Journal of the Korean Physical Society, Vol. 57, No. 4, October 2010, pp. 715∼724

Design of the Advanced Metadata Service System with AMGA for the Belle II Experiment

S. Ahn,* K. Cho, S. Hwang, J. Kim,† H. Jang, B. K. Kim, H. Yoon and J. Yu
Korea Institute of Science and Technology Information, Daejeon 305-806, Korea

Z. Drasal
Charles University, Prague 116 36

T. Hara, Y. Iida, R. Itoh, G. Iwai, N. Katayama, Y. Kawai, S. Nishida, T. Sasaki and Y. Watase
High Energy Accelerator Research Organization (KEK), Tsukuba 305-0801, Japan

R. Frühwirth and W. Mitaroff
Institute of High Energy Physics, Austrian Academy of Science, Vienna A-1050, Austria

R. Grzymkowski, M. Sitarz, H. Palka and M. Zdybal
H. Niewodniczanski Institute of Nuclear Physics, Krakow PL-31-342, Poland

M. Heck, T. Kuhr and M. Röhrken
Institut für Experimentelle Kernphysik, Universität Karlsruhe, Karlsruhe 76128, Germany

M. Bračko
J. Stefan Institute, Ljubljana 1000, Slovenia

S. Lee
Korea University, Seoul 136-701, Korea

C. Kiesling, A. Moll and K. Prothmann
Max-Planck-Institut für Physik, München 80805, Germany

H. Nakazawa
National Central University, Chung-li 32001, Taiwan

T. Fifield and M. E. Sevior
University of Melbourne, School of Physics, Victoria 3010, Australia

S. Stanič
University of Nova Gorica, Nova Gorica SI-5000, Slovenia

(Received 23 August 2010, in final form 6 September 2010)

The Belle II experiment is expected to produce 50 times more data than the existing Belle experiment. Such huge data production requires not only scalability with respect to the storage service but also scalability regarding the metadata service. There has already been a metadata service at the Belle experiment, but it is not proper for the Belle II experiment because it has scalability problems and it is not intended to be used in a distributed grid environment. To deal with these issues, we designed an advanced metadata service system based on AMGA, which provides efficient and scalable metadata searching. We have built testbed sites to test the correctness, performance and scalability of the advanced metadata service system, and it has been proved to be able to provide efficient metadata searching for the Belle II experiment.

PACS numbers: 07.05.Kf, 07.05.-t, 29.50.+v
Keywords: High energy physics, Belle II, Large data handling, Metadata service, AMGA
DOI: 10.3938/jkps.57.715

I. INTRODUCTION

The Belle II experiment [1,2] studies a precision measurement in the B meson system through the detector at KEK. Belle II will bring the searches of New Physics performed at the existing B factories [3] to a completely new level of sensitivity and significantly extends the reach of present experimental efforts in a way complementary to the energy frontier experiments. It is an international collaboration work where hundreds of researchers from more than 10 countries are participating.

The Belle II experiment, which will start operation in 2014, is expected to produce 50 times more data than the existing Belle experiment. The expected amount of storage is about 4.9 petabytes in 2014, and will increase to about 253 petabytes in 2019. We assume a trigger rate of 30 kHz corresponding to the highest luminosity of $L = 0.8 \times 10^{35}$ $cm^{-2}s^{-1}$, and the raw event size in the worst case is about 300 kilobytes. To manage and analyze these huge amounts of data, we are actively examining the feasibility of adopting grid infrastructures and technologies to the computing environment of the Belle II experiment [4].

In the Belle II experiment, metadata service is considered to be one of integral services because it may be common to see millions of files spread over geographically distributed environments. It provides an efficient mechanism to publish data with descriptive information (*i.e.*, metadata) about files or events and to locate the files using the metadata, as well. We are currently examining AMGA (ARDA Metadata Grid Application) [5,6] as metadata service software for Belle II. It is a part of the gLite [7] middleware software stack being developed for the EGEE (Enabling Grids for E-sciencE) [8] project. It is designed to provide an interface for accessing the metadata attributes of the files on the grid.

Data produced from the Belle II accelerator and computing farms will be primarily processed to a form that researchers can analyze easily and will be stored as files at multiple distributed grid sites. To locate files stored at multiple distributed sites with some descriptive information, we are planning to provide two types of metadata service: a file-level metadata service and an event-level metadata service. The file-level metadata service maintains information associated with each file, such as the total number of events in it, types, a file creation date, a used software version, *etc*, and it returns lists of files, as a form of GUID (Globally Unique Identifier) or LFN (Logical File Name) satisfying users' given conditions. The event-level metadata service maintains summarized information about physics values associated with each event, and it returns lists of files and of event numbers satisfying users' given conditions. Event-level metadata searching can eliminate the necessity to read all the files to find events of interest to a user, so it may speed up analyses of users that utilize only a small number of events among large datasets.

There has already been a metadata service at the present Belle experiment. The service manages only file-level metadata in a local database, which provides physical locations of files. In addition, there is a mechanism, so-called "index" files, for course-grained event-level searching. The index file has information on specific events that the analyzer has interested in, *i.e.*, event number. When a user reads an index file with the Belle analysis software framework, the file-level metadata service provides the physical location of the file first; then, he or she can access the events whose event number is listed in the index file. The file-level metadata and index files are all centralized at the KEK site.

The centralized metadata catalog service at the present Belle experiment may cause many problems if it is applied to the Belle II metadata service. First, it may have some reliability problem if a centralized metadata server would have a failure. Second, it is not intended to be used in a distributed grid environment, to which the Belle II experiment is moving. Third, it does not provide fine-grained event-level metadata searching, so it takes a very large amount of time to find events of interest to the users. Besides, the centralized metadata catalog is inappropriate to support fine-grained event-level metadata searching because it may result in severe performance and scalability problems. Belle II is expected to have 50 times more data, and the amount of event-level metadata is O (10^3) times larger than the amount of file-level metadata. As the amount of metadata increases, the centralized metadata catalog service would likely present unacceptable response times to users that try to access it. What makes it even worse is that the service may be accessed by hundreds of users across the world simultaneously.

To deal with these issues, considering performance, scalability and reliability, we present an advanced metadata service system for Belle II. First, we present distributed metadata service for Belle II where metadata are replicated into multiple sites to provide scalability, which relies on AMGA. Second, we have designed an optimized metadata schema for Belle II, which significantly reduces disk space and response time for event-level metadata searching. Third, we present some preliminary test results measured at the testbed for correctness, performance and scalability of the advanced metadata service system.

*E-mail: siahn@kisti.re.kr
†E-mail: jhkim@kisti.re.kr; Fax: +82-42-869-0759

Table 1. Expected storage size and number of files and events.

Type	Belle (Roughly, 2010)		Belle II (Expected, 2019)	
	storage size (petabytes)	# of files	storage size (petabytes)	# of files
Raw Data	1.3	6.5×10^5	196	10^8
Real Data (mDST)	0.14	1×10^5	8.7	4.3×10^6
MC Data (mDST)	0.6	8×10^5	42	21×10^6

This paper is organized as follows: Section II introduces the advanced metadata service system for the Belle II experiment. It details requirements, hierarchically structured file-level and event-level metadata schema, optimization methods, and distributed grid metadata service while focusing on efficient event-level metadata searching. Section III shows some test results for the advanced metadata service system. Section IV explains some related works, and finally Section V concludes with an outlook.

II. THE METADATA SERVICE SYSTEM AT BELLE II

1. Background

The data acquisition system records the raw data taken by the detector. The raw data are processed, and the derived information is stored in DST (Data Summary Tape) files, with more high-level compact information being stored in mDST (mini-DST) files. In addition to the real data, high-statistics samples of simulated data (Monte-Carlo, MC) are generated to derive physics quantities from the real data. Table 1 gives a summary of expected storage size and number of files that will be produced for Belle II [4]. A maximum file size was assumed to be 2 gigabytes. The Belle II file-level metadata service manages all three types of files in Table 1, and the event-level metadata only targets events in mDST files.

2. Requirements

We have defined some requirements for the metadata service system of Belle II. First, it should provide grid accessibility. The Belle II experiment is planned to utilize distributed grid infrastructures to process large-scale data. Therefore, it is necessary for all the involved services to provide access based on grid authentication. Second, it should provide reliable and robust service, which means that the metadata service should provide correct results and be able to support persistent availability. Third, it is necessary to provide scalability in terms of disk space and response time. When the experiment starts in 2014, the amount of metadata is expected to be a few terabytes, but it may increase to a hundreds of terabytes scale in 2020 if it is extended to an event-level metadata searching. Even though the total amount of metadata is scaled, it should be possible for hundreds of users to access the metadata service within an acceptable response time. Fourth, it should be possible to provide minimum authorization. For example, normal users are supposed to have permission to search metadata only, and managers should have all the proper permissions to manage metadata.

We are implementing a metadata service system based on placeAMGA, which satisfies most of these requirements. AMGA provides grid access and authentication through SSL (Secure Sockets Layer), GSI (Grid Security Infrastructure) [9] and VOMS (Virtual Organization Management Service) [10], which satisfies the first requirement. The replication feature of AMGA allows all or parts of metadata to be replicated into multiple sites automatically, which plays a decisive role in improving reliability and scalability, which are the second and the forth requirements. In addition, AMGA supports fine-grained access control based on the ACL (Access Control List) which satisfies the third requirement.

3. Hierarchically Structured Schema

We have defined the Belle II metadata schema to have hierarchically structured tables, supported by AMGA. The hierarchically structured tables can decrease management complexity and are very useful when replicating metadata in a structured way. AMGA uses a directory concept similar to the UNIX file system, which allows each metadata table to be mapped into a directory and all the related tables to be located at the same parent directory. Table 2 shows the hierarchical table structure for Belle II metadata. In Table 2, an "experiment" refers to a collection of data gathered until the accelerator is turned off after it is turned on once, and a "run" refers a collection of data gathered until the accelerator stops shooting beams after it starts to shoot once. Several "events" are produced from a collision of beams, and events produced in a "run" may be recorded in several files. These data are processed further to produce simulation data, called a "stream." Currently Belle has 10 "streams," and we are expecting 6 "streams" in Belle II.

Table 2. Hierarchical table structure for Belle II file-level metadata.

Directory Name in placeAMGA	Description
/belle/raw/E##/FC	File-level metadata tables raw Data (##: an "experiment" number, FC table in Figure 1)
/belle/data/E##/FC	File-level metadata tables Real mdst Data (##: an "experiment" number, FC table in Figure 1)
/belle/MC/generic/E##/FC /belle/MC/signal/E##/FC	File-level metadata tables Monte Carlo simulation data.
/belle/data/E##/EC/C**	Event-level metadata tables Real Data (##:"experiment" number, **: "condition id")
/belle/MC/generic/E##/EC/C**	Event-level metadata tables Monte Carlo simulation data. (##:"experiment" number, **: "condition id")
/belle/dataset	Event types (Dataset table in Figure 1)
/belle/skim	Skims (Skim table in Figure 1)
/belle/userinfo	User information (User table in Figure 1)
/belle/site	Site information (Site table in Figure 1)
/belle/software	Software information (Software table in Figure 1)

Table 3. File-level metadata schema of Belle II.

Attribute Name	Data Type in placeAMGA	Description
id	int	Unique ID
guid	varchar 40	Globally unique ID (Hex Format)
lfn	varchar(1024)	Logical file name
status	varchar 16	Good/bad/further values ...
events	int	Total number of events
datasetid	int	Dataset ID Foreign key to /belle/dataset
stream	int	Stream number (in case of MC)
runH	int	Highest run number
eventH	int	Highest event number in the highest run
runL	int	Lowest run number
eventL	int	Lowest event number in the lowest run
parentid	int 128	ID of parent files
softwareid	int	ID of software release version Foreign key to /belle/software
siteid	lint	Site ID where the file was created Foreign key to /belle/site
userid	int	ID of a user who creates the fileForeign key to /belle/user_info
log_guid	varchar 40	GUID of log file (Hex Format)

4. File-level Metadata Schema for Belle II

With the file-level metadata service, it is possible to retrieve lists of files satisfying users' given conditions. The conditions are composed of one or more conditional statements on an "experiment" number, a file type, a "run" number, a "stream" number, *etc.* The file-level metadata schema for Belle II is depicted in Fig. 1. The "FC" table stores information related to each file, such as a physical or logical location, total number of events in it, "run" number, type, file creation date, used software version, *etc.* The "Dataset" table stores information related to the type of files, and the "Skim" refers to a collection of events a certain user group is interested in. The "Site" table stores information about other metadata service sites, especially whether each site has metadata on a certain "experiment." The "Software" table has information about the used Belle II library version, and the "User" table stores grid certificates for users.

The attributes of the "FC" table, which plays the central role in file-level metadata searching, are detailed in Table 3. Table 3 includes an LFN attribute, which may be redundant because LFC (LCG File Catalog) service [11] has a mapping between GUID and LFN. However, it

Table 4. Example of event-driven metadata attributes.

Attributes	Data Type	Size (bytes)	Description
datasetid	smallint	2	Type
run	smallint	2	Run number
event	int	4	Event number
r2	float	4	r2 (range: 0.0 ∼ 1.0)
plus_charge	smallint	2	Number of + charged track (range: 0∼, mostly < 5)
minus_charge	smallint	2	Number of - charged track (range: 0∼, mostly < 5)
k_short	smallint	2	Number of K_S (range: 0∼, mostly < 7)
k_long	smallint	2	Number of K_L (range: 0∼, mostly < 7)
Row overhead		32	In case of PostgreSQL
Space consumption per row		54	

Table 5. Estimated disk space consumption of event-driven event-level metadata.

Description	Estimated number	Note
(A) Expected total Hadronic events in Belle II	1.92×10^{11}	[4]
(B) Expected MC events (6 streams)	11.5×10^{11}	= A × 6
(C) Bytes per events	54 bytes	Refer Table 4
(D) Total disk consumption	72.6 terabytes	= (A + B) × C

Table 6. Metadata attributes of condition-driven event-level metadata.

Attribute	Data Type	Description
datasetid	smallint	dataset ID (type)
stream	smallint	"stream" number
run	smallint	"run" number
eventbits	varying bits	a list of events satisfying a certain condition

is currently included to test the metadata service against the existing Belle data.

5. Event-level Metadata Schema for Belle II

The event-level metadata service maintains summarized information about physics variables associated with each event. With the event-level metadata service, it is possible to retrieve lists of files and event numbers satisfying users' given conditions. The conditions are composed of one or more statements on physics attributes. Currently, the physics attributes that should be maintained at the event-level metadata have been not decided yet, and they are still under discussion. In this paper, we present the event-level metadata schema with five candidate attributes, as shown in Table 4. They are valuable when searching for events with many tracks.

It is possible to define an event-level metadata table to have a row per event as shown in Table 4; we call it event-driven schema hereafter. This method was used in the ATLAS tag database [12]. However it requires too much disk space. For example, the total required storage space

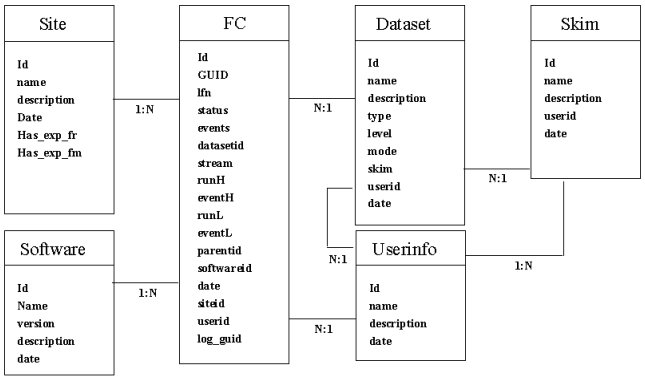

Fig. 1. File-level metadata schema of Belle II.

is estimated to be 72.6 terabytes, as shown in Table 5, assuming the five candidate physics attributes of Table 4. If the number of attributes is increased further, it will consume more disk space for metadata. In addition, it is possible to have an unacceptable response time because there are billions of rows to be searched.

To reduce required disk space and response time as much as possible, we have designed an optimized

Table 7. Example of tables and conditions for the condition-driven metadata.

Attribute	# tables	Table Name	Condition
plus_charge	5	/belle/MC/generic/E##/EC/C01	The number of plus charged track ≥ 1
		...	...
		/belle/MC/generic/E##/EC/ C05	The number of plus charged track ≥ 5
minus_charge	5	belle/MC/generic/E##/EC/C06	The number of minus charged track ≥ 1
		...	...
		/belle/MC/generic/E##/EC/C10	The number of minus charged track ≥ 5
k_short	7	/belle/MC/generic/E##/EC/C11	The number of $K_L \geq 1$
		...	...
		/belle/MC/generic/E##/EC/C17	The number of $K_L \geq 7$
k_long	7	/belle/MC/generic/E##/EC/C18	The number of $K_S \geq 1$
		...	...
		/belle/MC/generic/E##/EC/C24	The number of $K_S \geq 7$
r2	9	belle/MC/generic/E##/EC/C25	r2 ≥ 0.1
		...	...
		/belle/MC/generic/E##/EC/C33	r2 ≥ 0.9

Table 8. Estimated disk space consumption of condition-driven event-level metadata.

Description	Estimated number	Note
(A) Expected total Hadronic events in Belle II	1.92×10^{11}	[4]
(B) Expected MC events (6 streams)	1.15×10^{12}	= A × 6
(C) Number of conditions	33	Refer Table 7
(D) Total disk consumption	5.5 terabytes	= (A + B) / 8 bit × C

condition-driven schema as shown in Table 6. While each entry (row) stores physics values corresponding to an event with the event-driven schema, each entry (row) stores list of events satisfying certain conditions in each "run" with the condition-driven schema. For example, the "Number of + charged track" attribute in Table 4 is an integer ranging mostly from 0 to 5, so that the minimum information that should be kept about "Number of + charged track" attribute is five lists of events for each "run"; 1+, 2+, 3+, 4+, and 5+, where 5+ means a list of events having more than 5 plus charged tracks. By maintaining these five lists of events, it is possible to answer most users' queries; events having more than 6 plus charged tracks are very rare.

In Table 6, the "eventbits" stores a list of events in a bit string data type, so the required space for metadata can be reduced significantly. Table 7 shows an example of conditions and corresponding tables that we used to build event-level metadata at the testbed. For the five physics attributes of Table 4, 33 conditions were sufficient to answer most users' queries. Table 8 shows estimated required storage space for the condition-driven event-level metadata with the five candidate physics attributes. It only occupies about 5.5 terabytes of disk space, which is 13 times less than that for the event-driven event-level metadata.

On the user's query, "eventbit" values can be connected through logical operators. Currently, only a few databases, such as PostgreSQL, support varying bit data type and their logical operations, but many other databases, such as Oracle and MySQL, do not support this data type.

6. Distributed Grid Metadata Service

On large-scale grids, a centralized metadata catalog service may result in severe performance and scalability problems. Therefore, we are planning to provide the metadata service for Belle II through multiple distributed regional servers in order to increase performance, reliability and scalability. It will be made accessible through the grid that is able to coordinate the trust fabric for authentication in the international collaboration. To construct efficient and scalable grid metadata service, we have applied AMGA, which is one of the most frequently used metadata service technologies on the grid, into the Belle II metadata service.

The replication and the redirection features of AMGA play decisive roles in enabling the Belle II metadata service reliability and scalability. The replication feature of AMGA enables all or parts of metadata to be copied and

Table 9. Summary of file-level metadata built at the Belle II testbed.

	Description	Estimated Number	Note	Space
Background	(A) total experiments	30		
	(B) total streams	6		
	(C) runs per experiment	800		
	(D) total types	4		
Raw Data	(E) total files	10^8	Refer Table 1	33.9
	(F) files per experiment	3.3×10^6	= E / A	gigabytes
	(G) files per run	4,125	= F / C	
Real Data	(H) total files	4.3×10^6	Refer Table 1	1.41
	(I) files per experiment	143×10^3	= H / A	gigabytes
	(J) files per run	180	= I / C	
MC Data	(K) total files	21×10^6	Refer Table 1	7.74
	(L) files per stream	3.5×10^6	= K / B	gigabytes
	(M) file per stream & experiment	116.7×10^3	= L / A	
	(N) files per type	29.2×10^3	= M / D	
	(O) files per run	146	= M / C	

synchronized with multiple regional metadata servers. It is based on a master-slave model, which means that all the updates of the master are synchronized with the slaves automatically. Write access on metadata is only allowed at the master server, and it is only possible to access metadata read-only at the slave servers. The redirection feature of AMGA provides users with a single virtualized view of metadata that are actually dispersed geographically across multiple sites. If a directory is set to be redirected to an other server, then all the queries related to the directory are redirected automatically.

Initially, all the metadata of Belle II will be stored at the Tier 0 site (KEK), and several regional Tier 1 sites will host partial metadata by requesting the interested part of metadata to be replicated. Users are advised to access their closest regional Tier 1 server, and if that regional Tier 1 server does not have the metadata that the user requests, then the query is redirected automatically to other Tier 1 sites or to the Tier 0 site having proper metadata. In addition, it is possible for personal users or groups to construct their own metadata service after downloading part of their metadata of interest from the web site.

Figure 2 depicts an example of building a distributed metadata service for Belle II event-level metadata searching through replication and redirection of AMGA. In this example, the Tier 0 site has several placeAMGA servers hosting metadata. At Tier 0, the "AMGA server 1" stores metadata only on the "experiments" 7, 9, and 11 and has virtual directories on other "experiments" for redirection. If a user queries about "experiment" 13 to "AMGA server 1", then it just redirects it to "AMGA server 2," which has metadata on "experiment" 13. In Fig. 2, there are two metadata servers at the Tier 1 site. At the Tier 1, the "Local AMGA server 1" keeps metadata on "experiments" 7 and 13, which are replicated from Tier 0. The "Local AMGA server 2" has metadata on "experiment" 25 which is replicated from "AMGA server 4" of Tier 0. If "Local AMGA server 1" receives a query on "experiment" 25, it redirects the query to "Local AMGA server 2" at Tier 1 or to "AMGA server 4" at Tier 0.

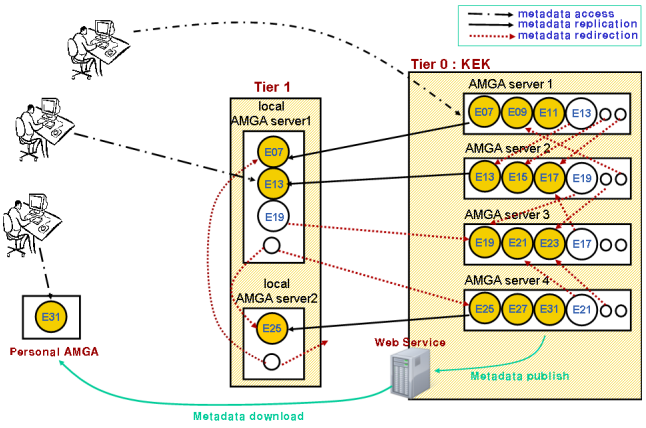

Fig. 2. (Color online) Example of building a distributed metadata service for Belle II metadata searching.

III. TEST RESULT

Currently, we have built two testbed sites; one is at KISTI (Korea Institute of Science and Technology Information), and the other is at Melbourne University. The KISTI site plays the role of a metadata master server, and the Melbourne site acts as a slave server, which means all the metadata are inserted at KISTI site and

Table 10. Efficiency of event-level metadata searching.

		Without event-level metadata	With event-level metadata
Conditions	Target	Monte Carlo stream 0 experiment 7	
	type	on_resonnance, uds	
	constraint	r2 $\geq$ 0.5	
Results	Retrieved # of events	2.4×10^6	2.4×10^6
	Time taken for event retrieval process	About 265 minutes	About 4 second
	Time taken for data caching	About 32 minutes	

then replicated into the Melbourne site. We confirmed that metadata at the KISTI site were replicated safely to the placeCityMelbourne site.

1. Efficiency of File-level Metadata Searching through AMGA

We have built a file-level metadata searching environment with emulated Belle II metadata based on the TDR (Technical Design Report) of Belle II data handling [4]. Table 9 gives statistical information on the built file-level metadata for Belle II. The emulated file-level metadata for Belle II occupied about 42 gigabytes of storage space.

To test whether AMGA ensures a proper response time for file-level metadata searching, we tried to measure the worst-case response time retrieving all file-level metadata for a random experiment of a random MC stream. Figure 3 shows the measured average response time taken per query while increasing the number of concurrent queries. The response time increased in proportion to the number of concurrent queries, but reasonable response time was observed even in the worst-case that the number of concurrent queries reached 50.

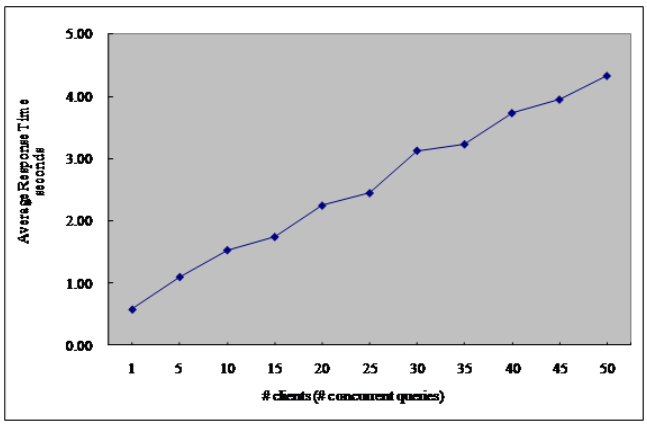

Fig. 3. Average response time for the worst-case query of file-level metadata searching.

2. Efficiency of Event-level Metadata Searching

The control of the event level provides users with efficient processing, such as events with K_0^S or K_0^L , or $e + e^- \rightarrow qq$ (q = u, d, s, c) processes. To show the effectiveness of the event-level metadata searching, we measured the time taken by the event retrieval process that searches files and events of user's interests. It is the first task in the user's analysis process.

Without the event-level metadata, the only way to carry out the event retrieval process is to access all files and see whether the events in the files satisfy the users' given conditions. We built and ran an application that searched all events satisfying certain conditions at the Belle data farm, as shown in Table 10. As a result, totally 2.4×10^6 events were retrieved, which took about 4.25 hours.

We also measured the time taken by the event retrieval process with the same conditions through the event-level metadata service. The number of retrieved events was the same, and it took only about 4 second to search the list of events. This means that the event-level metadata service can reduce the time taken by the event retrieval process by more than O (10^3) times. If the whole event-level metadata is built for the Belle, it is expected to take only several minutes to complete the event retrieval process, would take more than one month without it.

3. Performance Comparison between Event-driven and Condition-driven Metadata

We measured and compared the response times between with the event-driven metadata and the condition-driven metadata. To make the test simpler, we built only partial metadata for both methods.

For an event-driven metadata, we assumed one metadata table per type shown in Table 11, where the number of events was assumed to be 50 times that of Belle. To make the test simpler, we built only one metadata table targeting (E) in Table 11. It had 256 million events following the schema of Table 5. We set each metadata to have a random value within its possible range, and an index was created for an attribute (r2) to make searching faster.

For the condition-driven metadata test, we built one

Table 11. Number of events by type used for the tests.

Type	Estimated # events ($\times 10^9$)	Fraction (%)
(A) uds, on_resonance	2.56	40
(B) charm, on_resonance	0.7	11
(C) charged, on_resonance	1.4	22
(D) mixed, on_resonance	1.4	22
(E) uds, off_resonance	0.256	4
(F) charm, off_resonance	0.064	1
Total Hadronic MC Events in an "experiment" of an MC "stream"	6.4	100

Table 12. Response time taken for even-level metadata searching.

Query Targeting (E) in Table 11	Retrieved events / total events	Event-driven metadata (second)	Condition-driven metadata (second)
SELECT, r2 >= 0.9	10%	about 416	about 20
SELECT, r2 >= 0.8	20%	about 722	
SELECT, r2 >= 0.7	30%	about 977	
SELECT, r2 >= 0.6	40%	about 1208	
SELECT, r2 >= 0.5	50%	about 1446	
SELECT, r2 >= 0.4	60%	about 1749	
SELECT, r2 >= 0.3	70%	about 2029	
SELECT, r2 >= 0.2	80%	about 2311	
SELECT, r2 >= 0.1	90%	about 2659	
SELECT, r2 >= 0	100%	about 2849	

metadata table targeting one condition in Table 7. Since the size of the event lists does not differ with the conditions, it is not necessary to build tables for all the conditions in this test. The table was set to have metadata regarding the 6.4×10^9 events shown in Table 11.

The machine used for the test was a Xeon 1600 GHz server with 4 cores and 8 gigabytes of memory. Table 12 shows the used queries and the measured response time. In the case of event-driven metadata searching, the response time increased linearly with the number of retrieved events. In the case of condition-driven metadata searching, it took only about 20 seconds to retrieve the lists of events. The condition-driven metadata searching may reduce the response time by more than 20 times compared to event-driven metadata searching.

Table 13. Queries to measure the performance of condition-driven metadata searching by the number of conditions.

# of conditions	Query (SELECT)
1	plus_charge $\geq$ 1
2	plus_charge $\geq$ 1, minus_charge $\geq$ 1
3	plus_charge $\geq$ 1, minus_charge $\geq$ 1, k_short $\geq$ 2
4	plus_charge $\geq$ 1, minus_charge $\geq$ 1, k_short $\geq$ 2, k_long $\geq$ 2
5	plus_charge $\geq$ 1, minus_charge $\geq$ 1, k_short $\geq$ 2, k_long $\geq$ 2, r2 >= 0.5

4. Performance of Condition-driven Metadata Searching by Number of Conditions

It is possible for a user to join several conditions across various physics attributes in a query. To measure the performance of condition-driven metadata searching by number of conditions, we built five tables; each table corresponded to a condition for each physics attribute in Table 7. We used the queries shown in Table 13, and the same machine as that at the previous test was used.

Figure 4 shows the response time taken by the number of conditions used in a query. Since the size of the event list differs by events' types, the response time increased in proportion to the number of events included in each type. However, the number of conditions did not degrade performance. This means that the overhead coming from the size of data retrieved overwhelms the overhead coming from the size of the disc block to be loaded into memory.

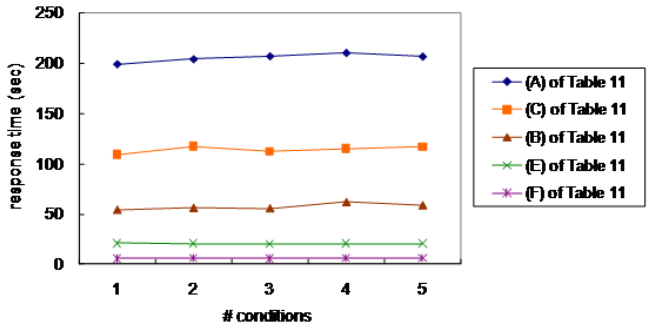

Fig. 4. (Color online) Response time of condition-driven metadata searching by the number of conditions.

IV. RELATED WORKS

In high energy physics, several LHC (Large Hadron Collider) experiments have implemented metadata catalogues specific to their needs. The catalogues are the ATLAS Metadata Interface AMI [12], the CMS experiment's RefDB [13] and the Alien Metadata catalogue [14] from the Alice experiments. They consist of a standard RDBMS backend and an adapter layer to allow access in a distributed computing environment. However, except ATLAS, they mostly provide file-level metadata searching only.

The ATLAS tag database [12] is an event-level metadata system designed to allow efficient identification and selection of interesting events for user analysis. By using queries on a relational database to make first-level cuts, the size of an analysis input sample may be greatly reduced; thus, the time taken for the analysis is reduced. For event-level tags, there is a central global database at CERN, hosted on Oracle. To ease the load on this central service, the database will be replicated to various other Tier-1 and Tier-2 sites.

Belle II metadata service is different from the ATLAS tag database in that the former supports condition-driven metadata searching and the latter supports event-driven metadata searching. As explained in the section II, the condition-driven metadata may reduce required disk space and response time significantly.

V. CONCLUSION

This paper proposed an advanced metadata service system for Belle II, which significantly reduces the disk space and the response time required for metadata searching. We also presented a distributed metadata service for the Belle II experiment to provide good performance and scalability, where the replication and the redirection features in AMGA play a decisive role. We have built a testbed site to test the correctness, performance and scalability of the advanced metadata service system.

In file-level metadata searching, the response time increased in proportion to the number of concurrent queries, but it showed very reasonable response time even in the worst-case query in which the number of concurrent queries reached 50. In event-level metadata searching, the optimized condition-driven metadata schema reduced the required disk space by as much as 13 times and the response time by more than 20 times. Also, we showed that a proper construction of event-level metadata service can reduce the time taken by the event retrieval process by the factor of thousand. If an event-level metadata system is built for the Belle experiment, it should only take a few minutes to complete the event retrieval process, which would take more than one month without it.

REFERENCES

[1] T. Hara and Belle II Computing Group, *Managed Grids and Cloud Systems in the Asia-Pacific Research Community* (Springer, New York, 2010), p. 109.
[2] M. Bracko, J. Phys. Conf. Ser. **171**, 52 (2009).
[3] A. Abashian and Belle Collaboration, Nucl. Instrum. Meth. A **479**, 117 (2002).
[4] Belle II Computing Group, *Belle II Technical Design Report* (to be published).
[5] N. Santos and B. Koblitz, Nucl. Instrum. Methods Phys. Res. **559**, 53 (2006).
[6] S. Ahn, N. Kim, S. Lee, D. Nam, S. Hwang, B. Koblitz, V. Breton and S. Han, Software Pract. Exper. **39** 1055 (2009).
[7] E. Laure, S. M. Fisher, A. Frohner, C. Grandi, P. Kunszt, A. Krenek, O. Mulmo, F. Pacini, F. Prelz, J. White, M. Barroso, P. Buncic, F. Hemmer, A. Di Meglio and A. Edlund , CMST **12**, 33 (2006).
[8] F. Gagliardi, B. Jones, F. Grey, M. E. Begin and M. Heikkurinen, Philos. Trans.: Math. Phys. Engin. Sci. **363**, 1729 (2005).
[9] I. Foster, C. Kesselman, G. Tsudik and S. Tuecke, in *Proceedings of 5th ACM Comp. and Commun. Security* (San Francisco, USA, 1998), p. 83.
[10] R. Alfieri, R. Cecchini, V. Ciaschini, L. dell'Agnello, Á. Frohner, A. Gianoli, K. Lõrentey and F. Spataro, Lect. Notes Comput. Sci. **2970**, 33 (2004).
[11] J. P. Baud, J. Casey, S. Lemaitre and C. Nicholson, in *Proceedings of High Perf. Distr. Comp. 14* (Wasington, USA, 2005), p. 91.
[12] J. Cranshaw L. Goosens, D. Malon, H. McGlone and F. T. A. Viegas, J. Phys. Conf. Ser. **119**, 072012 (2008).
[13] A. Afaq, A. Dolgert, Y. Guo, C. Jones, S. Kosyakov, V. Kuznetsov, L. Lueking, D. Riley and V. Sekhri, J. Phys. Conf. Ser. **119**, 072001 (2008).
[14] P. Buncic, P. Saiz and A. J. Peters, in *Proceedings of Comp. in High Energy and Nucl. Phys.* (La Jolla, USA, 2003), p. 10.