A Simple Mass Storage System for the SRB Data Grid

Michael Wan, Arcot Rajasekar,
Reagan Moore, Phil Andrews
San Diego Supercomputer Center

Outline

- Motivations for implementing a Mass Storage System in a data grid
- Simple MSS design
- An Overview of SRB architecture and features
- Simple MSS implementation

Motivations for a MSS

- Cost of licensing commercial MSS systems.
- Efficiency and performance –
  - Tightly integrated with SRB data grid functions
  - Use SRB feature such as file replication, server directed parallel I/O, latency management.
- Elimination of duplication of features
  - Better disk cache utilization
  - Single authentication scheme
  - SRB data grid abstractions - logical name space, storage repository abstraction, logical storage resource naming and metadata management
- Provide a storage system that spans remote caches and distributed archival devices

Simple MSS - Design Goal

- A distributed farm of disk cache resources backed by a tape library
  - Use a pool of distributed cache resources to form one large cache resource
  - A tape library system to control the mounting and dismounting of tapes.
  - Support Storage Tech silo running ACSLS software
- Physical location (on cache or tape) transparency
  - User uses same access mechanism to access data independent of data location

Simple MSS - Design Goal

- Files should always be staged to cache before I/O operations
  - Tool to manage disk cache
- Large files should be stored in segments.
  - handle files of very large size
  - parallel data transfer between tapes and cache system can be implemented (not yet implemented)

Simple MSS – Components Needed

- A client-server architecture
  - authentication scheme
  - a framework for client-server and server-server communication
  - A federated server system - cache and tape resources located on different hosts.
- A metadata server that maintains
  - logical POSIX-like name space
  - mapping of each logical file name to its physical location.

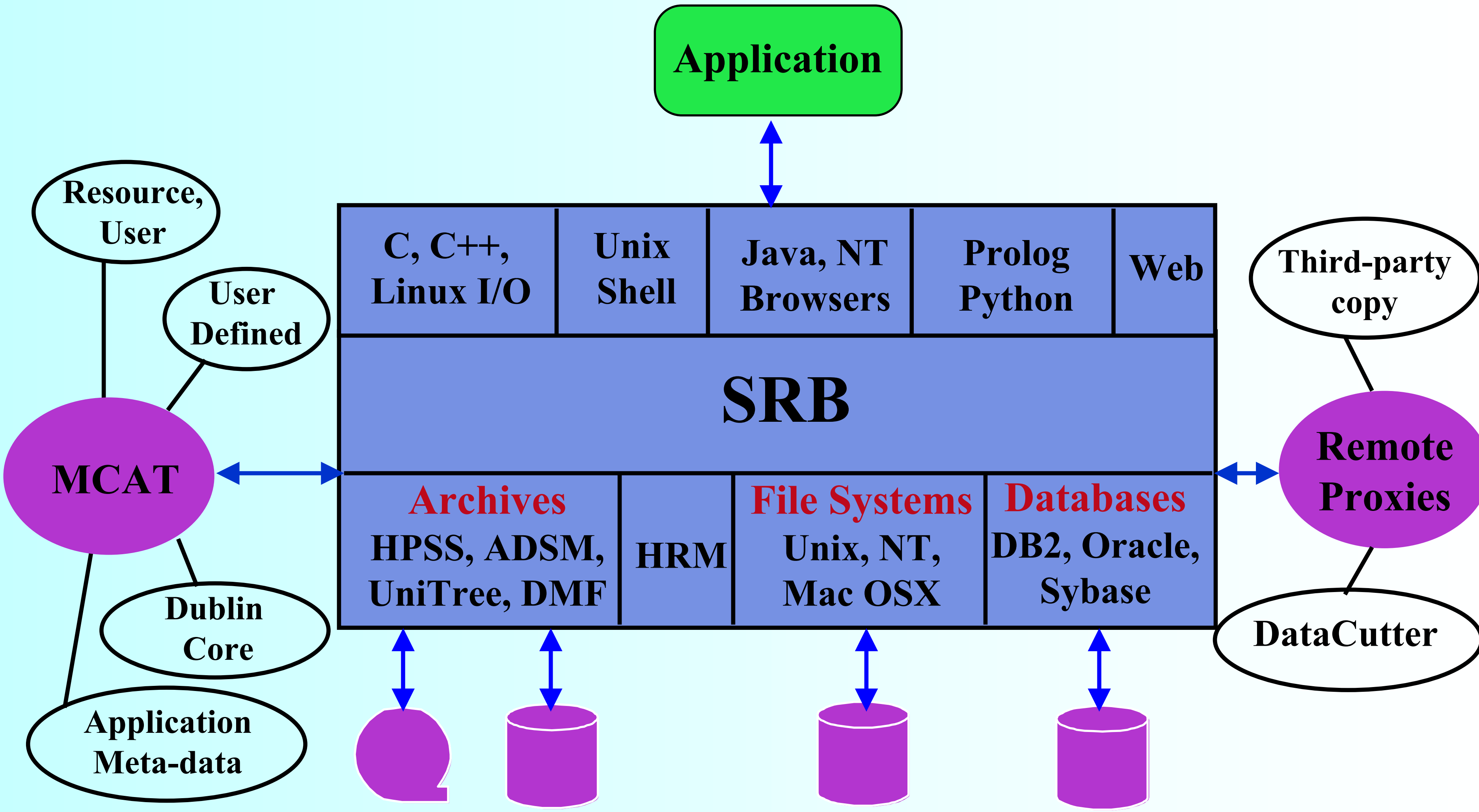
Simple MSS – Components Needed

- Server infrastructure and meta data that
  - allow files stored in the MSS to appear the same as other files stored on disk caches
  - translate user requests to physical actions using metadata
  - driver functions for basic tape I/O operations.
  - driver functions for basic cache I/O operations.
  - Functions to stage files from tape to cache and dump files from cache to tapes.
  - Data transfer between the cache system and clients.

SRB Architecture

- Federated middleware system
- Client/server model
  - Federated servers with uniform interfaces
  - Access to all resources in the federation
- MCAT metadata catalog
  - Information repository abstraction
  - Client access through SRB server

Simplified SRB Model



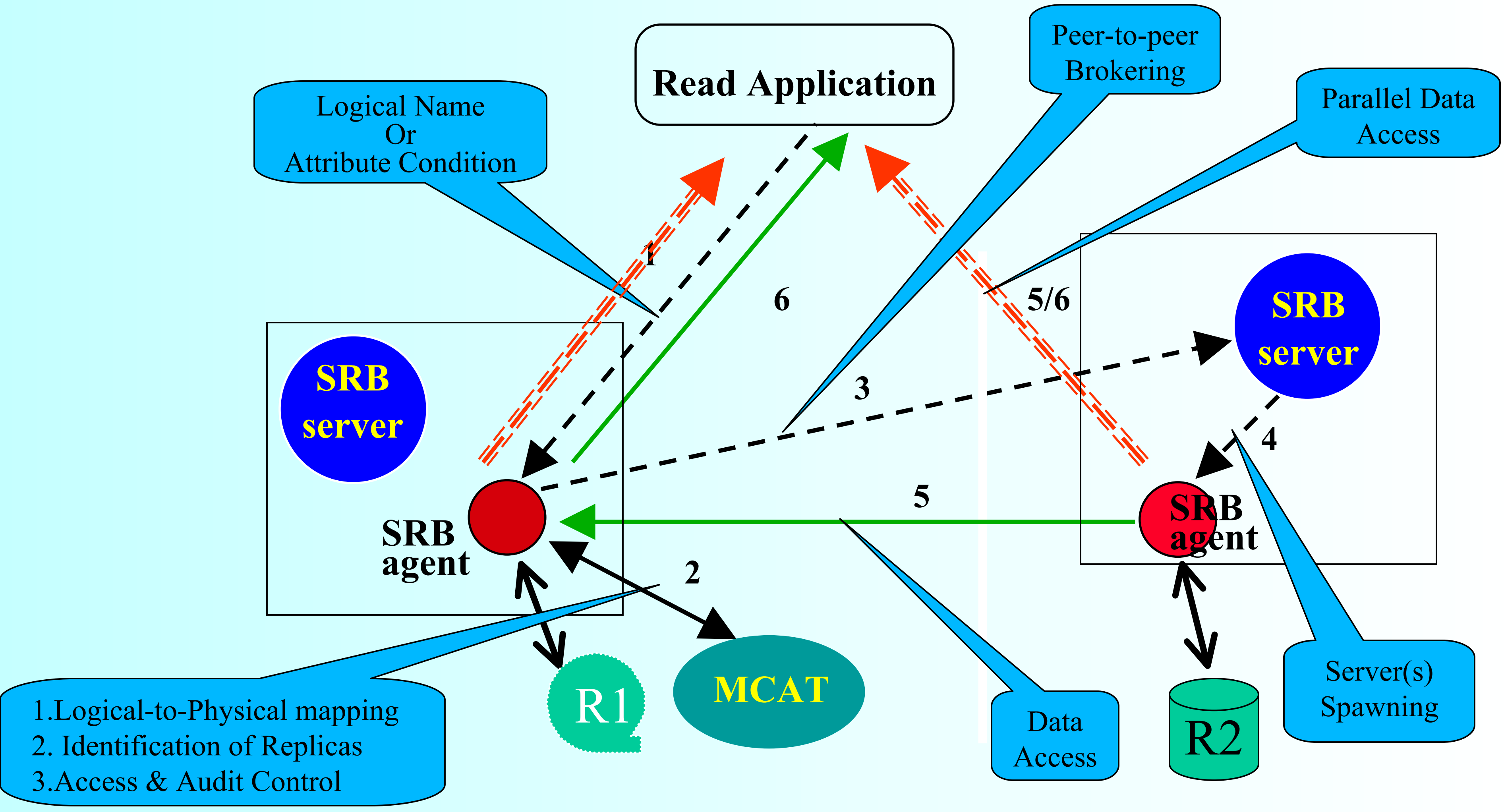
SRB Server Design

- Three layer architecture
  - Top layer (client abstraction layer)
    - Interacts with clients and other servers through tcp/ip sockets
    - User authentication
    - Handle function requests – parses requests and calls handlers in middle and bottom layers.
  - Middle layer (logical layer)
    - Input parameters are in their logical representations (logical name space, logical resource name)
    - Queries MCAT, translates from logical to physical representations (e.g., host address, type of resources, actual path where the file is located)
    - Calls functions in the bottom (physical) layer to access resources.

SRB Server Design

- Bottom layer (physical layer), handles three types of resources
  - File system
    - Drivers for the 17 POSIX functions (creat, open, read, write..)
    - FS supported : UNIX, HPSS, DPSS, UniTree, gridFTP (to be released), SRB's own tape library system (to be released)
    - Include/exfSw.h – defines driver function mapping
  - DB large objects
    - Similar to files except stored in DB
    - Drivers for Oracle, DB-2 and Illustra
  - DB tables
    - Access external DB tables (query, insert, …)

Federated SRB Operation



SRB Concepts and Features

- Abstraction of User Space
  - Single sign-on
    - No need for UNIX account on every systems
  - Multiple authentication schemes supported
    - Certificates, (secure) passwords, tickets, group permissions
  - Robust access control
    - User level, grant access to multiple users
    - Group level
    - Tickets

SRB Concepts and Features

- Abstraction of Data and Collections
  - Logical Name space -
    - UNIX like directories (collections) and files (data)
    - Mapping of logical name to physical attributes - host address, physical path.
    - Single logical name space mapped to data on multiple resources
    - POSIX like API for making collections (mkdir) and data creation (creat)
  - Data replication
    - replica on different resource
    - same logical name but different replica number

SRB Concepts and Features

- Virtualisation of Resources
  - Mapping of a logical resource name to physical attributes: Resource Location, Type & Access transparency
  - Client use a single logical name to specify a resource
  - Resource group (logical resource) – bundling of resources, automatic replication, transparent caching/archival storage

- Uniform Access Methods
  - Use same APIs, Command Line, GUI Browsers, Web-Access (Portal,WSDL, CGI) to access various resources

SRB Concepts and Features

- **Performance enhancement**
  - Client and server-driven Parallel I/O strategies
  - Interface with HPSS's mover protocol for parallel I/O
  - Container – physical grouping of small files for tape I/O

Containers

- Physical Grouping of Objects
- Similar to tar but many more features
  - Each inContainer object appears in logical name space
  - Can be accessed as normal object
  - Data objects in a container moved together
    - To aid access patterns
    - To take advantage of resource characteristics
  - Automatic staging(caching) and Archiving - sync to backend
- Family of Containers
  - New container automatically created when filled
- Containers for Collections
  - Associate a collection with a container

Access Control

- Access controls on logical names
  - Datasets
  - Collections
  - Resources
- Multi-level access
  - Read, Annotate, Write, Curate, Own
- Access control for users and groups
- Ticket-based access control
- Audit access

Simple MSS – Components Needed

- A tape library server to schedule, mount and dismount tapes.
- A tape database that tracks the usage of all tapes controlled by the MSS.
- A set of tape management utilities to initialize and migrate data

Simple MSS – Implementation

- Use many existing features in SRB
    - Federated client/server architecture
    - MCAT - logical POSIX-like name space
    - Parallel I/O for data movement
- Mechanisms added to the SRB
    - A tape library server for mounting and dismounting tapes
    - Drivers for Tape I/O in SRB server
    - metadata needed to manage files on the MSS.
    - functions to stage files from tape to cache.
    - Tape management functions and utilities.

Simple MSS – Implementation

- Compound resource
  - Appear as a single resource to user
  - Internally:
    - A pool of distributed cache resource for front end
    - A tape library system resource for back end

Simple MSS – Implementation

- Compound object
  - Object stored in compound resource
  - Behave as a single normal SRB object
  - I/O always always done on cache resource first
    - When object opened for read/write:
    - Check whether already on cache.
    - Stage to one of the cache resources.
  - Dirty bit set if object modified
  - API for synchronizing dirty copies to tape and purging cache copies by sys admin
  - File locking on object during I/O

Simple MSS – Implementation

- Tape library server
  - An independent server
  - For STK silo running ACSLS software
  - Use the same SRB server framework
    - Authentication scheme
    - Client-server function call
  - Scheduling, mounting and dismounting tapes
  - Handle multiple drive types
  - Handles only 3 function calls
    - mount tape
    - dismount tape
    - report priorities by drive types (based on number of drives idling and in use)

Simple MSS – Implementation

- Other supporting features
  - Drivers in SRB servers for tape I/O operations
  - A database schema and metadata that tracks tape usage
    - Tape labels controlled by MSS
    - Current tape positions
    - Bytes written for each tape
    - Full flag
  - Functions to insert, modify and query the tape meta data
  - Tape management utilities - tape initialization, data migration, tape metadata manipulation and query, etc.

Comparison to IEEE MSSRM

- Provides similar functionality
- Implementation - similarities and differences.
  - Differences attributed to SRB abstraction mechanisms
    - SRB MSS uses the underlying File System for managing data storage
    - Reference Model - maps bitfiles to the logical and physical volume abstractions
  - Name Server – Similar to SRB’s POSIX-like name space
  - Reference model’s Bitfile Server, Storage Server and Mover are combined into a single SRB resource server
  - Similar Migration-Purge facility

Simple MSS - Status

- First version released with SRB 2.0 (2/19/03)
- Validated through significant testing. About to enter production at SDSC
- Performance on SDSC resources
  - Cache to client - up to 40 Mbytes/sec
  - Tape to cache - 5-7 Mbytes/sec (3590 tape)

More Information

http://www.npaci.edu/DICE