Journal of Physics: Conference Series **396** (2012) 012029 doi:10.1088/1742-6596/396/1/012029

Belle-II High Level Trigger at SuperKEKB

S. Lee[1], R. Itoh[2], T. Higuchi[2], M. Nakao[2], S. Y. Suzuki[2], and E. Won[1]

[1] Department of Physics, Korea University, 1 Anam-dong, Seounbuk-gu, Seoul, Republic of Korea

[2] IPNS, High Energy Accelerator Research Organization, 1-1 Oho, Tsukuba, Japan

E-mail: shlee@hep.korea.ac.kr

Abstract. A next generation B-factory experiment, Belle II, is now being constructed at KEK in Japan. The upgraded accelerator SuperKEKB is designed to have the maximum luminosity of 8×10^{35} cm^{-2}s^{-1} that is a factor 40 higher than the current world record. As a consequence, the Belle II detector yields a data stream of the event size $\sim$1 MB at a Level 1 rate of 30 kHz. The Belle II High Level Trigger (HLT) is designed to reduce the Level 1 rate to 1/5 by performing the real time full event reconstruction and by applying the physics level event selection as the software trigger. In this paper, the development of the high level trigger system for Belle II and its performance is discussed.

1. Introduction

Although the Belle experiment [1] has solved many puzzles in the B physics area, the statistical uncertainty is still high. In order to search more CP violation sources, to test the Standard Model, and to search for evidences of New Physics, a larger statistics is necessary. In this context, the Belle II experiment [2] at the SuperKEKB accelerator is designed and planned to start in 2014. The target luminosity of Belle II is 8×10^{35} cm^{-2}s^{-1} which is 40 times larger than the luminosity recorded by Belle. As the luminosity increases, the amount of data taken from the detector gets huge, therefore, the online event processing becomes a computational challenge.

2. The Belle II High Level Trigger (HLT) system

The Belle II HLT system triggers the event data that is taken from the Belle II detector in software. It performs a full physics skim such as hadronic event selection so that uninterested events are discarded. Because software for the HLT system is intended to be the same with the offline software which is described in further section, the same event reconstruction codes are used consistently over the experiment.

The major role of the Belle II HLT system is not only the event selection but also passing information of selected events such as hits associated with reconstructed tracks to pixel detector readout processor. This feature enables us to reduce not only the event rate but also the event size at the pixel detector readout processor.

The input rate to HLT is expected to be 30 kHz with the size of 100 kB for one event so that 3 GB of data will come into HLT for every seconds. The goal of the HLT system is factor of 1/3 to 1/5 data reduction in order to the output to be less than 1 GB for every seconds.

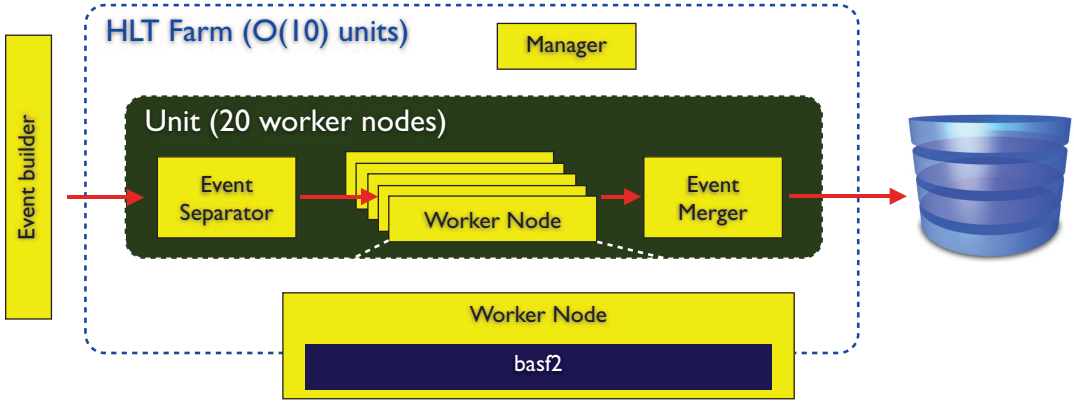

Figure 1. Illustration of the Belle II HLT system. The system consists of $\mathcal{O}(10)$ units and a unit contains 20 worker nodes that process the event data. The data taken from detector is reduced at sub-detector triggers first and event builder collects them to full events. The event data are passed to the HLT system and distributed to worker nodes by the event separator. After event processing, the event data are collected by event merger and sent to online storage.

The HLT system consists of $\mathcal{O}(10)$ units. A unit contains 3 special kinds of nodes and 20 worker nodes which are supposed to process the actual event data with multicore CPUs. At the beginning of the experiment, 2,000 CPU cores of 3 GHz clock are expected to be used in the HLT farm. The Fig. 1 demonstrates the Belle II HLT system.

3. The basf2 and hbasf2 frameworks

The basf2 is the software framework for Belle II [3] that is supposed to be used in both offline and online event processing simultaneously. It is designed to have a software pipeline architecture so that a large scale event processing can be done by combining a set of small modules for specific purpose. This modular structure makes the framework flexible and easy to maintain. The basf2 supports not only single processing but also parallel processing utilizing multicore CPU technology. But the basf2 is thought to be used in a single PC for general purpose so that there should be an extension for parallelization over the network to PC servers aiming to HLT purpose.

In this context, the hbasf2, the HLT software framework for Belle II, is being developed. The hbasf2 is a super-framework which wraps the basf2 framework. Since it runs the basf2 for the actual event processing inside, the flexible structure of the framework is still kept consistently. The main features of the hbasf2 are multi-node based parallelization and node management in the HLT farm as described in further sections.

In a global point of view, the basf2 framework can perform both single and multicore based parallel processing and the capability of parallel processing can be extended to multi-node based and grid by additional packages of hbasf2 and gbasf2, respectively.

4. Node management

Since the HLT farm consists of many PC servers which have particular roles as described in previous section, node management is an important feature of hbasf2. For the global management of nodes, there is a special node called manager node which is in charge of node initialization and management.

Once an input file that contains global HLT farm formatted as XML is provided from outside of the system, the manager node parses it and distributes individual node information to proper

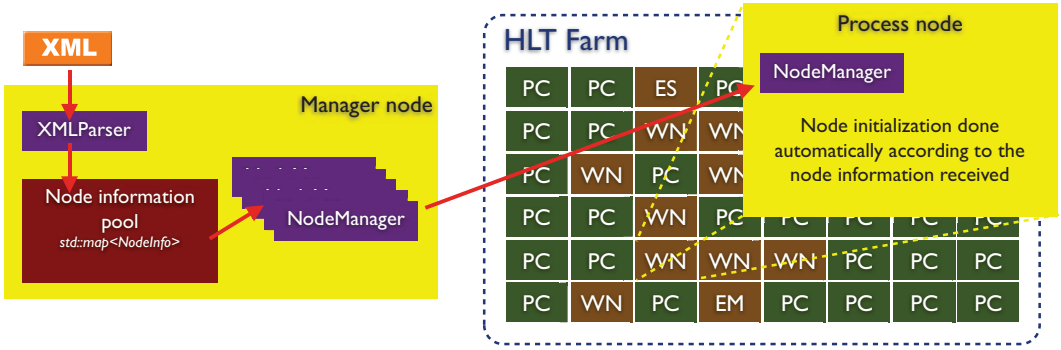

Figure 2. Input XML file contains the all information of HLT farm. Once XMLParser parses the XML file, node information are stored into internal node information pool in manager node. The individual information are sent to particular nodes by NodeManager and the nodes in HLT farm initialize themselves according to the information received.

nodes as described in the input file. The other nodes accept particular node information and initialize themselves accordingly. The node behaviors are predefined as PYTHON scripts same as a typical execution of basf2. The node management system is illustrated in Fig. 2.

5. Data flow

Between two nodes, the hbasf2 uses standard TCP socket for the data communication. The C++ style implementation has been developed and named B2Socket as a base class. Inherited classes from B2Socket, HLTSender and HLTReceiver, provide interfaces for the data transfer. Since they are separated processes from the framework even though controlled by the framework, there are shared memory based FIFOs (First-In-First-Out) for the interprocess communication.

The B2Socket based data communication is used not only for node information distribution described in previous section but also for the actual event data distribution. Because B2Socket utilizes the standard TCP socket, the data should be serialized for the communication. The event data are stored as objects at the framework level but serialized by using TMessage of ROOT [4] for the communication while the node information are serialized as plain text.

The interface between HLTSender/HLTReceiver and the basf2 framework is provided by modules, HLTInput and HLTOutput, specialized to this purpose. These modules access FIFOs for the data communication. The data flow between nodes in the HLT system is summarized in Fig. 3.

6. Performance tests

The hbasf2 framework has been implemented as discussed in previous sections and its performance has been tested in various ways although there are still enough rooms for the improvement.

6.1. Test environment

The performance tests has been done with a prototype of test bench. The test bench consists of 7 PC servers which have Intel Xeon CPUs and 24 GB memory for each. All machines are connected over 10 Gb ethernet and run on Scientific Linux 5.5 (kernel version of 2.6.18, x86_64). Although the test bench already has multicore capability, the tests has been done without using them in order to bound the performance to the multi-node parallelization.

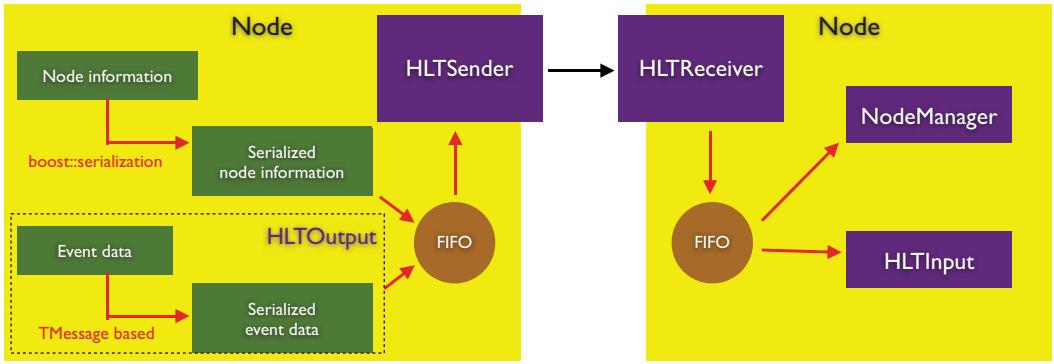

Figure 3. Data flow in the HLT system. There is only one FIFO drawn in the left node and it seems that two separated flows of management and data share the FIFO but they use different FIFOs in the actual implementation.

6.2. Unit tests

Before the global test of the framework, the components of the framework have been tested as unit tests such as B2Socket test for the data communication, FIFO test for test to the interprocess communication, and test of I/O module specialized to HLT purpose as a file I/O test.

From the B2Socket unit test, the transfer rate is evaluated as 392.4 MB/s for 1.4 GB data transfer. Although theoretical rate is up to 1.25 GB/s in the test environment, the performance of B2Socket is still reasonable. The performance is expected to be even better by further TCP tuning of the test bench in future.

For the FIFO unit test, we made three different situations that happen in three different kinds of nodes - event separator (one-to-many), worker node (one-to-one), and event merger (many-to-one). In worker nodes, a FIFO is shared by only one HLTSender/HLTReceiver and only one HLTOutput/HLTInput, respectively. In event separator, a FIFO is shared by one HLTOutput and many HLTSenders because the connection between HLTSender and HLTReceiver implies a point to point connection. On the other hand, a FIFO is shared by many HLTReceiver and one HLTInput in event merger. From the unit test of the FIFO in different situations, the performance of FIFO is evaluated as shown in Table 1. The performance is high enough as we expected.

In the reality of the HLT operation, all data are given by former component (event builder) and passed to online storage over the network. But for the HLT system test, the I/O of data relies on the local hard disk because there are no event builder and online storage available yet.

Table 1. FIFO performance. The cases listed are; one-to-one is that a process writes and another process reads; one-to-many is that a process writes and 4 processes read; many-to-one is that 4 processes write and a process reads, simultaneously.

Cases	Transfer rate (GB/s)
one-to-one	2.34
one-to-many	2.21
many-to-one	1.60

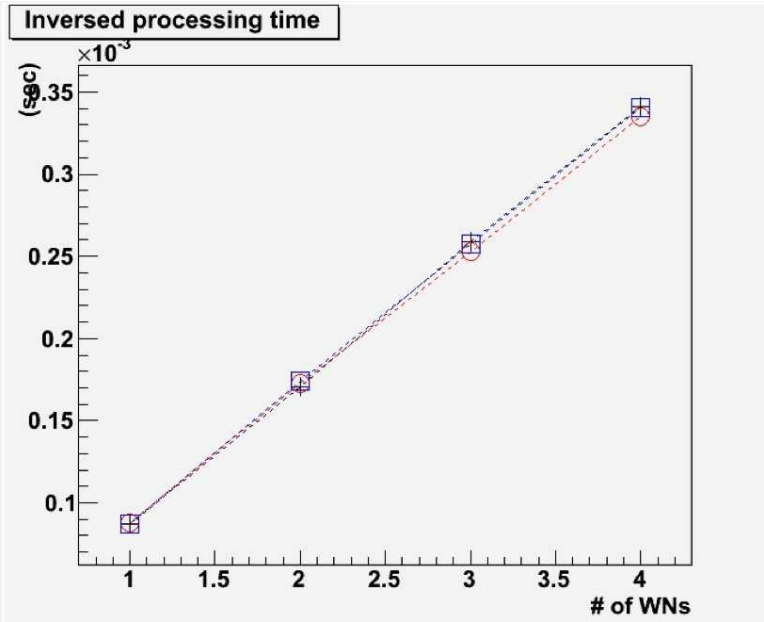

Figure 4. Linearity test for the performance up to 4 worker nodes. The vertical axis is inverses of elapsed time for arbitrary tests. The elapsed time for multiple worker nodes is averaged and the plot shows three tests altogether. Since arbitrary event processing has been used, the absolute value of this test does not represent the absolute performance of HLT system.

For this purpose, we made specialized I/O modules that read and write as serialized data. The I/O modules, SeqRootInput and SeqRootOutput, certainly make overheads during the test so that the overheads should be analyzed as well. From the test, the input rate of the module is measured 110.1 MB/s for data size of 416 MB. The output rate exceeds the input rate although the output rate is measured indirectly because it is hard to decouple the SeqRootOutput from the framework.

6.3. Maximum performance tests

After confirming the performance of each components, the maximum performance of the framework has been carried out in two steps. The first step is intended to represent ideal case and the other is to represent more realistic case.

For the ideal case, we remove hard disk I/O and data serialization. Event separator generates 5 MB dummy data and sends it 1,000 times to a worker node. Worker node only passes received data to event merger and event merger only receives them. From the test the ideal performance is measured 192.3 MB/s. The performance is thought to be bound to the transfer rate of B2Socket but the measured maximum performance is less than the rate of 392.4 MB/s measured in the unit test because the size of data is different. The measured rate of 392.4 MB/s is for 1.4 GB data with one operation but the data size is 5 MB for an operation in this test. Considering the dependance of transfer rate on data size, the measured maximum performance is believed to be reasonable.

For more realistic case, test data is read from local hard disk using SeqRootInput module discussed already and is written to the disk back using SeqRootOutput module. In addition, data serialization is also implied in all nodes. For the test, we used pre-generated test data of 5 kB size per event that is expected to be similar to the event data used in the real experiment. In this test, the maximum performance is measured to be 40.7 MB/s. The main source of the

overhead in this test is the serialization. The improvement of serialization is in progress so that the performance will get better if the optimized version is applied to the test in future.

6.4. Linearity test
The performance of the framework should be naturally increased as the number of worker nodes increases. For the linearity test, we configured that event separator generates events and worker nodes simulate them in the Belle II detector configuration designed so far. Although the test doesn't represent the reality of HLT, the linearity can be confirmed because the worker nodes actually consume processing time. As shown in Fig. 4, the linearity of the framework has been confirmed up to 4 worker nodes.

7. Summary
The Belle II HLT system is a software trigger that reduces event data to manageable level by the Belle-II experiment. A software framework for the HLT system, hbasf2, is designed and implemented. Although there is explicit software framework for HLT, the core framework for event processing, basf2, is shared in both online and offline so that the same software can be used in both sides. The hbasf2 supports node management in the system and data communication over network which enables multi-node based parallelization. The fundamental implementation of hbasf2 has been already developed and its performance is confirmed to be acceptable. The framework will be more efficient with further improvements including the optimization of object serialization and combination with multicore based parallelization.

Acknowledgments
We acknowledge support from KREONET/GLORIAD network, and by NRF Grant No. 2011-0030865.

References
[1] Abashian A *et al.* 2002 *Nucl. Instr. and Meth.* A**479** 117-232
[2] Abe K *et al.* 2010 Belle II Technical Design Report *Preprint* physics/1011.0352
[3] Moll A 2011 *J. Phys.: Conf. Series* **331** 032024
[4] Rademakers F and Brun R 1998 *Linux Journal* **51**