PROCEEDINGS OF SCIENCE

Computing at Belle II

Takanori Hara[*a]
[a]*High Energy Accelerator Research Organization (KEK), Tsukuba*
E-mail: takanori.hara@kek.jp

S. Ahn[b], M. Bračko[c], K. Cho[b], Z. Drasal[d], T. Fifield[e], R. Frühwirth[f], R. Grzymkowski[g], M. Heck[h], S. Hwang[b], Y. Iida[a], R. Itoh[a], G. Iwai[a], H. Jang[b], N. Katayama[a], Y. Kawai[a], C. Kiesling[i], B. K. Kim[b], J. H. Kim[b], T. Kuhr[h], S. Lee[j], W. Mitaroff[f], A. Moll[i], H. Nakazawa[k], S. Nishida[a], H. Palka[g], K. Prothmann[i], M. Röhrken[h], T. Sasaki[a], M. E. Sevior[e], M. Sitarz[g], S. Stanič[l], Y. Watase[a], H. Yoon[b], J. Yu[b], M. Zdybal[g]

[b]*Korea Institute of Science and Technology Information, Daejeon*
[c]*J. Stefan Institute, Ljubljana*
[d]*Charles University, Prague*
[e]*University of Melbourne, School of Physics, Victoria 3010*
[f]*Institute of High Energy Physics, Austrian Academy of Sciences, Vienna*
[g]*H. Niewodniczanski Institute of Nuclear Physics, Krakow*
[h]*Institut für Experimentelle Kernphysik, Universität Karlsruhe, Karlsruhe*
[i]*Max-Planck-Institut für Physik, München*
[j]*Korea University, Seoul*
[k]*National Central University, Chung-li*
[l]*University of Nova Gorica, Nova Gorica*

The Belle II experiment, a next-generation B factory experiment at KEK, is expected to record a two orders of magnitude larger data volume than its predecessor, the Belle experiment. The data size and rate are comparable to the ones of LHC experiments and requires to change the computing model from the Belle way, where basically all computing resources were provided by KEK, to a more distributed scheme. While we adopt existing grid technologies for our baseline design, we also investigate the possibility of using cloud computing for peaking resource demands. An important task of the computing framework is to provide easy and transparent access to data and to facilitate the bookkeeping of processed files and failed jobs. To achieve this we set up a metadata catalog based on AMGA and plan to use it in a bookkeeping service that is based on concepts implemented in the SAM data handling system used at CDF and D0.

13th International Workshop on Advanced Computing and Analysis Techniques in Physics Research,
ACAT2010
February 22-27, 2010
Jaipur, India

[*] Speaker.

1. Introduction

The Belle experiment [1] started taking data in 1999 to confirm the Kobayashi-Maskawa theory [2] and accumulated an integrated luminosity of more than 1 ab^{-1} during its 10-year operation. It was supported by a centralised computing facility at KEK, where almost all data processing, Monte-Carlo (MC) production and physics analysis was performed. The system was well designed for the amount of data and worked smoothly.

Now we plan the Belle II experiment, a next-generation B factory experiment at KEK, to search for new physics beyond the Standard Model which explains the behavior and interactions between elementary particles and aim to start in 2014. The Belle II computing system has to handle an amount of data eventually corresponding to 50 times the Belle level by the end of 2020. This means an amount of raw data of the order of 10^{10} events per year.

In order to achieve the physics goals, it is required that the raw data is processed without any delay to the experiment data acquisition, in addition to the production of MC events corresponding to at least 3 times of beam data. Moreover, computing power for physics analysis has to be provided. The computing resources required for these purposes increase faster than the projected performance of CPUs and storage devices. Under this situation, it can not be expected that one site will be able to provide all computing resources for the whole Belle II collaboration. To solve this problem, a distributed computing model based on the grid is adopted.

Figure 1 illustrates the concept of the Belle II computing model. Basically, KEK will host the main center that is responsible for raw data processing/archiving and play a role as one of the Grid site. Each Grid site allows users to produce ntuples from skimmed datasets and takes care of the MC production, possibly complemented by Cloud Computing facilities. Finally users analyze ntuples on local resources, where could be the non-grid site. More detailed explanation of the role of each site is presented in Section 1.1.

1.1 Grid-based Computing

Belle II will make extensive use of distributed computing solutions built upon gLite[3] middleware. The gLite middleware enables both data and CPU resources to be located at a variety of sites around the world. It also enables the movement of data between storage resources and allows jobs to run at sites with CPU resources. Therefore we plan to deploy most pieces of the gLite middleware used by the LHC experiments. In addition we plan to develop a new piece of middleware, the “Project Server”, which steers particular tasks amongst Belle II’s distributed compute and data resources. The project server as well as the data management is discussed in Section 1.3.

The Belle II computing system has to accomplish several tasks, e.g. the raw data processing, archiving the produced physics skimmed files as well as raw data, the MC production, and ntuple-level user analyses. However, not all tasks are equally suited for a distributed system. Because the raw data is produced at KEK, the location of the Belle II detector, this site is distinct from other centers. It will play the role of a main center and be responsible for the processing and storage of raw data. As MC production can be distributed easily we will use grid sites for it. We also envision to analyze the produced MC samples and replicated real data samples on the grid. The distribution of grid sites should reflect the geographical distribution of Belle II collaborators. Together with KEK the remote grid sites form the bulk of the Belle II computing resources and their deployment

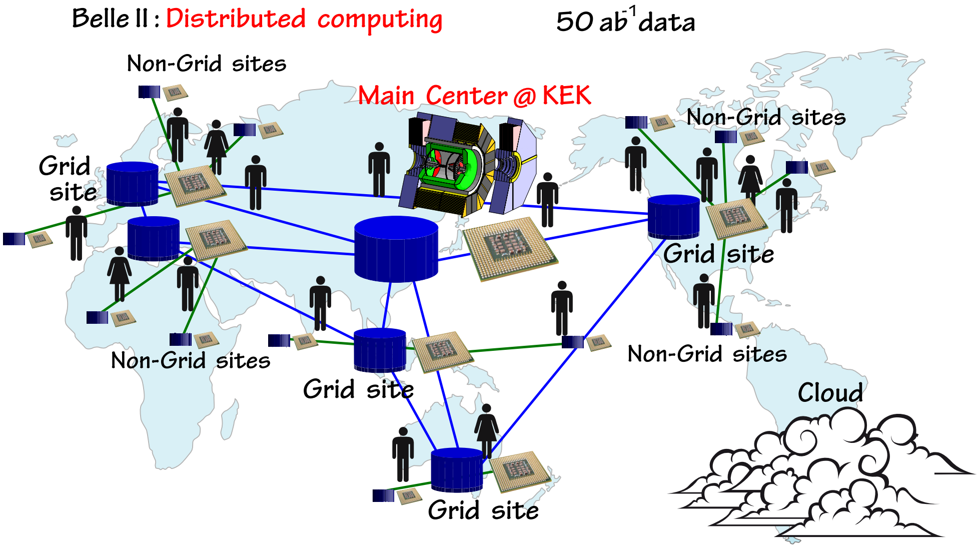

Figure 1: Concept of the Belle II computing model. The main center, KEK, is a core site for raw data processing of beam data and data archive. KEK also contributes a fraction of the MC production, but the main part will be produced and skimmed at the other Grid sites. Every analysis user can access to the nearest computer facility to analyze the datasets through the data management system explained in Section 1.3. In parallel with the Grid-based computing system, we are considering to incorporate Cloud Computing facilities for the MC production.

and operation is coordinated centrally. For efficiency and scalability reasons experiment-specific services will most likely be distributed globally to selected grid sites.

While ntuple-level analysis could be done on the grid, a fast turn-around time usually requires to have the ntuples available on resources local to the user. These local resources are ideally grid-enabled, but we explicitly include non-grid resources, like private clusters at institutes, desktops or laptops, for which we will provide means to install the software needed for ntuple analysis and for access to the Belle II grid system.

In order to clarify the tasks on each grid site, we categorize the computing centers. The classification into main center, grid, and local resources describes the tasks for which the resources are used. Physical sites may contribute to more than one task. For example KEK will provide resources for raw data processing, MC production and data analysis, and ntuple-level analysis. Figure 2 illustrates the relation between physical sites and their function.

1.2 Cloud Computing

An important fact we have to consider in the design of the computing model and the planning of resources is the variation of resource demands with time. There is a steady increase of required

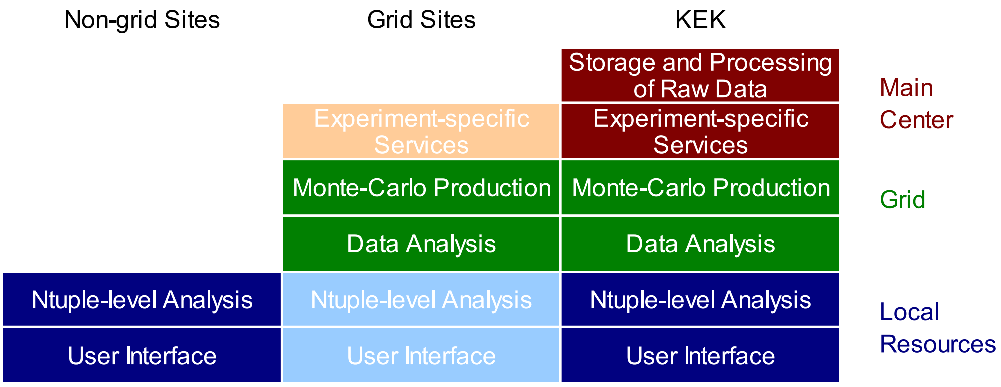

Figure 2: Tasks of computing facilities.

resources because of the growing data size. This is taken into account in the planning on a yearly basis. On top of this trend there are variations on a shorter time scale of months or weeks. One reason is that the raw data processing and the MC production can only start after a data taking period (which usually lasts several months) has finished and calibration constants have been determined. In order to have the data and MC ready for analysis in a timely manner it is advantageous to save CPU resources during the data taking phase and spend them when the processing phase starts. However CPU resources can not be easily saved because idle CPU time is lost and can not be regained. To some extent this issue can be solved by fair-share mechanisms when several virtual organizations share CPU resources at the same site. However, it is unlikely that the resource demands of all virtual organizations balance each other at all times.

Computing system are usually designed to match the expected peak demand. This has the risk of a resource shortage if the demand exceeds the expectation. Cloud computing can provide a solution to the issue of varying CPU resource demands. It offers the possibility to purchase CPU resources for a limited amount of time. This mitigates the problem of a CPU resource shortage due to unexpectedly high demand as financial resources can be transformed into CPU resources almost instantaneously.

As cloud computing is a relatively new technology, testing our concepts is needed prior to considering it as a solution. The results of our testing to date using the production of MC data on the Amazon Elastic Compute Cloud (Amazon EC2) are described below.

Table 1: Cost of full production runs on EC2.

		Cost in $				
Run	Number of Events	CPU	Storage	Transfer	Total	per 10^4 events
1	752,233	80.00	0.20	6.65	86.85	1.16
2	1,473,818	108.11	0.25	7.12	115.37	0.78
3	10,000,998	724.80	1.42	39.96	766.18	0.76

Table 1 summarizes the costs of the MC production on the cloud. It can be seen that the total cost is dominated by the investment in CPU cycles and that it scales quite well with the number of produced events. Recently, prices have dropped (as we expected) and we will continue to run more tests pushing the scale achievable upwards.

In contrast to publicly funded grid sites the cloud computing facilities are usually operated by commercial companies. This may change in the future as a clear trend towards cloud computing technologies can be observed in the grid computing development. However for now, we need to consider the avoidance of vendor lock-in with regards to data storage. Furthermore, we have to also design our software in a way that the technical issues of a vendor change are solvable with reasonable effort.

We see that this technology has a high potential and we feel obliged to keep the possibility of using it for Belle II. While our baseline computing resources will be provided by grid sites, cloud computing will be an option for peak demands in MC production or physics analysis.

1.3 Data Management

The data management system is essential to enable users to effectively and smoothly analyze datasets. If there is no reliable data management system, the files would be missed or processed twice, then the obtained result of the analysis is most likely wrong. Due to the limited resources and short time-line in the preparation of Belle II computing, we also adopt the key ideas of metadata and project structure from D0 and CDF experiments.

The basic concept of the data management shown in Fig. 3 is rather simple. When the user wants to start a project he/she calls the project client with a metadata query to identify the input files, a tarball containing the analysis code, and other necessary information to submit jobs. The project client then talks to AMGA, which allows to get a list of files, in form of GUIDs, that match the users' request and the LFC to get the list of matching SURLs. Based on the SURL it determines the sites where the data files are available. Then it submits the desired number of jobs to these grid sites. All jobs may be submitted to one site or they may be split to several sites.

To store the output file(s) on an SE and register them in the LFC and AMGA, a tool will be provided within the Belle II software. It selects either the nearest SE or a predefined one that the user specifies at job submission.

In addition to submitting the grid jobs, the project client will store the list of GUIDs, selected SURLs, and corresponding job identifiers in a local file. This allows to identify the project via the local project file.

Now the user can query the project client for the status of his jobs by providing the name of the project file. From the project file the client reads in the job identifiers and determines their status via standard grid mechanisms. The status of successfully finished and failed jobs will be recorded in the project file to avoid further unnecessary queries. Based on the status information in the project file the client can also easily either resubmit all failed jobs or generate a new project containing only the unprocessed files. This recovery mechanism simplifies the bookkeeping effort for the user enormously.

As an extension of this baseline system we investigate the idea of a dynamic assignment of input files to jobs, inspired by the success of this concept in the SAM [4] system employed by the CDF and D0 experiments. To realize this mechanism the project bookkeeping has to be moved from a local file to a central service, the project server. Now the project client just passes on the

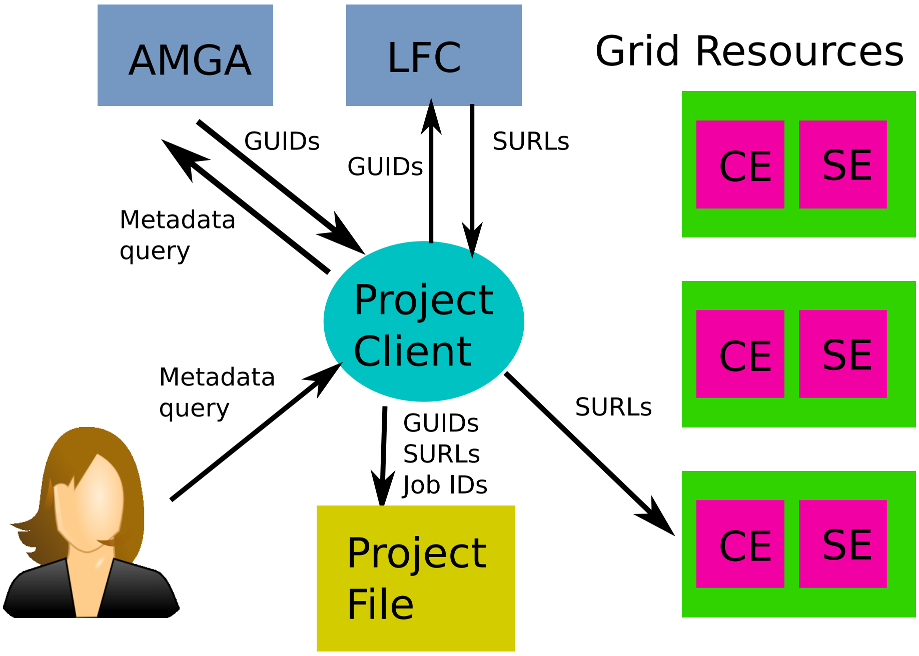

Figure 3: Outline of user interaction with the project client to submit jobs to grid sites for the analysis of a dataset defined by a metadata query. In order to submit jobs to the Grid site, user calls the project client with a metadata query to identify the input file, first. The project client then talks to AMGA, which provides the user with a list of files in the form of GUIDs, and gets the list of SURLs corresponding to the specified GUIDs from LFC. Based on this SURL it determines the sites where the data files are available.

metadata query to the project server when the user starts a project. The project server contacts AMGA and the LFC as the client did in the baseline model. Instead of a local file a database is used to store the list of GUIDs. The list is identified by a unique number, the project ID. It takes over the role of the project file name. The project server returns the project ID to the client together with a list of sites on which the data is available. The client prepares a single job script with the project ID in an environment variable. The desired number of jobs is submitted to the site (or sites) indicated by the project server with the same single job script.

In order to realize our concept, we have started making a test bed and now we are examining the feasibility of this system.

1.4 Summary

We will start the Belle II experiment in 2014, and the expected amount of data will be of the order of 10^{10} events per year. Due to the limited resources and short time-line in the preparation of the Belle II computing, it is necessary to design a technologically-feasible computing system including the data handling scheme as soon as possible. For the computing system, we adopt the grid-based distributed computing as a baseline, and cloud computing will be an option for peak demands in MC production or physics analysis. We are also considering integrating the concept of the project server which has been applied in D0/CDF experiments as a data handling system. We are now testing the idea of our data handling system. We documented the basic concept of our distributed computing as well as the data handling system in the Belle II Technical Design Report [5].

References

[1] A. Abashian, *et al*(Belle Collaboration), Nucl. Instrum. Methods. **A 479**, 117-232 (2002).

[2] M. Kobayashi and T. Maskawa, Prog. Theor. Phys. **49**, 652 (1973).

[3] E. Laure, S.M. Fisher, A. Frohner, C. Grandi, P. Kunszt, *et al* Comp. Methods in Science and Technology, **12**, 33-45, (2006)

[4] http://projects.fnal.gov/samgrid

[5] KEK Report 2010-1; http://arxiv.org/abs/1011.0352