SRB Replicated Data Management for Cooperative Computing

Arcot (Raja) Rajasekar
San Diego Supercomputer Center
sekar@sdsc.edu

What is SRB?

- SRB is an Intelligent Data Access System
- SRB provides federated access to datasets
- SRB provides protocol transparency to diverse and distributed storage systems
- SRB provides location transparency to distributed datasets
- SRB provides access transparency to remote user
The Storage Resource Broker is Middleware

- Application (SRB client)
- SRB Server
- Distributed Storage Resources:
  - Database systems (DB2, Oracle, Illustra, ObjectStore)
  - Archival storage systems (HPSS, ADSM, UniTree)
  - File systems (UNIX, NTFS, HTTP, FTP)

The SRB Process Model

- Application
- SRB Master
- SRB agents
- MCAT
**Federated SRB Operation**

- SRB server
- SRB agent

**SRB Concepts (1)**

- **Abstraction of User Space**
  - no domain dependence
  - no user accounts needed on remote servers
- **Abstraction of Resources**
  - Logical Resource Definitions - bundling
  - Resource type and Access protocol transparency
- **Abstraction of Data and Collections**
  - Persistent Identifier and Global Name Space
- **Uniform Access Methods**
SRB Concepts (2)

- Provide Scalability
  - Hosts
  - Resource Types
  - Resources
  - Collections
  - Data Objects - size and number
  - Users & Groups
  - Methods
  - MetaData

SRB Concepts(3)

- Provide Logical Abstractions
  - srbSpace - an abstract storage space
  - Resource Types - resource defined by properties
  - Resources - resource identified by name and type
    - multiple resources tied together as a single resource
  - Collections - abstraction over directory structure
    - distributed & curated
  - Datasets - identified by properties
  - Users - authenticated across hosts/networks
  - Domain - abstraction over physical domains
  - Metadata Schema/Attributes
**SRB Concepts (4)**

- Replication of Datasets
- Collections for logical co-location
- Containers for physical co-location
- Access Control Lists for Authorization
- Ticket-based Access
- Auditing
- Authentication and Encryption (SEA)
- Server-side proxy Operations
- Metadata-based Discovery
- Rich Interface - programmatic & interactive

---

**SRB Space**

DR - Data Repository
DL - Dig Library
MC - Meta Catalog
CP - Comp Process/SRB Client
**MCAT: Metadata Catalog**

- Stores metadata about
  - Data sets, Users, Resources, Proxy Methods,
- Maintains replica information for data & containers
- Provides “Collection” abstraction for data
- Provides “Global User” name space & authentication
- Provides Authorization through ACL & tickets
  - data, collection, resources and methods
- Maintains audit trail on data & collections
- Maintains metadata for methods and resources
- Provides Resource Transparency - logical resources
- Implemented as a relational database - Oracle or DB2

---

**SRB Replication Concepts**

- Replication is a core functionality in SRB
- Global Name Space (hierarchical)
  - local name independence
  - replica can reside in any type of resource
- Persistent Id: data movement independence
- Access Control at Replica Level
- Resource Access Control
- Replicas created using SRB or from outside
- Semantic Replicas & Syntactic Replicas
- Typing of Replicas: Archive, Cache, Temporary
**SRB Data Replication Support**

- **Synchronous Replication**
  - Replication via *Logical Resource* definition
  - integrated into *open/create & write* function
  - Can choose: \( k \) out of \( n \)
  - Associate replication with containers/collections
  - Consistency

- **Asynchronous Replication - Offline**
  - `srbObjReplicate API`, `sreplicate command`, GUI

- **Out of Band Replication - outside SRB**
  - Registering of Replicas using `srbRegisterReplica API`

---

**SRB Data Replication Support**

- **Choice at Read**
  - any replica
  - specific replica (by copy number)
  - round-robin
  - “nearest”
  - by resource characteristics
  - by timestamp or other characteristics
  - data itself may be identified by meta characteristics
    - user defined metadata & annotations
    - data type, owner, comments, ...
Data Replication

SRB API

- Programmatic API
  - High-level API
  - Low-level API
  - SRB Manager API
- Command Level Interface - Scommands
- Graphical User Interface - Java Browser, NT Browser
- Web Utilities
- Transparent Access
High & Low-level API

• Low-level API
  • talks to resource drivers
  • no registration of data sets in MCAT
  • no authentication through MCAT
  • User provides all information

• High-level API
  • Uses low-level API to access resources
  • Registers data management information in MCAT
  • Uses MCAT for authentication and meta information
  • Uses MCAT for resource and data discovery
  • Access/store data in remote SRB

Low-level API

• srbFileOpen(conn, storType, host, fileName, mode)
• srbFileCreate(conn, storType, host, fileName, mode)
• srbFileClose(conn, fd)
• srbFileUnlink(conn, storType, host, fileName)
• srbFileRead(conn, fd, buffer, length)
• srbFileWrite(conn, fd, buffer, length)
• srbFileSeek(conn, fd, offset, whence)
• srbFileSync(conn, fd)
• srbFileStat(conn, storType, host, fileName, statBuf)
• srbFileMkdir(conn, storType, host, dirName, mode)
• srbFileRmdir(conn, storType, host, dirName, mode)
• srbFileChmod(conn, storType, host, fileName, mode)
High-level API

- srbObjOpen(conn, objChar, mode, collectionName)
- srbObjCreate(conn, objName, objType, resourceName, collectionName, pathName, size)
- srbObjClose(conn, od)
- srbObjUnlink(conn, objChar, collectionName)
- srbObjRead(conn, od, buffer, length)
- srbObjWrite(conn, od, buffer, length)
- srbObjSeek(conn, od, offset, whence)
- srbObjMove(conn, objChar, collectionName, newResourceName, newPathName)
- srbObjReplicate(conn, objChar, collectionName, newResourceName, newPathName)
- srbObjProxyOpr(conn, Operation, sourceDesc, targetDesc)
- srbRegisterReplica(conn, objChar, collectionName, newResourceName, newPathName)

High-Level API (contd …)

- srbGetDatasetInfo(conn, objChar, collectionName, resultStruct, requiredNumber)
- srbGetMoreInfo(resDesc, resultStruct, requiredNumber)
- srbGetDataDirInfo(conn, conditionList, selectList, resultStruct)
- srbModifyDataset(conn, objId, collectionName, newValue1, newValue2, modifyType, resourceName, pathName)
- srbCreateCollect(conn, parentCollectionName, childCollectionName)
- srbListCollect(conn, CollectionName, flag, resultStruct)
- srbModifyCollect(conn, CollectionName, newValue1, newValue2, newvalue2, modifyType)
- srbModifyUser(conn, newValue1, newValue2, modifyType)
- srbSetAuditTrail(conn, setValue)
Scommands

- Sinit - initialize S-environment
- Sexit - clean up
- Sman - get manpage for Scommand
- Seat - display srbObject on screen
- Sput - copy local file into srbSpace
- Sget - copy srbObject to local space
- Sappend - append to srbObject
- Srename - change srbObject name
- Srm - remove srbObject
- Schmod - change/grant access to srbObject
- Scd - change collection
- Spwd - display current collection
- Sls - list collection
- Smkdir - make new collection
- Srmdir - remove old collection
- SgetD - get srbObject information
- SgetR - get resource information
- SgetU - get user information
- SmodD - modify srbObject info
- SmodU - modify user info
- Stoken - get native type information
- Scopy - copy srbObject in another collection and under another name
- Sreplicate - clone object in new resource - same internal id
- Smove - move srbObject to new collection or resource

Scommands (contd …)

- ingestUser - adding a new user or group
- ingestResource - adding a new resource
- ingestLogicalResource - making a new resource grouping
- addLogicalResource - adding to a resource grouping
- ingestLocation - adding new location information
- ingestToken - adding new native types
  (eg. resourceType, objectType, userType, domainName, ActionType, . . .)
Web Utilities

- Sgetw - copies a SRBobject into server site
- Sputw - copies local file in SRBspace
- Scatw - displays SRBobject on browser (handles types)
- Slsw - displays information of SRBobjects