Best practices for working with large-scale geospatial data

Joe Flasher
Open Geospatial Data Lead
Amazon Web Services
Agenda

Intros

AWS Public Dataset Program

Cloud-optimized storage

Customer examples

So many new things!
AWS Public Dataset Program
The AWS Open Data program expands access to data by staging it for analysis in the cloud.

https://opendata.aws
AWS public datasets

- NASA
- NYC Taxi & Limousine Commission
- National Institutes of Health
- DC.gov
- Internal Revenue Service
- Ilmatieteen Laitos, Meteorologiska Institutet, Finnish Meteorological Institute
- Met Office
- Emory

https://registry.opendata.aws
AWS public datasets
https://registry.opendata.aws
Earth on AWS
aws.amazon.com/earth
Cloud-optimized storage
Flipped data flow in the cloud

**Traditional approach:** Move data to computing

**Cloud approach:** Move computing resources to data

- Amazon Simple Storage Service (Amazon S3)
- Amazon Athena
- Amazon EC2
- Amazon EMR
“Cloud-optimized storage means getting to the bytes you want without having to access the bytes you don’t.”

Joe Flasher
Traditional GeoTIFF bundle
.tar
Cloud-optimized GeoTIFFs
Cloud-optimized GeoTIFFs
Staging data for analysis

Amazon S3 allows programmatic and precise access to data at planetary scale.

Landsat on AWS uses cloud-optimized GeoTIFFs that allow users to get only the data they need when they need it.

cogeo.org
Graph by Drew Bollinger (@drewbo19) at Development Seed

Landsat on AWS
Index patterns

Amazon S3 key index

External index

Internal index
Example: GOES-16 key naming

s3://noaa-goes16/ABI-L1b-RadF/2018/149/14/
OR_
ABI-L1b-RadF-M3C14_
G16_
s20181491430465_
e20181491441232_
c20181491441300.nc
Example: SpatioTemporal Asset Catalog (STAC)

The STAC specification aims to standardize the way geospatial assets are exposed online and queried. The initial focus is primarily remotely sensed imagery (from satellites, but also planes, drones, balloons, etc.), but the core is designed to be extensible to SAR, full motion video, point clouds, and hyperspectral, LiDAR, and derived data like NDVI, digital elevation models, mosaics, etc.

- Static catalog
- Catalog API
- Core metadata and extensions

https://github.com/radiantearth/stac-spec
landsat-tiler from Mapbox

https://viewer.remotepixel.ca/
Sentinel Hub from Sinergise

https://www.sentinel-hub.com/
What is the right object size?

<table>
<thead>
<tr>
<th></th>
<th>Small objects</th>
<th>Big objects</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Organization</strong></td>
<td>Built-in</td>
<td>Depends on the format</td>
</tr>
<tr>
<td><strong>Cost</strong></td>
<td>More requests, per-object fees, minimum object size fees</td>
<td>More data transfer (maybe)</td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td>More small requests</td>
<td>Larger requests, wasted data transfer (maybe)</td>
</tr>
</tbody>
</table>
Storage lessons learned

• Lots of small objects can present difficulty when you try to update them
  • Also want to watch for transaction overhead; it becomes large with object access time
  • Geospatial data is often tiled/gridded, and you can pack multiple levels into a single object

• S3 inventory is much better than listing objects in a bucket
  • Can easily be queried by Amazon Athena + S3 Batch Operations

• If you’re chaining a lot of processes, S3 transaction overhead may be costly
  • Can copy to something like Amazon EBS or Amazon EFS depending on architecture

• Do you need to process that data into multiple formats?
  • Take a look at using API Gateway + Lambda or Lambda@Edge to accomplish the same thing with no extra storage (trading processing for storage)

• If you have not seen it, S3 has increased the request rate performance
Putting data to work
Global Hydrology Resource Center Field Campaign Explorer (FCX)

FCX provides visualization and analytics capabilities for diverse coincident datasets, with a focus on airborne field campaigns.
GHRC Field Campaign Explorer

• Serverless cloud-native technologies
  • Amazon Athena for searching data stored in S3 buckets
  • Data framework with ingest and access APIs to Parquet files via the Athena query interface or cloud-native data storage technologies such as Zarr and cloud-optimized GeoTIFF
  • AWS Step Functions and Lambda to orchestrate and run data processing and rendering code without provisioning or managing servers, automatically scaling resources as needed
  • Scalable, efficient data access to support on-the-fly rendering and analytics
GHRC FCX architecture #1

Original Files & Metadata
Data Readers
Data Layers in Parquet Format
AWS Lambdas
S3

Amazon RDS
S3

Metadata Catalog
Glue Crawler
Glue Data Catalog

Front-end Web App
Cesium Map Platform
with Potree

VISAGE API
Amazon API Gateway

Parallel SQL GIS Queries
Amazon Athena

Selected Data Layers

On-the-fly 3D Tile Generation
3D Tiles
Tileset.json
S3

AWS Step Function

AWS Lambdas
GHRC FCX results

• Architecture #1
  • Data is stored in Athena in Parquet format
  • Most of the components are serverless
  • Lower operating cost
  • Scalable but slower as compared to architecture #2

• Architecture #2
  • Data is stored in Zarr and cloud-optimized GeoTIFF
  • Uses an AWS Fargate cluster—scalable and parallel generation of 3D point cloud files with ability to control number of parallel workers
  • Works well with larger datasets
  • Higher operating cost
Tilezen open-source terrain tiles

Mapzen released version 2 of its terrain tiles dataset

- Large-scale usage of AWS Batch and AWS Lambda
- ~3 billion tiles generated, 3.75 million per minute

- Total rendering of the world took less than 1 week, ~10x time improvement
- AWS Lambda was used as a dynamic testing mechanism before rendering
Amazon EC2 Spot Instances

• Offers spare compute capacity available on the AWS Cloud at steep discounts compared to on-demand instances
• Amazon EC2 instances at savings of up to 90% off the On-Demand price
• Spot Instances will automatically pause and resume your work around interruptions, so your applications can start right where they left off
• Works natively with a number of services like AWS Batch

In a geospatial context, this works very well for bulk reprocessing tasks and can work for forward processing depending on your architecture and latency tolerances
2018 announcements

- S3 Batch Operations (Preview)
- S3 Intelligent-Tiering
- AWS Transfer for SFTP
- EC2 Predictive Scaling
- AWS Snowball Edge Compute Optimized
- Amazon SageMaker Ground Truth
- AWS Marketplace for machine learning
- EMR Notebooks
- AWS Ground Station
- Free ML training
- S3 Glacier Deep Archive
- Amazon Elastic Inference
- AWS Inferentia
- AWS Step Functions service integrations
Final thoughts related to geospatial

• Lots of legacy code! Containers can be a good way to handle this and there are a number of options for running these including Amazon EC2, AWS Batch, Amazon ECS, AWS Fargate, and Amazon EKS—depending on your desired architecture.

• Data can be large! AWS Snowball (and AWS Snowball Edge) can be used to move large amounts of data to/from AWS.

• So many objects! S3 Lifecycle policies and S3 Batch Operations are good ways to handle movement/deletion of large numbers of objects.

• For more information about geospatial data: https://github.com/jflashe/reinvent-geospatial
Open data
is happy data
aws
Thank you!

Joe Flasher
jflasher@amazon.com

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Please complete the session survey in the mobile app.