An Intelligent Rule-Oriented Data Management System

Wayne Schroeder
San Diego Supercomputer Center,
University of California San Diego
Talk Outline

• Background
  • Brief Overview of the SDSC SRB
  • Current Projects/Usage
  • Activities/Plans

• Rule-Oriented Data Management System
  • iRODS Requirements/Planning
  • Architecture
  • Infrastructure Development
  • Collaborations/Plans
Using a Data Grid – in Abstract

Data Grid

- User asks for data from the data grid
- The data is found and returned
  - Where & how details are hidden
Using a Data Grid - Details

- The data is found and returned
Using a Data Grid - Details

- Data Grid has arbitrary number of servers
- Complexity is hidden from users
Storage Resource Broker
A Data Grid Solution

• Collaborative client-server system that federates distributed heterogeneous resources using *uniform interfaces* and *metadata*

• Provides a simple tool to integrate data and metadata handling – *attribute-based access*

• Blends browsing and searching

• Developed at SDSC
  - Operational for 7+ years;
  - Under continual development since 1997;
  - Customer-driven
Some SRB Features

The SRB is an integrated solution which includes:
- a logical namespace,
- interfaces to a wide variety of storage systems,
- high performance data movement (including parallel I/O),
- fault-tolerance and fail-over,
- WAN-aware performance enhancements (bulk operations),
- storage-system-aware performance enhancements ('containers' to aggregate files),
- metadata ingestion and queries (a MetaData Catalog (MCAT)),
- user accounts, groups, access control, audit trails, GUI administration tool
- data management features, replication
- user tools (including a Windows GUI tool (inQ), a set of SRB Unix commands, and Web (mySRB)), and APIs (including C, C++, Java, and Python).

SRB Scales Well (many millions of files, terabytes)

Supports Multiple Administrative Domains / MCATs (srbZones)

And includes SDSC Matrix: SRB-based data grid workflow management system to create, access and manage workflow process pipelines.
Recent SRB Release, April 28

• Any valid ASCII characters are now acceptable in SRB filenames, except a string of two quotes in a row
• Data integrity and vault management
• Quota System
• SRB Web Perl Portal
• SRB account management via grid-mapfile
• Real time data management
• New driver for NCAR MSS
• Completely reworked web site/documentation system (MediaWiki)
• Other new features
• Critical bug patches for in 3.4.0 included
• Other bugzilla fixes (about 35)
• MCAT Patch
Recent SRB Releases

- 3.4.1 April 28, 2006
- 3.4 October 31, 2005
- 3.3.1 April 6, 2005
- 3.3 February 18, 2005
- 3.2.1 August 13, 2004
- 3.2 July 2, 2004
- 3.1 April 19, 2004
- 3.0.1 December 19, 2003
- 3.0 October 1, 2003
- 2.1.2 August 12, 2003
- 2.1.1 July 14, 2003
- 2.1 June 3, 2003
- 2.0.2 May 1, 2003
- 2.0.1 March 14, 2003
- 2.0 February 18, 2003
SRB Projects

- Astronomy
  - National Virtual Observatory
- Data Grids
  - UK e-Science CCLRC
  - Teragrid
- Digital Libraries and Archives
  - National Archives and Records Administration
  - National Science Digital Library
  - Persistent Archive Testbed
- Ecological, Environmental, Oceanographic
  - ROADnet
  - Southern California Earthquake Center
  - SIO Digital Libraries
- Molecular Sciences
  - Synchrotron Data Repository
  - Alliance for Cellular Signaling
- Neuro Sciences
  - Biomedical Information Research Network
- Physics and Chemistry
  - BaBar
- Many others

Over 650 Tera Bytes in 106 million files
SRB Scalability

- Over 2 Petabytes World-wide
- Major SRB instances in the UK, Australia, Taiwan, US
  - United Kingdom - UK e-Science
  - Australia - APAC
  - Taiwan - Academia Sinica, NCHC
  - Europe - IN2P3, Italy, Norway
  - United States
- 660 Terabytes at SDSC
  - 100 Million files
  - SAM QFS, HPSS, Unix file system, SRB Bricks
### SDSC Hosted SRB Data

<table>
<thead>
<tr>
<th>Project</th>
<th>GBs of data stored</th>
<th>GBs of data stored</th>
<th>1000's of files</th>
<th>1000's of files</th>
<th>Users with ACLs</th>
<th>GBs of data stored</th>
<th>1000's of files</th>
<th>Users with ACLs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Grid</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NSF / NVO</td>
<td>17,800</td>
<td>51,380</td>
<td>51,380</td>
<td>8,690</td>
<td>80</td>
<td>93,252</td>
<td>11,189</td>
<td>100</td>
</tr>
<tr>
<td>NSF / NPACI</td>
<td>1,972</td>
<td>17,578</td>
<td>17,578</td>
<td>4,694</td>
<td>380</td>
<td>34,452</td>
<td>7,235</td>
<td>380</td>
</tr>
<tr>
<td>Hayden</td>
<td>6,800</td>
<td>7,201</td>
<td>7,201</td>
<td>113</td>
<td>178</td>
<td>8,013</td>
<td>161</td>
<td>227</td>
</tr>
<tr>
<td>Pzone</td>
<td>438</td>
<td>812</td>
<td>812</td>
<td>47</td>
<td>49</td>
<td>19,674</td>
<td>10,627</td>
<td>68</td>
</tr>
<tr>
<td>NSF / LDAS-SALK</td>
<td>239</td>
<td>4,562</td>
<td>4,562</td>
<td>16</td>
<td>66</td>
<td>104,494</td>
<td>131</td>
<td>67</td>
</tr>
<tr>
<td>NSF / SLAC-JCGS</td>
<td>514</td>
<td>4,317</td>
<td>4,317</td>
<td>563</td>
<td>47</td>
<td>15,703</td>
<td>1,666</td>
<td>55</td>
</tr>
<tr>
<td>NSF / TeraGrid</td>
<td>80,354</td>
<td>2,962</td>
<td>2,962</td>
<td>685</td>
<td>195,012</td>
<td>4,071</td>
<td>3,267</td>
<td></td>
</tr>
<tr>
<td>NIH / BIRN</td>
<td>5,416</td>
<td>3,366</td>
<td>3,366</td>
<td>148</td>
<td>13,597</td>
<td>13,329</td>
<td>351</td>
<td></td>
</tr>
<tr>
<td><strong>Digital Library</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NSF / LTER</td>
<td>158</td>
<td>233</td>
<td>233</td>
<td>6</td>
<td>35</td>
<td>236</td>
<td>34</td>
<td>36</td>
</tr>
<tr>
<td>NSF / Portal</td>
<td>33</td>
<td>1,745</td>
<td>1,745</td>
<td>48</td>
<td>384</td>
<td>2,620</td>
<td>53</td>
<td>460</td>
</tr>
<tr>
<td>NIH / AfCS</td>
<td>27</td>
<td>462</td>
<td>462</td>
<td>49</td>
<td>21</td>
<td>733</td>
<td>94</td>
<td>21</td>
</tr>
<tr>
<td>NSF / SIO Explorer</td>
<td>19</td>
<td>1,734</td>
<td>1,734</td>
<td>601</td>
<td>27</td>
<td>2,452</td>
<td>1,068</td>
<td>27</td>
</tr>
<tr>
<td>NSF / S Cec</td>
<td>15,246</td>
<td>1,737</td>
<td>1,737</td>
<td>52</td>
<td>153,159</td>
<td>3,229</td>
<td>73</td>
<td></td>
</tr>
<tr>
<td><strong>Persistent Archive</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NARA</td>
<td>7</td>
<td>63</td>
<td>63</td>
<td>81</td>
<td>58</td>
<td>2,703</td>
<td>1,906</td>
<td>58</td>
</tr>
<tr>
<td>NSF / NSDL</td>
<td>2,785</td>
<td>20,054</td>
<td>20,054</td>
<td>119</td>
<td>5,205</td>
<td>50,586</td>
<td>136</td>
<td></td>
</tr>
<tr>
<td>UCSD Libraries</td>
<td>127</td>
<td>202</td>
<td>202</td>
<td>29</td>
<td>190</td>
<td>208</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>NHPRC / PAT</td>
<td>101</td>
<td>474</td>
<td>474</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td>28 TB</td>
<td>6 mil</td>
<td>194 TB</td>
<td>40 mil</td>
<td>4,635</td>
<td>655 TB</td>
<td>106 mil</td>
<td>5,383</td>
</tr>
</tbody>
</table>
Case Study: SRB in BIRN
Federated SRB Operation

**Logical Name Or Attribute Condition**

**Read Application in Boston**

**Peer-to-peer Brokering**

**Parallel Data Access**

**SRB server**

**SRB agent**

San Diego

**SRB server**

Durham

**SRB agent**

R1

MCAT

R2

Data Access

Server(s) Spawning

1. Logical-to-Physical mapping
2. Identification of Replicas
3. Access & Audit Control

1. 2. 6. 3. 5. 5/6
IRODS - the Next Generation of Data Grid Technology
Moving Forward, a Two-Prong Plan

Maintain and Adapt SRB to New Usages:
SRB has reached a Stable Plateau
• Bug Fixes
• Some New Features
• Merge Features Developed by others
• Continue Testing
• Improve Documentation
• Continue Application Support
  • Existing and new Projects
  • Continue Answering User Queries

Chart New Areas
• Federation Research - ZoneSRB
  • Collaborative Data Grids
• Real-time Data Grids -
  • Virtual Object Ring Buffer
• Sensors and Video Streams
  • Collaborating Observatories
• SRB Workflows - New UI for Admins and users
  • Kepler actors, Matrix, etc
• iRODS - Adaptive Middleware Architecture
Continuing SRB Support

• 10 FTEs SRB
• 5 FTEs iRODS
• iRODS Developers Support SRB
Next generation Data Architecture

- SRB is quite complex – with too many functions and operations
  - The intelligence is hard-coded
    - extensions/modifications require extreme care
  - But, the modules are fairly robust and reusable
- AIM: Can we make SRB more flexible
  - Easy to customize at finer level
    - Example: Higher authentication for a particular collection
    - Example: Can we use stricter authorization for a collection
    - Example: Can we treat a particular resource differently
  - Currently- needs code changes
- Solution: Use rule-based architecture to provide flexibility
iRODS

- A New Paradigm in Middleware Development
  - Flexible Collection management
  - Can be customized at user/collection-levels, …
  - Language for Collection management
    - As in stored procedures, triggers (RDB)
  - Administrative ease
  - Lot of potential beyond SRB
    - adaptive middleware architectures
  - This will be a fully Open Source effort
Rule-Oriented Data Systems Framework

Client Interface

Rule Invoker

Rule

Engine

Current State

Admin Interface

Service Manager
Rule Modifier Module
Config Modifier Module
Metadata Modifier Module

Consistency Check Module
Consistency Check Module
Consistency Check Module

Confs

Rule Base

Meta Data Base

Resources

Resource-based Services
Micro Service Modules

Metadata-based Services
Micro Service Modules
Rule-oriented Data System
(Phase I Operational Model)
Rules and Constraints

- Rule-based
  - Lower-level Functions are composed of micro-services
  - Higher-level Functions are composed of rules of lower-level micro-services
  - Rules are interpreted using a rule engine
- Customizability
- Problems with rule composition
  - Integrity checks to make sure rules do not break higher-level functionalities
- Declarative programming
  - Rules define semantics
- Operational programming
  - Rule invocation provides procedural interpretation
- Rules can be used as “checks and balances” to make sure that collections are self-consistent
  - Example: Rule makes two copies of each files
  - Constraint checking: can be used to see if the collection is consistent with this rule
Rule Scalability and Decidability

Distinct Sets of Rules Applied in Different Ways
- Atomic
- Deferred (state flags)
- Compound
- Applied Using Micro-services

Granularity
- User Input to Influence Rule Expression
- Administration Enforcement
- Collection Consistency Management

Rule Properties
- Metadata Managing Execution (granularity, periodicity)
- Metadata Defining Result of Rule Execution
**Sample Rules**

\[
\text{ingestInCollection}(S) \leftarrow /* \text{store \& backup} */ \\
\quad \text{chkCond1}(S), \text{ingest}(S), \text{register}(S) \\
\quad \text{findBackUpRsrc}(S, \text{Coll}, R), \text{replicate}(S, R).
\]

\[
\text{ingestInCollection}(S) \leftarrow /* \text{store \& check} */ \\
\quad \text{chkCond2}(S), \text{computeClntChkSum}(S, C1), \\
\quad \text{ingest}(S), \text{register}(S), \\
\quad \text{computeSerChkSum}(S, C2), \\
\quad \text{checkAndRegisterChkSum}(C1, C2, S).
\]

\[
\text{ingestInCollection}(S) \leftarrow /* \text{store, check, backup \& check} */ \\
\quad \text{chkCond3}(S), \text{computeClntChkSum}(S, C1), \\
\quad \text{ingest}(S), \text{register}(S), \\
\quad \text{computeSerChkSum}(S, C2), \\
\quad \text{checkAndRegisterChkSum}(C1, C2, S), \\
\quad \text{findBackUpRsrc}(S, \text{Coll}, R), \text{replicate}(S, R) \\
\quad \text{computeSerChkSum}(S, C3), \text{checkAndRegisterChkSum}(C2, C3, S).
\]

\[
\text{ingestInCollection}(S) \leftarrow /* \text{store, check, backup \& extract metadata} */ \\
\quad \text{chkCond4}(S), \text{computeClntChkSum}(S, C1), \\
\quad \text{ingest}(S), \text{register}(S), \\
\quad \text{computeSerChkSum}(S, C2), \\
\quad \text{checkAndRegisterChkSum}(C1, C2, S), \\
\quad \text{findBackUpRsrc}(S, \text{Coll}, R), [\text{replicate}(S, R) \| \text{extractRegisterMetadata}(S)].
\]

\[
\text{ingestInCollection}(S) \leftarrow /* \text{just store} */ \text{ingest}(S), \text{register}(S).
\]

---

\[
\text{chkCond1}(S) \leftarrow \text{user}(S) \Rightarrow \text{‘adil@cclrc’}. \\
\text{chkCond2}(S) \leftarrow \text{coll}(S) \approx \text{‘/scec.sdsc/img/*’}. \\
\text{chkCond3}(S) \leftarrow \text{user}(S) \Rightarrow \text{‘@nara’}. \\
\text{chkCond3}(S) \leftarrow \text{user}(S) \Rightarrow \text{‘@salk’}. \\
\text{chkCond4}(S) \leftarrow \text{user}(S) \Rightarrow \text{‘@birn’}, \\
\text{datatype}(S) \Rightarrow \text{‘DICOM’}.
\]

**OprList** implies delay for later
or send to a CronJobManager

**Opr||Opr** implies do them in parallel

**Opr, Opr** implies do them serially
**New DataGrid Technology**

- Next Generation SRB -- iRODS: Intelligent Rule-Oriented Data Systems
- Customizable and Flexible – User Configurable
- Administratively Simpler – Admin Configurable
- Build upon the experience of SRB Data Grid
- Transition from SRB to iRODS
  - Client-level similarity
  - Meta Catalog transition
- Current NSF Funding
  - Information Technology Research
  - 2 years
  - ~ 2 FTEs
  - Simple proto-type in a year
  - Started September 2004
- Rule-based architecture
- Follow-on funding
  - NARA
  - NSF
iRODS Collaborations

• SRB/iRODS Developers
  • Arcot Rajasekar
  • Michael Wan
  • Wayne Schroeder
  • Other SRB Team Members

• Collaborative Development
  • UK e-Science
  • University of Queensland
  • University of Maryland
  • Others