Hadoop for High-Performance Climate Analytics

Use Cases and Lessons Learned

Glenn Tamkin (NASA/CSC)

Team: John Schnase (NASA/PI), Dan Duffy (NASA/CO), Hoot Thompson (PTP), Denis Nadeau (CSC), Scott Sinno (PTP)
Overview

• Scientific data services are a critical aspect of the NASA Center for Climate Simulation’s mission (NCCS). Modern Era Retrospective-Analysis for Research and Applications Analytic Services (MERRA/AS) …

  • Is a cyber-infrastructure resource for developing and evaluating a next generation of climate data analysis capabilities
  • A service that reduces the time spent in the preparation of MERRA data used in data-model inter-comparison
Vision

- Provide a test-bed for experimental development of high-performance analytics
- Offer an architectural approach to climate data services that can be generalized to applications and customers beyond the traditional climate research community

**MERRA Analytic Services**

John L. Salmon*, Daniel G. Dally†, Mark A. McCreary*, Chien S. Tung†, John H. Thompson*, Roger Gil†, and Christina M. Jiang†

*Office of Computational and Information Science and Technology
NASA Center for Climate Simulation (CCS), NASA Goddard Space Flight Center

**Hadoop for High-Performance Climate Analytics**
MERRA A/S Background

- Initially evaluated MapReduce and the Hadoop Distributed File System (HDFS) on representative collections of observational and climate data (MERRA)
  - Focused on a small set of canonical operations such as, average, minimum, maximum, and standard deviation operations over a given temporal and spatial extent
  - Built a cluster with available hardware (then acquired a custom cluster)
  - Implemented a prototype to process the data via MapReduce
  - Captured metrics and observed performance improvements as the number of data nodes and block sizes increase
Project Details

• MERRA/AS…
  • Leverages the Hadoop/MapReduce approach to parallel storage-based computation.
  • Uses a workflow-generated approach to perform analyses over the MERRA data
  • Introduces a generalized application programming interface (API) and web service that exposes reusable climate data services.
Why HDFS and MapReduce?

- Software framework to store large amounts of data in parallel across a cluster of nodes
  - Provides fault tolerance, load balancing, and parallelization by replicating data across nodes
  - Co-locates the stored data with computational capability to act on the data (storage nodes and compute nodes are the same – typically)
  - A MapReduce job takes the requested operation and maps it to the appropriate nodes for computation using specified keys

Who uses this technology?
- Google
- Yahoo
- Facebook

Many PBs and probably even EBs of data.

Hadoop for High-Performance Climate Analytics
Initial Use Case

• Create a time-based average over the monthly means for specific variables
• This example shows a seasonal average of temperature for the winter of 2000
• Focused on reducing the time spent in the preparation of reanalysis data used in data-model inter-comparison, a long sought goal of the climate community
MERRA Data

• The GEOS-5 MERRA products are divided into 25 collections: 18 standard products, 7 chemistry products
• Comprise monthly means files and daily files at six-hour intervals running from 1979 – 2012
• Total size of netCDF MERRA collection in a standard filesystem is ~80 TB
• One file per month/day produced with file sizes ranging from ~20 MB to ~1.5 GB
Map Reduce Workflow

A job containing map-reduce operations is submitted to Hadoop.

The resulting sequence files are sent to Hadoop for storage.

Hadoop distributes the mappers, reducers, etc. across the available nodes.

Hadoop File System (HDFS)

The mappers and reducers utilize the HDFS and the sequence files stored there to perform operations.
Ingesting MERRA data into HDFS

- Option 1: Put the MERRA data into Hadoop with no changes
  » Would require us to write a custom mapper to parse
- Option 2: Write a custom NetCDF to Hadoop sequencer and keep the files together
  » Basically puts indexes into the files so Hadoop can parse by key
  » Maintains the NetCDF metadata for each file
- Option 3: Write a custom NetCDF to Hadoop sequencer and split the files apart
  » Breaks the connection of the NetCDF metadata to the data
- Chose Option 2
Sequence File Format

- During sequencing, the data is partitioned by time, so that each record in the sequence file contains the timestamp and name of the parameter (e.g. temperature) as the composite key and the value of the parameter (which could have 1 to 3 spatial dimensions)
Data Set Descriptions

- Two data sets
  - MAIMNPANA.5.2.0 (instM_3d_ana_Np) – monthly means
  - MAIMCPASM.5.2.0 (instM_3d_asm_Cp) – monthly means

- Common characteristics
  - Spans years 1979 through 2012…..
  - Two files per year (hdf, xml), 396 total files

- Sizing

<table>
<thead>
<tr>
<th>Type</th>
<th>Raw Total (GB)</th>
<th>Sequenced Total (GB)</th>
<th>Raw File (MB)</th>
<th>Sequenced File (MB)</th>
<th>Sequence Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAIMNPANA</td>
<td>84</td>
<td>224</td>
<td>237</td>
<td>565</td>
<td>30</td>
</tr>
<tr>
<td>MAIMCPASM</td>
<td>48</td>
<td>119</td>
<td>130</td>
<td>300</td>
<td>15</td>
</tr>
</tbody>
</table>
Seasonal Averages – Operational Cluster

**MAIMNPANA.5.2.0 (sec)**

<table>
<thead>
<tr>
<th>Years</th>
<th>Period</th>
<th>Test</th>
<th>Operational</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2001</td>
<td>89.1</td>
<td>32.4</td>
<td>2.8</td>
</tr>
<tr>
<td>10</td>
<td>2001 - 2010</td>
<td>475.4</td>
<td>128.8</td>
<td>3.7</td>
</tr>
<tr>
<td>20</td>
<td>1991 - 2010</td>
<td>1026.6</td>
<td>245.2</td>
<td>4.2</td>
</tr>
<tr>
<td>All</td>
<td>1979 - 2011</td>
<td>1520.0</td>
<td>404.7</td>
<td>3.8</td>
</tr>
</tbody>
</table>

**MAIMCPASM.5.2.0 (sec)**

<table>
<thead>
<tr>
<th>Years</th>
<th>Period</th>
<th>Test</th>
<th>Operational</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2001</td>
<td>65.4</td>
<td>18.5</td>
<td>3.5</td>
</tr>
<tr>
<td>10</td>
<td>2001 - 2010</td>
<td>205.0</td>
<td>38.7</td>
<td>5.3</td>
</tr>
<tr>
<td>20</td>
<td>1991 - 2010</td>
<td>358.1</td>
<td>79.8</td>
<td>4.5</td>
</tr>
<tr>
<td>All</td>
<td>1979 - 2011</td>
<td>545.6</td>
<td>110.8</td>
<td>4.9</td>
</tr>
</tbody>
</table>
## Operational Node Configurations

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Bare1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>Dell R720</td>
</tr>
<tr>
<td>Processor Type</td>
<td>Intel Sandy Bridge</td>
</tr>
<tr>
<td>Processor Number</td>
<td>E5-2670</td>
</tr>
<tr>
<td>Processor Speed</td>
<td>2.60 GHz</td>
</tr>
<tr>
<td>Cores per Socket</td>
<td>8</td>
</tr>
<tr>
<td>Number of Sockets</td>
<td>2</td>
</tr>
<tr>
<td>Cores per Node</td>
<td>16</td>
</tr>
<tr>
<td>Main Memory</td>
<td>32 GB</td>
</tr>
<tr>
<td>Storage</td>
<td>12 by 3 TB drives = 36 TB RAW</td>
</tr>
<tr>
<td>Interconnect</td>
<td>Mellanox MT27500 FDR IB</td>
</tr>
<tr>
<td>Operating System</td>
<td>Centos 6.3</td>
</tr>
<tr>
<td>Kernel</td>
<td>2.6.32-279.5.1</td>
</tr>
<tr>
<td>Hadoop</td>
<td>0.20.2</td>
</tr>
<tr>
<td>java-6-sun</td>
<td>1.6.0_24</td>
</tr>
</tbody>
</table>
Open Source Tools

• Using Cloudera (CDH), the open source enterprise-ready distribution of Apache Hadoop.

• Cloudera is integrated with configuration and administration tools and related open source packages, such as Hue, Oozie, Zookeeper, and Impala.

• Cloudera Manager Free Edition is particularly useful for cluster management, providing centralized administration of CDH.
Customer Connections

• NASA ASP A.35 Wildland Fires RECOVER project.
• NSF DataNet Federation Consortium
• SIGClimate

• Others include: GSFC / LARC iRODS Testbed, CSC Climate Edge product line, Applied Science and Terrestrial Ecology Program climate adaptation projects, Direct Readout Laboratory Climate Data Records (CDRs), and NCA modelers
Next Steps

- Tune the MapReduce Framework
- Identify potential performance optimizations (e.g., modify block size, tweak I/O config)
- Complete canonical operations (e.g., add mappers/reducers)
- Try different ways to sequence the files
- Experiment with data accelerators
Conclusions and Lessons Learned

• Design of sequence format is critical for big binary data
• Configuration is key…change only one parameter at a time
• Big data is hard, and it takes a long time….
• Expect things to fail – a lot
• Hadoop craves bandwidth
• HDFS installs easy but optimizing is not so easy
• Not as fast as we thought … is there something in Hadoop that we don’t understand yet
• It’s all still cutting edge to a certain extent
• Ask the mailing list or your support provider