NSF CluE Project
Performance Evaluation of On-Demand Provisioning Strategies for Data Intensive Applications

PI: Chaitan Baru
Co-PI: Sriram Krishnan
San Diego Supercomputer Center
UC San Diego
Project Goals

- Investigate whether dynamic allocation of resources (processors/disk) can be more efficient than current static approach for large scientific data archives

- Corollary: Can scientific data portals/archives be effectively backend-ed by cloud platforms?
Acknowledgements

- Other team members:
  - Viswanath Nandigam
  - Christopher Crosby

- Funding:
  - NSF CluE / CISE; GEO/Earth Sciences; Office of Cyberinfrastructure
  - Intermural funding among NSF directorates
Outline

- Example application: OpenTopography.org
- Current implementation
- CluE experiments
Scientific data lifecycle
The application: OpenTopography.org

• Online access to high-resolution topographic data
  • Airborne
  • Terrestrial
LiDAR Data

D. Harding, NASA

Waveform Data

Portal

Full-featured DEM

Bare earth DEM

Point Cloud Dataset

NSF CluE PIs Meeting, Oct 5-6, 2009
Current system characteristics

- **Data**
  - Currently, 7 datasets (3 from EarthScope)
    - Each is independent
  - ~3TB (50% indexes); ~30 billion entries

- **Users**
  - 80-20 split
    - 80% want “standard” products, e.g. pre-computed DEMs, KML
    - 20% want to derive their own data products

- **Access**
  - “Online”, service-oriented access to data is highlight
Sample LIDAR Dataset

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Return Number</th>
<th>Number of Returns</th>
<th>Off Nadir Angle</th>
<th>Return Intensity</th>
<th>Classification Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>2074881.21</td>
<td>1557.35</td>
<td>1 5</td>
<td>5.48</td>
<td>14</td>
<td>V</td>
<td>1205 148544.74364 6135780.64</td>
</tr>
<tr>
<td>2074880.95</td>
<td>1526.20</td>
<td>2 6</td>
<td>5.45</td>
<td>0</td>
<td>G</td>
<td>1205 148544.76241 6135770.33</td>
</tr>
<tr>
<td>2074884.30</td>
<td>1529.68</td>
<td>1 5</td>
<td>5.47</td>
<td>15</td>
<td>G</td>
<td>1205 148544.78122 6135765.79</td>
</tr>
<tr>
<td>2074881.62</td>
<td>1524.31</td>
<td>1 5</td>
<td>5.44</td>
<td>80</td>
<td>B</td>
<td>1205 148544.76246 6135775.01</td>
</tr>
</tbody>
</table>

Column 1: Date - Day or week of acquisition  
Column 2: Time - GPS time stamp uniquely identifying laser pulse time  
Column 3 and 4: X and Y (Lat, Long)  
Column 5: Elevation.  
Column 6: Return number - Return number of this return.  
Column 7: Number of Returns - Number of returns for this pulse.  
Column 8: Off Nadir Angle - Angle between nadir and transmitted pulse  
Column 9: Return Intensity - Intensity of return pulse  
Column 10: Classification Code - Classification of return  

B-Blunder; G-Ground or water; V-Vegetation; S-Building/Structure; N-Not ground or water - Could be V or S
DB2 Parallel (Enterprise Edition) with spatial extender

Database cluster
- 8 x dual-core Intel Xeon 3.0 GHz, 8GB, 750GB disk
- 4 x 3TB disk arrays; 1.5TB/node
Partial declustering of LiDAR data sets

NSF CluE PIs Meeting, Oct 5-6, 2009
Data access patterns
Application Requirements

- Scale to:
  - More data sets
  - Larger data sets
  - Increased processing, e.g. “parameter sweeps” for DEM generation; hydrologic modeling; waveform LiDAR; ecological applications; …
  - More users
Experiences with DBMS

- **Database Pros**
  - Simple SQL-based querying of data
  - Robust production-quality software/hardware stack
  - High performance access to data (shared nothing platform)

- **But**
  - Loading, indexing and storage overhead
  - Scaling to very large configurations (price / performance)
  - ASCII data
MapReduce Implementation

Massage data; Convert to common coordinate system, etc.

Input point cloud data files (ASCII & Binary)

HDFS

MapReduce
- Subsetting
- Output Generation

User Request

Output

MapReduce
- DEM Generation
- Slope, aspect, curvature, etc. calculations
- Rendering

DEM output

NSF CluE PIs Meeting, Oct 5-6, 2009
DEM Generation

- Pushdown into DBMS
  - Generate grid file based on user’s request
  - Do “band join” between dataset and grid file and compute DEM

- MapReduce
  - Implement using HQL / HIVE
  - Programmed in MR
    - Use grid file, create overlapped data regions and compute DEM
    - Merge DEMs
The need for dynamic provisioning

- Provision data based on access patterns
  - DBMS provides much better response for smaller datasets
  - MapReduce may be better for much larger datasets

- Access patterns change:
  - Over time (after initial release of data)
  - When events occur
  - When results are published, etc.

- Personalized database for intensive analysis (myDB)

- Performance vs cost differential
  - DBMS on small configs vs Hadoop on large configs
Experiences with MapReduce

- SQL vs MapReduce
  - Programmer is the optimizer
- Investigating HIVE, but may not yet be suitable for complex SQL
  - We require spatial functions, “band joins”, outer joins
  - Need extensions to HIVE
Experiments

- “On-demand” database vs Hadoop
- Implement binary data formats
- Provision different data sets, or different parts of a data set, differently

Platforms to be used
- Google-IBM cluster
- 8-node lab cluster at SDSC
- UIUC CCT (Cluster Computing Testbed)
- SDSC Triton resource
- AWS for Education