BASF - Belle AnalysiS Framework

Ryosuke Itoh [a]

[a] *Institute for Particle and Nuclear Study, High Energy Accelerator Research Organization (KEK), 1-1 Oho, Tsukuba, Ibaraki 305, Japan*

An analysis framework for the BELLE experiment called BASF has been developed. The framework features the capabilities of the dynamic linking and the event-by-event parallel processing. The module for BASF can be written in any of Fortran, C and C++ with the unified data access by a memory management system called Panther. The scalability and I/O throughput of BASF was tested and proven to work for the BELLE experiment.

Key words: data analysis; framework; parallel processing; dynamic linking

1 Introduction

A common software framework to process event data is a "must" for the data analysis at HEP experiments. Such a framework is also required at the BELLE experiment which is the KEK's B-factory experiment. Followings are required for the framework at BELLE:

(i) Module and Path structure:
The original idea is from DSTANL by PEP4/TPC and the idea was extended by TOPAZ (FULL analysis system) and CDF (CDF analysis control package) independently. An analysis program is built by plugging a set of modules each of which has a relatively simple function. The path is used to determine the execution sequence of modules. The modules are placed on a path in the execution order. Multiple paths can be defined with the conditional branch to another path. The framework for BELLE is desired to have such a structure.

(ii) Integral data management:
The framework is desired to provide an unified method to access the event data from modules using a memory management software. This also provides the method to pass data between modules. The I/O capability for

data files is also provided and controlled by the framework. The framework also manages the histogram handling by modules with a histogram package.

(iii) Multi-language support:
The modules for the framework can be written in any of Fortran, C and C++. The modules written in different language have to be co-exist on the framework.

(iv) Integrated parallel processing mechanism:
A single job of BELLE data analysis requires at most 5000 VAX equivalent CPU power. To achieve this number, a parallel processing at some level is required since the processing power of a CPU is at most 500 VAX or so. Since most of the users of the framework are not experts in programming of parallel jobs, the parallel processing mechanism is desired to be implemented in the framework and hidden from the users.

(v) Flexible user interface:
"User's favirite" is widely different from the command line interface to the GUI. It is desired that the user interface is switchable according to the execution environment.

(vi) Runtime linkage of modules:
It is annoying to link all the modules and libraries with the framework whenever some modification is made to a module or library. It is desired to be able to choose the modules to be plugged into the framework at runtime.

BASF, which is the abbreviation of BELLE AnalysiS Framework, has been developed to satisfy these requirements.

2 Architecture

The global architecture of BASF is shown in Figure /refarch. BASF is made as a single program forking a multiple processes for the parallel processing when executed on a SMP-server. All the code was written in C++ based on object oriented design.

2.1 Module and Path

An analysis program consists of a set of modules plugged into BASF. A module is written as a class object in C++ with a set of member functions for the event processing. Modules written in Fortran and C are encapsulized to a class object by the wrapper functions.

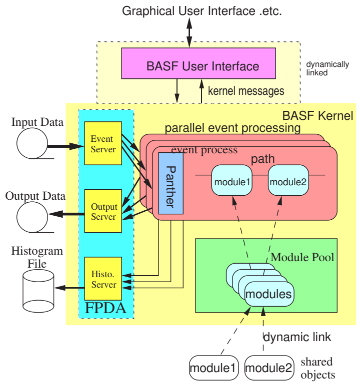

Fig. 1. Schematic view of BASF architecture

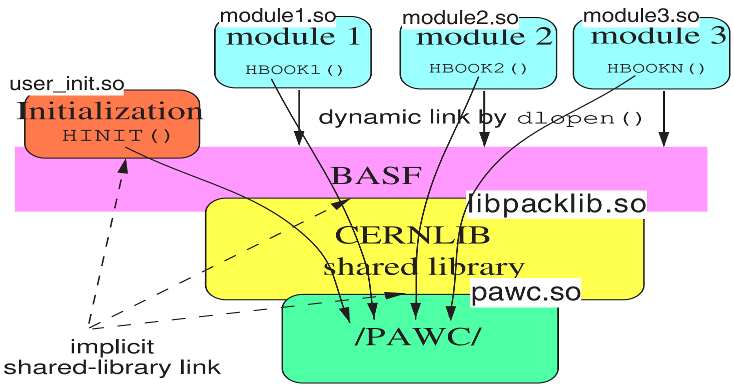

Fig. 2. Dynamic linking by BASF

The execution sequence of modules is defined by creating a path. A path is defined as a chain of modules with a condition descriptor in which conditional branches to other paths can be defined. Each path has a status variable which can be modified by each module in the path, and the conditional branches are defined with respect to the status.

2.2 Dynamic link of modules

The modules are linked with BASF by the dynamic link. A module is made as a shared object and the BASF links the module when requested at run time using `dlopen()` system call. Common utility libraries such as CERNLIB, Data Management Package, etc are also made as shared libraries and the external references from modules are resolve at run time.

The global initialization function and the user interface subsystem are also dynamically linked at run time so that the users can modify the initialization behavior and can choose their own user interface.

Figure 2 shows the structure of the dynamic linking by BASF.

2.3 Data Management

A home made memory management system called Panther[1] is employed for the data management in BASF. To cope with the mixed language environment, non object oriented approach was taken. Panther is based on the entity relationship model like ADAMO[2] and the data structure is given by the data definition language. For the histogram accumulation, HBOOK4 is used as the histogramming package.

The data file I/O is implemented inside BASF and is managed using the I/O functions of Panther. To improve the I/O performance, the event data are staged in a large memory buffer and the access to event data from the analysis program is done through the buffer. The actual I/O with devices is done asynchronously by the double buffering method with a large block length.

2.4 Parallel processing support

BASF supports the event-by-event parallel processing facility on an SMP-server. The mechanism is based on the FPDA system reported at CHEP95 [3]. The main program of BASF spawns off multiple event processes using `fork()` after the initialization procedure with several server processes (event server, output server, histogram server, etc). When a bunch of events are read from a data file, they are first placed in the buffer on the shared memory space. Then pointers to event data on the shared memory are passed to event processes by the event server. On the SMP-server, the event processes are executed on multiple CPU's processing different event simultaneously and therefore the event-by-event parallel processing can be performed. The event output is also handled in the reverse sequence by the output server. The histograms/N-tuples in each event process are accumulated independently in the local histogram buffer of the event process. When the processing is completed or the histogram buffere becomes full, the buffer contents are sent to the histogram server by `msgsnd()` call. The histogram server collects the histogram contents from all the event processes and write the integrated contents onto a histogram file. This can be done during the event processing also by sending a signal to the histogram server so that the user can "spy" the histogram contents.

When BASF is executed on a single CPU machine, BASF does not fork off event processes, instead, it runs as a single process program. This enables the use of the debugger for the program development.

For Monte Carlo simulations using parallel processing, BASF spawns off the seed server instead of the event server to provide consistent random number seed to each event processes.

2.5 User Interface

To realize the "switchable" user interface, the user interface subsystem is completely separated from the kernel of BASF. The BASF kernel works as a message-driven server to the user interface subsystem. The message to control the kernel consists of simple string messages. The user interface subsystem is built as a separate shared object and linked with BASF at run time. This enables to choose the user interface by each user and to develop a new user interface without changing BASF itself.

3 Performance of BASF

BASF is currently working on Sun UltraSparc with Solaris 2.5.1, DEC Alpha with Digital UNIX 4.0 and Pentium with Linux 2.0.x.

The performance of BASF was tested by measurering the scalability of parallel processing and the maximum I/O throughput on a Sun UltraEnterprise 6000 server system with 28 CPU's. A SONY DIR-1000M and DTF tape drives are connected to the server for the input and output devices, respectively.

Scalability was tested by running three different application programs implemented on BASF. One is the fast detector simulation of the BELLE detector together with the event generator. The output of the simulation was fed into an simple analysis module which accumulates some histograms. These three components are made as different modules and plugged into BASF by the dynamic link. The second is the GEANT simulation of BELLE. This is also made as a module and put into BASF together with the event generator module. The output of GEANT was written into the SONY DIR-1000 tape. The third is the preliminary DST production program. The output of the second program was used as the input and the cluster reconstruction of calorimeter was done which is made as a module for BASF. The results were written to a DTF tape. The typical event size of one input event is 50KB while the size was 70KB for output.

Fig. refscale shows the measured scalability for these three applications. The relative performance was calculated by the elapsed time of the application measured as a function of number of event processes. As seen from the figure, the scalability was confirmed to be ensured up to the expected number of event processes for all the applications.

The I/O throughrate was measured by running BASF without plugging any modules. The data were read from a DIR-1000 tape and distributed to 25 event

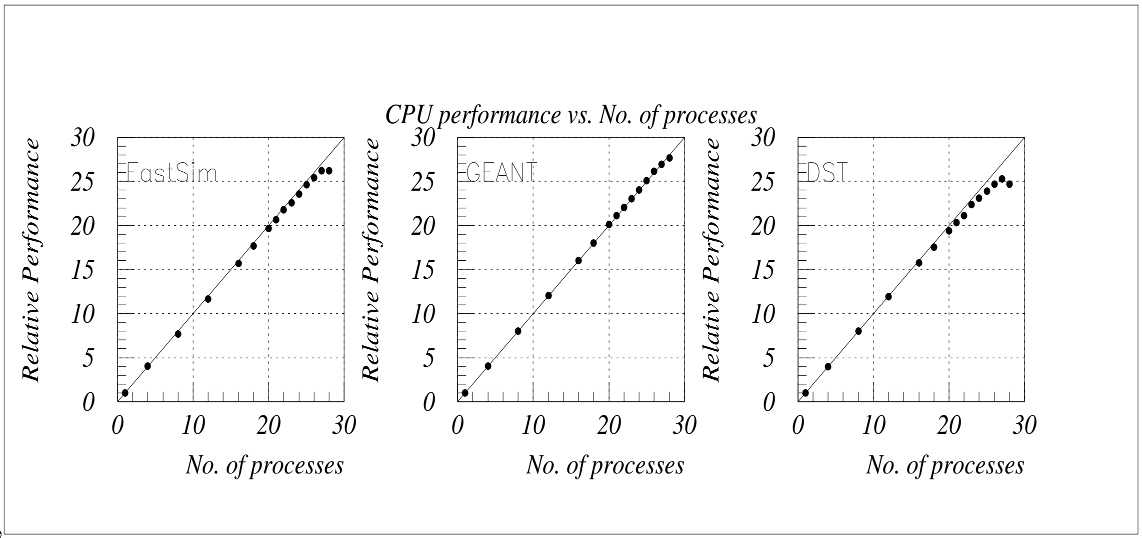

Fig. 3. Scalability of parallel processing

processes. The data were then copied to the output buffer and written to a DTF tape. The measured throughrate was 15.7MB/sec without the output turned on while 9.8MB/sec with the output. These values are consistent with the maximum transfer rate of the I/O devices (DIR-1000 and DTF).

4 Summary

- The analysis framework for the BELLE experiment, BASF, has been developed.
- BASF has following features; 1) Module and Path structure, 2) dynamic linking of modules, 3) support for Fortran, C and C++ programming, 4) unified data access method by Panther, 5) integrated event-by-event parallel processing capability on the SMP-server, 6) separated user interface subsystem.
- The scalability of the parallel processing of BASF was tested on SUN Ultra-Enterprise 6000 with 28 CPU's with three different cases, namely, the fast detector simulation, the GEANT simulation and the DST produciton. The scalability was confirmed to be kept up to the expected number of event processing processes.
- The I/O throughrate of BASF was measured using high speed tape drives for the input and output. The measured throughrate was consistent with the maximum transfer rate of tape drives.

References

[1] S.Nagayama, "Panther Reference manual", BELLE-note 131 (1996)

[2] S.M.Ficher and P.Palazzi, Comp. Phys. Comm. **57**, 169 (1989)

[3] R.Itoh and S.Ichizawa, Proceeding of the International Conference on Computing in High Energy Physics'95, 733 (1996)