SRM-iRODS Interface Development

WeiLong UENG
Academia Sinica Grid Computing
wlueng@twgrid.org
What is iRODS

- Integrated Rule-Oriented Data-management System
- From SRB (Storage Resource Broker) to iRODS
- A community-driven, open source, data grid software solution
iRODS Architecture

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 data*
iRODS features

- High-performance network data transfer
- A unified view of disparate data
- Support for a wide range of physical storage
- Easy back up and replication
- Manages metadata
- Controlled access
- Policies, Rules and Micro-services
- Workflows
- Management of large collections
iRODS Applications

• Data grids
  • Project level data sharing - Share data

• Digital libraries
  • Specify data context, provide standard services - Publish data

• Persistent archive
  • Build reference collections - Preserve data

• Real-time sensor systems
  • Manage real-time data distribution - Federate data

• Workflow systems
  • Integrate client- & server-side workflows - Analyze data
Why SRM?

- Storage Elements (SE) can use different type of technologies
  - CASTOR, dCache, DPM, BeStMan, etc.
  - DRM (Disk Resource Manager)/TRM (Tape Resource Manager)/HRM (Hierarchical Resource Manager)
- Grid middleware needs to access files with an uniform interface
  - Manage storage resources
  - Not a file transfer protocol
What is SRM?
What is SRM?

• **Storage Resource Managers (SRMs) are middleware components**
  • whose function is to provide
    • dynamic space allocation
    • file management on shared storage resources on the Grid
  • Different implementations for underlying storage systems are based on the same SRM specification
SRM features

- Provides space management
- Provides an uniform access interface
- Manages DRM/Tape/HRM
- Does not transfer files itself.
- Manage the life time of file
SRMs role in grid

- SRMs role in the data grid architecture
  - Shared storage space allocation & reservation
    - important for data intensive applications
  - Get/put files from/into spaces
    - archived files on mass storage systems
  - File transfers from/to remote sites, file replication
  - Negotiate transfer protocols
  - File and space management with lifetime
  - support non-blocking (asynchronous) requests
  - Directory management
  - Interoperate with other SRMs
SRM: Main concepts

- Space reservations
- Dynamic space management
- Pinning file in spaces
- Support abstract concept of a file name: Site URL
- Temporary assignment of file names for transfer: Transfer URL
- Directory management and authorization
- Transfer protocol negotiation
- Support for peer to peer request
- Support for asynchronous multi-file requests
- Support abort, suspend, and resume operations
- Non-interference with local policies
Site URL and Transfer URL

- **Provide: Site URL (SURL)**
  - URL known externally – e.g. in Replica Catalogs
  - e.g. `srm://fct01.grid.sinica.edu.tw:8443/axis/services/srm?/AS/home/wlueng.ASGC/testFile1.dat`

- **Get back: Transfer URL (TURL)**
  - Path can be different from SURL – SRM internal mapping
  - Protocol chosen by SRM based on request protocol preference
  - e.g. `gsiftp://t-ap20.grid.sinica.edu.tw:2811/AS/home/wlueng.ASGC/testFile1.dat`

- **One SURL can have many TURLs**
  - Files can be replicated in multiple storage components
  - Files may be in near-line and/or on-line storage
  - In a light-weight SRM (a single file system on disk)
    - SURL may be the same as TURL except protocol

- **File sharing is possible**
  - Same physical file, but many requests
  - Needs to be managed by SRM implementation
Transfer protocol negotiation

- **Negotiation**
  - Client provides an ordered list of preferred transfer protocols
  - SRM returns first protocol from the list it supports
  - Example
    - Client provided protocols list: bbftp, gridftp, ftp
    - SRM returns: gridftp

- **Advantages**
  - Easy to introduce new protocols
  - User controls which transfer protocol to use

- **How it is returned?**
  - The protocol of the Transfer URL (TURL)
SRM v2.2 Interface

- **Data transfer functions** to get files into SRM spaces from the client's local system or from other remote storage systems, and to retrieve them
  - `srmPrepareToGet`, `srmPrepareToPut`, `srmBringOnline`, `srmCopy`

- **Space management functions** to reserve, release, and manage spaces, their types and lifetimes.
  - `srmReserveSpace`, `srmReleaseSpace`, `srmUpdateSpace`, `srmGetSpaceTokens`

- **Lifetime management functions** to manage lifetimes of space and files.
  - `srmReleaseFiles`, `srmPutDone`, `srmExtendFileLifeTime`

- **Directory management functions** to create/remove directories, rename files, remove files and retrieve file information.
  - `srmMkdir`, `srmRmdir`, `srmMv`, `srmRm`, `srmLs`

- **Request management functions** to query status of requests and manage requests
  - `srmStatusOf{Get,Put,Copy,BringOnline}Request`, `srmGetRequestSummary`, `srmGetRequestTokens`, `srmAbortRequest`, `srmAbortFiles`, `srmSuspendRequest`, `srmResumeRequest`

- **Other functions include** Discovery and Permission functions
  - `srmPing`, `srmGetTransferProtocols`, `srmCheckPermission`, `srmSetPermission`, etc.
Interoperability in SRM v2.2

Client
User/application

dCache
Fermilab
CASTOR

DPM
BNL
SLAC
LBNL

xrootd

SRB (iRODS)

Disk

BeStMan

SRM

HPSS

STAR

INFN

GRID

VANDERBILT UNIVERSITY

STORE

BNL
SLAC
LBNL

EGEE

SDSC
SINICA
LBNL

CCLRC

Science

CERN

Fermilab

Star

BNL

SLAC

LBNL
When iRODS met SRM

- Make iRODS an archival system of gLite-based e-Infrastructure.
- Support flexible lifetime policy for files
- Impose the VO-based resource policy and security control to iRODS as the Grid infrastructure.
SRM-iRODS implementations
SRM-iRODS Architecture
Use Case: putting files

1. Create a new LFN entry in LFC, return a SURL.
2. srmPrepareToPut (SURL)
3. Transfer the file to iRODs use GridFTP
4. srmPutDone (SURL)
Use Case: getting files

1. Query the file catalog to retrieve the SURL from the LFN.
2. srmPrepareToGet (SURL)
3. Transfer the file (read)
4. srmReleaseFile (SURL)
Information in Auxiliary File Catalog

- AMGA server, it stores partial filecatalog, resource and iRODS host information...
  - Users Information
  - Resources Information
  - Files Information
  - Space Metadata
  - Resource States
  - ...

Enabling Grids for E-sciencE
Architecture Overview

Put a file (SURL) into the Core.

URL -> Path

Default space is available? If yes, some files need to be uploaded.

Auxiliary filecatalog (AMGA)

Data server management

iCAT Server (GSI enabled)

Non iCAT (+DSI)

Non MES+DSI

SRB storage space
Architecture Overview (cont.)

Return TURL

Web Service

Data server management

Core

Auxiliary Filecatalog (AMGA)

Yes, default space is ready and this file has been set to pinned.

Non MES+DSI

Non iCAT (+DSI)

Non iCAT (+DSI)

SRB storage space

Non MES+DSI

iCAT Server (GSI enabled)
Architecture Overview (cont.)

Web Service

Core

Auxiliary Filecatalog (AMGA)

Data server management

iCAT Server (GSI enabled)

Non iCAT (+DSI)

Non iCAT (+DSI)

Non MES+DSI

SRB storage space

Upload a file (gridftp)
Architecture Overview (cont.)

Web Service

Core

Auxiliary Filecatalog (AMGA)

Data server management

iCAT Server (GSI enabled)

Non iCAT (+DSI)

Non MES+DSI

SRB storage space

put file done

SURL -> Path

Set this file has been uploaded and unpinned
The Status of PutDone
Return the Status of PutDone

Web Service
Core

Data server management

Auxiliary filecatalog (AMGA)

iCAT Server (GSI enabled)

Non iCAT (+DSI)
Non iCAT (+DSI)
Non MES+DSI
SRB storage space
Support Flexible File/Space Types

- SRM system has a caching mechanism and has to take care of SRM issues like file lifetime, space management, etc.
  - Permanent space
  - Volatile space
  - Durable space
- Implementation
  - Use AMGA as auxiliary catalog and record all space usage, space type, and some file metadata inside.
Checking Disk Status

Web Service

Core

Update status of each resource

• Auxiliary Filecatalog
  • (AMGA)

Resource info

Data server management

Logical Resource info

iRODSInfoServer

Resource info

iCAT Server (GSI enabled)

Resource info

Storage space

Resource info

Storage space
Checking Disk Status

• How to get the disk usage of the space?
  • Need to know the free and used space on iRODS server
  • iRODS provide the mechanism to monitor resource usage: SL_DISK_SPACE

• We need to know the usage
  • Space management

• Implementation
  • iRODSInfoServer:
    • Deployed on iRODS master server
Progress

• **Space Management Functions**
  - srmReserveSpace
  - srmReleaseSpace
  - srmUpdateSpace
  - srmGetSpaceMetaData
  - srmChangeSpaceForFiles
  - srmGetSpaceTokens

• **Permission Functions**
  - srmSetPermission
  - srmCheckPermission
  - srmGetPermission

• **Directory Functions**
  - srmMkdir
  - srmRmdir
  - srmRm
  - srmLs
  - srmMv

• **Data Transfer Functions**
  - srmPrepareToGet
  - srmBringOnline
  - srmPrepareToPut
  - srmCopy
  - srmStatusOfCopyRequest
  - srmReleaseFiles
  - srmPutDone
  - srmAbortRequest
  - srmSuspendRequest
  - srmResumeRequest
  - srmGetRequestSummary
  - srmGetRequestTokens

• **Discovery Functions**
  - srmGetTransferProtocols
  - srmPing
SRM API: srmPing

`srmPing()`: used to verify the responsiveness of the service, to retrieve the SRM version and other internal information.
SRM API: srmPrepareToPut

`srmPrepareToPut()`: used to write files into the storage. Upon the client’s request, SRM prepares a TURL so that client can write data into the TURL.

Lifetime (pinning expiration time) is assigned on the TURL.

- Target space token and SURLs
- Asynchronous operation (typically)
  - Request token returned by SRM service
  - Request status may be checked through `srmStatusOfPutRequest()` with the returned request token.
SRM API: 
srmPrepareToGet

`srmPrepareToGet()`: used to bring files upon the client’s request. It assigns TURL so that client can access the file.

- Source SURLs
- Asynchronous operation (typically)
  - Request token returned by SRM service
  - Request status may be checked through `srmStatusOfGetRequest()` with the returned request token.
- Similar function: `srmBringOnline()`, bring files online but do not return TURLs.
SRM API: srmStatusOfPut/GetRequest

- `srmPrepareOfPutRequest()`: used to check the status of the previously request `srmPrepareToPut`. Client can get target TURLs if the status is **SRM_SUCCESS**.

- `srmPrepareOfGetRequest()`: used to check the status of the previously request `srmPrepareToGet`. Client can get source TURLs if the status is **SRM_SUCCESS**.
SRM API: srmPutDone and srmReleaseFiles

`srmPutDone()`: used to notify the SRM that the client completed the file transfer(s) to the TURL(s). This should normally follow `srmPrepareToPut`.

`srmReleaseFiles()`: used to release pins on the previously requested “copies” of the SURLs. This function normally follows `srmPrepareToGet` and `srmBringOnline` functions.
SRM API: srmReserveSpace and srmGetSpaceMetadata

- **srmReserveSpace()**: used to reserve a space in advance for the upcoming requests to get some guarantee on the file management.

- **srmGetSpaceMetadata()**: used to get information of a space. Space token must be provided, and space tokens are returned upon a completion of a space reservation through `srmReserveSpace`.
Synchronous and Asynchronous

SRM service provides two class of methods:

- Asynchronous methods (non-blocking call)
- Synchronous methods (blocking call)
Asynchronous Operations

CopyClient 1
CopyClient 2
CopyClient 3
CopyClient n

Push case
Pull case
Progress

• The 1st stage:
  • Core Functions
    • Space Management Functions.
    • Permission Functions.
    • Directory Functions.
    • Data Transfer Functions.
    • Discovery Functions.
  • AMGA DB Schema
  • iRODS Server Manager
    • iRODSInfoServer
Progress (Cont.)

• 2nd stage
  • Internal space management functions
    • Use a thread to recycle expired space
  • Asynchronous operation
    • Space functions
    • Transfer functions
References

- SRM working group:
- iRODS:
  - https://www.irods.org/
- AMGA:
  - http://amga.web.cern.ch/amga
- Globus:
  - http://www.globus.org
- CoG:
- Axis:
  - http://ws.apache.org/axis/