Visualization with WSRF
A Concrete Example

The Usable Grid Workshop

Monday, March 14, 2005
By Pascal Kleijer
Outline

- Project Overview
- Current Status
- WSRF Application
- Benchmarking
- Conclusion
- Demo
Project Overview

• **Background**
  – Essential tool for scientific simulations
  – Critical demand for coping with data on the Grid

• **Objectives**
  – Visualization large-scale distributed data
  – Generalized Grid visualization services

• **Mission**
  – Contribution to the improvement of usability and practicability of the Grid
Current Status

• Visualization Functionalities for Massive Data
  – **Extension**: molecular structures, iso-surfaces, etc.
  – **Application**: high-speed visualization of huge MD simulation results (3 million atoms)

• Visualization Grid Service Prototyping
  – WSRF-based interactive visualization system
  – Integrated parallel visualization module and 3rd party software (PyMOL) into a partial visualization service.
Example of Massive Data Visualization

- Motion of 1 million molecules (3 million atoms)

Data courtesy: Institute for Molecular Science
Grid Visualization Service Framework

Integrated WSRF-based remote visualization environment

NAREGI Middleware
- Workflow Tool
- Information Service
- GridMPI

Demo System
- GVS Client (Reference Implementation)
- Particular Client Conformed to Standard Interface

Through a standard interface
Integration of various visualization software

Integration of other existing visualization software

- Coupled Sim. Visualization Service
- Post-proc. Visualization Service
- DB Visualization Service
- Molecular Visualization Service
- CFD Visualization Service
- Particular Visualization Service

Parameter Integration of other existing visualization software

Images

- Adapter
- Coupled Sim. Parallel Visualizer
- Coupled Simulation
- Adapter
- Coupled Sim. Parallel Visualizer
- Coupled Simulation
- Adapter
- Post-proc. Parallel Visualizer
- Computation Results
- Adapter
- DB Parallel Visualizer
- DB (DataGrid, etc.)
- Adapter
- Molecular Visualization Software (PyMOL, etc.)
- Adapter
- CFD Visualization Software
- Adapter
- Particular Visualization Software

Empowered by Innovation
NII
WSRF Application

The Grid

User Machine
- GVS Client
  - Visualization WSRF Service
  - Globus (Passive)

Server Machine
- GVS Server
  - GVS Server Adapter
  - Visualization Service
  - Globus (Active)

Grid Service Machine
- PyMOL Adapter
  - Visualization Service
  - Globus (Active)
WSRF Application

User Machine
  GVS Client
  Visualization WSRF Service
  Globus (Passive)

Server Machine
  GVS Server
  GVS Server Adapter
  Visualization Service
  Globus (Active)

Grid Service Machine
  GVS Client
  Visualization WSRF Service
  Globus (Passive)
  GVS Server
  GVS Server Adapter
  Visualization Service
  Globus (Active)
  Python / PyMOL
  PyMOL Adapter
  Visualization Service
  Globus (Active)

The Grid
Benchmarking

• Objectives
  – Evaluate the impact of the Grid Layer on P2P applications.
  – Analyze the feasibility of real-time applications over the grid.
  – What is the performance state of the Grid Services compared to Web Services.

→ We only intend to benchmark the middleware
Benchmarking (2)

• Setup
  – 2 Values tested
    • Initialization Time
    • Image Sending Time
  – 3 Configurations
    • Local
    • LAN
    • Internet
  – 2 Models
    • Globus (WSRF Grid Service)
    • Tomcat (Web Service, HTTP Streaming)
Benchmarking (3)

• **Data Set**
  – Type: Reovirus Core
  – PDB: 1EJ6 (Bio. Unit)
  – Size: +2 millions Atoms
  – Coloring: Residue Seq.
  – Radius: ~315 Å
  – Image: 512x512 pixels, ~200 KB JPEG

• **Configuration**
  – GVS Client + GVS Server
## Benchmarking (4)

<table>
<thead>
<tr>
<th>Target:</th>
<th>User</th>
<th>Service</th>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Local:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>2x P3 933 MHz</td>
<td>2x P3 933 MHz</td>
<td>8x Xeon 2.4 GHz</td>
</tr>
<tr>
<td>RAM</td>
<td>1.8GB / 2 CPU</td>
<td>1.8GB / 2 CPU</td>
<td>2.3 GB / 2 CPU</td>
</tr>
<tr>
<td><strong>LAN:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>PM 1.3 GHz</td>
<td>2x Xeon 3.06 GHz</td>
<td>16x Xeon 2.8 GHz</td>
</tr>
<tr>
<td>RAM</td>
<td>512 MB</td>
<td>688 Mb / 2 CPU</td>
<td>1.0 GB / 2 CPU</td>
</tr>
<tr>
<td><strong>Internet:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>PM 1.3 GHz</td>
<td>2x Xeon 3.06 GHz</td>
<td>16x Xeon 2.8 GHz</td>
</tr>
<tr>
<td>RAM</td>
<td>512 MB</td>
<td>688 Mb / 2 CPU</td>
<td>1.0 GB / 2 CPU</td>
</tr>
</tbody>
</table>
Benchmarking (5)

• **Results:** *Initialization*

![Bar chart showing times in seconds for Globus and Tomcat, with categories for Local, LAN, and Internet, and specific times indicated.]
Benchmarking (6)

- **Results**: Images

![Bar chart showing time in seconds for different protocols]

- **Globus**: 0.899 (Local), 2.158 (Internet), 0.36 (LAN)
- **Tomcat**: 0.013 (Local), 0.043 (LAN), 0.615 (Internet)
Conclusions

• WSRF Cons
  – Slow,
  – Uns suited for large-scale Data,
  – Needs Polling.

• Needs
  – Streaming (not GridFTP or GridMPI),
  – Binary Transport,
  – Lower Latency.
Conclusions (2)

• Real-time over Grid?
  – Yes, but it’s still immature.

• Cost of porting/wrapping to the Grid?
  – Could seriously be improved
  – Isn’t too difficult once the initial phase is assimilated.