SRB SYSTEM AT BELLE/KEK

Y. Iida, I. Adachi, N. Katayama, S. Kawabata, A. Manabe, T. Sasaki, S. Y Suzuki, S. Yashiro, Y. Watase, KEK, Tsukuba, Japan
S. Honma, H. Kuraishi, T. Nakajima, Fujitsu, Tsukuba, Japan
K. Ishikawa[3], ISE, Chiba, Japan
Ma Mei, IHEP, Beijing, China

Abstract

The Belle experiment has accumulated more than 1PB of raw and processed data stored on tape with the rate of 1TB/day. The processed, compactified data, together with Monte Carlo simulation data for the final physics analysis amounts to more than 100TB. The Belle collaboration consists of more than 55 institutes in 14 countries and at most of the collaboration institutions, active physics data analysis programs are being undertaken. To meet their storage and data distribution demands, we have tried to adopt a Storage Resource Broker, SRB. We have installed the SRB system at KEK, Australia, and other collaborating institutions and have started to share data. In this talk, experiences with the SRB system is discussed and the performance of the system when used for data processing and physics analysis of the Belle experiment is demonstrated.

INTRODUCTION

The processed data of a huge quantity generated by the Belle experiment is stored on the storage at the Belle computing system in KEK which is a very secure network. The Belle collaboration which consists of about 400 members at 57 institutes in 13 countries, need to access the storage via many login machines, in order to get the data. Since the data analysed by each collaboration institutes is stored their storage, the management of Belle data is very complex. So they want to make a better file sharing environment.

The Storage Resource Broker (SRB) from San Diego Supercomputer Center (SDSC) provides a uniform interface for connecting to various types of data storage over a network and accessing replicated data sets. SRB, in conjunction with the Metadata Catalog (MCAT), provides a way to access data sets and resources based on their attributes rather than their names or physical locations. And we can use it as global file system. It is because we can always see as the same file system even if we use various OS, even if we connect from somewhere. It provides parallel I/O (multiple threads each sending a data stream on the network) for transferring large files, and Container (a lot of small files into one larger file) and/or bulk load and bulk unload for small files.

Before applying to the Belle experiment, we have measured the performance and tested basic functionality of SRB on an independent test system.

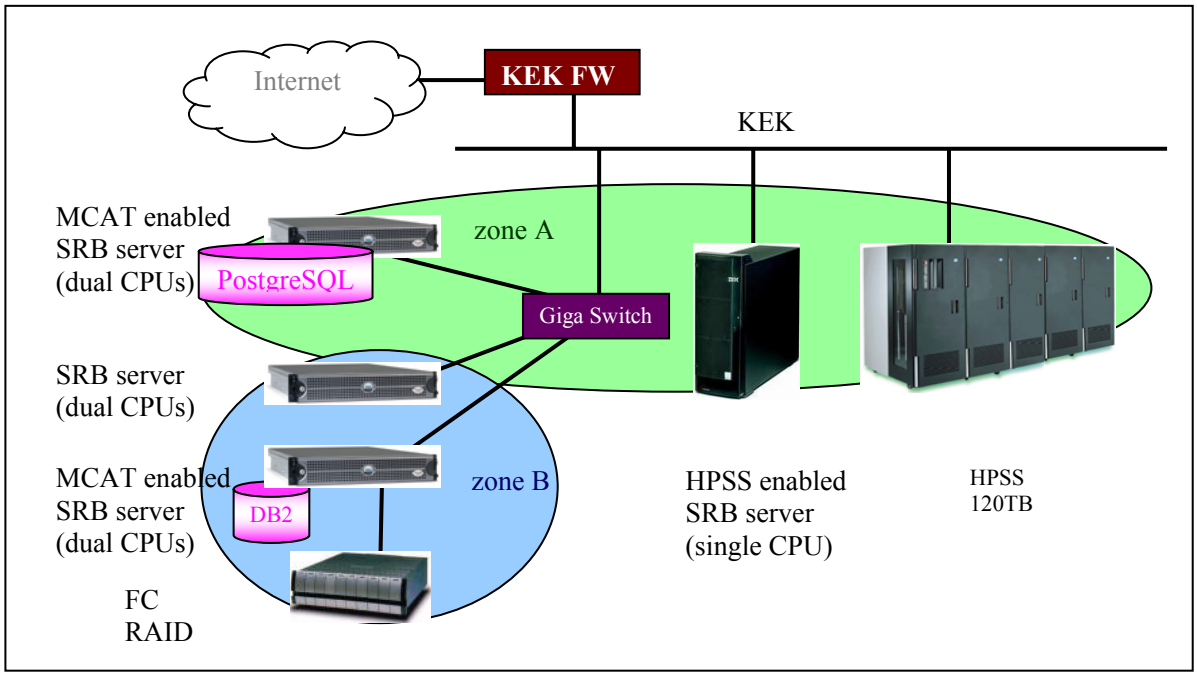

Figure 1: test environment

BASIC PERFORMANCE MEASUREMENT

Test environment

SRB supports the federation of MCAT to enable access to resources and data across zones, while a zone consists of one or more SRB servers along with one MCAT-enabled server

We built two zones to test the federation of MCAT in out test system. Zone A consists of a MCAT SRB server with Postgres and also a HPSS enabled SRB server. The other zone, Zone B, consists of a MCAT SRB server with DB2 and a SRB server with the Fibre Channel RAID as shown in Figure 1. For all servers, SRB version 3.1 has been installed. In Zone A, the HPSS system and the HPSS enabled SRB server are connected by Gigabit Ethernet with KEK LAN which is always congested, while the servers in Zone B is connect to the same Giga bits Ethernet switch. Using this system, performance measurements for basic file transfer commands in SRB are carried out and the results are compared with ftp or pftp as a reference. Here, pftp is the command provided by IBM as a part of HPSS and supports parallel transfer.

We decided to measure the performance for a large single file case and also smaller files so that following two file sets are used for this performance study.

- mixed files: 68files, 928MB in total, from 4.7KB to 101MB for each file
- larger file: single file in 1GB size

For larger file case, parallel I/O mode in SRB is also tested, while "Containers" and/or "Bulk load" options are tested for transferring small files. A "Container" is a way to put together a lot of files into one large file to improve performance. And "Bulk load" is designed to improve the efficiency of ingesting a large number of small files by registering up to several hundreds files with MCAT with a single call instead of the normal mode of registering one file at a time, also separate threads for registration and data transfer are used. The detail of these two options is well described in the SRB manual.

SRB also supports parallel I/O and four TCP/IP ports are used simultaneously as a default and this number can be changed with the command option. Here we use the default, four, for this option implicitly.

Transfer mixed files

We measure the performance of file transfer from local disc to HPSS (Zone A) and to remote local disk (Zone B).

The measurements were done with following 4 types of transfer: "Bulk load" with "Container" (Sbload –c), "Bulk load" (Sbload), "Container" (Sput –c) and ftp/pftp. The measured results are shown in Table 1.

Table 1: Results of transfer mixed files

Zone	Commands	Avg_rate (MB/sec)
Zone B	Sbload –c	30
	Sbload	12
	Sput –c	14
	ftp	36
Zone A	Sbload –c	11
	Sbload	3
	Sput –c	14
	Pftp	14

The results show that the "Sbload -c" case gives the best performance among the SRB commands in Zone B. It is almost comparable with one with ftp even though overhead of MCAT handling is expected when data is stored on a usual UNIX file system.

In the HPSS case, the "Sput –c" case gives the best result and "Bulk load" are not efficient as the UNIX file system case. This is due to the characteristic of HPSS which is designed for storage of larger files. The performance for the "Sbload" case looks worse. Also even though we attempted many trials, the measurements were very fragile. This is due to the HPSS system which used for this measurement is the busy production system and many users are accessing it. We need further investigation on HPSS case in the future.

Transfer larger file

We also measured the performance for the single larger file transfer case. In this case, we measured the performance for two cases, UNIX file system and HPSS as the storage, with the Sput command with single stream transfer and also parallel stream transfer mode. The results are compared with ftp for UNIX file system case, and pftp for HPSS. The results are shown in Table 2.

Table 2: Results of transfer larger file

Zone	Commands	Avg_rate (MB/sec)
Zone B	Sput –m	30
	Sput	23
	ftp	34
Zone A	Sput –m	19
	Sput	7
	pftp	17

As expected, the parallel stream mode gave the better result than the single stream case. However, it was a bit slower than usual single stream ftp. For HPSS case, the result seems slightly slower than the UNIX file system case because of the same reason as above.

SRB performance consideration

MCAT has the extra time that is required for the database query. Database performance varies widely depending upon the tuning (indices that are built). The sources of decreased performance include:

- Initiation time: This includes the time needed to access the SRB metadata catalog to map from the logical name to the physical resource that is being accessed. This also includes the time required to authenticate the user and authenticate messages between SRB servers

- Transfer time: This is minimized by either use of parallel I/O streams, or by using bulk load and bulk registration to minimize the number of messages.

The time spent in accessing the SRB metadata catalog is strongly dependent upon whether the database has been tuned. Typical tuning operations for Oracle include:

- Indexing of the table structures
- Setting memory size
- Setting network parameters
- Caching of intermediate results

We will tune above parameters for our DB2 and also Postgres to obtain the best performance in our environment.

BELLE SRB SYSTEM

SRB system at Belle

After the tests of functionally and performance measurements in the CRC system, SRB was implemented into the Belle computer system. Those two systems are operating completely independently. In Belle SRB system (see Fig.2), there are two SRB servers and one SRB client at KEK, and one SRB server at Tohoku University connected by Super SInet which provides 1Gbps DWDM. One of them at KEK is PostgreSQL based MCAT enabled SRB server. Other SRB server along with it. A part of HSM disc which is used as Belle storage system is registered as SRB resource in this MCAT. These are connected by Giga bits Ethernet .

We used the Grid Security Infrastructure (GSI) authentication. GSI is based on public key encryption, X.509 certificates, and the Secure Sockets Layer (SSL) communication protocol. It is also secure against network eavesdropping and somewhat less vulnerable against compromised hosts as only temporary delegation certificates are stored in files.

Federation with Australian SRB system

The SRB server is also working at the Melbourne University which is one of the Belle collaboration institutes. This SRB server constitutes one SRB system with MCAT enabled SRB server in Australian National University (ANU). It is federated with Belle SRB system at KEK. Thereby, while each SRB system maintained control of the data and resource registered in local SRB system, it became possible to share the data and resource registered in remote SRB system.

Belle software with SRB

The primary application for the Belle experiment is the Belle Analysis Software Framework (BASF). This application is used for simulation, filtering of events, and analysis. Belle has extended BASF to dynamically load I/O subsystems as C++ objects. It was quit simple to add SRB support as a new I/O class using the following SRB client APIs: srbConnect, srbObjOpen, srbObjCreate, srbObjStat, srbObjRead, srbObjWrite and srbobjClose. They then tested and compared I/O performance using SRB, Belle's own TCP/IP protocol, and NFS.

Belle test results

We have tested the mechanisms on a small scale test bed. It seems this test is the preliminary examinations for using SRB in Belle analysis.

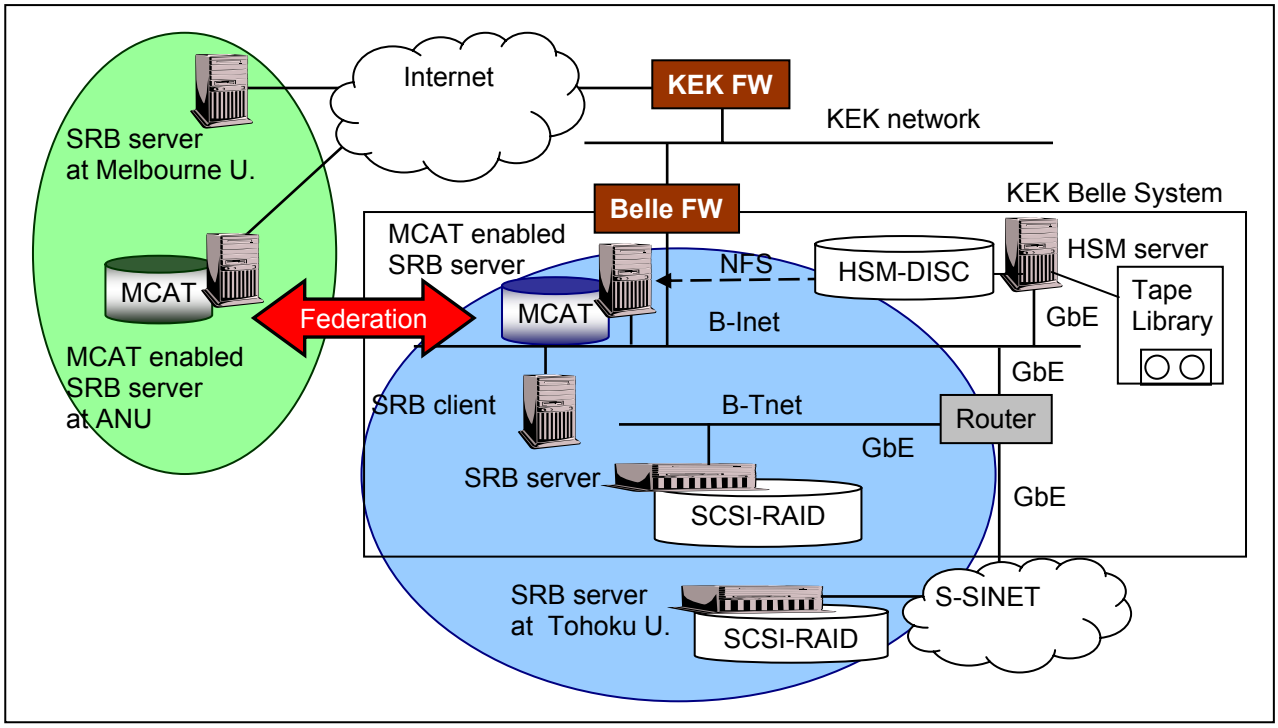

Figure 2: Belle SRB system

Table 3: Belle test results (preliminary)

Protocol	Resource	Elapsed time	Utilization
SRB	Local SCSI-RAID	10:22	53.3%
SRB	Remote HSM (NFS)	13:13	41.8%
UNIX read	Local SCSI-RAID	5:44	90.0%
Belle TCP/IP	Remote HSM (NFS)	6:24	86.1%

This result showed that there is no problem reading and writing remote data by SRB client APIs within Belle software. Belle's own (simple) TCP/IP transfer is about 40% faster then SRB. If it just blocking, double buffering can fix the problem, otherwise more detailed tests are necessary. GSI authentication and MCAT federation look promising in order to share data with collaboration institutes and management its own resources.

SUMMARY

SRB is now working in the Belle experiment with BASF, the Belle analysis software framework, and performance measurements have been done. Zone federation to Australian institution has been established and distributed file sharing over the Internet with Grid authentication is established.

We have also built the independent test environment for SRB at KEK before deployment in the Belle experiment, and measured the performance using different options for SRB commands. The functionally of SRB is very promising for any HEP experiments. Also the preliminary results show that SRB is efficient even for transferring smaller size files, however, we need further study to understand the situation and improve the performance.

ACKNOWLEDGEMENT

We would like to thank S. Chen, G. Kremenk, A. Rajasekar and R. Moore at San Diego Super Computer Center for providing us SRB and their great support on installing and using it.

Also we are grateful to W. Kreoger and A. Hasan at SLAC for helping us to start over SRB at KEK. They stayed at KEK and helped us very much.

We also thank G. Moloney for his enthusiastic work in Australia side.

We wish to thank S. Yamamoto at IBM Japan for supporting construction of HPSS enabled SRB server.

We express our thanks to National Institute of Informatics for their supporting SuperSINET and to KEK Computing Research Center for HEPnet-J, which enabled efficient data sharing among collaborators.

REFERENCES

[1] http://www.npaci.edu/DICE/SRB/index.html
[2] http://www.globus.org