Virtualization of the Data Life Cycle

Address Communication Challenges

Reagan Moore
rwmoore@renci.org
http://irods.diceresearch.org
Advancing Scholarly Communication

• Establish social consensus
  • Shared specification of collection properties
    • Type of material
    • Semantics for interpreting the material
    • Allowed data formats
  • Shared specification of management policies

• Build a collection
  • Enforce management policies
  • Apply procedures
  • Validate collection properties

• Share with a broader community
  • And iterate
Communication Scale

- Institutional repositories - digital libraries
  - Intellectual holdings that advance local research and education initiatives
- Data grids
  - Shared collections within a project or between projects
- Preservation environments
  - Reference collections for use by future researchers
- Challenges
  - Hundreds of millions of files
  - Tens of petabytes of data
  - Distributed data sources, repositories, users
  - Policies unique to each collection
NOAO Zone Architecture
Carolina Digital Repository

**Architecture:**
- Web interface
- Fedora digital library middleware
- iRODS data grid

**Supports:**
- Registration of file into iRODS
- Generation of FOXML
- Registration into Fedora
- Query through Fedora
- Synchronization of catalogs

From Conceptualizing Policy-Driven Repository Interoperability (PoDRI) Using iRODS and Fedora (Pcolar, Davis, Zhu, Chassanoff, Hou, Marciano)
Extensible Environment, can federate with additional research and education sites. Each data grid can use different vendor products.

Policy to coalesce authentic records from independent data grids. Choose whether write to central archive, or use soft links.
Data Life Cycle

• What differentiates these data management applications?

• Can a collection migrate through the multiple stages of the data life cycle within the same generic framework?

• Can scholarly communication be extended to broader communities with minimal effort?
Preservation is a Stage in the Data Life Cycle

Each data life cycle stage re-purposes the original collection

Project Collection
Private

Data Grid
Shared

Data Processing Pipeline
Analyzed

Digital Library
Published

Reference Collection
Preserved

Distribution Policy
Service Policy

Description Policy
Re-preserved Policy

Data Processing Pipeline
Analyzed

Service Policy

Stages correspond to addition of new policies for a broader community

Virtualize the stages of the data life cycle through policy evolution

Interoperability across data life cycle representations
Scholarly Communication through Re-purposing of Collections

• Each stage represents a **consensus** by a user community on the **purpose** of the collection, the **policies** that will be used to control desired collection **properties**, the **procedures** that will enforce the **policies**, and the metadata or **state information** that is needed for a self-consistent system.

• **Virtualization** of the data life cycle corresponds to tracking the evolution of the **policies**, **procedures**, and **state information** and providing mechanisms for each new user community to re-purpose the collection.
Policy-based Data Environments

- **Purpose** - reason a collection is assembled
- **Properties** - attributes needed to ensure the purpose
- **Policies** - control for ensuring maintenance of properties
- **Procedures** - functions that implement the policies
- **State information** - results of applying the procedures
- **Assessment criteria** - validation that state information conforms to the desired purpose
- **Federation** - controlled sharing of logical name spaces

These are the necessary elements for a sustainable collection
Policy-based Preservation - Authenticity

- **Purpose** - Maintain authenticity of records
- **Properties** - Define template for required representation information
- **Policies** - Extract and register representation information for each file on ingestion
- **Procedures** - Parse record / XML file to extract metadata
- **State information** - Register representation information into metadata catalog
- **Assessment criteria** - Compare registered metadata with template defining required values

A preservation environment should automate each of these steps
User

Can Search, Access, Add and Manage Data & Metadata

iRODS Data System

iRODS Data Server
Disk, Tape, etc.

iRODS Rule Engine
Track policies

iRODS Metadata Catalog
Track information

*Access data with Web-based Browser or iRODS GUI or Command Line clients.
iRODS - Policy-based Data Management

- Turn policies into computer actionable rules
- Compose rules by chaining standard operations
  - Standard operations (micro-services) executed at the remote storage location
- Manage state information as attributes on namespaces:
  - Files / collections /users / resources / rules
- Validate assessment criteria
  - Queries on state information, parsing of audit trails
- Automate administrative functions
  - Minimize labor costs
Migration of Procedures

**Access Interface**

Map from actions requested by the access method to a standard set of Micro-services.

**Standard Micro-services**

Map the standard Micro-services to standard operations.

**Data Grid**

**Standard Operations**

Map the operations to protocol supported by the operating system.

**Storage Protocol**

**Storage System**
# Data Grid Clients

<table>
<thead>
<tr>
<th>API</th>
<th>Client</th>
<th>Developer</th>
<th>I/O Libraries</th>
<th>API</th>
<th>Client</th>
<th>Developer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browser</td>
<td>DCAPE</td>
<td>UNC</td>
<td></td>
<td>I/O Libraries</td>
<td>PHP - DICE</td>
<td>DICE-Bing Zhu</td>
</tr>
<tr>
<td></td>
<td>iExplore</td>
<td>DICE-Bing Zhu</td>
<td></td>
<td></td>
<td>C API</td>
<td>DICE-Mike Wan</td>
</tr>
<tr>
<td></td>
<td>JUX</td>
<td>IN2P3</td>
<td></td>
<td></td>
<td>C I/O library</td>
<td>DICE-Wayne Schroeder</td>
</tr>
<tr>
<td></td>
<td>Peta Web browser</td>
<td>PetaShare</td>
<td></td>
<td></td>
<td>Jargon</td>
<td>DICE-Mike Conway</td>
</tr>
<tr>
<td>Digital</td>
<td>Akubra/iRODS</td>
<td>DICE</td>
<td></td>
<td>Portal</td>
<td>Pyrods - Python</td>
<td>SHAMAN-Jerome Fusillier</td>
</tr>
<tr>
<td>Library</td>
<td>Dspace</td>
<td>MIT</td>
<td></td>
<td></td>
<td>EnginFrame</td>
<td>NICE / RENCI</td>
</tr>
<tr>
<td></td>
<td>Fedora on Fuse</td>
<td>IN2P3</td>
<td></td>
<td>Tools</td>
<td>Archive tools</td>
<td>NOAO</td>
</tr>
<tr>
<td></td>
<td>Fedora/iRODS module</td>
<td>DICE</td>
<td></td>
<td></td>
<td>Big Board</td>
<td>RENCI</td>
</tr>
<tr>
<td></td>
<td>Islandora</td>
<td>DICE</td>
<td></td>
<td></td>
<td>visualization</td>
<td></td>
</tr>
<tr>
<td>File System</td>
<td>Davis - Webdav</td>
<td>ARCS</td>
<td></td>
<td>File-format-identifier</td>
<td>GA Tech</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dropbox / iDrop</td>
<td>DICE-Mike Conway</td>
<td></td>
<td>icommands</td>
<td>DICE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FUSE</td>
<td>IN2P3, DICE</td>
<td></td>
<td>Pcommands</td>
<td>PetaShare</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FUSE optimization</td>
<td>PetaShare</td>
<td></td>
<td>Resource Monitoring</td>
<td>IN2P3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OpenDAP</td>
<td>ARCS</td>
<td></td>
<td>Sync-package</td>
<td>Academica Sinica</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PetaFS (Fuse)</td>
<td>Petashare - LSU</td>
<td></td>
<td>URSpace</td>
<td>Teldap - Academica Sinica</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Petashell (Parrot)</td>
<td>PetaShare</td>
<td></td>
<td>Web Service</td>
<td>VOSpace</td>
<td>NVOA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Shibboleth</td>
<td>King's College</td>
</tr>
<tr>
<td>Grid</td>
<td>GridFTP - Griffin</td>
<td>ARCS</td>
<td></td>
<td>Workflows</td>
<td>Kepler</td>
<td>DICE</td>
</tr>
<tr>
<td></td>
<td>Jsga</td>
<td>IN2P3</td>
<td></td>
<td></td>
<td>Stork</td>
<td>LSU</td>
</tr>
<tr>
<td></td>
<td>Parrot</td>
<td>Notre Dame-Doug Thain</td>
<td></td>
<td></td>
<td>Taverna</td>
<td>RENCI</td>
</tr>
<tr>
<td></td>
<td>Saga</td>
<td>KEK</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
iRODS Distributed Data Management
Format of an iRODS Rule

- Action | Condition | MS$_1$, ..., MS$_n$ | RMS$_1$, ..., RMS$_n$

- Action
  - Name of action to be performed
  - Name known to the server and invoked by server

- Condition – condition under which the rule applies

- Micro-services - Chain of micro-services to be executed

- Recovery micro-service - If any micro service fails, recovery micro-service(s) executed to maintain transactional consistency

- Example of MS/RMS
  - createFile(*F)  removeFile(*F)
  - ingestMetadata(*F,*M)  rollback
Demonstration

- Data grid in North Carolina at RENCI
- Icommands user interface (file manipulation)
- System state information
- Rule base controlling the data grid (policies)
- Composition of rules from micro-services
- Interactive execution of server-side workflows
Assessment Criteria

- NARA Electronic Records Archive capabilities list
  - 853 defined capabilities
  - Mapped to 174 computer actionable rules
  - Mapped to 212 state information attributes
- RLG/NARA Trusted Repository Audit Checklist
  - Mapped to 105 computer actionable rules
  - Included 66 rules specific to preservation
- ISO Mission Operations Information Management System repository audit checklist
  - 106 policies for operation and control
  - Mapped to 52 computer actionable rules
Examples of Assessment Criteria

• Specify
  • a template that governs the representation information required for a specific record series
  • content of a Submission Information Package (SIP)
  • content of an Archival Information Package (AIP)
  • number of replicas

• Verify
  • compliance of SIP with specification
  • compliance of AIP with specification
  • compliance with required replica number
  • integrity of the replicas
Funding

• **First generation Data Grid - Storage Resource Broker (SRB)**
  - DARPA Massive Data Analysis System (1996)
  - DARPA/USPTO Distributed Object Computation Testbed (1998)
  - NARA Persistent Archive (1999)
  - Application driven development (2000-2005)

• **Second generation Data Grid - iRODS**
  - NARA supplement to NSF SCI 0438741, “Cyberinfrastructure; From Vision to Reality” - “Transcontinental Persistent Archive Prototype” (TPAP) (2005)
  - NSF SDCI 0721400, "SDCI Data Improvement: Data Grids for Community Driven Applications” (2007)
Development Team

- DICE team
  - Arcot Rajasekar - iRODS Development Lead
  - Mike Wan - iRODS Chief Architect
  - Wayne Schroeder - iRODS Product Mgr., Developer
  - Bing Zhu - Fedora, Windows
  - Mike Conway - Java (Jargon)
  - Paul Tooby - Documentation, Foundation
  - Sheau-Yen Chen - Data Grid Administration
  - Reagan Moore - PI

- Preservation
  - Richard Marciano - Preservation Development Lead
  - Chien-Yi Hou - Preservation Micro-services
  - Antoine de Torcy - Preservation Micro-services
Foundation

• Data Intensive Cyber Environments Foundation
  • Nonprofit open source software development
  • Promotes use of iRODS technology
  • Supports standards efforts
  • Coordinates international development efforts
    • IN2P3 - quota and monitoring system
    • King’s College London - Shibboleth
    • Australian Research Collaboration Services - WebDAV
    • Academia Sinica - SRM interface
iRODS is a "coordinated NSF/OCI-Nat'l Archives research activity" under the auspices of the President's NITRD Program and is identified as among the priorities underlying the President's 2009 Budget Supplement in the area of Human and Computer Interaction Information Management technology research.

Reagan W. Moore
rwmoore@renci.org
http://irods.diceresearch.org

NSF OCI-0848296 “NARA Transcontinental Persistent Archives Prototype”
NSF SDCI-0721400 “Data Grids for Community Driven Applications”