Data Sharing with Storage Resource Broker®

Enabling Collaboration in Complex Distributed Environments

White Paper

Table of Contents

Introduction....3
The Case for SRB....4
SRB Architecture....4
SRB Design....4
Terminology....4
Components....6
Metadata....7
SRB Features....8
SRB Highlights....8
Virtualization....8
Services....9
Security....9
Expanding into the Enterprise....10
Performance....11
Automation....13
Additional Interfaces....13
Additional Drivers....13
Tickets....14
Reports....14
Conclusion....14
About Nirvana Storage....14
About General Atomics....14

The Storage Resource Broker

Introduction

This paper describes the architecture and capabilities of the Storage Resource Broker (SRB). SRB is advanced software that enables the federation of enterprise data. Powerful yet easy-to-use, SRB helps collaborating knowledge workers to discover, share and manage high-value files through simple interfaces.

SRB provides applications and clients with a uniform interface to access heterogeneous distributed storage resources, including file systems, databases, and archival storage. The use of a metadata catalog (MCAT) provides an enterprise-wide, "collection"-oriented view of data, and SRB provides a large number of interfaces to enable discovery, presentation and management of data collections.

General Atomics developed SRB in a joint effort with the San Diego Supercomputer Center. The product responded to a need for researchers in collaborative projects to share access to files on remote storage systems in diverse, heterogeneous networked operating environments. In these academic, supercomputing and government projects, SRB formed the foundation for the world's most advanced and demanding data grid implementations. Some of those *Innovator* projects, which were conducted at leading research institutions in the U.S. and span efforts in neuroscience, particle physics, cosmology, and other advanced research, are described separately in SRB Cases, available at http://www.nirvanastorage.com/solutions/cases.htm .

Needs for performance tuning, secure operation, tracking, data persistence and auditing, ease of use and improved data discovery led to the product enhancements which produced the commercial release of SRB in 2003. Experience with early implementations of commercial SRB led to further functional enhancements, improved federation access, and expanded storage support, and SRB 2004, the product described in this paper.

A culmination of intense development in demanding research environments, SRB provides a unified view and simple access to distributed data from heterogeneous sources to accredited end-users, enabling productive collaboration and efficient data exchange within a secure, scalable, easily administered environment.

The Case for SRB

From small, relatively simple ILM implementations across two or three storage subsystems within a single facility, to data grid implementations spanning geographically diverse enterprises with heterogeneous legacy compute environments, SRB enables the federation of users and data into effective collaborative groups. SRB enables easy access across a variety of storage devices and provides improved discovery, access and management of data files. Users and administrators alike benefit from an enterprise-wide, transparent view of all data.

The Storage Resource Broker precisely addresses the needs of IT professionals administering complex environments and seeking to provide a transparent but secure user view of all enterprise data: access to diverse, heterogeneous data, support for a wide variety of storage systems, and configuration flexibility to accommodate rapid growth and increasing complexity.

SRB is client/server middleware that connects applications with diverse data resources including file systems on Storage Area Networks (SANs) or Network Attached Storage devices (NAS), and online Content Addressed Storage (CAS) disk archives such as EMC Centera. Using SRB, many layers of data from disparate sources and formats can be assembled and grouped into logical Data Collections, for transparent access by authorized users.

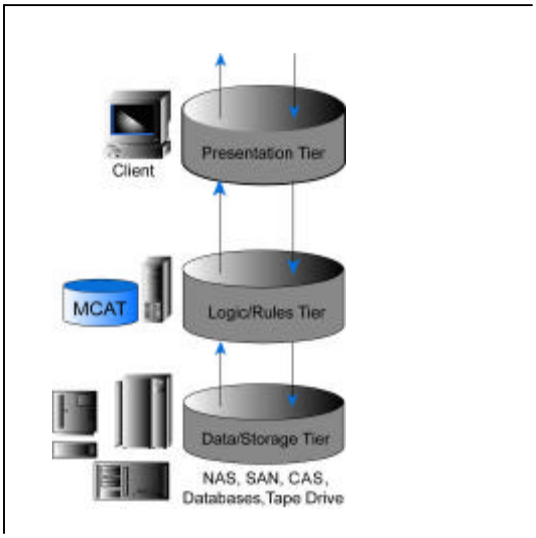
The result is an effective mechanism for managing rapidly-expanding data stores. SRB scales easily, from single storage subsystems to very large enterprise-wide federations, and provides multiple user interfaces, including Windows gateways and simple point-and-click Java applications. SRB is designed to access varied storage platforms, so it enables expanding operations at every level: departmental, organizational, enterprise, cross-enterprise, and in the public sector community, state and federal entities. SRB can easily scale into a full enterprise or cross-enterprise collaborative data grid implementation. It is a mature and stable product, under continuous development since 1995, and has been implemented at the core of some of the world's most advanced image repositories, digital libraries, archives and data grids.

SRB Architecture

SRB Design

SRB offers a simple user presentation, in effect a "Collection hierarchy" – an extension of a standard directory hierarchy – of files residing on heterogeneous data sources. The design can be described conceptually as three major layers: the SRB Clients (Presentation Tier), the Metadata Catalog or MCAT (Logic Tier), and the SRB Agents with drivers to the storage systems (Data Tier), illustrated as follows:

Figure 1: SRB Design



Terminology

Data Objects

Every piece of data managed or accessed through SRB is represented as a Data Object. Examples of Data Objects are: images, metadata files, databases, spreadsheets, database queries, URLs, or others. Data Objects can physically reside anywhere within an SRB Federation including file systems, tape drives, tape libraries, relational databases, or archives. A Data Object is not necessarily the same as a file, although it can be. Furthermore, Data Objects do not necessarily have a one-to-one relationship to the underlying data. A Data Object can in fact point to several (replicated) pieces of data (a one-to-many relationship). Therefore, a Data Object can be viewed as a logical entity, whereas a file refers to a physical entity. There are numerous attributes automatically associated with each Data Object in SRB: name, data type, size, physical path, creation

timestamp, modification timestamp, last access timestamp, custom attributes, etc.

Collections

An SRB Collection is (much like a folder or directory in a file system) an object that contains other Collections or Data Objects. A Collection is used to organize Data Objects into a logical hierarchy that is easily accessible and understood by both user and administrator. For example, a Collection named "Project X" can be used to store all the Data Objects related to project X, independent of where that data is physically stored or how it is physically structured. This logical Collection view is maintained transparently through SRB so that users need not keep track of the physical locations of their data. Collections in SRB possess the same attributes as Data Objects.

Global Namespace

The entire logical hierarchy comprising all Collections, Sub-Collections, and Data Objects is called a Global Namespace, also referred to as a *Logical Namespace*. By definition, a Global Namespace can span multiple heterogeneous and distributed storage systems and data centers.

Storage Resources

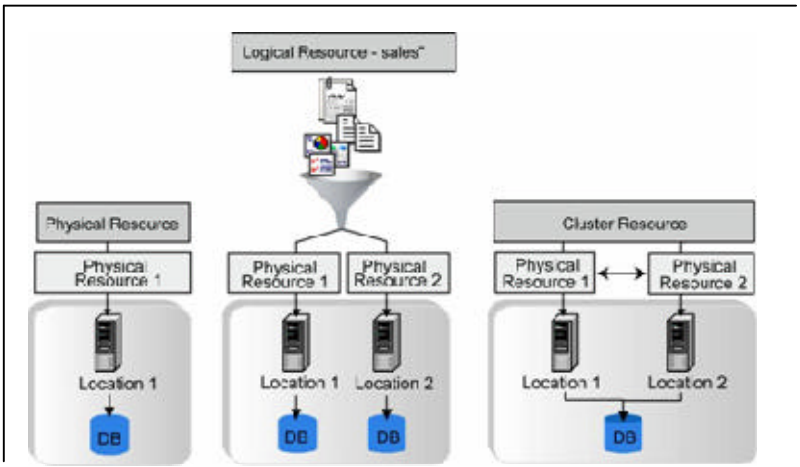
Every piece of data must ultimately be on a physical storage system. In SRB, the mapping between those storage systems and the Data Objects is done using Resources. There are three types of Resources: *Physical Resources*, *Logical Resources*, and *Cluster Resources*.

Physical Resources represent abstractions of places where Data Objects are physically stored. Such "places" include file systems on UNIX, Linux, or Windows; relational databases like Oracle, Sybase, or DB2; tape drives and libraries; Content Addressed Storage (CAS) systems; web or FTP servers etc.

Logical Resources are used to group one or more Physical Resources together, making it transparent to the SRB User where data is physically stored. Logical Resources can be useful for many reasons: (a) data can be transparently stored to several underlying resources simultaneously (replication), (b) Resources can be switched out or added, (c) load balancing can be managed among several Resources, and (d) SRB Containers can be used as a mechanism to automatically archive and stage the data in the correct archival or cache resource – - and all this can be accomplished transparently to SRB Users.

Cluster Resources enable the efficient management of clustered file systems. Behind the scenes, the Cluster Resource might in fact have many Physical Resources attached. When data is read or written on the Cluster Resource, the Cluster will iterate through its attached Physical Resources until it finds a working Resource. Cluster Resources provide SRB with an elegant automatic fail-over from one server to the next if those servers are seeing the same data.

Figure 2: Comparison of SRB Resources



Components

SRB is implemented as a distributed software layer using a client / server architecture. The combination of all SRB Servers, Clients, and storage systems is called a Federation. Every Federation must have at least one master server connected to a Metadata Catalog (MCAT). SRB Agents can be installed optionally on other servers besides the master server. The client interfaces provide easy administration, navigation, and access to all the data within the Federation. Each SRB configuration contains an MCAT and a number of client interfaces.

MCAT

The Metadata Catalog (MCAT) includes attributes required for the implementation of the Global Namespace and the mapping of Data Objects to Storage Resources within the Federation. The MCAT is used to determine where a given Data Object is physically located, and which drivers handle and access the data of interest. Furthermore, the MCAT contains all file attributes, metadata, access control lists, Storage Resource information, and user data, and also handles queries.

Agents / Locations

Optionally, an SRB Federation maintained in MCAT can be expanded to span multiple servers within a department, the entire enterprise, or even beyond the enterprise and across administrative domains. This is achieved by installing additional software called **SRB Agents** on the respective servers. An Agent provides a full range of services such as data handling, load balancing, data discovery, legacy systems integration, and proxy services.

Each SRB Agent is listening to a unique IP address and port number combination (a **Net Address**) on the server machine for requests from other Agents or SRB Clients. The combination of a unique port number and IP address (or hostname) is called an **SRB Location**. Therefore, several SRB Locations (or unique port number/IP address combinations) can coexist on a single server machine if they are monitoring different port numbers or IP addresses.

Drivers

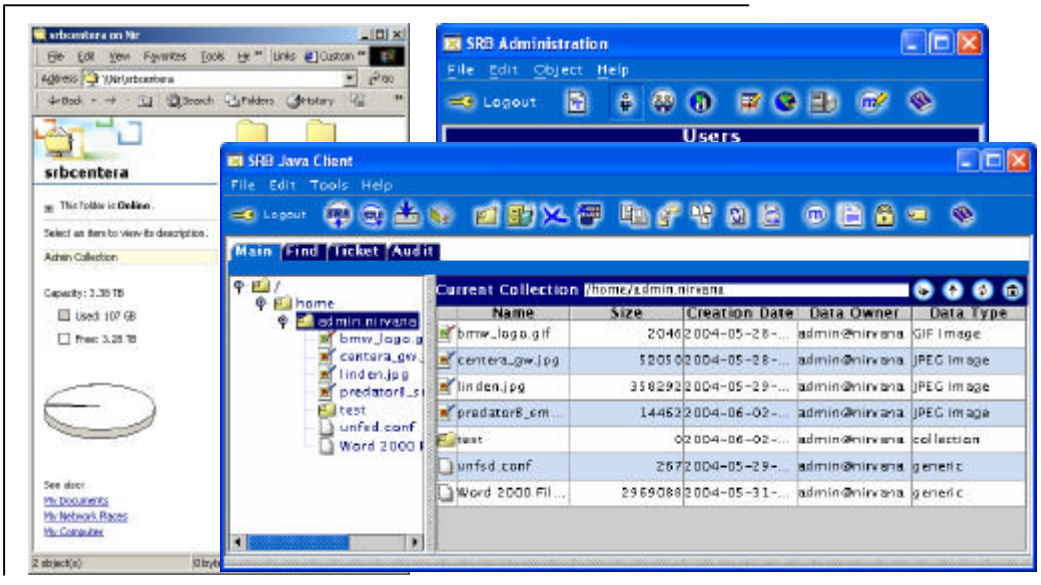
The MCAT or an SRB Agent may contain any number of drivers to interface with various storage systems. SRB is configured with drivers to EMC Centera and a number of file systems. Additional drivers are separately available and new drivers are continually under development. As data stores grow, and as the complexity of a data store increases (due to reorganization, collaboration, acquisition, etc.), SRB enables continuing data transparency and greatly simplifies migration to new storage formats and media technologies.

Client Interfaces

SRB includes the following interfaces:

- Windows (CIFS) Gateway
- UNIX/Linux (NFS) Gateway
- Internet (WebDAV) Gateway
- Grid (GridFTP) Gateway
- Java Admin
- Java Client
- Web Client
- Shell commands (for scripting)
- SDK with APIs (in C and Java)

Figure 3: SRB User Interfaces



Windows applications mainly access data using the Windows (CIFS) Gateway, which essentially creates a Windows network share using a Samba server with an SRB Module in the backend. The CIFS Gateway makes it completely transparent to applications where data is physically stored.

For applications running on UNIX/Linux the NFS Gateway will provide the same functionality as the CIFS Gateway on Windows. It enables the mounting of SRB Collections as a local directory on any UNIX/Linux workstation or server.

The entire administration of SRB can be performed through a central console (Java Admin) and allows for such tasks as the creation of new Storage Resources, the changing of access permissions, user management, Metadata Scheme customization, etc.

A user may also interface with SRB through a Java Client that can be installed on any of the supported platforms. The GUI looks familiar to anyone who has used a file manager to copy and move files and folders. Compared to the Gateways it has additional SRB-specific functionality like data migration, data discovery, or the viewing of audit trails.

The Shell commands contain utilities used for the maintenance of MCAT (Mcommands) and are usually only necessary during the installation or upgrade process of MCAT.

A number of additional clients are available as an option (see Expanding into the Enterprise).

SRB Federations often begin as production prototypes, an MCAT installed on a single server with one or two storage systems attached, sometimes with a CAS disk system serving as an online archive. This initial prototype can then be easily expanded through the addition of SRB Agents to encompass a more complex, enterprise-wide federation serving many users and spanning a wide variety of legacy processors and storage.

Metadata

The MCAT at the heart of an SRB Federation keeps track of several different metadata attributes. They are generally grouped into system-level and user-level attributes.

System-level attributes are automatically maintained by MCAT and are mostly queried, displayed, and used for data management. System-level attributes are maintained on all SRB Objects including Data Objects, Collections, Storage Resources, Locations, Users, Groups, and Domains. Examples for system-level attributes are file path, Data Object size, resource type, etc.

User-level attributes are created and maintained by applications or users and can be attached to both Data Objects and Collections. They also are useful for query, display, and data management purposes. User-level attributes are access controlled on a per-attribute-basis so that SRB Administrators can create access policies such as "Engineering can modify the revision attribute, Analysts can read it, and the Public can not even see it". Other examples for user-level attributes might be customer name, keywords, projects, etc.

Data discovery is made much easier through SRB's metadata attributes. Additionally, Collection metadata can ease discovery of Data Collections with attributes such as author or project name. SRB keeps access control lists in every user-level metadata attribute, ensuring a high level of security. Different access levels such as 'read' or 'write' are supported, assuring that metadata is only viewed and modified by users authorized to do so.

Metadata Schemes

To make SRB management easier, administrators can group different metadata attributes into SRB Metadata Schemes. Metadata Schemes provide "attribute groupings" that may be specific to certain types of Data Objects or specific user groups within the federation.

SRB Features

SRB Highlights

SRB is configured to provide an affordable entry point compatible with the requirements of most operating environments, and complementary to both current online storage options and legacy systems. At every step, users pay only for what they need. A typical entry-level SRB configuration includes the following features:

- 20TB MCAT
- 3 SRB Users (super + 2 custom)
- EMC Centera driver
- Disk driver (for UNIX/ Linux/ Windows file systems)
- Windows (CIFS) and UNIX/ Linux (NFS) Gateways
- Java Admin
- Java Client
- Mcommands (for MCAT maintenance)

SRB provides Windows, UNIX and Linux Gateways to a very wide variety of storage systems.

Virtually every component of the configuration is upgradeable, so the offering represents a scaleable path to very large (multi-petabyte) installations, with a wide variety of networked storage and server configuration options.

Supported Systems SRB

- **Operating Systems**
  AIX, HP UX, SGI Irix, Linux, Mac OS X, Sun OS, Sun Solaris, Windows
- **File Systems**
  all supported operating systems
- **Archives / HSM**
  EMC Centera, DiskXtender 2000, DiskXtender UX (formerly Unitree)
- **Databases**
  DB2, MSSQL, Oracle, Postgres, Sybase

SRB users can add SRB Agents, upgrade MCAT and server capacities, and also add optional Policy/HSM and synchronization daemons, as required. Also available are Web Client, Scommands (command-line client), Acommands (command-line administration), the SRB Software Development Kit with APIs in C and Java, SRB Container technology for reducing latencies and complexities, additional reporting mechanisms and MCAT reports, Ticketing for time- and access-count restricted access, bulk operations and the planned “zone” feature.

Virtualization

When using SRB, applications, clients, and administrators immediately benefit from the following levels of abstraction, which ease administration, data discovery, and data access:

Information Repository Virtualization

The information repository (or MCAT) where the Federation’s metadata is stored is transparently accessed and managed by SRB. This allows for a seamless exchange of the underlying information repository over time. Currently there is support in SRB for many relational databases (see *Supported Systems SRB* above). The entire information repository can be exported and imported using XML. SRB dynamically generates SQL to perform various operations on the MCAT.

Storage Virtualization

The underlying technologies for storing data change every few years. SRB makes it transparent to its clients which particular storage technology underlies an SRB environment. For SRB Clients it is no longer necessary to remember a server’s hostname, IP address, operating system, access protocol, or even whether a system is an archival storage system or a cache system. SRB hides those complexities and alleviates data storage complications by defining a common set of operations for manipulating data through its drivers.

Data Virtualization

Data within an SRB Federation is virtualized through a mapping mechanism from the physical path to a logical name within SRB. Every Data Object in SRB has its unique Global Identifier that is independent from the physical data path. SRB associates any metadata such as access control lists, audit trails, checksums, or user-level metadata with the logical name (or Global Identifier). This enables SRB to function across heterogeneous and distributed storage systems.

Discovery

In the event that the logical name or pathname to a Data Object is unknown, data can be discovered and accessed solely through an object’s attributes (creation date, size, frequency of access, customer name, keywords, projects, etc.).

Access Virtualization

Single-sign-on and access through one common set of APIs provide for a complete access transparency. Clients do not require previous knowledge of a system’s APIs or security mechanisms.

Services

The MCAT and the Global Namespace enable the transitioning of services such as backup or migration from a local administrative domain into the enterprise-wide SRB Federation. The following paragraphs give an overview of the most important services implemented in SRB.

Data Load

There are two mechanisms to load data into SRB: registration and ingestion. Registration leaves the data untouched and only registers a pointer to the data in MCAT without transferring any of the data contents. Ingestion is similar to Registration but also transfers the data to an SRB Storage Resource.

Data Retrieval

Data is as easily retrieved from SRB as it is loaded. It is completely transparent where the data is coming from because SRB and TCP/IP handle the routing of the data streams.

Replication

Data replication – the process of creating and maintaining synchronized copies of a Data Object throughout an SRB Federation – has a number of uses and benefits: it enables disaster protection with instantaneous failover; caching is achieved by allowing clients to access a local replica rather than a remote one; and load balancing works by distributing synchronized replicas across multiple servers.

Persistent Migration

Using the Global Namespace as a logical view of federation-wide data, SRB Clients can migrate data to new Storage Resources without affecting the way users see data organized in their Collections.

Backup And Restore

Unlike Replication, a backup is not automatically overwritten when the original data is modified. This protects against accidental or malicious destruction of data. SRB creates copies of Data Objects or entire Collections in selected backup Collections. The restore process recovers deleted or wrongfully modified objects to an earlier point in time.

Secure Queries

SRB Objects such as Resources, Users, Data Objects, Collections, or Metadata Schemes can be queried through the MCAT. There are several hundred attributes that allow the creation of very complex SQL-style queries. At no point in time will SRB Clients gain access to any object/information for which they do not have at least ‘read’ access.

Security

SRB provides authentication schemes to protect data (Challenge Response Mechanism, Grid Security Infrastructure - GSI, Kerberos) enabling Single sign-on throughout the whole organization. SRB’s authorization service maintains Access Control Lists (ACL’s) on various SRB Objects: Data Objects, Collections, Resources, and Metadata Attributes. Support is also built in for data encryption (GSI, Kerberos), User management (users, groups, domains) as well as Audit trails.

Authentication

There are a number of built-in mechanisms that ensure the authenticity of an SRB User: challenge response mechanism (password is not sent across the network), Grid Security Infrastructure (using public and private keys), and Kerberos (sophisticated ticket management mechanism to provide single sign-on for multiple networked services). When using the CIFS and NFS Gateways, administrators are free to set up any supported authentication mechanism for their end-users, such as the integration with Microsoft Active Directory (AD) or Lightweight Directory Access Protocol (LDAP).

Authorization

Once users are authenticated to ensure that they really are who they claim to be, SRB performs authorization services on every SRB Object requested. This ensures that users gain access only to the appropriate objects. The level of access may vary depending on the Access Control List (ACL).

Users / Groups / Domains

Users within an SRB Federation have assigned roles and only one *Super User* has the power to do and see everything. Others can be designated to manage a subset of the Federation and users can be assigned to manage Data Collections or to be data readers.

In order to ease administration of an SRB Federation, administrators can create domains and groups of users. Access control lists can then be created referencing SRB Domains and Groups.

Access Control Lists

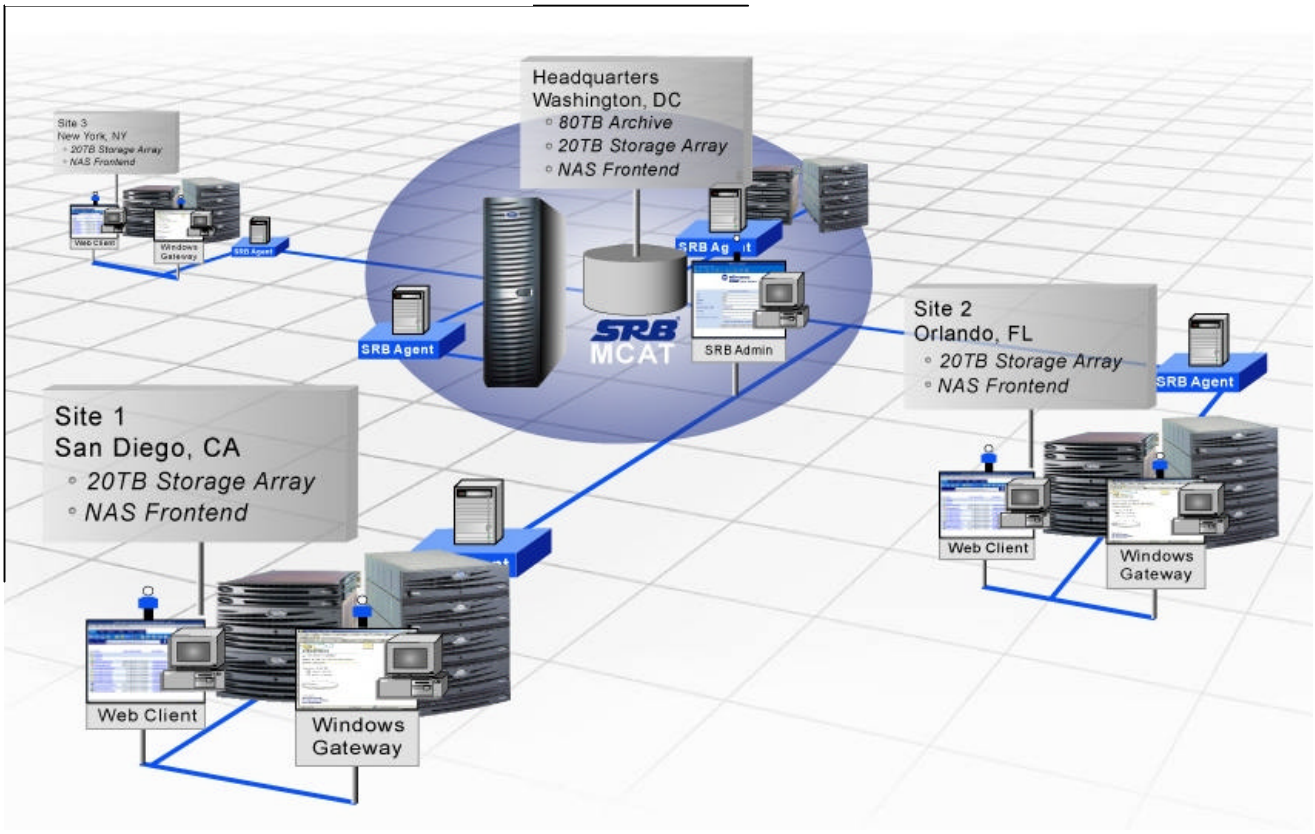
Access is controlled (authorization service) on Data Objects, Collections, Resources, and Metadata Attributes. There are various levels of access for each object ranging from ‘null’ (no access) to ‘all’ (full access). A fine-grained security mechanism enables access control on a per-user-level. In order to ease user and access control

management, access can also be controlled for entire SRB Groups or Domains.

Audit Trails

Every transaction within an SRB Federation can be audited. An audit trail will typically contain information such as date of transaction, success or error code, user performing transaction, type of transaction, and notes. Audit trails, like everything else stored inside MCAT, can be easily queried and filtered.

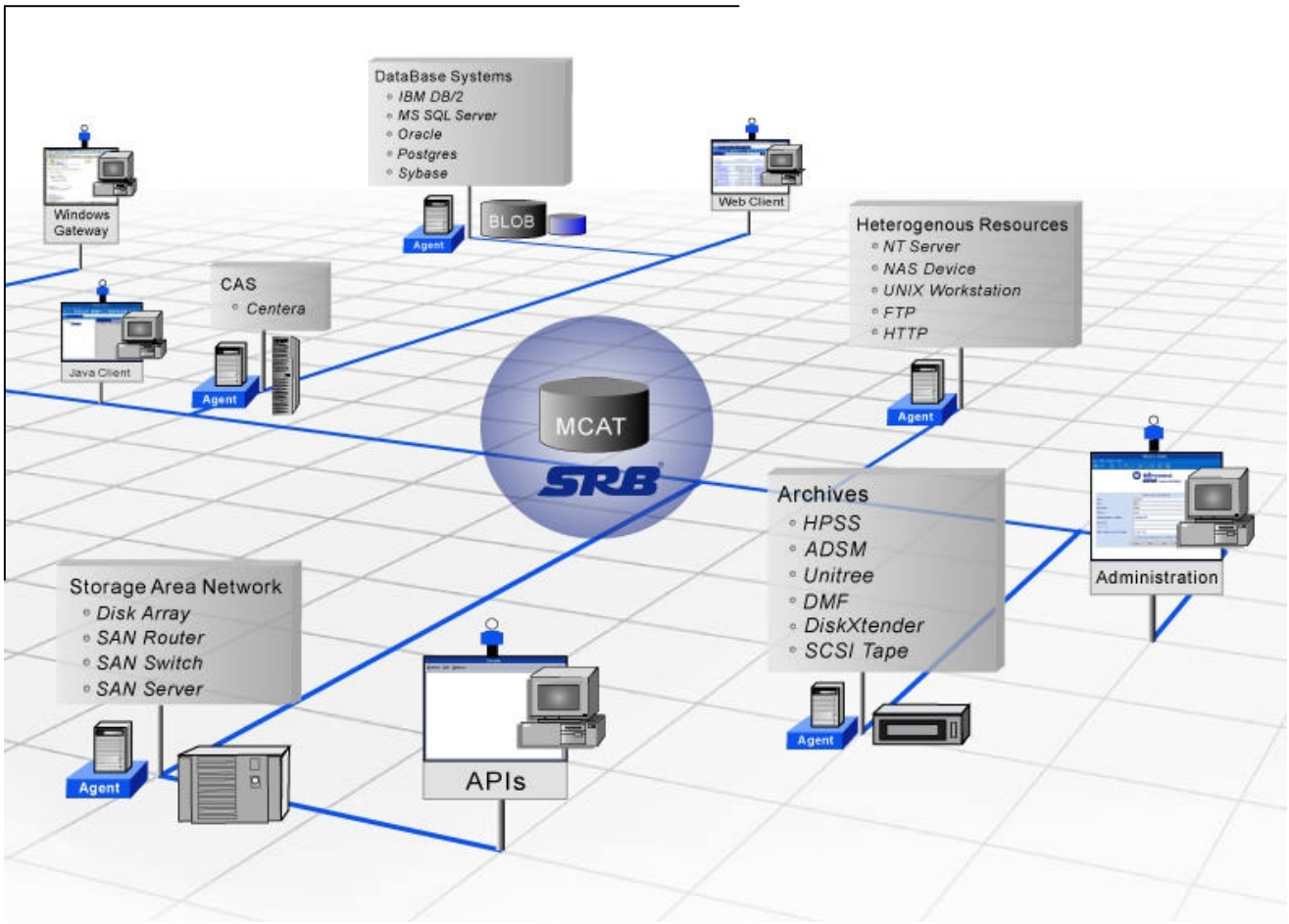
Figure 4: SRB Production Prototype



Expanding into the Enterprise

An organization evaluating data grid and collaborative networking alternatives may face a large and very complex migration to an eventual global solution, which be intimidating. But SRB users have found the evaluation, test and implementation of the product to be startling straightforward and conveniently incremental. A typical evaluation for a qualified organization can be arranged in a few days, with installation, including the necessary training, typically taking 2-3 days following the software download. A 60-day evaluation for the software is permitted, and there may be some consulting and additional training during this period, depending on the experience level of staff. If the capabilities of SRB and the business requirements are deemed to be a match as a result of this exercise, the next step can be a production prototype, where a subset of the projected data federation can be put into production, testing everything about the collaborative data sharing strategy except eventual scale. Starting with a production prototype, as illustrated above in Figure 4, SRB can be gradually expanded into an enterprise-wide data grid. Figure 5: SRB Enterprise Implementation illustrates the capabilities of the product in federating the most complex of data environments. Capacity can be increased, more resource types and drivers can be added, additional servers (or Agents) can be attached, data management processes can be automated, and performance can be scaled. At every step, SRB users pay only for what they need. In taking this next step – implementing an enterprise-wide, or even cross-enterprise SRB Federation, the IT organization can be assured that SRB is functionally a match for the collaborative data needs of the organization, and that the product's scalability has been proved in the most demanding compute environments over nine years.

Figure 5: SRB Enterprise Implementation



Data Collaboration and Management Suite

SRB is an advanced data grid implementation that enables simplified information management in ad hoc federations of users. Not only does it enable cross-platform access to data, but it also facilitates the discovery of the information residing on very different, usually incompatible, platforms crossing administrative domains. SRB is a comprehensive solution for managing all the information in an organization. This includes table databases, LOBJ (Large Object) databases, file systems, tape drives, and archives. To allow viewing of table databases together with typical documents (spreadsheets, letters, forms), SRB can display customizable reports dynamically generated from database queries.

Performance

Numerous performance advantages include prefetch for Containers, support for a network of distributed caches, support for multiple archive resources in Containers, and Collection-based (hierarchical) management of distributed digital objects. Parallel I/O and bulk operations are also supported by SRB.

Parallel I/O

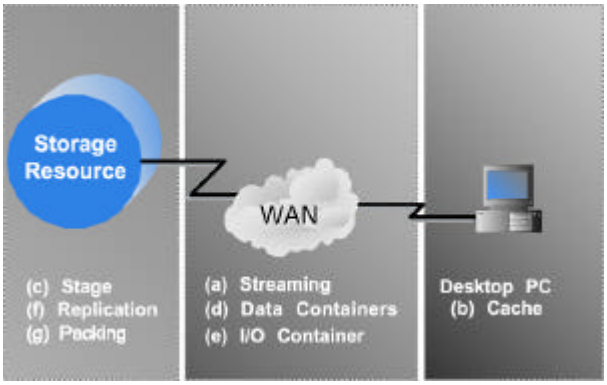
The SRB communication protocol is a thin protocol implemented on top of TCP/IP. This protocol supports parallel communications using multiple sockets and a single static port initiated by an SRB Client. This greatly enhances the utilization of bandwidth, especially when sending larger files across Wide Area Networks (WANs). SRB's parallel I/O mechanism can be configured to use a single static port through a firewall since the request is initiated by the client. Because the SRB Server (or rather its administrator) has a better understanding of the server's bandwidth availability and speed of the attached storage system, the server controls the

number of parallel I/O streams. In SRB's parallel I/O mechanism not only Data Objects are transferred using the parallel mechanism but also any other data (like a large query result) is transported in parallel.

Latency Minimization

SRB features many mechanisms to reduce latencies that occur during any data transfer:

Figure 6: Latency Management



(a) *Streaming…*is the continuous transfer and receipt of data without any noticeable lag time.
(b) *Caching…*keeps local copies of frequently accessed data on fast storage systems or memory.
(c) *Staging…*temporarily writes Data Objects stored on archival storage media to disk for faster access.
(d) *Data Containers…*transparently aggregate multiple Data Objects into one large file that can be stored and transferred more efficiently than multiple smaller files.
(e) *Input/Output Containers*…allow for a remote execution of multiple commands in large batches.
(f) *Replication*…keeps multiple synchronized copies of Data Objects at different sites.
(g) *Packing*…aggregates multiple Data Objects into a single buffer before sending the files over the network reducing latencies associated with multiple requests.
Furthermore, SRB can pre-spawn processes on the server. The administrator can determine the size of the 'SRB Server pool'. Performance is improved by reducing latency during the initial connect to SRB.

Scalability

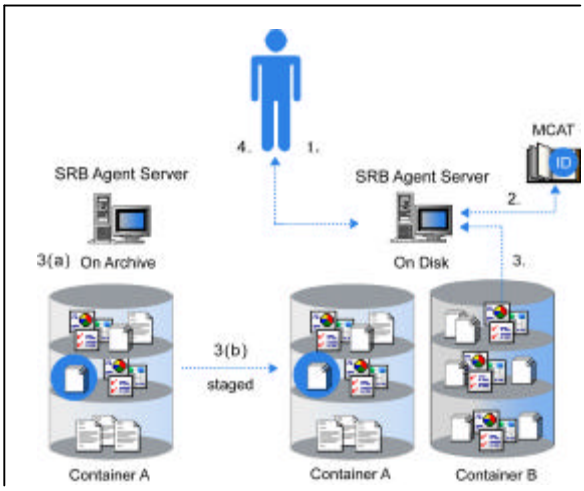
SRB's architecture – comprised of Clients, Agents, and the MCAT database – allows for nearly infinite scalability by adding more Agents as more capacity is needed and scaling the MCAT database for higher transaction rates using the database vendor's mechanisms (like Oracle Real Application Clusters).

Containers

Three major problems typically occur with distributed archival storage systems. The first is a high latency rate when retrieving data from tape. The second is file storage limitations. Archives are designed to store small numbers of very large files and data Collections are more likely to contain large numbers of small files. Finally, if the data is distributed over a Wide Area Network (WAN), serious latencies can result when transferring a large number of files. Through its patented Container technology, SRB can overcome all of these obstacles.

SRB manages Containers so that archives can be transparently integrated with all other enterprise-wide data. SRB incorporates Containers so it can manage distributed Data Collections across multiple storage systems.

Figure 7: SRB Container Design



Additionally, SRB manages Containers transparent to end-users or applications by presenting in-Container Objects just like any other SRB Data Object in the Global Namespace.

Data Objects are physically packed into a Container before being stored into an archive. This assures that associated Data Objects will reside on the same tape while decreasing the number of Data Objects managed by the archive. When retrieving these Data Objects at a later time, the correct Container is automatically staged to disk and all the read access happens against the staged copy. Since related Data Objects are retrieved together, access time can improve dramatically.

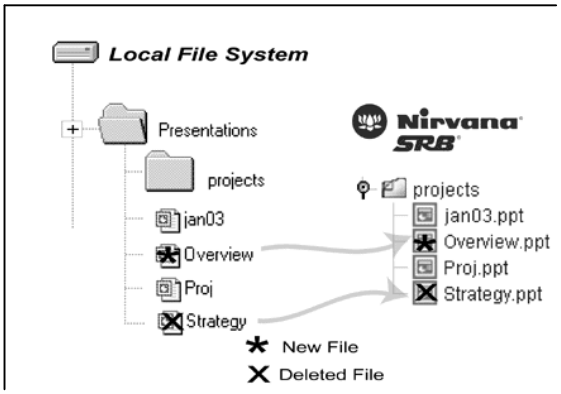
Automation

SRB has the ability to automatically launch helper processes or daemons upon start-up. Two of those daemons are particularly useful when it comes to easing the data transfer into the SRB Federation or automating data management.

Directory Synchronization Daemon

As files are added to, or deleted from, local directories, the *Directory Synchronization* (DS) Daemon running in the background detects these activities. When changes are noted to the local directory structure, the new information is registered and updated in the MCAT. This ensures that even data outside the control of SRB, which is modified by third party applications, can be brought into the SRB Global Namespace without affecting such third party applications.

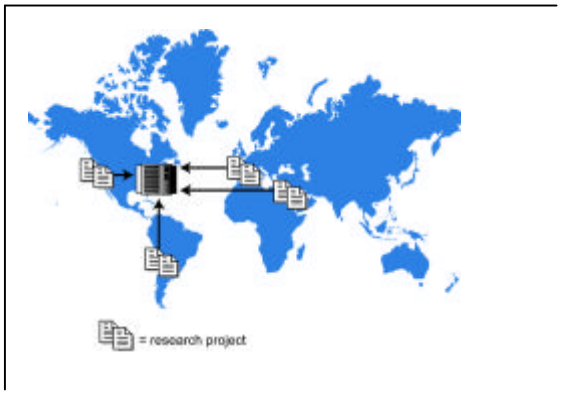
Figure 8: DS Daemon



Hierarchical Storage Management Daemon

A *Hierarchical Storage Management* (HSM) Daemon is a policy engine that routinely queries the MCAT. For example, an administrator sets a policy to migrate data from distributed Locations to one particular archival storage system with the criterion that all data over 10MB be included. In another example, data associated with a specific project and that has been accessed fewer than five times on average per day would be migrated. Besides migration, other actions can be performed on the data including replication, deletion, backup, or simply reporting.
The policies are extremely flexible and can contain standard metadata, customized metadata or expressions. All this happens behind the scenes, transparent to the end-user or applications, so that the Global Namespace does not change.

Figure 9: HSM Daemon



Additional Interfaces

The following client interfaces exist and can be added to SRB as an option: Web Client, Scommands (command-line client), Acommands (command-line administration), and Software Development Kit (SDK).

The SRB Web Client runs on any system that supports Internet Explorer or Netscape Navigator browsers. The SRB Web Client is therefore accessible from anywhere on the internet and does not require any client-side software installation.

In order to enable automation of certain tasks and to write scripts, a command line-based interface (Scommands) is also available. The Scommands essentially extend the functionality of local UNIX commands ('ls', 'cd', 'pwd', etc.) into the Global Namespace and are especially useful for terminal sessions or scripting.

Command line-based administration is optionally available through the Acommands. They are used for administrative tasks that require bulk operations or scripting.

The SDK allows application developers to integrate the SRB Global Namespace and all of SRB's services and functionality into their own applications. The SDK is available in both C and Java and comes with detailed documentation and examples.

Additional Drivers

A parallel disk driver is available that extends the standard SRB disk driver with extra performance using parallel threads and various caching mechanisms optimized for both random and sequential data access.

SRB includes relational database resources to permit DBMS-style row-based access to data stored in databases. SRB includes an HTML, XML or customized template interface to databases so that queries can be returned in any format and

records within one format can easily be transferred into another format.
Most relational databases can also be used to store binary large objects (LOBJs). SRB supports drivers for such large object databases in order to store Data Objects directly into database tables.

A driver to SCSI tape drives and tape libraries is also natively supported by SRB. Since tape storage is still relatively more cost effective compared to disk, SRB makes tape management also very straightforward and therefore reduces Total Cost of Ownership (TCO) for tape archives significantly.

Internet data sources can be seamlessly integrated with drivers for FTP and HTTP servers.

Tickets

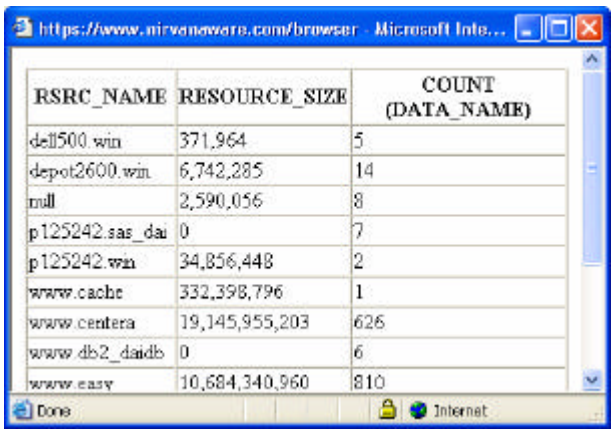
SRB employs an additional authentication mechanism whereby data-sharing Tickets can be sent out to internal or external SRB Users. The ticket then grants controlled access to Data Objects or entire Collections. Additional restrictions such as time limits and limits on the number of accesses can be built into every Ticket.

Reports

In combination with relational database drivers, the MCAT can be used to report on various SRB Objects. Examples are:

- Resource Usage (over time)
- Data Object Access Pattern
- Data Ownership report (storage by user)
- Valid Tickets
- Audit Trails (for certain Data Objects or entire Collections)

Figure 10: Resource Usage Report



Conclusion

In complex, distributed environments, SRB enables a more powerful use of valuable data. It brings ease of use, data persistence, improved integration, enabled collaboration, true scalability, better administrative function, and integrated security benefits to data storage use and management. SRB can accelerate high-value research and development projects, speed organizational integration, and span disparate islands of high-value enterprise data. SRB's simple and intuitive interfaces permit the easy management of different systems throughout a department or global organization and in the public sector, at the community, state, or federal agency level, reducing complexity to simplicity in both ease-of-use as well as administration.

Nirvana Storage developed SRB in cooperation with the San Diego Supercomputer Center, and the product is currently operational in over 100 governmental, scientific, and research data federations, where it is used to support collaborative projects (over 400TB and 50 Million files) among networked institutions. Some of this SRB experience is described in papers entitled "The SDSC Storage Resource Broker," and "SRB Case Studies", both of which are available on request. SRB is ideally suited to complex computing environments where knowledge workers would benefit from simplified, easy access to shared, high-value data in heterogeneous working environments.

About Nirvana Storage

Nirvana Storage is a division of General Atomics, the founder of the San Diego Supercomputer Center (SDSC). Nirvana Storage specializes in advanced information management systems for complex storage environments.

About General Atomics

General Atomics (GA) and its affiliated companies comprise one of the world's leading resources for high-technology systems development. General Atomics is a leader in areas such as nuclear fission and fusion or Unmanned Aerial Vehicles (Predator). Founded in 1955, GA is based in San Diego, CA, and has a total employment of approximately 4,000.

Nirvana Storage
3550 General Atomics Ct
San Diego, CA 92121-1122
T: (858) 455-2500
F: (858) 455-2529
E: srb@nirvanastorage.com
W: www.nirvanastorage.com