Policy-Driven Distributed Data Management

Reagan W. Moore
University of North Carolina at Chapel Hill
rwmoore@renci.org
http://irods.diceresearch.org

NSF OCI-0848296 “NARA Transcontinental Persistent Archives Prototype” (2008-2012)
NSF SDCI 0721400 “Data Grids for Community Driven Applications” (2007-2010)
Policy-based Data Management

• Turn management policies into computer actionable rules
  – Dynamically evolve the rule base

• Turn management processes into remotely executable computer procedures
  – Can apply a workflow at the storage system to filter, subset, manipulate data
  – Minimize the amount of data pulled over the network
  – Automate administrative tasks

• Develop assessment criteria
  – Automate validation of collection properties
  – ISO MOIMS-rac
Overview of iRODS Data System

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 state

*Access data with Web-based Browser or iRODS GUI or Command Line clients or DSpace or Fedora or Kepler workflow or WebDAV or user level file system.
# Data Management Systems

**iRODS - integrated Rule-Oriented Data System**

<table>
<thead>
<tr>
<th>Data Management Environment</th>
<th>Conserved Properties</th>
<th>Control Mechanisms</th>
<th>Remote Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management Functions</td>
<td>Assessment Criteria</td>
<td>Management Policies</td>
<td>Management Procedures</td>
</tr>
<tr>
<td>Data Management Infrastructure</td>
<td>State Information</td>
<td>Rules</td>
<td>Micro-services</td>
</tr>
<tr>
<td>Physical Infrastructure</td>
<td>Database</td>
<td>Rule Engine</td>
<td>Storage System</td>
</tr>
</tbody>
</table>
Policies Control Processes

- Control execution of preservation processes based on conditions specified by:
  - Collection, user group, data type, storage system, time, any persistent state information attribute

- Administrative processes:
  - Retention, disposition, distribution, replication, deletion, registration, synchronization, checksum creation, addition of users, addition of resources, migration

- Ingestion / Access processes:
  - Metadata extraction, logical organization, derived data product generation, redaction, time-dependent access controls, IRB approval flags

- Validation processes:
  - Authenticity checks, integrity validation, chain of custody, repository trustworthiness, audit trails
Policy Enforcement Points

- User creation
- User deletion
- Collection creation
- Collection deletion
- Data object creation
- Data object open
- Data object ingestion
- Data object retrieval
- Data object deletion
- Data object registration
- Data object purge

- Zone renaming
- Resource specification
- Number of replicas on resource
- Number of I/O streams
- Trash management
- Public operations
- Host access control
- Physical path name
- Metadata catalog access
Persistent State Attributes

- ZONE_ID
- ZONE_NAME
- ZONE_TYPE
- ZONE_CONNECTION
- ZONE_COMMENT
- USER_ID
- USER_NAME
- USER_TYPE
- USER_ZONE
- USER_DN
- USER_INFO
- USER_COMMENT
- USER_CREATE_TIME
- USER_MODIFY_TIME
- RESC_ID
- RESC_NAME
- RESC_ZONE_NAME
- RESC_TYPE_NAME
- RESC_CLASS_NAME
- RESC_LOC
- RESC_VAULT_PATH
- RESC_FREE_SPACE
- RESC_INFO
- RESC_COMMENT
- RESC_CREATE_TIME
- RESC_MODIFY_TIME
- RESC_GROUP_RESC_ID
- RESC_GROUP_NAME
- DATA_ID
- DATA_COLL_ID
- DATA_NAME
- DATA_REPL_NUM
- DATA_VERSION
- DATA_TYPE_NAME
- DATA_SIZE
- DATA_RESC_GROUP_NAME
- DATA_RESC_NAME
- DATA_PATH
- DATA_OWNER_NAME
- DATA_OWNER_ZONE
- DATA_CHECKSUM
- DATA_EXPIRY
- DATA_MAP_ID
- DATA_COMMENTS
- DATA_CREATE_TIME
- DATA_MODIFY_TIME
- DATA_ACCESS_TYPE
- DATA_ACCESS_NAME
- DATA_TOKEN_NAMESPACE
- DATA_ACCESS_USER_ID
- DATA_ACCESS_DATA_ID
- DATA_REPL_STATUS
- DATA_STATUS
- COLL_ID
- COLL_NAME
- COLL_PARENT_NAME
- COLL_OWNER_NAME
- COLL_OWNER_ZONE
- COLL_MAP_ID
- COLL_INHERITANCE
- COLL_COMMENTS
- COLL_CREATE_TIME
- META_DATA_ATTR_NAME
- META_DATA_ATTR_VALUE
- META_DATA_ATTR_UNITS
- META_COLL_ATTR_NAME
- META_COLL_ATTR_VALUE
- META_COLL_ATTR_UNITS
- META_NAMESPACE_COLL
- META_NAMESPACE_DATA
- META_NAMESPACE_RESC
- META_RESC_ATTR_NAME
- META_RESC_ATTR_VALUE
- META_RESC_ATTR_UNITS
- META_USER_ATTR_NAME
- META_USER_ATTR_VALUE
- META_USER_ATTR_UNITS
- META_USER_ATTR_ID
- USER_GROUP_ID
- USER_GROUP_NAME
- RULE_EXEC_ID
- RULE_EXEC_NAME
- RULE_EXEC_REI_FILE_PATH
- RULE_EXEC_USER_NAME
- RULE_EXEC_ADDRESS
- RULE_EXEC_TIME
- RULE_EXEC_FREQUENCY
- RULE_EXEC_PRIORITY
- RULE_EXEC_ESTIMATED_EXE_TIME
- RULE_EXEC_NOTIFICATION_ADDR
- RULE_EXEC_LAST_EXE_TIME
- RULE_EXEC_STATUS
- TOKEN_NAMESPACE
- TOKEN_ID
- TOKEN_NAME
- TOKEN_VALUE
- TOKEN_VALUE2
- TOKEN_COMMENT
- AUDIT_OBJ_ID
- AUDIT_USER_ID
- AUDIT_ACTION_ID
- AUDIT_COMMENT
- AUDIT_CREATE_TIME
- AUDIT_MODIFY_TIME
- AUDIT_ACTION_ID
- AUDIT_COMMENT
- AUDIT_CREATE_TIME
- AUDIT_MODIFY_TIME
Generic Data Management Steps

• Logical Arrangement
  – Organization of material into collections

• Metadata
  – Descriptive / provenance / context

• Processes for manipulating the data
  – Calibration / coordinate projection / physical data

• Policies for managing the data
  – Administrative / Access / Redaction / Validation

• Access mechanisms
  – Web / workflow / digital library

• Workflows for data analysis
  – Server side remote procedures / client side
Extensible Environment, can federate with additional research and education sites. Each data grid uses different vendor products. Each data grid manages selected NARA digital holdings (5 TBs, 6.4 million files)
Federation

- Federation of data grids controlled by federation policies
  - Local policies always enforced
  - Requests are forwarded to the remote data grid for execution
  - Multiple types of federation
    - Master-slave data grids
    - Central archive data grids
    - Chained data grids
    - Peer-to-peer data grids
    - Deep archives
Example Deep Archive

• Policies that can be enforced:
  – No external writes allowed. All data transfer initiated by pull from the Deep Archive into a staging area
  – No record deletion allowed.
  – All updates create a new version of a record.
  – Periodically validate authenticity, integrity
  – Parse audit trails to show that policies have remained consistent over time.
  – Compare holdings with submission agreements
Using a Data Grid - Interoperability

- User asks for data
- Data request goes to iRODS Server #1
- 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
Architecture

• Highly extensible, modular architecture
  – Generic infrastructure
  – Open source software
  – Peer-to-peer servers interact to form a data grid
  – Support data sharing, data publication, data preservation, data processing pipelines, real-time sensor networks

• Layered architecture
  – Clients
  – Rules
  – Micro-services
  – Storage drivers
  – Structured information resource drivers
Data Virtualization

- Access Interface
- Standard Micro-services
- Data Grid
- Standard Operations
- Storage Protocol
- Storage System

Map from the actions requested by the access method to a standard set of micro-services. The standard micro-services are mapped to the operations supported by the storage system.
iRODS - Integrated Rule Oriented Data System

1. Generic infrastructure
2. Shared collection assembled from data distributed across remote storage locations
3. Server-side workflow environment in which procedures are executed at remote storage locations
4. Policy enforcement engine, with computer actionable rules applied at the remote storage locations
5. Validation environment for assessment criteria
6. Consensus building system for establishing a collaboration (policies, data formats, semantics, shared collection)
Applications

• Archives
  – NARA Transcontinental Persistent Archive Prototype
  – NASA Center for Computational Sciences archive
  – Carolina Digital Repository - institutional repository

• Data Grids
  – NASA Jet Propulsion Laboratory Planetary Data System data grid
  – NSF Temporal Dynamics of Learning center - cognitive science shared collection data grid
  – Australian Research Collaboration Services data grid
  – KEK high energy physics data grid
  – Cinegrid - management of distributed film repositories

• Digital Libraries
  – French National Library - Fedora/iRODS infrastructure

• Sensor data
  – NSF Ocean Observatories Initiative - manage real-time sensor data
Summary

• Preservation is a social process, requiring a consensus between the expectations of the submitter and capabilities of the archivist.
• Consensus is expressed as assertions on properties of the archived records.
• Assertions -> assessment criteria -> policies -> procedures -> state information -> validation queries
• A preservation environment needs to validate and enforce the consensus.
Additional Slides

- iRODS technology
- Use cases
Building a Shared Collection

Have collaborators at multiple sites, each with different administration policies, different types of storage systems, different naming conventions.

Assemble a self-consistent, persistent distributed shared collection to support a specific purpose.
The iRODS Data Grid installs in a “layer” over existing or new data, letting you view, manage, and share part or all of diverse data in a unified Collection.
User Interfaces

- C library calls - Application level
- Unix shell commands - Scripting languages
- Java I/O class library (JARGON) - Web services
- SAGA - Grid API
- Web browser (Java-python) - Web interface
- Windows browser - Windows interface
- WebDAV - iPhone interface
- Fedora digital library middleware - Digital library middleware
- Dspace digital library - Digital library services
- Parrot - Unification interface
- Kepler workflow - Grid workflow
- Fuse user-level file system - Unix file system
integrated Rule-Oriented Data System

Client Interface

Admin Interface

Rule Invoker

Rule Base

Rule

Engine

Current State

Resources

Resource-based Services

Metadata-based Services

Micro Service Modules

Micro Service Modules

Rule Modifier Module

Config Modifier Module

Metadata Modifier Module

Service Manager

Consistency Check Module

Consistency Check Module

Consistency Check Module

Consistency Check Module

Metadata Persistent Repository

Resource-based Services

Metadata-based Services
Applications

- Data grids - sharing data
- Digital libraries - publishing data
- Persistent archives - preserving data
- Processing pipelines - analyzing data
- Real-time data management - federation
- Integrated workflows - server and client side

- Switch applications by switching management policies
  - Building reference policy sets for each type of application
Scale

- Tens of millions to hundreds of millions of files
- Hundreds of terabytes to petabytes of data
- Hundreds of metadata attributes
- Hundreds of collaborators
- Tens to hundreds of policies
- Distributed internationally
- Federations of tens of data grids
- Thousands to tens of thousands of users
<table>
<thead>
<tr>
<th></th>
<th>As of 12/11/2006</th>
<th></th>
<th>As of 2/25/2008</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Data_size</strong></td>
<td><strong>Count (files)</strong></td>
<td><strong>Curators</strong></td>
<td><strong>Data_size</strong></td>
</tr>
<tr>
<td></td>
<td>(in GB)</td>
<td></td>
<td></td>
<td>(in GB)</td>
</tr>
<tr>
<td><strong>Data Grid</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NSF/NVO</td>
<td>110,615.00</td>
<td>16,381,466</td>
<td>100</td>
<td>88,216.00</td>
</tr>
<tr>
<td>NSF/NPACI</td>
<td>35,909.00</td>
<td>7,458,960</td>
<td>380</td>
<td>43,684.00</td>
</tr>
<tr>
<td>PZONE</td>
<td>24,755.00</td>
<td>14,208,012</td>
<td>68</td>
<td>29,851.00</td>
</tr>
<tr>
<td>NSF/LDAS-SALK</td>
<td>163,706.00</td>
<td>176,897</td>
<td>67</td>
<td>211,542.00</td>
</tr>
<tr>
<td>NSF/SLAC-JCSG</td>
<td>18,494.00</td>
<td>1,945,302</td>
<td>55</td>
<td>26,100.00</td>
</tr>
<tr>
<td>NSF/TeraGrid</td>
<td>269,332.00</td>
<td>7,300,999</td>
<td>3,267</td>
<td>286,390.00</td>
</tr>
<tr>
<td>NCAR</td>
<td>2.00</td>
<td>8</td>
<td>2</td>
<td>76,255.00</td>
</tr>
<tr>
<td>LCA</td>
<td>1,834.00</td>
<td>39,611</td>
<td>2</td>
<td>4,544.00</td>
</tr>
<tr>
<td>NIH/BIRN</td>
<td>18,921.00</td>
<td>18,499,588</td>
<td>385</td>
<td>20,400.00</td>
</tr>
<tr>
<td>Others</td>
<td>8,013.00</td>
<td>161</td>
<td>227</td>
<td>8,013.00</td>
</tr>
<tr>
<td><strong>Digital Library</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NSF/LTER</td>
<td>257.00</td>
<td>41,152</td>
<td>36</td>
<td>260.00</td>
</tr>
<tr>
<td>NSF/Portal</td>
<td>2,620.00</td>
<td>53,048</td>
<td>460</td>
<td>2,620.00</td>
</tr>
<tr>
<td>NIH/AFCS</td>
<td>733.00</td>
<td>94,686</td>
<td>21</td>
<td>733.00</td>
</tr>
<tr>
<td>NSF/SIO Explorer</td>
<td>2,681.00</td>
<td>1,201,719</td>
<td>27</td>
<td>3,053.00</td>
</tr>
<tr>
<td>NSF/SCEC</td>
<td>168,931.00</td>
<td>3,545,070</td>
<td>73</td>
<td>168,933.00</td>
</tr>
<tr>
<td>LLNL</td>
<td>8,176.00</td>
<td>335,540</td>
<td>5</td>
<td>18,934.00</td>
</tr>
<tr>
<td>CHRON</td>
<td>932.00</td>
<td>830,354</td>
<td>5</td>
<td>13,278.00</td>
</tr>
<tr>
<td><strong>Persistent Archive</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NARA</td>
<td>4,713.00</td>
<td>5,992,817</td>
<td>58</td>
<td>5,036.00</td>
</tr>
<tr>
<td>NSF/NSDL</td>
<td>5,699.00</td>
<td>50,446,490</td>
<td>136</td>
<td>8,618.00</td>
</tr>
<tr>
<td>UCSD Libraries</td>
<td>5,080.00</td>
<td>1,077,202</td>
<td>29</td>
<td>5,210.00</td>
</tr>
<tr>
<td>NHPRC/PAT</td>
<td>3,756.00</td>
<td>527,695</td>
<td>28</td>
<td>2,575.00</td>
</tr>
<tr>
<td>RoadNet</td>
<td>2,057.00</td>
<td>712,534</td>
<td>30</td>
<td>3,886.00</td>
</tr>
<tr>
<td>UCTV</td>
<td>7,111.00</td>
<td>2,045</td>
<td>5</td>
<td>7,140.00</td>
</tr>
<tr>
<td>LOC</td>
<td>9,921.00</td>
<td>252,046</td>
<td>8</td>
<td>6,644.00</td>
</tr>
<tr>
<td>EarthSci</td>
<td>3,306.00</td>
<td>499,137</td>
<td>5</td>
<td>6,317.00</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>877 TB</td>
<td>131 million</td>
<td>5479</td>
<td>1.04 PB</td>
</tr>
</tbody>
</table>
Use Cases (1)

• RENCI Data Grid (per Ray Idaszak)
  – Build collaboration space for visualization
  – Use existing RENCI Virtual Organization nodes (i.e. videoconferencing nodes, aka Dell PC w. Windows)
  – Add 1.5TB Seagate drives where needed
  – Repurpose Dell PCs

• Demonstrate data grid that manages a collection of visualization data files distributed across:
  – All RENCI locations

• Use iRODS policies to automate replication of data to Europa Center Data Direct cache
RENCI Data Grid

Staff
Generate new visualization data

User
Access and display content

iRODS Data System

RENCI @ UNC Asheville
RENCI @ UNC Charlotte
RENCI @ Duke University
RENCI @ NC State University
RENCI @ ECU
RENCI @ UNC Chapel Hill
RENCI @ UNC Health Sciences Library
RENCI @ UNC Asheville
RENCI @ UNC Charlotte
RENCI @ Duke University
RENCI @ NC State University
RENCI @ ECU
RENCI @ UNC Chapel Hill
RENCI @ UNC Health Sciences Library
RENCI @ Europa Center

iRODS Metadata Catalog
Use Cases (2)

• NARA Transcontinental Persistent Archive Prototype
  – Federate 7 independent iRODS data grid: Each data grid manages its own resources and metadata catalog, applies its own policies
  – Use iRODS federation mechanism to establish the policies under which data can be shared between the data grids.
  – Control operations that a remote user is allowed to do within your data grid
Overview of iRODS Architecture

Preserving Electronic Records with iRODS

Archivists can use iRODS for preserving Electronic Records, from Appraisal to Access, with Rules enforcing trustworthy repository criteria with audits.
Challenges

- Building a consensus on management policies for the shared collection
- Translating service level agreements for shared use of resources into computer actionable rules
- Translating assessment criteria into computer executable procedures
- Defining federation policies for sharing data between data grids / institutions
iRODS Micro-services and Rules – Part 2
Rule Flow

Application Client Call

Find Appropriate Rules

Select First/Next Rule

Condition Check

Execute Recovery MicroService/Action

Server Call

Failure: No More Rules

Execute Next MicroService/Action

Success

Success: No More MS/A
How the Rule Engine Works:

A: C1 | M1 M2 | R1 R2
A: C2 | M3 M4 | R3 R4
A: C3 | M5 M6 M7 | R5 R6 R7
A: C4 | M8 M9 | R8 R9

Execute A
Check C1 (success)
 Execute M1 (success)
 Execute M2 (fail)
 Execute R2
 Execute R1 /*R1 is also executed!*/
Check C2 (fail)
Check C3 (success)
Execute M5 (success)
Execute M6 (success)
Execute M7 (success)
A succeeds

/* C4 is not even checked */
Some Sample Rules

- rule for querying the iCAT
- acCreateUser (default policy in core.irb)
- acDataDeletePolicy (not a default – can be turned on at admin’s discretion)
Easy way of Querying iCAT: iQuest (1)

- iCommand utility for querying the iCAT
- It is in pseudo-SQL format
  - SQL is a query language for databases
  - Stands for structured query language
- You view the whole iCAT as one large table
  (iCAT has more than 20 tables)
  - You give conditions for picking rows from the “universal” table
  - You give a list of column names to pick values in the rows
  - SELECT DATA_NAME
    WHERE  DATA_NAME like ’%.txt’
    AND COLL_NAME = ’/myzone/home/me’
Easy way of Querying iCAT: iQuest (2)

- The iquest command:
  
  iquest [format] selectQuery

- Samples:

  iquest "SELECT DATA_NAME WHERE DATA_NAME like '%.txt'"
  
  iquest "File %s has %-2.2s copies"
      "SELECT DATA_NAME, DATA_REPL_NUM"

- Complicated Example:

  iquest "User %-9.9s uses %14.14s bytes in %8.8s files in '%s'
      "SELECT USER_NAME, sum(DATA_SIZE),
            count(DATA_NAME),  RESC_NAME"

  User sekar has 25342 bytes in 342 files in demoResc
  User sekar has 34529 bytes in 412 files in tapeResc
How to query in iRule (1)

- Two Micro-services:
  - msiMakeQuery(*colList, *cond, *queryStr)
    - Takes a list of columns and a condition string and creates a pseudo-SQL query-string
    - Alas! Does not do formats; but don’t despair!!
  - msiExecStrCondQuery(*queryStr, *genQOut)
    - Takes the query-string executes it in iCAT and returns the answer-table in an internal structure

- Sample-rule: Given a condition get the answer-table

  acExecMyQuery(*C,*T) ||
  msiMakeQuery("DATA_NAME,COLL_NAME",*C,*S)##
  msiExecStrCondQuery(*S,*T) | nop

- But *Q is an internal structure and not printable!!
How to query in iRule (2)

• So, to print,
  – we need to take the values out of the structure
    • msiGetValByKey(*Row, *ColName, *Value)
    • Given a row of the table, and a column name, it returns the value of that column.
  – How do we print a value?
    • writeLine(*where, *what)
    • writeLine (stdout, ”Hello World!”)
  – How to get a row from the table (of rows)
    • Use the forEachExec system micro-service
      
forEachExec(*T, msiGetValByKey(*T, DATA_NAME, *Value)##
      
writeLine(File Name is *Value) , nop )
How to query in iRule (3)

Finally, we can put all together:

```plaintext
myRule(*Cond)
{
  msiMakeQuery("DATA_NAME,COLL_NAME",*Cond,*S);
  msiExecStrCondQuery(*S,*T);
  forEachExec(*T) /* for each row in answer table T */
  {
    msiGetValByKey(*T, DATA_NAME, *DV);
    msiGetValByKey(*T, COLL_NAME, *CV):
    writeLine(File *DV is in Collection *CV)
  }
}
```
acCreateUser

• Used by iRODS when an administrator creates a new user.

• Flexibility to add “new” features when creating users
  – Create a trash bin
  – Add user to groups based on her domain
  – Verify the user in a list or external database or with some community authentication system
  – Allocate storage and quotas
  – Notify someone about this new user (may be the domain manager)
  – Send the new user some emails about how to use irods
acCreateUser – by default

acCreateUser {
  ON ($otherUserName == anonymous) {
    msiCreateUser ::: msiRollback;
    msiCommit;
  }
  OR {
    msiCreateUser ::: msiRollback;
    acCreateDefaultCollections ::: msiRollback;
    msiAddUserToGroup(public) ::: msiRollback;
    msiCommit
  }
}
acCreateDefaultCollections

acCreateDefaultCollections
{
    acCreateUserZoneCollections
}
acCreateUserZoneCollections
{
    
    msiCreateCollByAdmin(/$rodsZoneProxy/home,
                        $otherUserName );
    msiCreateCollByAdmin(/$rodsZoneProxy/trash/home,
                        $otherUserName );

}

• Creates two collections a ‘home’ and a ‘trash’
acDataDeletePolicy

• Can be used to disallow deleting files from a collection

acDataDeletePolicy
{
  ON ($objPath like /myzone/home/sekar/*)
  {
    msiDeleteDisallowed; /*sets a disallow flag */
  }
  OR
  {
    nop;
  }
}
From Policies to Rules

• Write the policy with clear “keywords” that define side-effects that can be performed by micro-services.
• Identify recovery mechanisms for failure
• Create high-level signatures for the micro-services – split complicated micro-services
• Form a workflow based on the micro-services and test various paths
• Search existing rules/micro-services which can be used.
• Code micro-services, if needed, and unit test
• Write and test the rules
Development Efforts

• DICE team
  – Arcot Rajasekar - iRODS development lead
  – Mike Wan - iRODS chief architect
  – Wayne Schroeder - iRODS developer
  – Bing Zhu - Fedora, Windows
  – Lucas Gilbert - Java (Jargon), DSpace
  – Paul Tooby - documentation, foundation
  – Sheau-Yen Chen - data grid administration

• Preservation
  – Richard Marciano - Preservation development lead
  – Chien-Yi Hou - preservation micro-services
  – Antoine de Torcy - preservation micro-services
Features in Next Release

- Support for mySQL as the iCAT metadata catalog
- Support for Kerberos authentication
- Support for resource monitoring system
- Multi-tasking the batch server (irodsReServer) for more robust job execution.
- A new resource class - Compound Resource for a class of resources that support only put/get type functions
  - (e.g., ftp, HPSS parallel I/O, etc)
- Better support for writing micro-services - consolidation of data structures used by micro-services, more helper routines.
- Better Jargon support for iRODS - parallel I/O, metadata support, etc.
- Multi-threading put/get of small files (if it can be done in time for the release)
- Better support for restricted listing of collections (ACLs).
Foundation

• Data Intensive Cyber-environments
  – Non-profit open source
  – Promote use of iRODS technology
  – Coordinate with standards efforts
  – Coordinate international development efforts
    • IN2P3 - quota and monitoring system
    • King’s College London - Shibboleth
    • Australian Research Collaboration Services - WebDAV
    • Academia Sinica - SRM interface
Prioritize Development

• Generic infrastructure
  – Turn specific requests into generic framework

• Assign importance
  – Bug fixes
  – Funded development
  – Multiple requests
  – Critical need to meet major demonstration

• Incorporate community supplied mods
  – Generic infrastructure
  – Compliance with iRODS modular design
More Information

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