Distributed Technologies for Remote Access of HDF Data

Svetlana G. Shasharina, Chuang Li, Nanbor Wang, Rooparani Pundaleeka, David Wade-Stein

Funded by DOE/SBIR Grant No DE-FG03-02ER83852 and Tech-X
Outline

• Problem addressed
• Candidate solutions
• Evaluation
• Status of the chosen solution
• Future directions
One cannot have and does not need all the data

- Data gets generated at remote sites (supercomputers at National Energy Research Scientific Computing Center - NERSC, clusters, experiments)
- Data is typically large
- Analysis and visualization need to be local
- Hard to get all data locally (limited storage, bandwidth etc)
- If the data is local - use API (like HDF5) to get what you need
- One needs to make the same API to work on remote data
HDF5 allows to store and access simulations data locally

- HDF5 is a format and API
  - Binary
  - Hierarchical (datasets for multidimensional arrays, images, grids; attributes for metadata
  - Allows to query and access metadata, full datasets, partial datasets (hyberslabs)
  - Efficient (parallel I/O)

- HDF5 is used in plasma physics, earth scientists, medical fields, neutronics, HEP/NP etc

- Our goal was to make HDF5-like API for remote data access
Distributed technologies allow making remote APIs from local APIs

- Start from “language-neutral” interface
- Generate client stubs (proxies of working objects), server stubs that need to be implemented to be working objects
- Stubs can be in different languages
Technologies used in evaluated solutions

• gsSoap (Web Service):
  – C/C++ bindings
  – DIME - binary blurbs of data get transmitted rather than verbose SOAP

• CORBA(TAO):
  – C++ bindings
  – Well suited for binary data

• Globus:
  – Promoted heavily
  – C/C++ binding
  – GridFTP comes naturally to avoid SOAP for large data
H5WS is Globus-Based Web Service
Test setting

- LAN - 2 Tech-X Linus boxes (grid and storage2)
- ESnet - NERSC (davinci) and PPPL (grid0)
- WAN - NERSC to Tech-X

<table>
<thead>
<tr>
<th>Setup</th>
<th>Bottleneck bandwidth (Mbyte/sec)</th>
<th>RTT (msec)</th>
<th>BDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAN</td>
<td>12.5</td>
<td>0.27</td>
<td>3.4 Kbytes</td>
</tr>
<tr>
<td>ESnet</td>
<td>125</td>
<td>72</td>
<td>9 Mbytes</td>
</tr>
<tr>
<td>WAN</td>
<td>0.19</td>
<td>162</td>
<td>31.4 Kbytes</td>
</tr>
</tbody>
</table>
Test scenario

- Locate the server and establish connection.
- Query the metadata of target dataset.
- Using the metadata, implicitly or explicitly allocate dynamic client memory for accommodating the dataset.
- Retrieve the dataset into the client memory.
- Disconnect from server.
- Release the memory and clean all the resources.
- GridFTP scenario has an extra step - saving the extracted dataset in a file on the server.
H5WS (GT4+GridFTP) has too much overhead and loses in LAN, while CORBA (TAO) saturates the connection, gSoap is halfway.

Data Access Throughput over LAN

- GT4+GridFTP/P1
- TAO
- gSoap+DIME

throughput (MB/s)

data size (MB)
gSoap and CORBA (TAO) wins over 1 stream H5WS in Esnet tests, but the throughput is much lower than the bottleneck (BDP is much higher than the TCP window = 64KB)
H5WS with many streams has the highest throughput in Esnet (still much less than the bottleneck)
H5WS wins and close to bottleneck throughput in WAN tests
Tests conclusions

- H5WS (GT4 +GridFTP) does not make sense for LAN
- H5WS overall wins in remote settings (both ESnet and WAN)
- Using multiple streams in LAN and WAN does not give advantage, but makes sense for Esnet
- That is why we proceeded with H5WS
H5WS status

• Allows
  – Query remote HDF5 file if a dataset is there
  – Query the dataset about its dimensions and rank
  – Extract dataset to the client machine
  – Extract hyberslabs (arbitrary cubes of data)

• C++ and command client interface
Future directions and conclusion

• Visualization clients wrapping C++ clients (VisIt, possibly)
• Reconsidering Globus (heavy-weight, not well suited for C/C++ bindings as adding external libs and includes in the implementation requires hacking,)