Re-organizing Earth Observation Data Storage to Support Temporal Analysis of Big Data

Christopher Lynnes*
NASA Goddard Space Flight Center

*NASA Civil Servant
Earth Observing System Data and Information System (EOSDIS)

EOSDIS

Capture and Clean

- Process
- Archive
- Transform
- Distribute

*Subset, reformat, reproject

Research
Applications
Education
Earth Observing System Data and Information System (EOSDIS)

- Capture and Clean
- Process
- Archive
- Transform*
- Distribute

*Subset, reformat, reproject

Research
Applications
Education
Earth Observing System Data and Information System (EOSDIS)

Capture and Clean

EOSDIS
- Distribute
- Transform*
- Archive
- Process

Research
Applications
Education

*Subset, reformat, reproject
Earth Observing System Data and Information System (EOSDIS)

EOSDIS

Capture and Clean

Distribute

Transform*

Archive

Process

*Subset, reformat, reproject

Research

Applications

Education
Earth Observing System Data and Information System (EOSDIS)

Capture and Clean

1. Process
2. Archive
3. Transform*
4. Distribute

*Subset, reformat, reproject

Research
Applications
Education
Earth Observing System Data and Information System (EOSDIS)

Capture and Clean

1. Process
2. Archive
3. Transform*
4. Subset, reformat, reproject
5. Distribute

Research
Applications
Education
Current Data Organization

<table>
<thead>
<tr>
<th>File Name</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIRS.2017.11.03.L3.RetStd_IR001.v6.0.31.0.G17310121421.hdf</td>
<td>2017-11-06T17:17:47</td>
</tr>
</tbody>
</table>
Current Data Organization
How does time slice organization affect analysis performance?

**Data Set:** North America Land Data Assimilation System

**Temporal Resolution:** Hourly

**Spatial Resolution:** 0.125 deg resolution (464 x 224)

**Variable:** Air Temperature @ 2m

**Calculation:** Average over time at each grid point

**Hardware:** MacBook Air

**Software:** *ncra* from netCDF Command Operators (nco)
How does time slice organization affect analysis performance?
Hmmm...what if we pre-aggregate?

For 2 years of data...

Original Thin-sliced data: 17544 files
Aggregated into Yearly Files: 2 files

<table>
<thead>
<tr>
<th>Data organization</th>
<th>Number of files</th>
<th>Elapsed time to process</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Hour / File</td>
<td>17544</td>
<td>461 s</td>
</tr>
<tr>
<td>1 Year / File</td>
<td>2</td>
<td>66 s</td>
</tr>
</tbody>
</table>
Meanwhile, back at the ranch archive...
EOSDIS archive volumes are slated to grow quickly over the next several years.
EOSDIS migration to the cloud brings several benefits

Large Volume Data Storage
All datasets stored in common Web Object Storage archive

Scalable Compute
Provision based on need
Cost by use

Cloud Native Compute
Cloud compute services enhance implementation

End-User Processing
Scientists bring algorithms to the data.
“Scalable Compute” comes with a catch...
Cloud-based Data Parallelism
Cloud-based Data Parallelism
## A user journey through data analysis on the cloud

<table>
<thead>
<tr>
<th>Processor</th>
<th>Data org.</th>
<th>No. of files</th>
<th>Storage Type</th>
<th>Elapsed time</th>
</tr>
</thead>
<tbody>
<tr>
<td>MacBook</td>
<td>1 Hr / File</td>
<td>17544</td>
<td>Local SSD</td>
<td>461 s</td>
</tr>
<tr>
<td>t2.xlarge</td>
<td>1 Hr / File</td>
<td>17544</td>
<td>Local SSD</td>
<td>97 s</td>
</tr>
<tr>
<td>MacBook</td>
<td>1 Yr / File</td>
<td>2</td>
<td>Local SSD</td>
<td>66 s</td>
</tr>
<tr>
<td>t2.xlarge</td>
<td>1 Yr / File</td>
<td>2</td>
<td>Network</td>
<td>56 s</td>
</tr>
<tr>
<td>t2.xlarge multi-proc.</td>
<td>1 Yr / File</td>
<td>2</td>
<td>Local SSD</td>
<td>20 s</td>
</tr>
<tr>
<td>2 * t2.xlarge multi-proc.</td>
<td>1 Yr / File</td>
<td>2</td>
<td>Local SSD</td>
<td>11 s</td>
</tr>
</tbody>
</table>

`t2.xlarge = 4 vCPU, 8 GB memory, $0.1856/hr
SSD = Solid State Drive`
Journey Cost in Time and Treasure

1.5 Days from a standing start*

*Thanks, Anaconda and nco!
Summary: How to run fast

1. Process on fast cloud CPUs
2. Reorganize the data (space-time tiles)
3. Get data onto fast storage
4. Use all the CPUs on the virtual machine
5. Use multiple virtual machines
Summary: How to run fast

1. Process on fast cloud CPUs
2. Reorganize the data (space-time tiles)
3. Get data onto fast storage
4. Use all the CPUs on the virtual machine
5. Use multiple virtual machines
Why not reorganize ALL the data in the cloud?

1. First Rule of Archive Club: Nobody modifies the original data in Archive Club.
2. But: a second copy of all the data costs a lot of money
3. Live data streams mean ever-changing tiles
4. Users may be confused by the quasi-duplication
Data Bursting

❖ Manual Curation
➢ Burst Based on User Requests / Votes
➢ Data Expeditions

❖ Automatic Curation
➢ Event-triggered
➢ “Data finds Data”*

Data Bursting Opportunities

• Multi-dataset suites for studying Earth systems
• Bespoke gridding / projection schemes
• Rapid assembly of data suites in response to events
Data Bursting Challenges

• Reproducibility:
  – freeze-dry suites and store in low-temperature storage?

• Provenance:
  – bind to or place inside data?

• Choosing:
  – lightweight proposal process?
  – base on data impact?
# Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>EOSDIS</td>
<td>Earth Observing System Data and Information System</td>
</tr>
<tr>
<td>nco</td>
<td>netCDF Command Operators</td>
</tr>
<tr>
<td>netCDF</td>
<td>Network Common Data Form</td>
</tr>
<tr>
<td>SSD</td>
<td>Solid State Drive</td>
</tr>
</tbody>
</table>