Policy-Based Data Management

Reagan W. Moore
Arcot Rajasekar
University of North Carolina at Chapel Hill
rwmoore@email.unc.edu
Topics

• Policy-based data management
• Integrated Rule-Oriented Data System (v3.0)
• Federation
  – Soft Links
  – Realizable objects
  – Policy-encoded objects
  – Virtualization of collection life cycle
• DataNet Federation Consortium
• LifeTime Library
Policy-based Data Sharing

Consensus on Policies and Procedures controls the shared data
Policy-based Data Environments

- **Purpose** - reason a collection is assembled
- **Properties** - attributes needed to ensure the purpose
- **Policies** - controls for enforcing desired properties, mapped to computer actionable rules
- **Procedures** - functions that implement the policies, mapped to computer actionable workflows
- **Persistent state information** - results of applying the procedures, mapped to system metadata
- **Assessment criteria** - validation that state information conforms to the desired purpose, mapped to periodically executed policies
Data Virtualization

- **Access Interface**
- **Policy Enforcement Points**
- **Standard Micro-services**
- **Standard I/O Operations**
- **Storage Protocol**
- **Storage System**

Map from the actions requested by the client to multiple policy enforcement points.

Map from policy to standard micro-services.

Map from micro-services to standard Posix I/O operations.

Map standard I/O operations to the protocol supported by the storage system.
iRODS Distributed Data Management
Policy Enforcement Points

- Currently have 71 locations within iRODS framework where policies are checked.
  - Each action may involve multiple policy enforcements points

- Policy enforcement points
  - Pre-action policy (selection of storage location)
  - Policy execution (file deletion control)
  - Post-action policy (derived data products)
### Policy Enforcement Points (71)

<table>
<thead>
<tr>
<th>ACTION</th>
<th>PRE-ACTION POLICY</th>
<th>POST-ACTION POLICY</th>
</tr>
</thead>
<tbody>
<tr>
<td>acCreateUser</td>
<td>acPreProcForCreateUser</td>
<td>acPostProcForCreateUser</td>
</tr>
<tr>
<td>acDeleteUser</td>
<td>acPreProcForDeleteUser</td>
<td>acPostProcForDeleteUser</td>
</tr>
<tr>
<td>acGetUserbyDN</td>
<td>acPreProcForModifyUser</td>
<td>acPostProcForModifyUser</td>
</tr>
<tr>
<td>acTrashPolicy</td>
<td>acPreProcForModifyUserGroup</td>
<td>acPostProcForModifyUserGroup</td>
</tr>
<tr>
<td>acAclPolicy</td>
<td>acChkHostAccessControl</td>
<td>acPostProcForDelete</td>
</tr>
<tr>
<td>acSetCreateConditions</td>
<td>acPreProcForCollCreate</td>
<td>acPostProcForCollCreate</td>
</tr>
<tr>
<td>acDataDeletePolicy</td>
<td>acPreProcForRmColl</td>
<td>acPostProcForRmColl</td>
</tr>
<tr>
<td>acRenameLocalZone</td>
<td>acPreProcForModifyAVUMetadata</td>
<td>acPostProcForModifyAVUMetadata</td>
</tr>
<tr>
<td>acSetRescSchemeForCreate</td>
<td>acPreProcForModifyCollMeta</td>
<td>acPostProcForModifyCollMeta</td>
</tr>
<tr>
<td>acRescQuotaPolicy</td>
<td>acPreProcForModifyDataObjMeta</td>
<td>acPostProcForModifyDataObjMeta</td>
</tr>
<tr>
<td>acSetMultiReplPerResc</td>
<td>acPreProcForModifyAccessControl</td>
<td>acPostProcForModifyAccessControl</td>
</tr>
<tr>
<td>acSetNumThreads</td>
<td>acPreProcForDataObjOpen</td>
<td>acPostProcForOpen</td>
</tr>
<tr>
<td>acVacuum</td>
<td>acPreProcForObjRename</td>
<td>acPostProcForObjRename</td>
</tr>
<tr>
<td>acSetResourceList</td>
<td>acPreProcForCreateResource</td>
<td>acPostProcForCreateResource</td>
</tr>
<tr>
<td>acSetCopyNumber</td>
<td>acPreProcForDeleteResource</td>
<td>acPostProcForDeleteResource</td>
</tr>
<tr>
<td>acVerifyChecksum</td>
<td>acPreProcForModifyResource</td>
<td>acPostProcForModifyResource</td>
</tr>
<tr>
<td>acCreateUserZoneCollections</td>
<td>acPreProcForModifyResourceGroup</td>
<td>acPostProcForModifyResourceGroup</td>
</tr>
<tr>
<td>acDeleteUserZoneCollections</td>
<td>acPreProcForCreateToken</td>
<td>acPostProcForCreateToken</td>
</tr>
<tr>
<td>acPurgeFiles</td>
<td>acPreProcForDeleteToken</td>
<td>acPostProcForDeleteToken</td>
</tr>
<tr>
<td>acRegisterData</td>
<td>acNoChkFilePathPerm</td>
<td>acPostProcForFilePathReg</td>
</tr>
<tr>
<td>acGetIcatResults</td>
<td>acPreProcForGenQuery</td>
<td>acPostProcForGenQuery</td>
</tr>
<tr>
<td>acSetPublicUserPolicy</td>
<td>acSetReServerNumProc</td>
<td>acPostProcForPut</td>
</tr>
<tr>
<td>acCreateDefaultCollections</td>
<td>acSetVaultPathPolicy</td>
<td>acPostProcForCopy</td>
</tr>
<tr>
<td>acDeleteDefaultCollections</td>
<td></td>
<td>acPostProcForCreate</td>
</tr>
</tbody>
</table>
iput ../src/irm.c  checks 10 policy hooks

srbbbrick14:10900:ApplyRule#116:: acChkHostAccessControl
srbbbrick14:10900:GotRule#117:: acChkHostAccessControl
srbbbrick14:10900:ApplyRule#118:: acSetPublicUserPolicy
srbbbrick14:10900:GotRule#119:: acSetPublicUserPolicy
srbbbrick14:10900:ApplyRule#120:: acAclPolicy
srbbbrick14:10900:GotRule#121:: acAclPolicy
srbbbrick14:10900:ApplyRule#122:: acSetRescSchemeForCreate
srbbbrick14:10900:GotRule#123:: acSetRescSchemeForCreate
srbbbrick14:10900:execMicroSrvc#124:: msiSetDefaultResc(demoResc,null)
srbbbrick14:10900:ApplyRule#125:: acRescQuotaPolicy
srbbbrick14:10900:GotRule#126:: acRescQuotaPolicy
srbbbrick14:10900:execMicroSrvc#127:: msiSetRescQuotaPolicy(off)
srbbbrick14:10900:ApplyRule#128:: acSetVaultPathPolicy
srbbbrick14:10900:GotRule#129:: acSetVaultPathPolicy
srbbbrick14:10900:execMicroSrvc#130:: msiSetGraftPathScheme(no,1)
srbbbrick14:10900:ApplyRule#131:: acPreProcForModifyDataObjMeta
srbbbrick14:10900:GotRule#132:: acPreProcForModifyDataObjMeta
srbbbrick14:10900:ApplyRule#133:: acPostProcForModifyDataObjMeta
srbbbrick14:10900:GotRule#134:: acPostProcForModifyDataObjMeta
srbbbrick14:10900:ApplyRule#135:: acPostProcForCreate
srbbbrick14:10900:GotRule#136:: acPostProcForCreate
srbbbrick14:10900:ApplyRule#137:: acPostProcForPut
srbbbrick14:10900:GotRule#138:: acPostProcForPut
srbbbrick14:10900:GotRule#139:: acPostProcForPut
srbbbrick14:10900:GotRule#140:: acPostProcForPut
Overview of iRODS Architecture

User w/Client
Can Search, Access, Add and Manage Data & Metadata

iRODS Middleware

iRODS Data Server
Disk, Tape, etc.

iRODS Rule Engine
Track Policies

iRODS Metadata Catalog
Track information

Access distributed data with Web-based Browser or iRODS GUI or Command Line clients.
iRODS Extensible Infrastructure

- **Clients** – specific to discipline and life cycle state
- **Policies** – specific to discipline
- **Procedures** – specific to discipline
- **Remaining infrastructure is** *generic*
  - Network transport
  - Authentication / Authorization
  - Distributed storage access
  - Remote execution
  - Metadata management
  - Message passing
  - Rule engine
  - Parallel I/O
  - Single Sign-on
  - Protocol mediation
  - Deferred/periodic
  - Catalog
  - Debugging/progress
  - Policy control
Generic Capabilities

• Replication
• Registration of files into the data grid
• Synchronization of remote directory
• Managed file transport (iDrop)
• Automated metadata extraction
• Queries on metadata, tags
• Server-side workflows (loop over result sets)
• Parallel I/O streams & RBUDP transport
Policies

- Retention, disposition, distribution, arrangement
- Authenticity, provenance, description
- Integrity, replication, synchronization
- Deletion, trash cans, versioning
- Archiving, staging, caching
- Authentication, authorization, redaction
- Access, approval, IRB, audit trails, report generation
- Assessment criteria, validation
- Derived data product generation, format parsing
- Federation of independent data grids
Highly Controlled Environment

• All accesses are authenticated
  – GSI / Kerberos / Challenge-response / Shibboleth

• All operations are authorized
  – ACLs on files, storage
  – User groups, storage groups

• All policies evaluate a constraint
  – Constraints based on persistent state information and session information
iRODS Version 3.0

- September 30 production 3.0 release
- Features
  - New rule engine
    - Strong parameter typing
    - Optimized performance (thousands of rules)
    - Expanded rule programming language
  - Rule versioning
  - Distributed rule base management
  - Soft links to external resources
  - New transport management for large files
  - Improved Java interface
  - Windows native port (C++ compilation)
  - Synchronization interface (iDrop)
Applications

• Data grids – PB-size distributed collections
  – Astronomy – NOAO, CyberSKA, LSST
  – High Energy Physics – BaBar, KEK
  – Earth Systems – NASA (MODIS data set)
  – Australian Research Collaboration Service
    – Plant biology – iPlant Collaboratory

• Institutional repositories
  – Carolina Digital Repository

• Libraries
  – Texas Digital Libraries
  – Seismology - Southern California Earthquake Center

• Archives
  – Ocean Observatories Initiative
Expectations

• Data collection sizes will increase
  – Now petabytes, soon exabytes
    • 1 PB/year = 32 MB/sec
    • 1 PB/day = 11.6 GB/sec
  – Need to do data analyses at the storage system

• Integration of data manipulation with storage controllers
  – Analyses do not go over the network
Applications

• Digital libraries
  – Continuous indexing of contents

• Scientific data collections
  – Extraction of features from data sets
  – Creation of derived data products

• Archives
  – Transformative migrations
  – Validation of assessment criteria
Federation

• Federate multiple storage systems
  – Build data grid to create shared collection

• Federate multiple data grids
  – Establish trust between data grids
    • Authenticate access based on user home data grid
    • Authorize operation based on data home data grid
  – Cross-register users between data grid

• Federate multiple data management systems
  – Register remote data through a soft link
Federation through Shared Collection

Client

Institution
iRODS
controlled
workflows

Shared Collection

Storage

Institution
iRODS
controlled
workflows

Storage

Shared name spaces for collaborative environment
Shared Collection

• Shared name spaces
  – User name space (single sign-on)
  – Resource name space (logical names)
  – Collection name space (collection hierarchy)
  – File name space (logical names)

• Shared policies and procedures
  – Micro-service name space (versions)
  – Rule name space (versioned rules)
Data Grid Federation

Client

Data Grid
iRODS
controlled workflows

Data Grid
iRODS
controlled workflows

Shared Collection

Storage

Storage

Storage

Storage

Consensus on Policies and Procedures controls the shared data across federation
Soft Link

Steps
1) Register access information
2) When access, instantiate a copy in the data grid
3) If copy is deleted, go back to the original site
Soft Links (iRODS version 3.0)

- Use `ireg` command to register access to data within a remote repository
- Can access data in web pages, ftp, z39.50, srb data grid, irods data grid
- Requires definition of micro-service object:
  - `resc_type` mso
  - `data_type` mso
- Requires creation of mso resource and mso resource group for data caching
Soft Link (iRODS 3.0)

ireg -D mso -R msoResc14 -G msoRescGroup
"//http://irods.org/pubs/iRODS_Fact_Sheet-0907c.pdf"
/tempZone/home/rods/webfile

Remote object is registered as webfile in the directory /tempZone/home/rods

Can retrieve a copy of the web page

iget webfile
myTestRule {
# Input parameters are:
#  inRequestPath - the string sent to the remote URL
#  inFileMode - the cache file creation mode
#  inFileFlags - the access modes for the cache file
#  inCacheFileName - the full path of the cache file on the local system
# No output parameters
# Output is the name of the file that was created
  msiobjget_http(*Request, *Mode, *Flags, *Path);
  writeLine("stdout","Retrieved file *Path from *Request");
}

*Path = "/home/reagan/Vaulttest/webfile"

OUTPUT ruleExecOut
Realizable Objects

• Soft links are implemented through micro-service objects.
  – A micro-service is defined that manages the required access protocol

• Realized Objects
  – Register into iRODS the workflow needed to create a derived data product
  – Accessing the link causes the derived data to be generated and registered as a replica
  – Implemented using compound resource concept
  – Elevates provenance information to first class object
Active Objects
Provenance as a First Class Object

Object A
Object B
Object C
Object D

workflow depends on workflow depends on workflow depends on

Change object D

System automatically re-computes objects C, B, A when object A is accessed
Policy-Encoded Objects

• Micro-services exist for loading policies
• The workflow associated with a realizable object can load the policies under which the object may be used
  – Enforce redaction
  – Transform to required format
  – Verify distribution
  – Assert audit trails on use
  – Object specific policies
Virtualization of Collection Life Cycle

• Define policies that implement desired management objectives
• As user community broadens, develop new consensus on required management objectives
• Define the new policies needed to meet the new objectives
• Migrate collection onto new policies
• Community specific policies
Community-Based Collection Life Cycle

Each life cycle stage re-purposes the original collection

Project Collection
Private
Local Policy

Data Grid
Shared
Distribution Policy

Data Processing Pipeline
Analyzed
Service Policy

Digital Library
Published
Description Policy

Reference Collection
Preserved
Representation Policy

Federation
Sustained
Re-purposing Policy

Stages correspond to addition of new policies for a broader community
Virtualize the collection life cycle through policy evolution
LIfeTime Library

• Student personal digital library
  – Manage course material
  – Photograph collections
  – Video collections
  – Reference collection (soft links to information)

• Choose favorite access mechanism
  – iDrop (synchronize local directory, share data)
  – iDrop cloud browser (add tags, metadata)
  – Unix tools (execute personal rules)
LifeTime Library

Access Interface

Policy Enforcement Points

Standard Micro-services

Standard I/O Operations

Storage Protocol

Storage System

Multiple clients

Policies to automate replication turn on versioning set audit trails enforce strict ACLs replicate metadata

Micro-services to create thumbnails extract metadata assign organization
DataNet Federation Consortium

- Implement national data grid
  - Federate existing discipline-specific data management systems to enable national research collaborations
- Enable collaborative research on shared data collections
  - Manage collection life cycle as the user community broadens
- Integrate “live” research data into education initiatives
  - Enable student research participation through control policies

Cyber-infrastructure Partners:
Univ. of North Carolina, Chapel Hill
Univ. of California, San Diego
Arizona State University
Drexel University
Duke University
University of Arizona
University of South Carolina

Science and Engineering Initiatives:
Ocean Observatories Initiative
the iPlant Collaborative
CUAHSI
CIBER-U
Odum Social Science Institute
Temporal Dynamics of Learning Center

National Science Foundation Cooperative Agreement: OCI-0940841

Policy-based data management
Open Source Software

• Community driven software development
  – Focus on features required by user communities
  – Focus on bug-free software
  – Focus on highly reliable software
  – Focus on highly extensible software
  – Approximately 3-4 software releases per year

• Distributed under a BSD license
  – International collaborations on software development
  – IN2P3 (France), SHAMAN (UK), ARCS (Australia), Academia Sinica (Taiwan)
Capability Summary

- Clients: 50
- Policy enforcement points: 71
- Micro-services: 256
- Persistent state information: 209
- Posix I/O protocol operations: 22
- Types of data objects: 3
  - data, database, soft links
  - adding realizable & policy encoded objects
iRODS - Open Source Software

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

NSF OCI-0940841 “DataNet Federation Consortium”
NSF OCI-1032732 "SDCI Data Improvement: Improvement and Sustainability of iRODS Data Grid Software for Multi-Disciplinary Community Driven Application"
NSF OCI-0848296 “NARA Transcontinental Persistent Archives Prototype”
NSF SDCI-0721400 “Data Grids for Community Driven Applications”