Policy-driven Distributed Data Management (iRODS)

Richard Marciano
marciano@unc.edu

• Professor @ SILS / UNC
• Chief Scientist for Persistent Archives and Digital Preservation @ RENCI
• Director of the Sustainable Archives & Library Science Lab @ DICE Center
iRODS: integrated Rule-Oriented Data System

• It is adaptive middleware that provides a flexible, extensible and customizable data grid architecture.
• It supports extensibility and customizability by encoding operations into sequences of micro-services.
• It is a data grid software system developed by the Data Intensive Cyber Environments (DICE) group and collaborators.
• It introduces management policies (sets of assertions that communities make about their digital collections) which are implemented as machine-actionable rules and state information.
• At its core, a Rule Engine interprets the rules to decide how the system is to respond to various requests and conditions.
• It is open source under a BSD license.
Data Management Applications
(What do they have in common?)

- Data grids
  - **Share data** - organize distributed data as a collection
- Digital libraries
  - **Publish data** - support browsing and discovery
- Persistent archives
  - **Preserve data** - manage technology evolution
- Real-time sensor systems
  - **Federate sensor data** - integrate across sensor streams
- Workflow systems
  - **Analyze data** - integrate client- & server-side workflows
Evolution of Data Grid Technology

- **Shared collections**
  - Enable researchers at multiple institutions to collaborate on research by sharing data
  - Focus was on performance, scalability

- **Digital libraries**
  - Support provenance information and discovery
  - Integrated with digital library front end services

- **Preservation environments**
  - Support preservation policies
  - Build rule-based data management system

- **Differ in choice of management policies**
Using a Data Grid - Details

- User asks for data
- Data request goes to iRODS Server
- Server looks up information in catalog
- Catalog tells which iRODS server has data
- 1st server asks 2nd for data
- The 2nd iRODS server applies rules
Generic Infrastructure

- Data grids manage data distributed across multiple types of storage systems
  - File systems, tape archives, object ring buffers
- Data grids manage collection attributes
  - Provenance, descriptive, system metadata
- Data grids manage technology evolution
  - At the point in time when new technology is available, both the old and new systems can be integrated
Data Grids

- **Data virtualization**
  - Provide the persistent, global identifiers needed to manage distributed data
  - Provide standard operations for interacting with heterogeneous storage system
  - Provide standard actions for interacting with clients

- **Trust virtualization**
  - Manage authentication and authorization
  - Enable access controls on data, metadata, storage

- **Federation**
  - Controlled sharing of name spaces, files, and metadata between independent data grids
  - Data grid chaining / Central archives / Master-slave data grids / Peer-to-Peer data grids
Each community implements different management polices
- Community specific preservation objectives
- Community specific assertions about properties of the shared collection
- Community specific management policies

Need a mechanism to support the socialization of shared collections
- Map from assertions made by collection creators to expectations of the users
Preservation Rules

• **Authenticity**
  • Rules that quantify required descriptive metadata
  • Rules that verify descriptive metadata is linked to records
  • Rules that govern creation of AIPs

• **Integrity**
  • Rules that verify records have not been corrupted
  • Rules that manage replicas
  • Rules that recover from corruption instances
  • Rules that manage data distribution

• **Chain of custody**
  • Persistent identifiers for archivists, records, storage
  • Rules to verify application of access controls
  • Rules to track storage location of records
Rule Execution Modes

- **Immediate Execution**
  - acCreateUser, acDeleteUser (see core.irb for others)

- **Delayed and Periodic Execution**
  - acPostProcForPut|delayExec((<PLUSET>1h</PLUSET>,msiSysChksumDataObj,nop)|nop

- **Remote Execution** (‘for parking micro-services’) 
  - remoteExec( yellow.unc.edu, null, msiDataObjChksum(*OjbName,verifyChecksum,*Status),nop)

- **Future Execution Modes (to be specified)**
  - Parallel execution (e.g. a checksum and a replication)
  - Broadcast execution
  - Mixing modes

- **General Syntax of delayExec:**
  - delayExec
    - Hints,
    - Micro-service-chains,
    - Recovery-micro-service-chains
Rationale for a “Delayed Execution Service”

Post-processing operations
• Synchronization of replicas
  • After changing one of the copies... a “dirty bit” is turned on to indicate that all further reads and writes should be done to the modified copy. A synchronization command can be executed at a future time to update all of the replicas.
• Validation of checksums
  • Even after a file has been stored successfully, it may become corrupted. Thus, checksums must be continually revalidated to ensure integrity, preferably at a frequency that is four times faster than the expected degradation rate. A timestamp is needed for each object for when the checksum was last validated.
• Placement of files within a distributed storage environment
  • A logical resource name can be used to identify the locations of the multiple storage systems where the copies will reside. If a new physical resource is added to the logical resource name, then a copy would need to be created at the new location. A change flag is needed to denote that the replication operation should be executed.
• Extraction of metadata
  • Metadata can be extracted upon ingest or in a deferred way.
• Conversion of formats
  • A service that would detect obsolescence could trigger differed conversion services.
<table>
<thead>
<tr>
<th>Date</th>
<th>Project</th>
<th>5/17/02</th>
<th>6/30/04</th>
<th>11/29/07</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GBs of data stored</td>
<td>1000Os of files</td>
<td>GBs of data stored</td>
<td>1000Os of files</td>
</tr>
<tr>
<td>Data Grid</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NSF / NVO</td>
<td>17,800</td>
<td>5,139</td>
<td>51,380</td>
<td>8,690</td>
</tr>
<tr>
<td>NSF / NPACI</td>
<td>1,972</td>
<td>1,083</td>
<td>17,578</td>
<td>4,694</td>
</tr>
<tr>
<td>Hayden</td>
<td>6,800</td>
<td>41</td>
<td>7,201</td>
<td>113</td>
</tr>
<tr>
<td>Pzone</td>
<td>438</td>
<td>31</td>
<td>812</td>
<td>47</td>
</tr>
<tr>
<td>NSF / LDAS-SALK</td>
<td>239</td>
<td>1</td>
<td>4,562</td>
<td>16</td>
</tr>
<tr>
<td>NSF / SLAC-JCSG</td>
<td>514</td>
<td>77</td>
<td>4,317</td>
<td>563</td>
</tr>
<tr>
<td>NSF / TeraGrid</td>
<td>80,354</td>
<td>685</td>
<td>2,962</td>
<td>282,536</td>
</tr>
<tr>
<td>NIH / BIRN</td>
<td>5,416</td>
<td>3,366</td>
<td>148</td>
<td>20,400</td>
</tr>
<tr>
<td>NCAR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Digital Library</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NSF / LTER</td>
<td>158</td>
<td>3</td>
<td>233</td>
<td>6</td>
</tr>
<tr>
<td>NSF / Portal</td>
<td>33</td>
<td>5</td>
<td>1,745</td>
<td>48</td>
</tr>
<tr>
<td>NIH / AfCS</td>
<td>27</td>
<td>4</td>
<td>462</td>
<td>49</td>
</tr>
<tr>
<td>NSF / SIO Explorer</td>
<td>19</td>
<td>1</td>
<td>1,734</td>
<td>601</td>
</tr>
<tr>
<td>NSF / SCEC</td>
<td>15,246</td>
<td>1,737</td>
<td>52</td>
<td>168,931</td>
</tr>
<tr>
<td>LLNL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHRON</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Persistent Archive</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NARA</td>
<td>7</td>
<td>2</td>
<td>63</td>
<td>81</td>
</tr>
<tr>
<td>NSF / NSDL</td>
<td>2,785</td>
<td>20,054</td>
<td>119</td>
<td>7,499</td>
</tr>
<tr>
<td>UCSD Libraries</td>
<td>127</td>
<td>202</td>
<td>29</td>
<td>5,205</td>
</tr>
<tr>
<td>NHPRC / PAT</td>
<td>2,576</td>
<td></td>
<td></td>
<td>2,576</td>
</tr>
<tr>
<td>RoadNet</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UCTV</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Earth Sci</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>28 TB</td>
<td>6 mil</td>
<td>194 TB</td>
<td>40 mil</td>
</tr>
</tbody>
</table>
Data Virtualization

Data Access Methods (C library, Unix, Web Browser)

Data Collection

Storage Repository
- Storage location
- User name
- File name
- File context (creation date, …)
- Access controls

Data Grid
- Logical resource name space
- Logical user name space
- Logical file name space
- Logical context (metadata)
- Access constraints

Data is organized as a shared collection
### Policy Virtualization

Data Access Methods (Web Browser, DSpace, OAI-PMH)

#### Data Collection

**Storage Repository**
- Storage location
- User name
- File name
- File context (creation date, …)
- Access controls

**Data Grid**
- Logical resource name space
- Logical user name space
- Logical file name space
- Logical persistent state
- Logical rule name space
- Logical micro-service name
iRODS Rule Syntax

- **Event | Condition | Action-set | Recovery-set**
  - **Event** - triggered by synchronous operation or asynchronous operations, or queued rule, or periodic rule
  - **Condition** - composed from tests on any attributes in the persistent state information
  - **Action-set** - server-side workflow composed from both micro-services and rules
  - **Recovery-set** - recovery workflow used to ensure transaction semantics and consistent state information
iRODS Rules

• Rule condition is a test on any metadata attribute
• Action set generates metadata from tracking remote operations
• Recovery set enforces consistency of metadata

• In distributed environment, must periodically verify compliance of system with desired properties
Types of Rules

• **Authenticity**
  - Extract required descriptive metadata and register into iCAT
  - Verify presence of required descriptive metadata

• **Integrity**
  - Automate resource selection and data distribution
  - Automate creation of replicas
  - Verify records have not been corrupted
  - Automate retention and disposition policy

• **Chain of custody**
  - Periodically parse audit trails for compliance with policy
  - Monitor storage utilization
  - Time-dependent access controls
Federation

• Set of policies that govern interactions between independent data grids
  • Sharing of name spaces
  • Control of procedures that remote users may invoke
  • Tracking of application of procedures (which data grid holds the resulting state information)