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

UCSD

UNC

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.

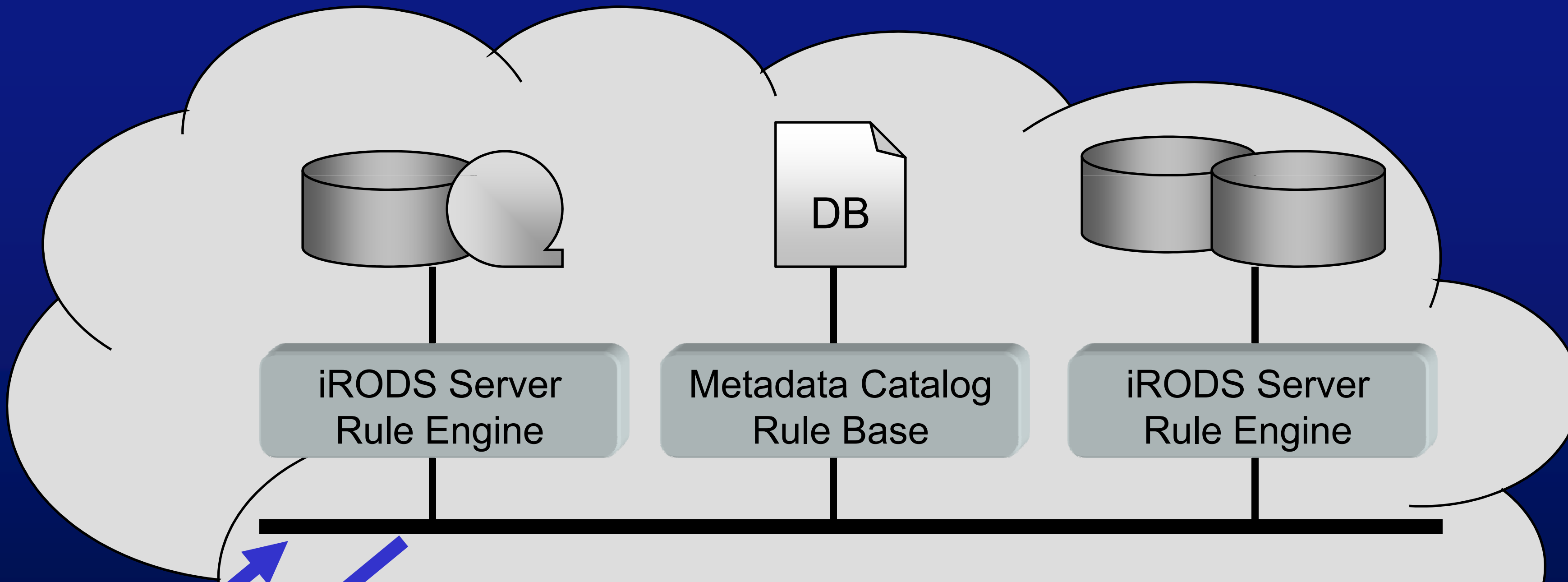
Data Management Applications
(What do they have in common?)

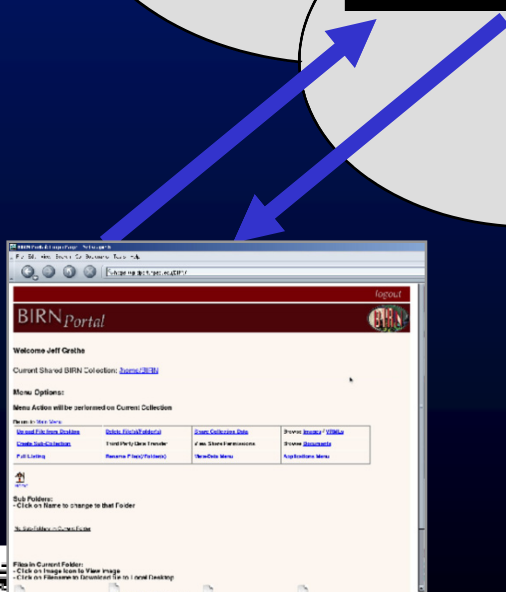
- **Data grids**
  - **Share data** - organize distributed data as a collection
- **Digital libraries**
  - **Publish data** - support browsing and discovery
- **Persistent archives**
  - **Preserve data** - manage technology evolution
- **Real-time sensor systems**
  - **Federate sensor data** - integrate 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

Observations of Production 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

Date	5/17/02		6/30/04			11/29/07		
Project	GBs of data stored	1000Õs of files	GBs of data stored	1000Õs of files	# Curators	GBs of data stored	1000Õs of files	# Curators
Data Grid								
NSF / NVO	17,800	5,139	51,380	8,690	80	88,216	14,550	100
NSF / NPACI	1,972	1,083	17,578	4,694	380	39,697	7,590	380
Hayden	6,800	41	7,201	113	178	8,013	161	227
Pzone	438	31	812	47	49	28,799	17,640	68
NSF / LDAS-SALK	239	1	4,562	16	66	207,018	169	67
NSF / SLAC-JCSG	514	77	4,317	563	47	23,854	2,493	55
NSF / TeraGrid			80,354	685	2,962	282,536	7,257	3,267
NIH / BIRN			5,416	3,366	148	20,400	40,747	445
NCAR						70,334	325	2
LCA						3,787	77	2
Digital Library								
NSF / LTER	158	3	233	6	35	260	42	36
NSF / Portal	33	5	1,745	48	384	2,620	53	460
NIH / AfCS	27	4	462	49	21	733	94	21
NSF / SIO Explorer	19	1	1,734	601	27	2,750	1,202	27
NSF / SCEC			15,246	1,737	52	168,931	3,545	73
LLNL						18,934	2,338	5
CHRON						12,863	6,443	5
Persistent Archive								
NARA	7	2	63	81	58	5,023	6,430	58
NSF / NSDL			2,785	20,054	119	7,499	84,984	136
UCSD Libraries			127	202	29	5,205	1,328	29
NHPRC / PAT						2,576	966	28
RoadNet						3,557	1,569	30
UCTV						7,140	2	5
LOC						6,644	192	8
Earth Sci						6,136	652	5
TOTAL	28 TB	6 mil	194 TB	40 mil	4,635	1,023 TB	200 mil	5,539

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)

UCSD

UNC