

# BaBar Experiment Data Distribution using the Storage Resource Broker

Wilko Kroeger<sup>1</sup>, Jean-Yves Nief<sup>2</sup>, Adil Hasan<sup>3</sup>

1. SLAC Computing Services, California, USA.

2. CC-in2p3, Lyon, France.

3. CCLRC e-Science Data Management Group, RAL, UK.

## Abstract

This paper describes the layout and experience in distributing large volumes of data from SLAC to CC-in2p3 and RAL. We describe the experience using the Storage Resource Broker to distribute more than 120TB of data from SLAC to CC-in2p3 and the experience in using a Grid tool in a production environment. We also describe the plans for extending the system to distribute BaBar data to RAL using the SRB making use of the new features of federation.

## 1. Introduction

The BaBar experiment [1] is dedicated to high precision measurements at B-meson energies. The experiment has collected more than 2 billion events of interest since 1999. The experiment comprises collaborators geographically distributed in Europe, Canada and the USA. From early on in the experiment it was considered important to distribute the experimental data to collaborating computing centres in order to maximise the computing potential available to the experiment, as well as providing easy local access to the data.

To date, BaBar data has been distributed from the experiment at SLAC, USA to computing centres in France, the UK, Italy and Germany. With France, the UK and the US providing unrestricted collaboration-wide access to the data for analysis whilst Germany provides access through applications submit on the Grid.

All computing centres hold partially overlapping data-sets with the BaBar experiment computing centre at SLAC holding the entire data-set and the CC-in2p3 computing centre in Lyon, France holding the second largest data-set.

In this paper we start with a description of the Storage Resource Broker [2] followed by a description of how the software was applied to the BaBar experiment to distribute data between two of the largest BaBar collaboration data centres (SLAC and CC-in2p3). We then describe the experience gained in distributing data from SLAC to CC-in2p3 using the SRB for more than a year and follow with a description of future plans for the SRB within the BaBar experiment.

## 2. The Storage Resource Broker

The Storage Resource Broker (SRB) [2] is developed and distributed by the San Diego Supercomputing centre and is a client-server middleware software product that provides uniform access to a heterogeneous set of data resources distributed across a network. It consists of three components, a Metadata Catalogue (MCAT), SRB servers that provide access to the storage resources, and SRB clients.

A simple example illustrates the working of the SRB. A user wishes to extract a file, the user knows the logical file name (which should be a name meaningful to the user and does not necessarily have to correspond exactly to the physical file name) and knows the SRB system that holds the data.

The user only requires a configuration file with information on the SRB system of interest and the client SRB applications in order to connect to an SRB server that is part of the SRB system. The server contacts the MCAT in order to find the physical location of the requested file and redirects the client to an SRB server that can provide access to the resource that holds the file. This SRB server then reads the file and transfers the file to the client. Server to server, in addition to server to client, data transfers are also possible.

The MCAT is implemented in relational database technology (Oracle, Postgres, DB2 etc) and is used to store the logical-to-physical file mapping as well as other metadata associated with the files (such as file size, check sum, etc), users (such as username, group, etc) and logical-to-physical resources managed by the SRB (such as unix file system, a tape resource

etc). For each file there exists metadata information on the logical name of the file, the physical name of the file, the resource within which the file resides, the users who can access the file and more metadata information related to the file itself (such as checksum, size, replicas, etc). The logical name allows the user to build up a hierarchy of collections or groupings of files meaningful to the project.

The SRB provides access control through username and password which are stored in configuration files accessible by the client or server. The password can be stored in clear-text, or encrypted (in v3.3.1 and onwards). However, it's worth mentioning that communication between client and server, or server and server automatically encrypts the password providing a level of security even in the case of clear-text passwords.

The SRB also provides support for access with X509 certificates using the Grid Security Infrastructure (GSI). The SRB provides SRB system-level, group-level and SRB domain-level administration providing different levels of administration within an SRB system.

Groups of users can be created within the SRB to permit, or exclude, users from accessing certain collections. Restricted access can be granted to users that are not registered within the SRB by means of *tickets*. Where a registered user can create a ticket to allow an unregistered user access to all or a subset of the data the registered user has access to for a limited period of time. The ticket issuer can restrict access permissions and also delete the ticket before its' lifetime has expired.

The SRB comes with a set of command line utilities (called S-commands). These tools provide UNIX like directory commands (Sls, Smkdir, Scat), programs to access and modify metadata of files stored in the SRB and programs to extract a file from the SRB or import a file into the SRB. The SRB also provides an API implemented in the C, Python and Java languages that allows users to write customized tools.

For production systems the file transfer using multiple streams and the bulk-mode features are also of great interest. In the case of large files the ability to split the file into multiple parallel streams can improve transfer rates by making more efficient usage of the network. The bulk-mode features are useful when transferring large numbers of small files where, in the case of import and export, the small files are concatenate into a large file and the large file is transferred reducing the connect/disconnect overhead. In the case of

registering metadata information in the MCAT the bulk-mode tends to open one connection to the database, perform all the operations and then close the connection thus reducing the number of database connections.

The latest version of SRB (v3) introduced the federation of multiple SRB systems. An individual SRB system, called a SRB *zone*, consists of a single MCAT and a set of SRB servers and resources that use that MCAT.

Federation means that multiple zones can interact with each other and allows a user to access files across these zones. This allows two sites to maintain their own zones, but from the users perspective it appears as if there were just one SRB system.

### 3. The SRB in BaBar

The prime motivation for looking at the SRB was to find a tool that could provide a sufficiently generic solution that would be able to efficiently distribute data from SLAC to CC-in2p3 with minimal maintenance and management. We chose the computing centres SLAC and CC-in2p3 as there was some interest within those two centres in using Grid tools to distribute data and also because the large quantity of data distributed between SLAC and CC-in2p3 would provide the best test of the system.

We chose the SRB for a number of reasons: (i) maturity of the product, the SRB has been used for a number of years; (ii) access to the SRB developers, through the bugzilla system new features and bugs are relayed to the developers providing an acceptable turnaround from report to implementation; (iii) a large user-base, there are more than five large-scale projects making use of the SRB which provides a steady stream of feedback and new feature requests to the SRB team.

The BaBar experiment possesses its' own bookkeeping system that holds metadata information on the data files. The metadata comprises of processing information as well as information on the contents of the file, all of which is necessary in order for the physicists to analyse the data contained within the file.

New data produced by the experiment were stored into the High Performance Storage System (HPSS) tapes. A daily cron job obtained the list of newly produced data files from the bookkeeping system and registered in the MCAT the new files recording the logical file name as recorded in the bookkeeping system, the type of data file and the universally unique identifier (uuid) used to identify the file. The

MCAT provides a number of pre-defined empty attributes that can be used to hold user-defined metadata. These pre-defined attributes were used to hold the uuid and file check sum.

The advantage of using the same logical file name as used in the BaBar bookkeeping system is to make integration between the two systems simpler. At the moment a user can query the bookkeeping system, extract a list of files of interest and pass that list of logical file names to the SRB to export the data.

The information stored in the MCAT allowed the following queries that were found to be important as far as data distribution was concerned, to be made:

- Find all files added since a certain date.
- Find all files belonging to a specific logical group (or collection).
- Find all files stored on a particular physical resource.
- Find a file with a given uuid.

The native MCAT metadata allowed the first three queries to be made and the user-defined MCAT metadata allowed the fourth to be made.

For the set of data collected by the BaBar experiment up to 2004 the system that was used to distribute data from SLAC to CC-in2p3 since late 2003 is shown in figure 1. We made use of the SRB release 3.0.1, which contained bulk-mode operations that we considered necessary to efficiently manage and distribute the large numbers of files (typically thousands) regularly produced by the experiment.

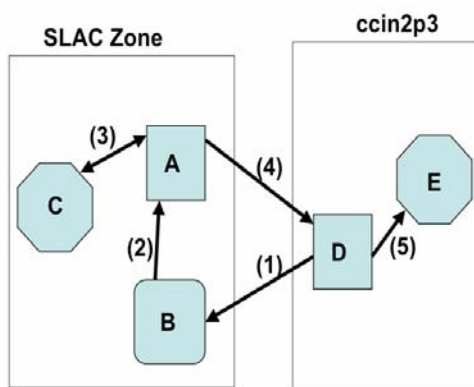


Figure 1: Schematic of the system for distributing BaBar 2002-2004 data.

The SRB system comprised only the SLAC machines A, B (from figure 1) with server B being MCAT enabled and server A being an SRB server able to access the tape storage system C. The ccin2p3 server D consisted of SRB client applications. The numbered arrows

show the typical flow of data. A request for the most recently registered data is made from CC-in2p3 to the MCAT enabled server (B in figure 1). This is followed by the MCAT enabled server re-directing the request to a server holding the data (or capable of accessing the data from the HPSS tape store C), in this case A in figure 1. The data is then transferred from A to D at CC-in2p3 and archived to tape (E in figure 1) after validation checks were performed on the transferred data (ie check sums calculated at CC-in2p3 and compared with the values stored in the SRB).

In reality six servers were used for the export and import of data at SLAC and two servers were used at CC-in2p3 to import the data. The SRB client need not connect to the SRB MCAT enabled server, but can connect to any SRB server in which case the SRB server would redirect lookup request to the MCAT enabled server adding a small amount of extra communication (which may not be noticeable if a lightly loaded SRB server is selected).

The BaBar data sample collected up to 2004 resulted in more than 120TB of data, corresponding to approximately 200,000 files being distributed from SLAC to CC-in2p3. The peak tape-to-tape transfer rate from SLAC to CC-in2p3 was around 3TB/day (on average the tape-to-tape transfer rate was around 2TB/day). Such high rates were readily achievable due to the SRB providing drivers for the HPSS tape system and the bulk-mode operations reducing the connect/disconnect overhead. In addition the implementation of multiple parallel SRB requests in the python scripts created by CC-in2p3 also added to the improved transfer performance.

## 4. Observations from the First BaBar SRB Production system

The experience of running a production distribution system using the SRB resulted in a number of observations that we believe are sufficiently general and may be of interest to other projects embarking on distributing large volumes of data regularly.

### 4.1 Workflow and Automation

In many cases the task of large-scale data distribution tends to be a well-defined process that happens at regular intervals. We found it important to identify the tasks involved in the successful distribution of data. The identification of the workflow for data distribution simplified the process of creating

scripts to automate the data distribution process. Through the course of the production cycle we identified a number of tasks that we list below.

- Identification of files to transfer.
- Identification of physical location of files.
- Identification of total size of files to transfer.
- Reservation of space at target site (enough to allow the transfer to take place which means that the reserved space does not need to equal or exceed the total size of files to be transferred).
- Reservation of space at source site, again enough to allow the transfers to take place.
- Preparation of files for transfer (ie staging from tape to disk, or tarring a set of files together, etc).
- Transfer of files.
- Validation of transferred files (ie check sums or any validation checks to ensure the files are usable).
- Storage of transferred files.
- Removal of successfully stored files from the source site.

Some of these tasks were performed by one SRB operation, for example the determination of files to copy automatically gave the total size of the sample and the SRB operation to transfer the data automatically found physical location of the files.

It is worth pointing out that an important ingredient in the automation scripts was robustness, such that the scripts could handle problems with outages or system failures gracefully.

Validation was another key component in the automated scripts. Although the SRB contained its own validation procedure for the transferred files we needed to put in place our own validation procedures as we considered data distribution to include the preparation of the file for transfer right up to the archiving of the file at the target site.

To this end, we stored the file check sums, generated at the time the file was produced, in the user-defined metadata attributes in the MCAT. We performed a checksum before the file was transferred and after the file was archived at the target site.

Another point worth mentioning is the evolutionary nature of the identification of workflow tasks as only during the production did we encounter problems or new requirements that had not been apparent during testing. So, any scripts to automate the process need to be modular enough to allow extensibility.

Identification of the workflow for data distribution also has the advantage of providing a template for other users of the system that may not wish to, or cannot, use the scripts developed by another site.

## 4.2 Monitoring and Trouble-shooting

Although we had a small manageable number of servers as part of the SRB data distribution system we found that the implementation of monitoring provided a number of benefits.

- Allowed problems with servers to be more quickly identified.
- Allowed identification of bottlenecks (ie if one server were heavily loaded).

For the BaBar system we had rudimentary monitoring of the SLAC server in place (cpu utilisation, disk space, hpss system) which eased trouble-shooting, but we could clearly see that a more integrated system would allow simpler trouble shooting and would make future production planning simpler. To this end audit information on the number of files transferred as a function of time, user and location would also help in the planning of new or upgrade systems.

We also developed a number of scripts to aid in troubleshooting. These simple scripts were able to transfer a simple file with a lot of verbose output in order to track down file-transfer problems. We also started to develop test applications capable of running a battery of tests against an SRB server in order to track down problems with a server.

Of the small number of problems that we observed during production one of the most common was a restart of the database back-end causing a connection with the SRB server to be broken resulting in an un-responsive SRB. In principle, the newest version of SRB (3.3.1) should get around this problem by closing unused connections with the database.

Even with the limited amount of monitoring of SRB servers that we had implemented at SLAC we noticed that monitoring plays a key role in troubleshooting applications.

## 5. Future Plans

For the 2005 production of BaBar data we aim to expand the SRB data distribution system building on the experience already gained.

We plan to implement an SRB system at CC-in2p3 and also an SRB system at RAL for BaBar (making use of the Data Management Group SRB services). We plan to make use of zone feature of SRB to create three separate

zones for each computing centre and federate them. We have already carried out many tests using zones at CC-in2p3 and SLAC.

The SRB zones will allow a user at another site to be able to copy data seamlessly from any one of the three computing centres. It also provides a backup for the SLAC computing centre allowing SLAC to easily copy data from either RAL or CC-in2p3 if any of the original data gets lost (through tape breakage) or corrupt. Federation will be achieved by registering all zones within each SRB MCAT and registering all publicly available files in all MCATs using the existing SRB federation synchronisation tools. Figure 2 shows a schematic of the three SRB zones.

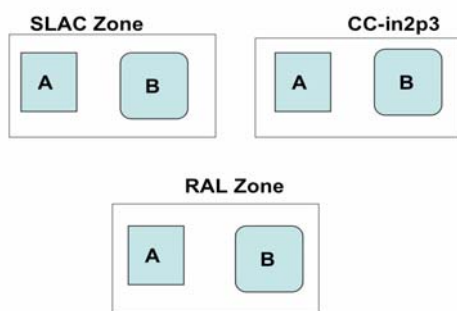


Figure 2. Schematic drawing depicting the three SRB zones. Server A corresponds to a plain SRB server and server B to an MCAT-enabled SRB server.

Each zone has its own metadata catalogue and its own set of SRB servers (A and B in figure 2). SRB federation will allow each site the autonomy to manage the data at its own site as it sees fit.

We also plan to implement greater centralised monitoring and auditing of the SRB systems to facilitate troubleshooting and planning. In most cases this requires integration of existing monitoring systems into a uniform framework with the information readily accessible by each site. We also plan to continue the development of testing tools that test the functionality of the SRB, these tools will also be of considerable help when troubleshooting.

At SLAC, where a majority of the data is export from, we plan to make use of the SRB compound resources feature where a logical resource comprising of a tape resource and more than one disk resource can be created. This will allow load balancing (initially in a simple round-robin manner) over the import export servers and will make the inclusion of new servers simpler. We may also introduce

compound resources at the other computing centres as well.

Finally, we plan to improve the integration between the BaBar bookkeeping system and the SRB to allow more seamless access to the experiment specific metadata. This would in principle improve the speed of data transfers. Currently, the two systems are integrated only at the script level. The SRB provides a database access interface to allow SQL queries to a different database to be stored and invoked from within the SRB.

## 6. Summary

The BaBar experiment has used the Storage Resource Broker in production for more than two years to distribute more than 200,000 files of BaBar data from SLAC to CC-in2p3. This note describes the set-up used to distribute the BaBar data. We also describe the production experience, some of which is relevant to other systems, and we describe the future plans to increase the number SRB systems potentially allowing users the ability to copy data seamlessly from any of the contributing computing centres as well as providing a more accessible backup system for the SLAC data store.

- [1] **The BaBar Detector Nucl. Instrum. Methods. A479 (2002) pp1-116**
- [2] **The Storage Resource Broker**  
<http://www.sdsc.edu/srb>