iRODS
Policy-Driven Data Preservation

Integrating Cloud Storage and Institutional Repositories

Reagan W. Moore
Arcot Rajasekar
Mike Wan

{moore,sekar,mwan}@diceresearch.org
http://irods.diceresearch.org
Abstract

• The integrated Rule-Oriented Data System (iRODS) organizes distributed data into a sharable collection. The data may reside in cloud storage, in institutional repositories, in tape archives, in laptop file systems. We will demonstrate the enforcement of management policies across the multiple storage locations, access mechanisms ranging from web browsers to Fedora to Web-DAV to EnginFrame interfaces, and types of assertions that can be made on data in cloud storage.
Projects

• TPAP - Transcontinental Persistent Archive Prototype
  – Preservation research in collaboration with NARA
• DCAPE - Distributed Custodial Archival Preservation Environment
  – Collaboration with State Archives to build preservation policies
• PoDRI - Policy Driven Repository Interoperability
  – IMLS collaboration with Duraspace to integrate Fedora / iRODS
• OOI - Ocean Observatories Initiative
  – NSF collaboration to archive sensor data
• CDR - Carolina Digital Repository
  – UNC-CH collaboration to build an institutional repository
• TUCASI - Triangle Universities Center for Advanced Studies, Inc.
  – Collaboration to build cyberinfrastructure between UNC, Duke, NCSU, RENCI
Overview of iRODS Data System

User

Can Search, Access, Add and Manage Data & Metadata

iRODS Server

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.
Integrated Rule Oriented Data System

• Software to organize distributed data into a sharable collection, while enforcing management policies
  – Manage properties of the shared collection

• Developed by DICE Center
  – University of North Carolina at Chapel Hill
    • 6 staff, 4 students
  – University of California, San Diego
    • 5 staff

• Funded by
  
  *NSF OCI-0848296 “NARA Transcontinental Persistent Archives Prototype”
  *NSF SDCI-0721400 “Data Grids for Community Driven Applications”*
Preservation Concept

• Maintain properties of records while utilizing new storage systems
  – Records are permanent
  – Federate storage across cloud, archives, file systems

• Implications - infrastructure independence
  – Active management of records
  – Multiple indirection mechanisms to protect records from dependencies upon technology
  – Validation of state information and parsing of audit trails to verify assertions about records
Data Virtualization

Map from tasks requested by the access method to a standard set of micro-services.

The standard micro-services are mapped to standard operations.

Operations are mapped to the protocol of the storage system.

View clients and storage as ephemeral.

- **Access Interface**
- **Standard Micro-services**
- **Data Grid**
- **Standard Operations**
- **Storage Protocol**
- **Storage System**
Policy-based Record Management

- Express policies as computer actionable rules
  - Define explicit locations in data management framework where policies will be enforced

- Express procedures as remotely executable micro-services
  - Create new preservation services by linking micro-services into a workflow

- Manage state information to track the application of preservation procedures
  - Maintain audit trails to track evolution of policies and procedures
Input With Replication

**Client**

- /<filesystem>
- Rule Base

**Institutional Repository**

- Data
- Rule Base
- Metadata

**Cloud Storage**

- Cache
- Data
- Rule Base
- Metadata

**icat**

- Metadata

**Replication rule added to rule base**

**Logos:**
- D.I.C.E
- iRODS
- T.N.C.C.
- UCSD
- NSF
- National Archives
- NCCS
- The University of North Carolina at Chapel Hill
Policies Implemented Outside of the (Cloud) Storage

• Automation of preservation procedures
  – Descriptive metadata extraction
  – Creation of archival form (AIP)
  – Transformative migration

• Automation of administrative functions
  – Distribution, replication, retention, disposition
  – Report generation - usage, performance, error tracking

• Periodic validation of assessment criteria
  – Trustworthiness, integrity, authenticity, chain of custody
  – Parsing of audit trails for policy compliance over time
Management Framework

• Instrument data management infrastructure at all locations where policy should be enforced (65 hooks)
  – File create, open, read, write, delete
  – Collection create, delete
  – User create, modify, delete, group
  – Resource create, modify, delete, group
  – Metadata file modify, collection modify, descriptive
  – ACL modify

• Support pre-processing policy
  – Authorization, selection, redirection

• Support post-processing policy
  – Audit trails, redaction, derived product generation
iRODS Distributed Data Management
Types of Rules

• Synchronous rules applied by framework at management hook locations
  – Stored in rule base; core.irb file

• Asynchronous rule that are queued for deferred or periodic execution
  – Batch system to manage queue

• Interactively executed rules defined by a user
  – Executed through irule command
iRODS Rules

• Server-side workflows
  Action | condition | workflow chain | recovery chain

• Condition - test on any attribute:
  – Collection, file name, storage system, file type, user group,
    elapsed time, IRB approval flag, descriptive metadata

• Workflow chain:
  – Micro-services / rules that are executed at the storage system

• Recovery chain:
  – Micro-services / rules that are used to recover from errors
Checksum Validation Rule

myChecksumRule{
    msiMakeQuery("DATA_NAME, COLL_NAME, DATA_CHECKSUM", *Condition, *Query);
    msiExecStrCondQuery(*Query, *B);
    assign(*A, 0);
    forEachExec (*B) {
        msiGetValByKey(*B, COLL_NAME, *C);
        msiGetValByKey(*B, DATA_NAME, *D);
        msiGetValByKey(*B, DATA_CHECKSUM, *E);
        msiDataObjChksum(*B, *Operation, *F);
        ifExec (*E != *F) {
            writeLine(stdout, file *C/*D has registered checksum *E and computed checksum *F);
        }
        else {
            assign(*A, *A + 1);
        }
    }
    ifExec(*A > 0) {
        writeLine(stdout, have *A good files);
    }
}

*Condition can be COLL_NAME like ‘/ils161/home/moore/genealogy/%’
Highly Extensible

• To integrate cloud storage
  – Wrote an S3 driver to execute the cloud storage protocol
  – Implemented cloud storage as a compound resource
    • Added disk cache in front of the cloud storage to enable enforcement of policies across all cloud accesses
  – All iRODS clients can then be used to access data stored in the cloud
    • Web browser, WebDav, FUSE, Kepler & Taverna workflows, Unix shell commands, C libraries, EnginFrame portal
  – Can write policies to distribute data between cloud, file systems, archives, laptops.
  – Can federate other data grids with the data grid that uses the cloud storage
NARA Transcontinental Persistent Archive Prototype

• Use data grid technology to build a preservation environment

• Conduct research on preservation concepts
  – Infrastructure independence
  – Enforcement of preservation properties
  – Automation of administrative preservation processes
  – Validation of preservation assessment criteria

• Demonstrate preservation on selected NARA digital holdings
  – Integration of generic infrastructure with preservation technologies (Cheshire, MVD, JHOVE, Pronom, Fedora, Dspace)
Extensible Environment, can federate with additional research and education sites. Each data grid uses different vendor products.
iRODS is a "coordinated NSF/OCI-Nat'l Archives research activity" under the auspices of the President's NITRD Program

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”