Applying Apache Hadoop to NASA’s Big Climate Data

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), Savannah Strong (CSC)
Overview

• The NASA Center for Climate Simulation (NCCS) is using Apache Hadoop for high-performance analytics because it optimizes computer clusters and combines distributed storage of large data sets with parallel computation.

• We have built a platform for developing new climate analysis capabilities with Hadoop.
• Hadoop is well known for text-based problems. Our scenario involves binary data. So, we created custom Java applications to read/write data during the MapReduce process.

• Our solution is different because it: a) uses a custom composite key design for fast data access, and b) utilizes the Hadoop Bloom filter, a data structure designed to identify rapidly and memory-efficiently whether an element is present.
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.
Background

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
Example Use Case - WEI Experiment

- The Wei team used MERRA data to study four intensively irrigated regions: northern India/Pakistan, the North China Plain, the California Central Valley, and the Nile Valley.

- Seasonal rates of evapotranspiration with and without irrigation over the studied areas were then compared to assess the impact of irrigation.

- The data required for these calculations include average daily precipitation, evapotranspiration, temperature, humidity, and wind at different tropospheric levels at six-hourly time steps from 1979 to 2002.

- This early-stage data reduction—average values for environmental variables over specific spatiotemporal extents—is the type of data assembly that historically has been performed on the scientist’s workstation after transfers from public archives of large blocks of data.

Example Use Case - WEI Experiment

Wei, et al.
- ~8.4 TB transferred from archive to local workstation (weeks)
- Clipping, averaging performed by Fortran program on local workstation (days)

MERRA/AS
- Clipping, averaging performed by MERRA/AS (~28 hrs)
- Only ~35 GB final product transferred to local workstation (minutes)

- Significant time savings in data wrangling,
- rapid screening over monthly means files takes minutes, and
- there's a possibility of folding Dr. Wei's modeling algorithm back into the CDS API...
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.

Mapper
Determines the fields for processing.

Reducer
Performs the operation on the sorted fields from the reducer.

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

The mappers and reducers utilize the HDFS and the sequence files stored there to perform operations.

MERRA Monthly Data Files

MERRA NetCDF files are processed into Hadoop sequence files.

NetCDF-Hadoop Sequence File Converter

The resulting sequence files are sent to Hadoop for storage.

Hadoop splits and distributes the sequence files across the HDFS.

Hadoop File System (HDFS)
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 (allows smaller block sizes)
  » 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)
Bloom Filter

• A Bloom filter, conceived by Burton Howard Bloom in 1970, is a space-efficient probabilistic data structure that is used to test whether an element is a member of a set. False positive retrieval results are possible, but false negatives are not; i.e. a query returns either "inside set (may be wrong)" or "definitely not in set".

• In Hadoop terms, the BloomMapFile can be thought of as an enhanced MapFile because it contains an additional hash table that leverages the existing indexes when seeking data.
Bloom Filter Performance Increase

- The original MapReduce application utilized standard Hadoop Sequence Files. Later they were modified to support three different formats called Sequence, Map, and Bloom.

- Dramatic performance increases were observed with the addition of the Bloom filter (~30-80%).

<table>
<thead>
<tr>
<th>Job Description</th>
<th>Host</th>
<th>Sequence (sec)</th>
<th>Map (sec)</th>
<th>Bloom (sec)</th>
<th>Percent Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read a single parameter (&quot;T&quot;) from a single sequenced monthly means file</td>
<td>Standalone VM</td>
<td>6.1</td>
<td>1.2</td>
<td>1.1</td>
<td>+81.9%</td>
</tr>
<tr>
<td>Single MR job across 4 months of data seeking &quot;T&quot; (period = 2)</td>
<td>Standalone VM</td>
<td>204</td>
<td>67</td>
<td>36</td>
<td>+82.3%</td>
</tr>
<tr>
<td>Generate sequence file from a single MM file</td>
<td>Standalone VM</td>
<td>39</td>
<td>41</td>
<td>51</td>
<td>-30.7%</td>
</tr>
<tr>
<td>Single MR job across 4 months of data seeking &quot;T&quot; (period = 2)</td>
<td>Cluster</td>
<td>31</td>
<td>46</td>
<td>22</td>
<td>+29.0%</td>
</tr>
<tr>
<td>Single MR job across 12 months of data seeking &quot;T&quot; (period = 3)</td>
<td>Cluster</td>
<td>49</td>
<td>59</td>
<td>36</td>
<td>+26.5%</td>
</tr>
</tbody>
</table>
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>
Applying Apache Hadoop to NASA's Big Climate Data

MERRA Cluster Components

Head Nodes
- Namenode
  - /merra 5TB
  - /hadoop_fs 1TB
- JobTracker
  - /mapred 1TB

Data Nodes
- Data Node 1
  - /hadoop_fs 16TB
  - /mapred 16TB
- Data Node 2
  - /hadoop_fs 16TB
  - /mapred 16TB
- Data Node 34
  - /hadoop_fs 16TB
  - /mapred 16TB

FDR IB

LAN

MERRA Data
- 180TB Raw

MERRA Data
- 180TB Raw
## 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>
Other Apache Contributions...

- **Avro** – a data serialization system
- **Maven** – a tool for building and managing Java-based projects
- **Commons** – a project focused on all aspects of reusable Java components
  - **Lang** – provides methods for manipulation of core Java classes
  - **I/O** - a library of utilities to assist with developing IO functionality
  - **CLI** - an API for parsing command line options passed to programs
  - **Math** - a library of mathematics and statistics components
- **Subversion** – a version control system
- **Log4j** - a framework for logging application debugging messages
Other 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.
Next Steps

- Tune the MapReduce Framework
- Try different ways to sequence the files
- Experiment with data accelerators
- Explore real-time querying services on top of the Hadoop file system:
  - Apache Drill
  - Impala (Cloudera)
  - Ceph,
  - MapR…
Conclusions and Lessons Learned

• Design of sequence file format is critical for big binary data
• Configuration is key…change only one parameter at a time for tuning
• 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
• Ask the mailing list or your support provider