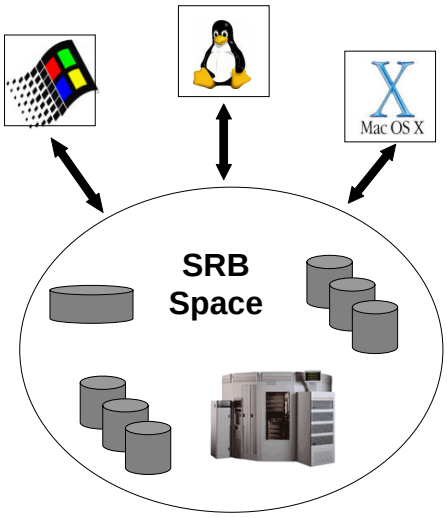
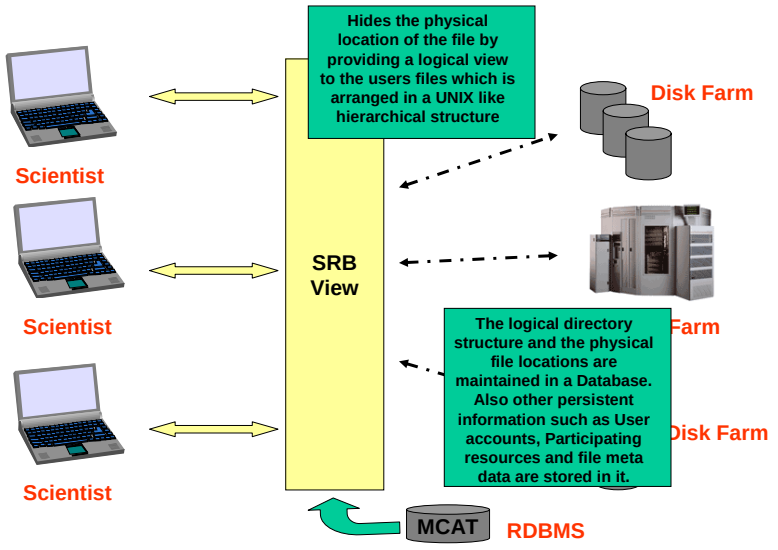
SDSC Storage Resource Broker (SRB) Introduction and Applications

based on material by Arcot Rajasekar, Reagan Moore et al
San Diego Supercomputer Center, UC San Diego

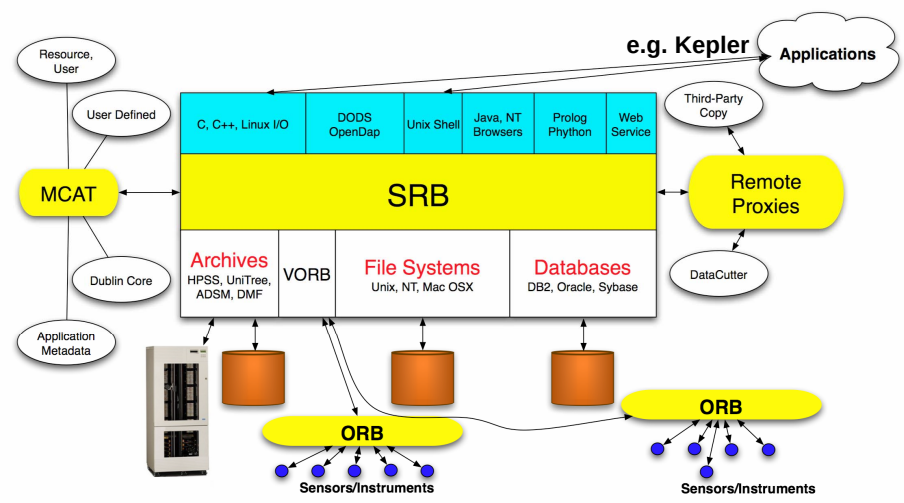
- A distributed file system (Data Grid), based on a client-server architecture.
- It's also more: It provides a way to access files and computers based on their attributes rather than just their names or physical locations.
- It replicates, syncs, archives, and connects heterogeneous resources in a logical and abstracted manner.



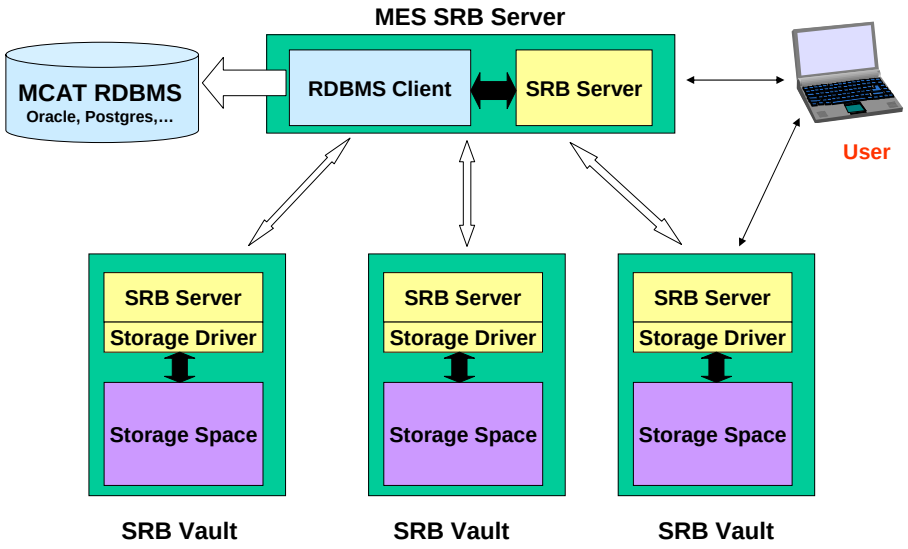
SRB Logical Structure



SRB Block Diagram



SRB Physical Structure



SRB Communication

User File Request

a) SRB Client sends request for file to SRB server.

b) SRB Server contacts MCAT Enabled Server (MES).

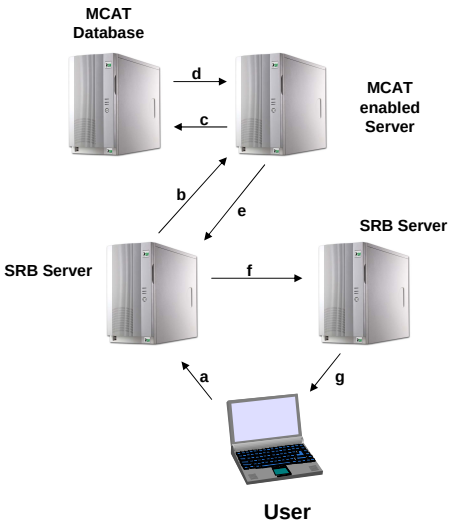
c) MES translates query into SQL and sends to database hosting MCAT

d) Database query returned to MES

e) Location of file etc returned to SRB Server A.

f) SRB Server A contacts SRB Server B hosting data file.

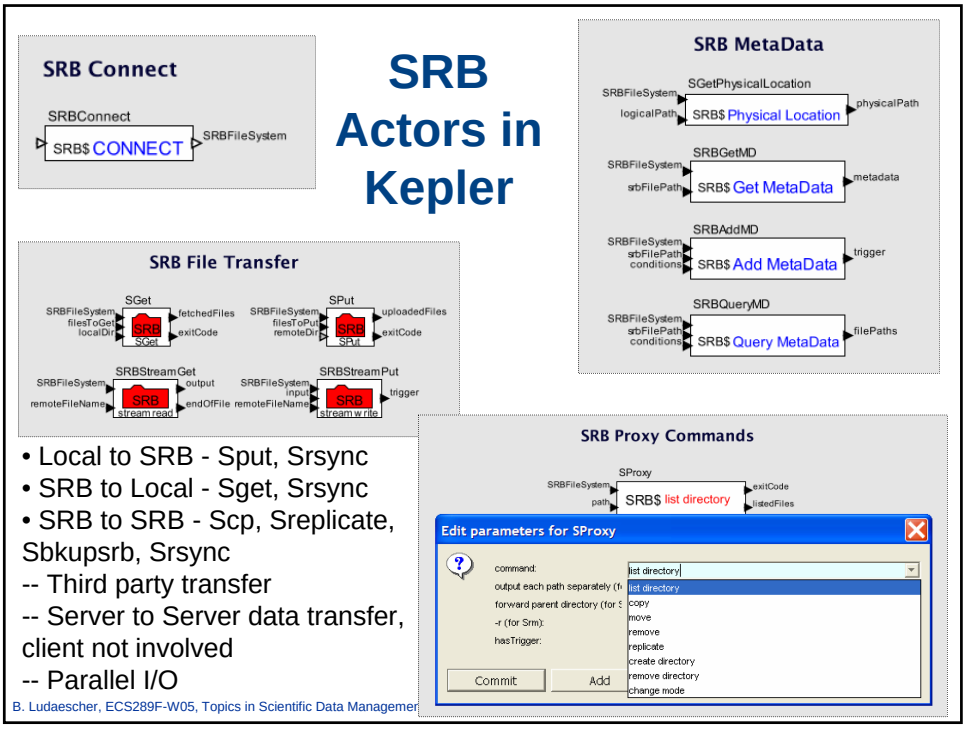
g) Data file transferred to user.



SRB Access Interfaces

- Scommands
  - Unix file system like interface
  - Versions available for Unix, Dos and Mac
- inQ –
  - Windows Explorer style interface
  - Version available for Windows
- MySRB
  - Web Browser interface
- Client access API
  - C API
  - JARGON – Java API to SRB.
  - MATRIX – SRB Workflow management system
  - KEPLER – SRB actors

SRB Actors in Kepler



- Local to SRB - Sput, Srsync
- SRB to Local - Sget, Srsync
- SRB to SRB - Scp, Sreplicate, Sbkupsrb, Srsync

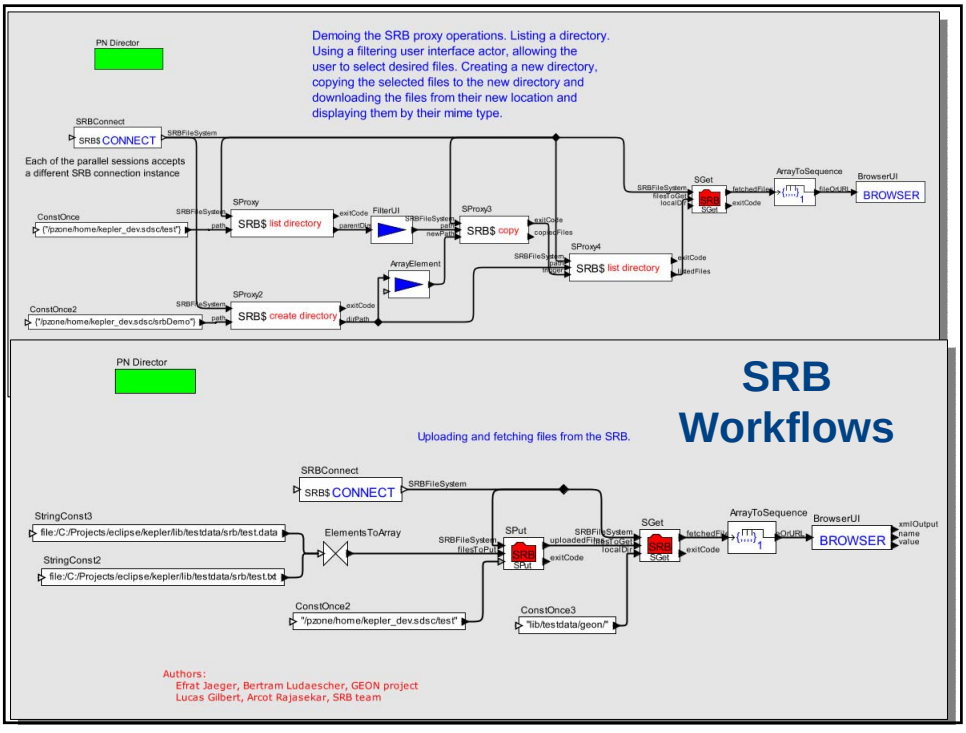
-- Third party transfer

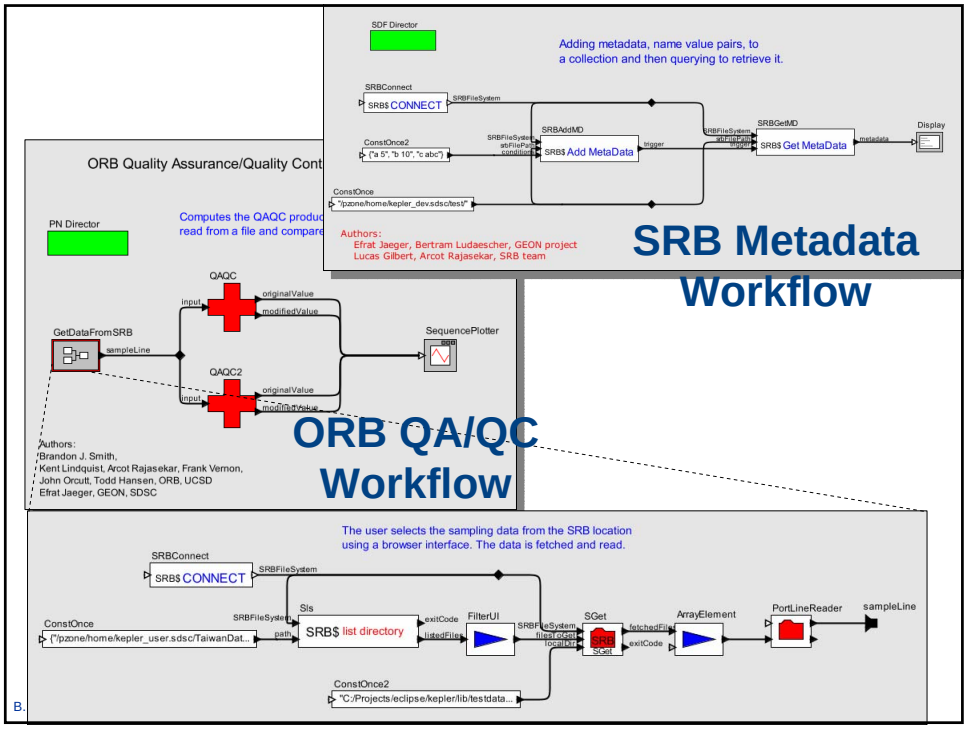
-- Server to Server data transfer, client not involved

-- Parallel I/O

B. Ludaescher, ECS289F-W05, Topics in Scientific Data Managemer

SRB Workflows





SRB User Interfaces

Scommands – primary interface to SRB

Bash like commands for interacting with SRB.

Versions available for Unix, Windows and Mac

Can also be used for scripting for batch processing.

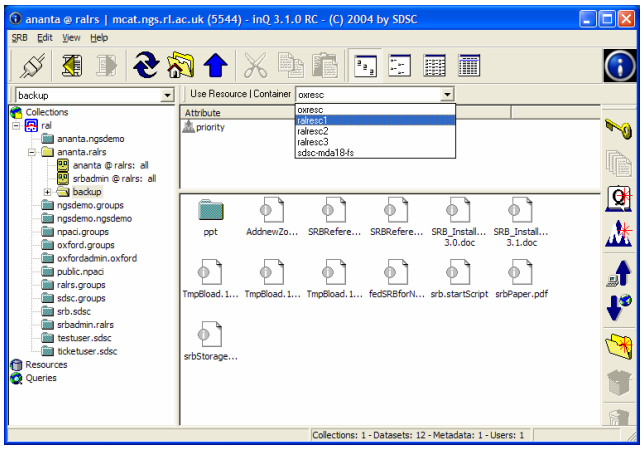
```
mcat@grid-data10:~/backup
[mcat@grid-data10 backup]$ Sinit
[mcat@grid-data10 backup]$ Sls
/ral/home/srbadmin.ralrs:
  3.1.0 readme.txt
  Beginners User Guide.doc
  metaFile
  metaFile2
  metaFileManyData
  sqlnet.log
[mcat@grid-data10 backup]$ Smkdir testDir
[mcat@grid-data10 backup]$ Scd testDir
[mcat@grid-data10 backup]$ Sput SRB3.1.0.tar
[mcat@grid-data10 backup]$ Sls
/ral/home/srbadmin.ralrs/testDir:
  SRB3.1.0.tar
[mcat@grid-data10 backup]$ Sls -l
/ral/home/srbadmin.ralrs/testDir:
  srbadmin    0 ralresc1              27176960 2004-05-08-
19.15 % SRB3.1.0.tar
[mcat@grid-data10 backup]$
```

SRB User Interfaces

- InQ – Windows Explorer Style interface to SRB

• Support for drag and drop between Windows mounted filesystems

• Provisions for access control enforcements, file replication, metadata entry and metadata query.

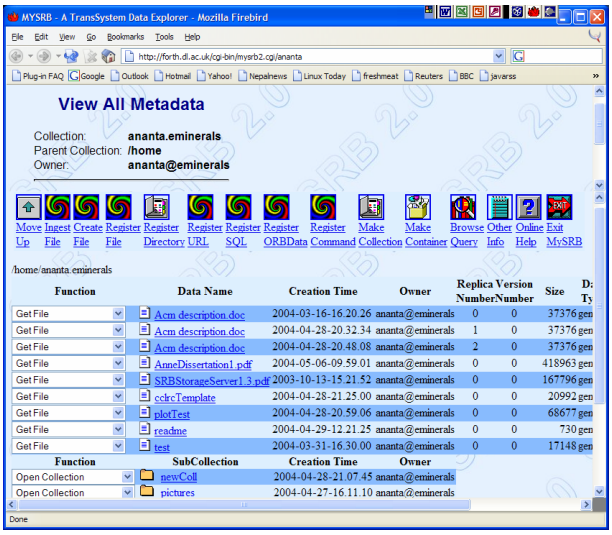


SRB User Interfaces

- My SRB Interface – Web browser interface to SRB

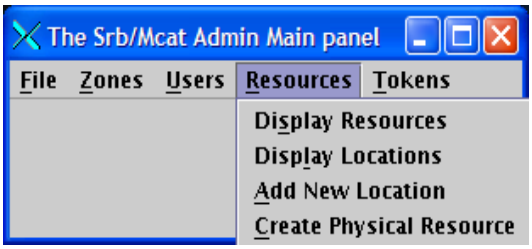
• Web based interface to SRB space.

• Works through port 80 and hence works around firewall issues.



SRB Admin Tool

- SRB Admin Tool

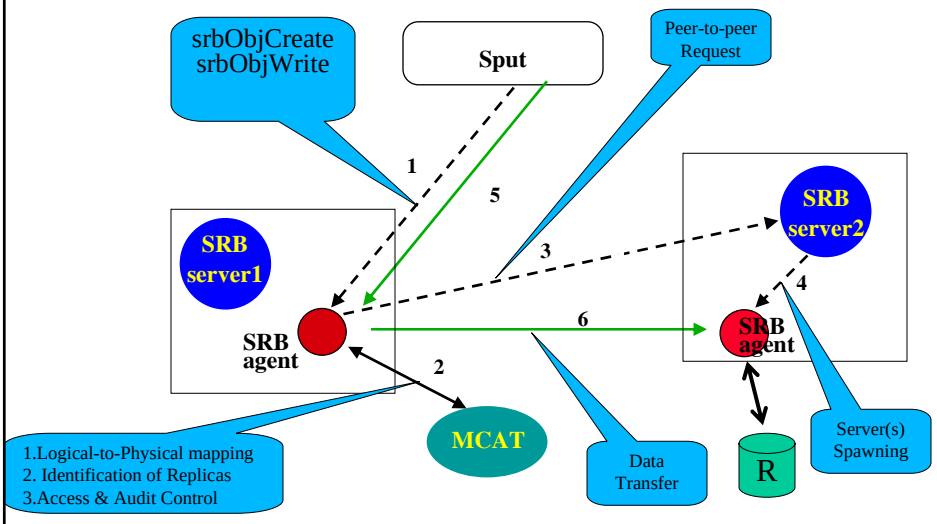


- For managing:
- Users
- Domains
- Resources
- Collaborating Machines
- Collaborating Zones (version 3 series)

Behind the scenes

- **Behind the scenes SRB provides many other functionalities in managing files and resources**
- Supports grouping of multiple physical resources into a logical resource.
- Support for direct Client Server parallel file transfers for performance improvements
- Support for bulk transfer of multiple small files into SRB server
- Supports grouping of multiple files into ‘containers’ which is then manageable for insertion and retrieval from Mass Storage systems.
- Fine Grained Access Control
- Meta Data Query and File Replication between resources

Sput – serial mode



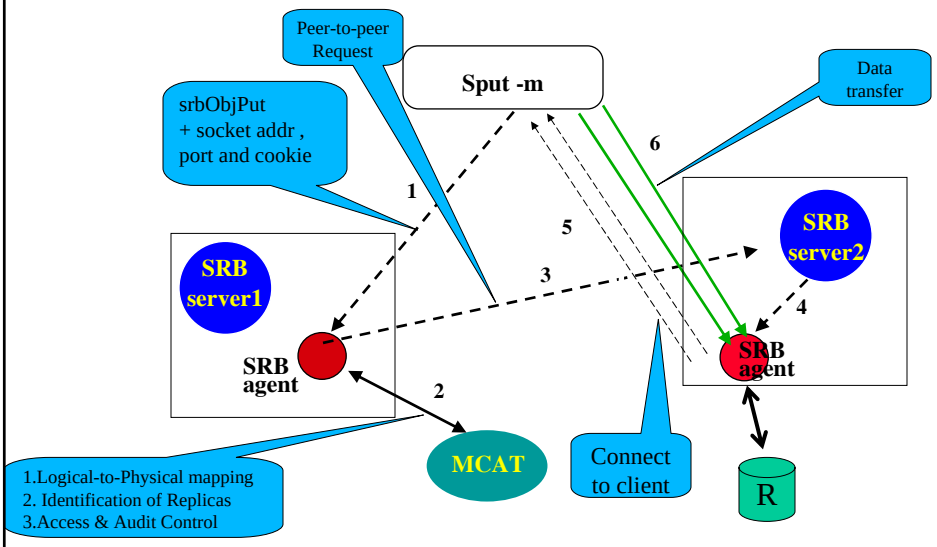
Serial Mode Data Transfer

- Simple to Implement and Use
  - Unix-like API – srbObjCreate, srbObjWrite
- Performance Issue
  - 2 hops data transfer
  - Single data stream
  - One file at a time – overhead relatively high for small files
    - MCAT interaction – query and registration
    - Small buffer transfer
- Large files – Single Hop, multiple data streams
- Small files – Single Hop, multiple files at a time

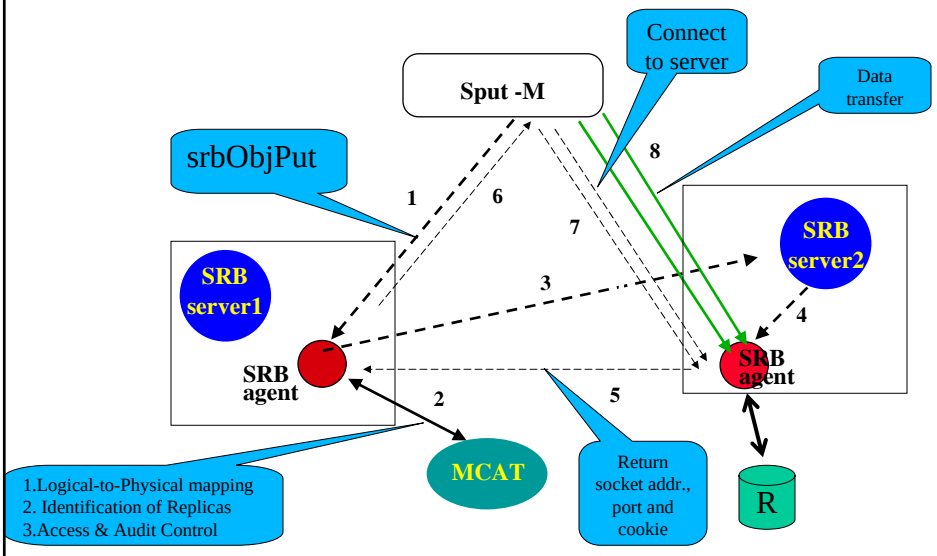
Parallel Mode Data Transfer

- For large file transfer
  - multiple data streams
  - Single hop data transfer
- Two sub-modes
  - Server initiated
  - Client initiated (for clients behind firewall)
- Up to 5 times speed up for WAN
- Two simple API functions
  - srbObjPut and srbObjGet
- Use –m (Server initiated), -M (Client initiated) options
- Available to all Scommands involving data transfer
  - As an option – Sput, Sget, Srsync
  - Automatic – Sreplicate, Scp, Sbkupsrb, SsyncD, Ssyncont

Parallel mode Data Transfer – Server Initiated



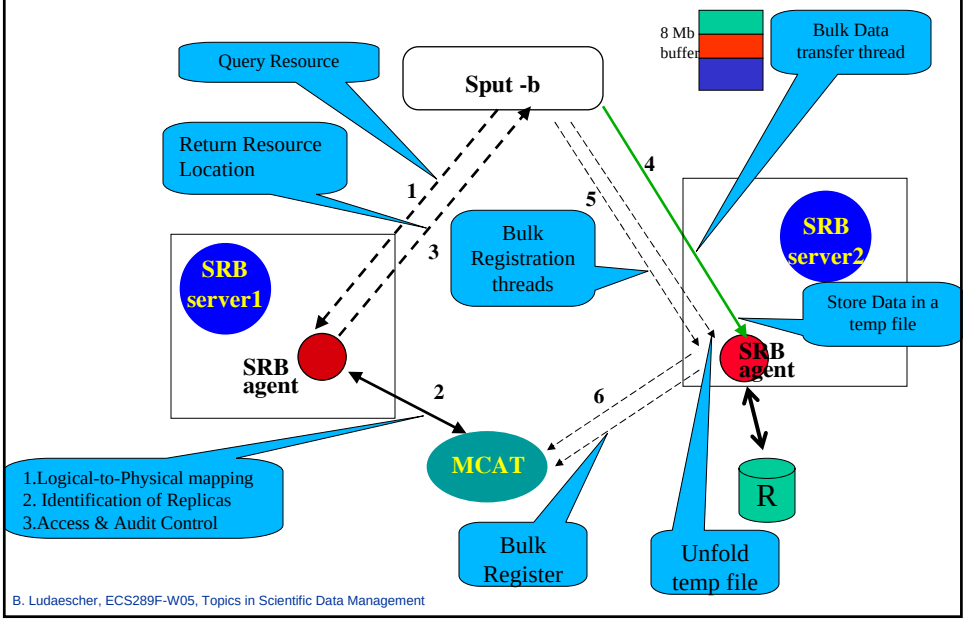
Parallel mode Data Transfer – Client Initiated



Small files Data Transfer (Bulk operation)

- Upload/download large number of small files
  - One file at a time – relative high overhead
    - MCAT interaction, Small buffer transfer
    - <= 0.5 sec/file for LAN, > 1 sec/files for WAN
- Bulk Operation
  - Bulk data transfer
    - transfer multiple files in a single large buffer (8 Mb)
  - Bulk Registration
    - Register large number of files (1,000) in a single call
  - Multiple threads for transfer and registration
  - Single Hop
  - 3-10 times speedup
  - All or nothing type operation
  - Specify -b in Sput/Sget

Bulk Load Operation



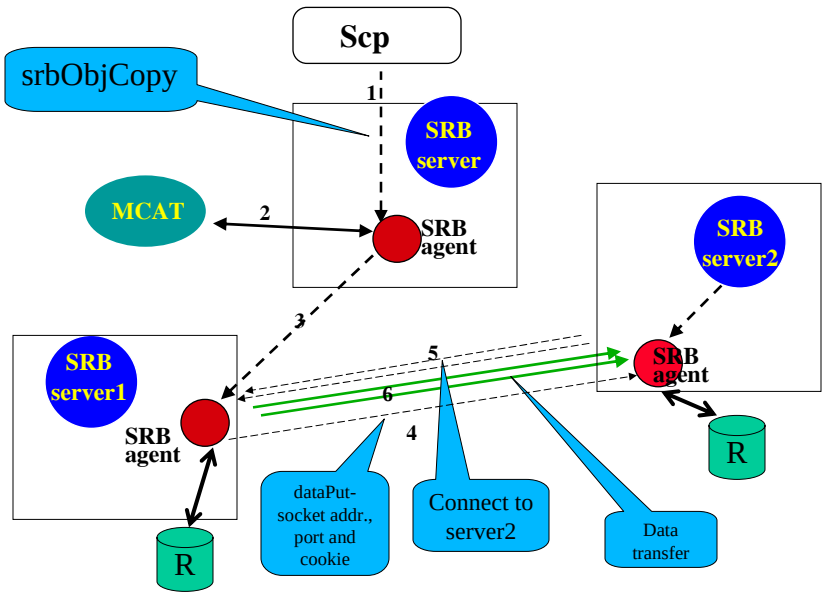
Container - Archival of Small files

- Performance issues with storing/retrieving large number of small files to/from tape
- Container design
  - physical grouping of small files
  - Implemented with a Logical Resource
    - A pool of Cache Resource for the frontend resource
    - An Archival Resource for the backend resource
  - Read/Write I/O always done on Cache Resource and sync to the Archival Resource
    - Stage to cache if a cache copy does not exist
    - The entire container is moved between cache and archival and written to tape
    - Bulk operation with container - faster

Examples of using container

- Make a container with name “myCont”
  - Smkcont -S cont-sdsc myCont
- Put a file into “myCont”
  - Sput -c myCont myLocalSrcFile mySRBTargFile
- Bulk Load a local directory into “myCont”
  - Sbload -c myCont myLocalSrcDir mySRBTargColl
- Sync “myCont” to archival and purge the cache copy
  - Ssyncont -d myCont
- Download a file store in “myCont”
  - Sget mySRBsrcFile myLocalTargFile
- Slscont - list existing containers and contents

Third Party Data Transfer



Other useful Data Management Scommands

- Srsync, Schksum -
  - Data synchronization using checksum values
  - similar to UNIX's rsync
- Sreplicate, Sbkupsrb
  - generate multiple copies of data using replica
  - Replica - multiple copies of the same file
    - same Logical Path Name - e.g., /home/srb.sdsc/foo
    - replica on different resources
    - Each replica has different replNum
    - Most recently modified flag

Commands Using Checksum

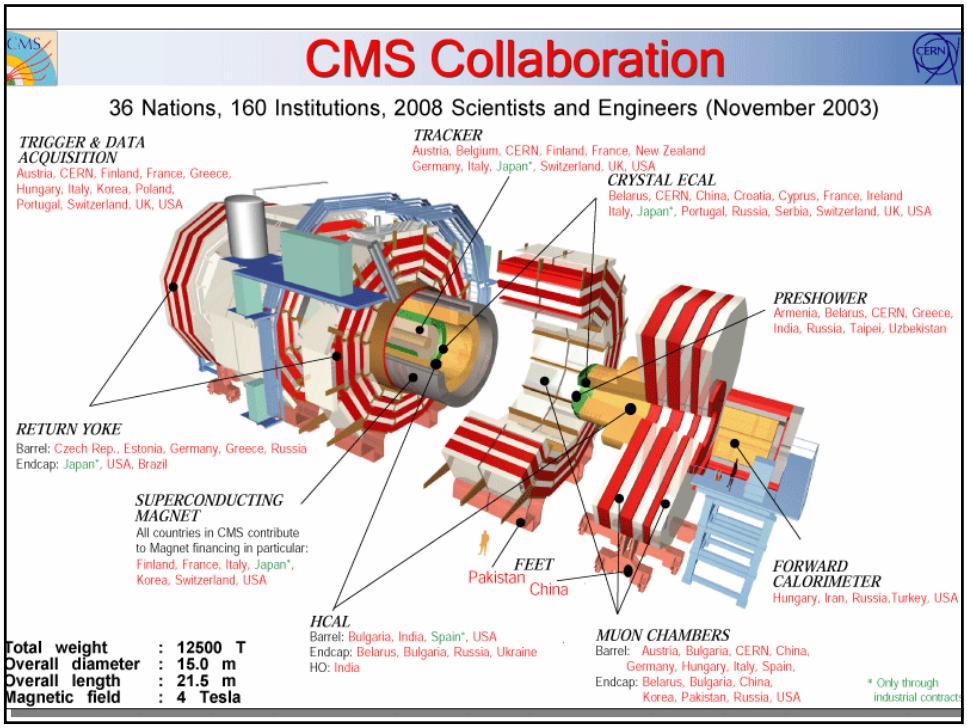
- Registering checksum values into MCAT
  - at the time of upload
    - Sput -k - compute checksum of local source file and register with MCAT
    - Sput -K
      - checkum verification mode
      - After upload, compute checksum by reading back uploaded file
      - Compare with the checksum generated with locally
  - Existing SRB files
    - Schksum
      - compute and register checksum if not already exist
    - Srsync - if the checksum does not exist

Srsync

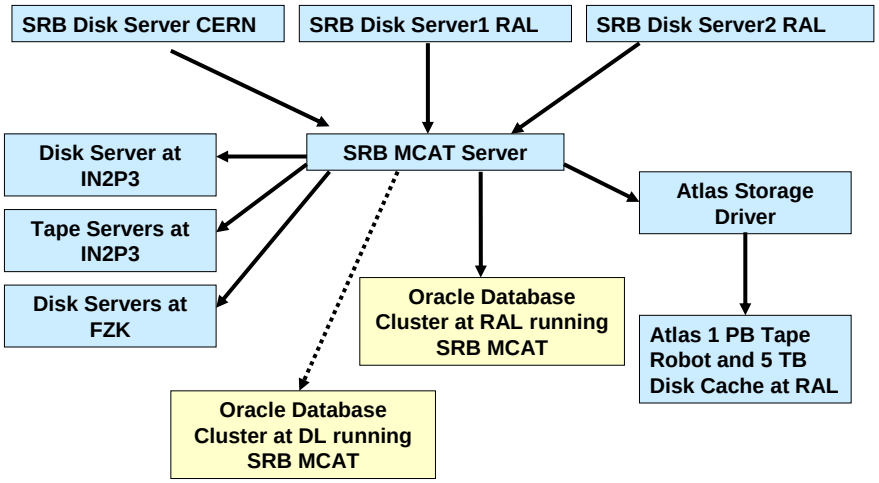
- Synchronize the data
  - from a local copy to SRB
    - Srsync myLocalFile s:mySrbFile
  - from a SRB copy to a local file system
    - Srsync s:mySrbFile myLocalFile
  - between two SRB paths.
    - Srsync s:mySrbFile1 s:mySrbFile2
- Similar to rsync
  - compare the checksum values of source and target
  - upload/download source to target if
    - target does not exist or checksum differ
  - Save checksum values to MCAT
- Some Srsync options
  - -r --- recursively Synchronizing a directory/collection
  - -s --- use size instead of checksum value for determining synchronization
    - Faster - no checksum computation
    - Less accurate
  - -m, -M --- parallel I/O

Sreplicate, Sbkupsrb

- Generate multiple copies of data using replica
- Sreplicate - Generate a new replica each time
- Sbkupsrb
  - Backups the srb data/collection to the specified backupResource with a replica
  - If an up-to-date replica already exists in the backupResource, nothing will be done



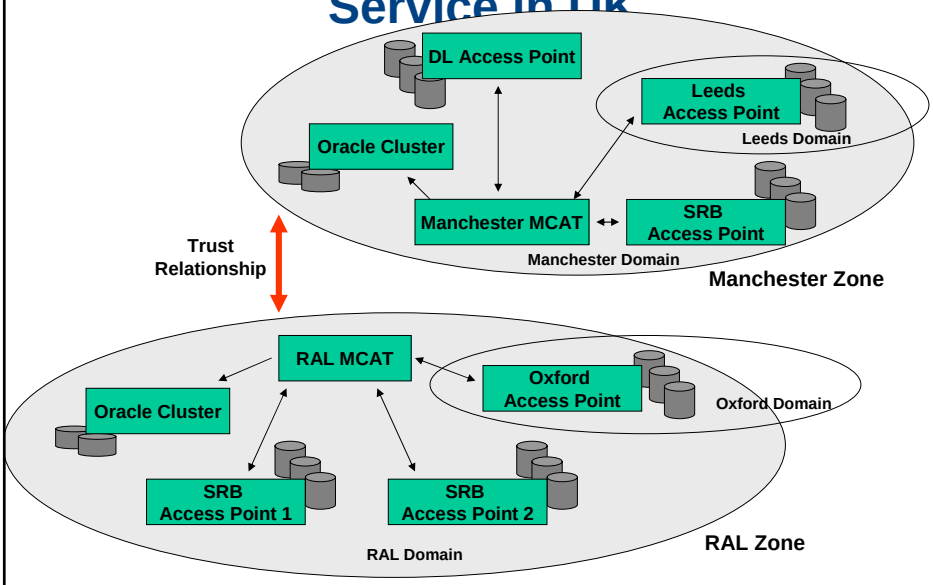
CMS SRB



Back then about 260,000 connection were seen per day, 48 TB of data, 1 million data files, 70 Physical Resources, 18 Countries involved

B. Ludaescher, ECS289F-W05, Topics in Scientific Data Management

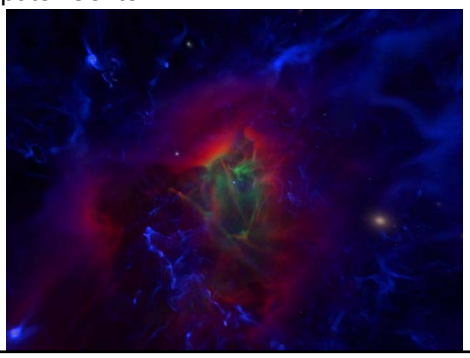
SRB Deployment on National Grid Service in UK



Hayden Planetarium Project

"A Search for Life: Are We Alone?"

- The animations was done for the new planetarium show "A Search for Life: Are We Alone?" narrated by Harrison Ford.
- The show opened Saturday, March 2nd.
- Sites involved in the project :
- AMNH = American Museum of Natural History
- NCSA = National Center for Supercomputing Applications
- SDSC = San Diego Supercomputer Center
- University of Virginia
- CalTech, NASA, UCSD



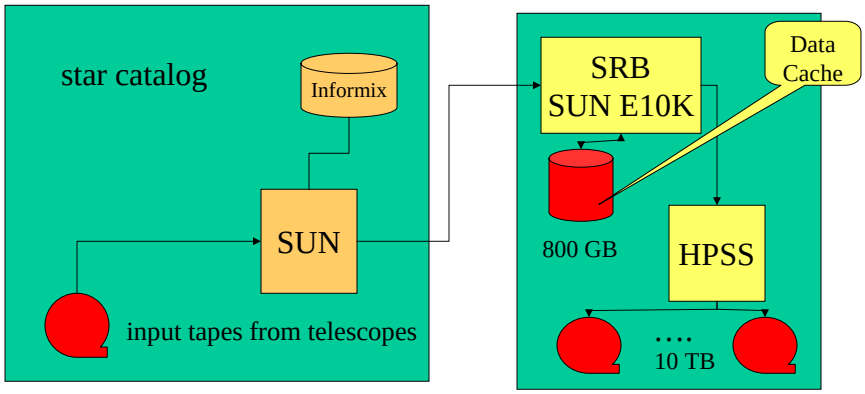
Hayden Data Summary

- ISM = Interstellar Medium Simulation
  - run by Mordecai Mac Low of AMNH at NCSA : 2.5 Terabytes sent from NCSA to SDSC. Data stored in SRB (HPSS, GPFS).
- Ionization :
  - Simulation run at AMNH, 117 Gigabytes sent from AMNH to SDSC. Data stored in SRB.
- Star motion:
  - Simulation run at AMNH by Ryan Wyatt.38 Megabytes sent from AMNH to SDSC.
- Data
  - total 3 * 2.5 TB = 7.5 TB
- Files
  - 3 * 9827 files + miscellaneous files
- Duration
  - December 2001, January, February 2002

Digital Sky / NVO

- 2MASS (2 Microns All Sky Survey):
  - Bruce Berriman, IPAC, Caltech; John Good, IPAC, Caltech, Wen-Piao Lee, IPAC, Caltech
- NVO (National Virtual Observatory):
  - Tom Prince, Caltech, Roy Williams CACR, Caltech, John Good, IPAC, Caltech
- SDSC – SRB :
  - Arcot Rajasekar, Mike Wan, George Kremenek, Reagan Moore

Digital Sky Data Ingestion



•http://www.ipac.caltech.edu/2mass
•The input data was on tapes in a random order.
•Ingestion nearly 1.5 year - almost continuous
•SRB performed a spatial sort on data insertion. The disc cache (800 GB) for the HPSS containers was utilized.

Digital Sky Data Ingestion & Sorting

- 4 parallel streams (4 MB/sec per stream), 24*7*365
- Total 10+TB, 5 million, 2 MB images in 147,000 containers.
- Ingestion speed limited by input tape reads
  - Only two tapes per day can be read
- work flow incorporated persistent features to deal with network outages and other failures.
- C API was utilized for fine grain control and to be able to manipulate and insert metadata into Informix catalog at IPAC Caltech.

•Sorting of 5 million files on the fly
•Input tape files: temporal order
•Stored SRB Containers: spatial order
  •Scientists view/analyze data by neighborhood
•Data Flow:
  •Files from tape streamed to SRB
  •SRB puts them in proper 'bins' (containers)
  •Container cache-management a big problem
  •Files from a tape may go into more than 1000 bins
  •Cache space limitations (300-800GB) made for a lot of trashing
  •SRB Daemon managed cache - watermarks