastructure but in many cases it would be more convenient to access the files in place.
- There are various extensions that enable directly reading HDF5 files on S3 (S3VFD, S3FS for Python, Fuse file system), but in general these have very limited performance.
- The approach the Data Carousel pursued is to use a HDF Data service developed by the HDF Group: HSDS (Highly Scalable Data Service)
-

Intro to HSDS

- HSDS (Highly Scalable Data Service) is a REST-based service for HDF data
- Runs as a set of containers on Docker or Kubernetes
  - Number of containers can be scaled up or down
  - More containers == better performance
- Accessing HDF data on S3 using HSDS is faster since:
  - The number of requests to S3 is less (metadata is consolidated)
  - Server caches recently accessed data
  - Data access can be parallelized across multiple containers
- Client libraries for Python (h5pyd) and C/C++ (HDF5 lib plugin)
- Interactive web pages can use REST API directly
- Initially developed under NASA ACCESS 2015 grant

HSDS Data Schema

- HSDS uses a sharded schema (similar to Zarr) where each object's meta data is stored as a JSON object and each chunk is stored as a binary blob
- Data access is accelerated since the service doesn't need to search through a larger object to find content
- Completely converting 84K Terra Fusion files to the HSDS schema would be quite a project and would require 2.4PB additional storage
- Instead, only the metadata (links, attributes, types, etc) is converted. Chunk information is stored as links back to the original file.
- Server can extract these chunks without using the HDF5 library

Primary AWS services

Amazon Simple Queue Service (SQS)

- Is the basis for a “thin”, well defined interface between the data carousel and any number of users’ computing capacity.
- Provides or clean separation of common costs and user costs.

Amazon Elastic Container Services (ECS)

- In the prototype, ECS form the basis for elastic computing that can dispatch computing as the staging of each object from Glacier occurrs.
- ECS can manage a cluster of EC2 instances with autoscaling or severless with Fargate to run containers without managing the underlying hardware

User Interaction With Carousel

- Submit a Request
  - Interactions via REST interface
  - Eventually provide a python library
- Deploy a Job
  - Dockerized job code
  - Runs in researcher's AWS account

Submit Job

```
{
  "username": "bengal1@illinois.edu",
  "job_description": "Climate Marble Test",
  "query": "MISR_Path='P125' and Year='2010'"
}
```

Submit Job


```
{
  "username": "bengal1@illinois.edu",
  "job_description": "Climate
  "query": "MISR_Path='P12
}
```

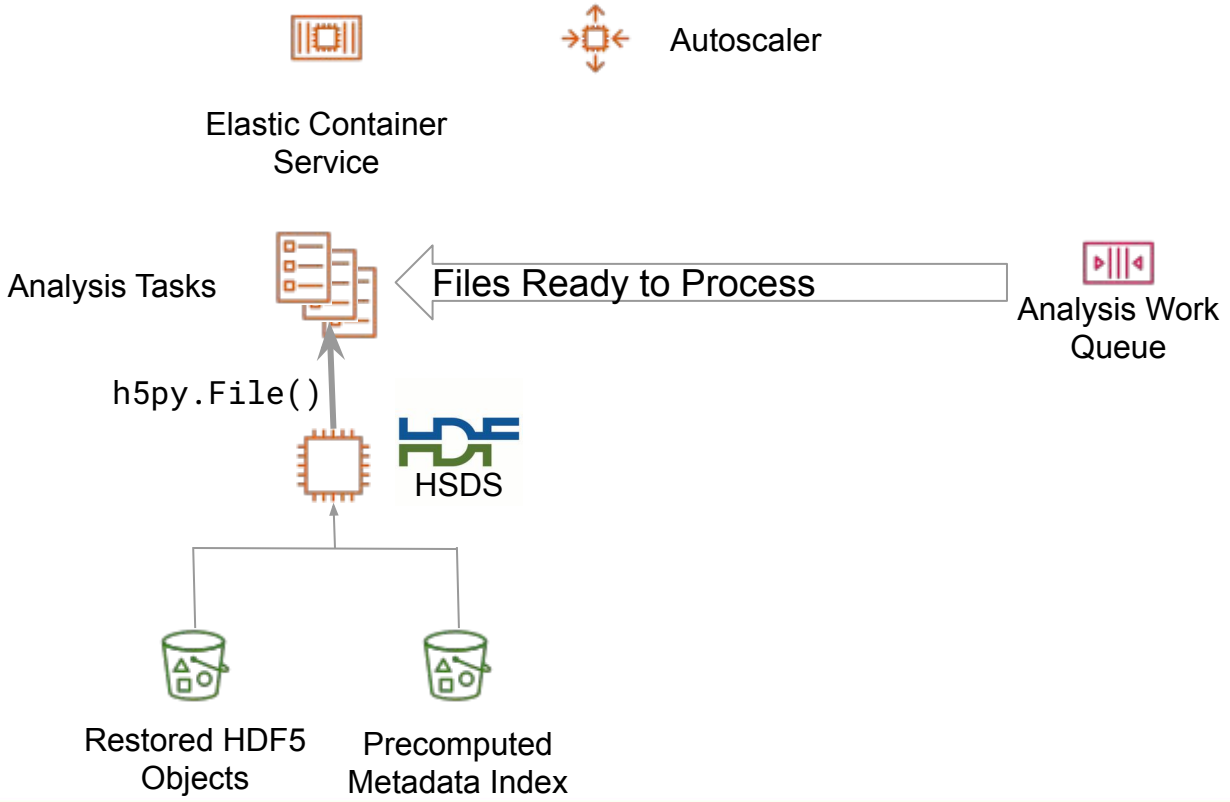


```
{
  "body": {
    "job-id": "2030d1cb-bf05-11ea-8678-d18cff52d3af",
    "file-count": 233,
    "expected-run-start-time": "2020-07-20 21:18:52.767564"
  }
}
```

Job Environment

- SQS Work Queue presents files ready for analysis
- Researcher is responsible for an HSDS instance, using centrally managed metadata and reading objects from the global restore bucket.
- Elastic Container Service to run the job tasks
- Autoscaler:
  - Scale cluster down to zero when no files are available to work on
  - Scale up to a level to insure all files are processed during the window
- Results persisted to researcher's S3 bucket, or downloaded to home environment

Batch Job Environment



Job Work Queue

- User sets up SQS Work Queue
- Receive messages as files become available

```
{
  "terra-file": "/terrafusion/P108/TERRA_BF_L1B_O10204_20011118010522_F000_V001.h5",
  "year": 2001,
  "month": 5
}
```

Job Environment

- CloudFormation template to set up batch compute environment:
  - SQS Work Queue
  - HSDS Instance
  - Elastic Compute Service for jobs
    - Autoscaling from zero to as many workers as desired
    - Scale back to zero while waiting for new files to be restored

Reactions and Community Input

Resources

https://github.com/ncsa/datacarousel	• Docker image for ingesting and cataloging the dataset. • Lambda functions for job submission and on file restored • CloudFormation templates
https://github.com/BenGalewsky/ClimateMarble	• Example job that can be run against the data carousel
https://www.ideals.illinois.edu/handle/2142/107186	• Whitepaper on Data Carousel concept and architecture
https://github.com/HDFGroup/hsds https://aws.amazon.com/blogs/big-data/power-from-wind-open-data-on-aws/	• HSDS software • AWS Big Data Blog about HSDS

Resources provided By...

- National Aeronautics and Space Administration (NASA) through contract number NNX13AL96G via the ACCESS program.
- NCSA DIrectorate (University of Illinois)
- Amazon Web Services
- The HDF group

For Follow-Up

petravic@illinois.edu or bengal1@illinois.edu

ILLINOIS

NCSA | National Center for Supercomputing Applications