Infiniband Event-Builder Architecture Test-beds for Full Rate Data Acquisition in LHCb

Enrico Bonaccorsi and Juan Manuel Caicedo Carvajal and Jean-Christophe Garnier and Guoming Liu and Niko Neufeld and Rainer Schwemmer
CERN, 1211 Geneva 23, Switzerland
E-mail: `jean-christophe.garnier@cern.ch`

Abstract. The LHCb Data Acquisition system will be upgraded to address the requirements of a 40 MHz readout of the detector. It is not obvious that a simple scale-up of the current system will be able to meet these requirements. In this work we are therefore re-evaluating various architectures and technologies using a uniform test-bed and software framework.

Infiniband is a rather uncommon technology in the domain of High Energy Physics data acquisition. It is currently mainly used in cluster based architectures. It has however interesting features which justify our interest : large bandwidth with low latency, a minimal overhead and a rich protocol suite.

An InfiniBand test-bed has been and set-up, and the purpose is to have a software interface between the core software of the event-builder and the software related to the communication protocol. This allows us to run the same event-builder over different technologies for comparisons.

We will present the test-bed architectures, and the performance of the different entities of the system, sources and destinations, according to their implementation. These results will be compared with 10 Gigabit Ethernet testbed results.

1. Introduction

LHCb [1] is one of the four large experiments at CERN which takes data from proton-proton collisions at the LHC. Data flows from hardware readout boards (TELL1 [2]) through a large Gigabit Ethernet network to an Event-Filter Farm (EFF) of 2000 nodes running the selection of the physics events [3]. The total event size, determined by the average number of electronics channels activated by a collision, is quite small: 60 kB according to what was observed during the first year. The data acquisition and data handling in LHCb are hence faced with a very high rate of rather small events. The current readout rate is 1 MHz and is planned to be upgraded to the collision rate of 40 MHz by 2016. This means that there would be no more rate reduction before the EFF. Because not all LHC beam-crossings contain particles, the event rate in the Event-Builder will be 30 MHz. Each event out of the 30 MHz will be about 100 KiB, taking into account the non-Gaussian distribution of fragment sizes from the readout-boards. An "event" is the aggregation of fragments. We expect around 1000 sources (corresponding to the readout-boards in the current system). An average fragment size would then be 100 bytes per source. We intend to use fragment coalescing to reduce the message rate using a protocol called Multi Event-fragment Protocol (MEP) [4]. Most of these figures are assumptions and may change

Table 1. Effective theoretical IB maximum data rates

	SDR	**DDR**	**QDR**	**FDR**	**EDR**
1X	2 Gbit/s	4 Gbit/s	8 Gbit/s	14 Gbit/s	25 Gbit/s
4X	8 Gbit/s	16 Gbit/s	32 Gbit/s	56 Gbit/s	100 Gbit/s
12X	24 Gbit/s	48 Gbit/s	96 Gbit/s	168 Gbit/s	300 Gbit/s

until the year 2016.

We are currently investigating two interesting industrial standard technologies with major commercial companies: 10 Gigabit Ethernet (GbE) and InfiniBand (IB). We will focus here on the InfiniBand technology, and how we could implement an Event-Builder compatible with both 10 GbE and IB. The aim is to find the optimal event-building protocol over either 10 GbE or IB. We are currently running a push protocol, i.e. sources send data to destinations regarding a very simple control protocol. The control protocol consists of recording credits per destinations, and consuming them as events are sent. If a farm destination has no credits, it is not eligible for more events. In this push protocol, the control traffic is very low compared to the useful data flow. The second protocol that we want to test is a pull protocol.

Any of these push and pull protocol can come in more sophisticated development. For instance, they can be implemented using a barrel shifter in order to insert a delay between frames, in order to do not overload links and buffers. Another possibility is to implement a partial pull protocol. The destination farm asks for only a subset of the event, to get a partial picture in order to perform a first analysis. When it considers the event interesting, it asks for the full picture, reading all fragments from all sources. A partial readout could be implemented using a mix of both push and pull protocols. Only a few sources would send their data to a destination, pushing them. If the destination finds the partial picture interesting, then it requests the complete picture pulling it. These protocols are implemented as the Application layer of the OSI model. They should not be dependent on the underlaying protocols, which could be InfiniBand or 10 GbE in the case of our upgrade.

Section 2 presents InfiniBand characteristics and its software stack. Section 3 presents the test-bed. Section 4 presents the study and implementation of Event-Builders with various protocols. Section 5 presents the results.

2. InfiniBand presentation

IB is used in High Performance Computing (HPC), for large cluster interconnect, and the interest of its usage in High Energy Physics is growing. It currently provides a maximum bandwidth of 40 Gb/s while an impressive bandwidth increase is foreseen for the next few years 1. Acknowledging such perspectives, it is an interesting technology to develop next generation Data Acquisition and Event Builder systems, challenging 10 GbE.

A 40 Gb/s event-building network is very interesting for our purpose and should match our requirements. IB latency is low. It is up to six time lower than for 10 GbE. Latency is the transit time between source and destination processes. It is not an issue in the case of a push protocol if buffers are large enough, as once the event is read out from the detector, the building and processing are not real-time tasks. It is however very important in the case of pull protocols. The farm requesting for an event is idling waiting for every fragments of the requested events. The higher latency is, the more the CPU time of the farm is wasted during this idle state. As long as latency is low, buffering in InfiniBand devices, switches and Host Channel Adapters (HCA), is not very large.

InfiniBand is a switched fabric topology, contrary to Ethernet which is a hierarchical switched

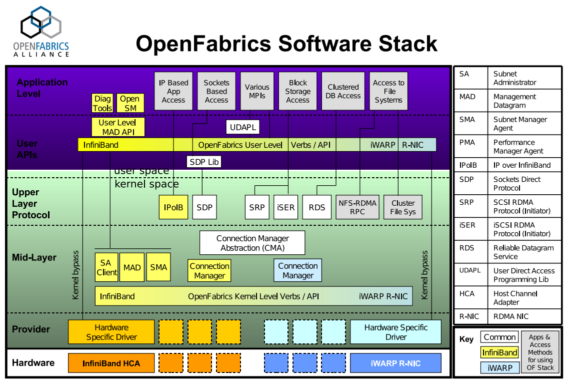

Figure 1. The OFED software stack. On the left and in blue boxes are InfiniBand tools.

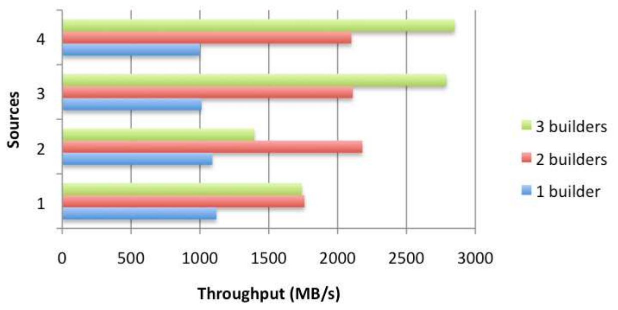

Figure 2. Aggregate throughput of the Event-Builder processes with a varying number of sources and destination, and a packet size of 100 kB.

Table 2. Basic performance test results

	RDMA read	RDMA write
Bandwidth average (MB/s)	2081	2581
Bandwidth peak (MB/s)	2248	2589
Min latency (μs)	8.88	4.74
Typical latency (μs)	10.13	5.03
Max latency (μs)	135.6	67.49

network. This mainly means that switches are usually connected with several uplinks in order to guarantee an homogeneous bandwidth for the entire network. Traffic is spread over several links. Topologies that are usually used in HPC are fat-trees, mesh or torus. Such topologies might not be useful for data-flows that characterize data acquisition networks where data flows top-down.

The open source standard is the OpenFabrics Enterprise Distribution (OFED$^{\mathrm{TM}}$) [5]. Figure 1 presents the software stack. It is made of user level protocols, kernel level protocols, Remote Direct Memory Access (RDMA) services and by-pass mechanisms. The OFED purpose is not only dedicated to InfiniBand and it provides tools for other link technologies. Many protocols and mechanisms are interesting to develop and Event-Builder. After a preliminary study, a few were selected and studied more carefully, implementing at first push event-builders.

3. Test-bed configuration

The test-bed is made of eight commodity servers interconnected with a QLogic 12300 InfiniBand switch. It features 36 4x QDR ports. The line rate is about 32 Gb/s. The test-bed network topology is very simple, each node is connected to one port of the switch. All the InfiniBand traffic passes through the switch. Our commodity servers are Dell R710, with Intel®Xeon®E5520 processors. They are two processors of four cores, with a clock of 2.27 GHz. They are equipped with QLogic QLA 7340 HCAs, which are QDR 4x HCAs. The theoretical bandwidth of each link of our test-bed is therefore 32 Gb/s, or 4 GB/s. The operating system configuration was optimized as usual for the use of high-speed networking devices, enlarging buffer sizes to 16 MB. The fabric is configured with the OFED 1.5.2. The OFED provides a few tools to perform basic RDMA tests, and monitoring of the system. Table 2 shows the performance of RDMA read and write over InfiniBand. These tools are implemented with verbs, which is an API allowing to access directly to the HCA.

Although the test-bed is very basic, it is enough to start testing InfiniBand and to implement different Event-Builders. The results that we can obtain with this will only give an idea of how InfiniBand can perform by extrapolation. It requires to be verified on larger systems, for example renting a cluster from a HPC center.

4. Event-Builder Test-bed Implementation

The selected technologies for studying InfiniBand Event-Builders are IP over InfiniBand (IPoIB) [6], Socket Direct Protocol (SDP) [7], Reliable Datagram Socket (RDS) [8], Message Passing Interface (MPI) [9] and Verbs [10].

IPoIB consists of using the TCP/IP stack on top of the InfiniBand frames. It is a by-pass mechanism of the Kernel standard API. If system calls are performed on the InfiniBand interface, IPoIB will be used automatically. It means basically that a TCP/IP application running on Ethernet networks can run on InfiniBand without any change in the source code. It is entirely compatible. It however adds extra layers in the network access, and this extra complexity has a cost in term of performance.

SDP was first designed for InfiniBand and it is now a generic transport protocol for RDMA fabrics. It provides a by-pass mechanism for TCP stream sockets, so that an application which uses TCP sockets over Ethernet networks can use this protocol with no change in the source code. It ensures good portability for application sources.

RDS is a connectionless protocol, which offers reliable datagram transfer. It was first implemented in the OFED and it has been included in the Linux kernel 2.6.30. RDS is based on IPoIB. Its main advantage is the reliability. LHCb's concern is that Event-Builder sources cannot implement connected protocol because they are hardware readout boards. Datagram protocols are the only solution and bringing reliability to physics data transfer can be interesting.

IPoIB, RDS and SDP usages are the same, and they are available both for InfiniBand and for Ethernet. It means that to test these three protocols, it is not required to implement an interface between core software and network layers.

MPI has not been investigated yet but it is an interesting field. Vendors affirm that it is the way to get the best performance. But implementing an Event-Builder using MPI is not very interesting currently as our readout boards cannot run complex protocols. It will be tested anyway for comparison, and if the advantage is clear it is still possible to go to a two stage Event-Builder, using an interface between the hardware readout boards and the destination farm.

All the previous tools are nice for their portability or their friendly way of programming because any programmer is used to sockets to manage protocol stacks. However they all bring layers and interfaces on top of InfiniBand and impact performance.

InfiniBand features at the link layer are very interesting. Verbs refer to a list of functions defined by the OFA which must exist in any InfiniBand device. A common set of verbs have been adopted by most of InfiniBand vendors. It is a very powerful interface as it allows to have a direct interface between application and the HCA. It provides a full access to every InfiniBand functionality.

The CBM experiment at GSI already carried out a very interesting work with InfiniBand and verbs [11, 12]. They accepted to share with us their software so we could compare our results. They designed their test-bed using Mellanox technologies, but they already performed tests on

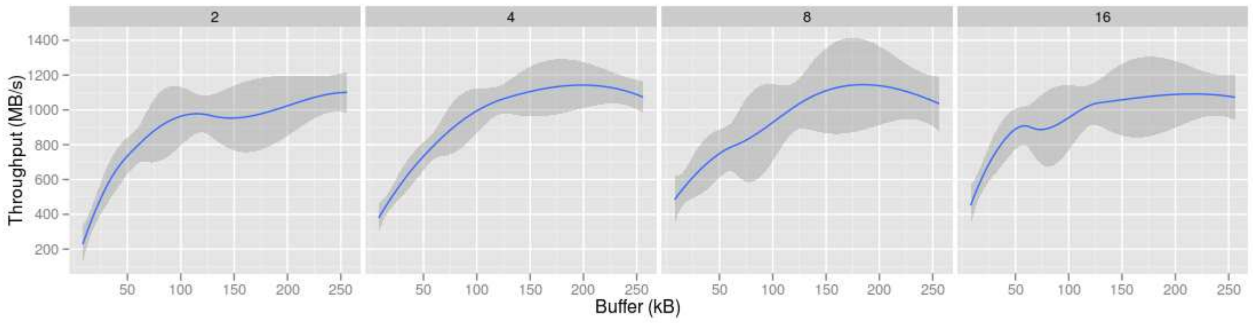

Figure 3. Throughput, per queue, of the CBM verb software, each plot are for respectively 2, 4, 8 and 16 sending queues, receiving queues are each time the double, 4, 8, 16 and 32.

QLogic clusters so this software is portable. This software is very thorough and covers many aspects of networking and event-building.

5. Results

In order to test IPoIB and to go further than the results presented by Domenico Galli in Real Time 2010 [13], a push protocol Event-Builder has been implemented based on datagram sockets.

The event-builder is very simple as the aim is currently to characterize network performance. They were unfortunately not as good as expected for a 32 Gb/s network, and network links were showing a lot of congestion. In the best conditions, the total throughput over the network hardly reaches 3 GB/s, as shown in figure 2. IPoIB implements a nice compatibility to the socket interface, but this is an overhead in complexity and performance can only be lower.

Testing the CBM verb implementation, figure 3 shows much more interesting results. Using large buffers, throughput is exceeding 1 GB/s per node. Hence we observed a total network throughput of about 8 GB/s. CPU usage remains low, varying around 12.5%, i.e. one CPU per node is fully used for networking operations. It shows that a verb implementation can give good results.

6. Conclusion

It is only the beginning of the LHCb InfiniBand study. IPoIB and verb implementations have been tested. The first results tend to show that a specific implementation of the Event-Builder, made out of verbs, is the best solution in comparison to IPoIB performance. Other protocols like SDP and RDS have to be tested, and vendors uphold that MPI, and more particularly their optimized MPI, gives the best performance.

Acknowledgments

This research project has been supported by a Marie Curie Initial Training Network Fellowship of the European Community's Seventh Framework Programme under contract number (PITN-GA-2008-211801-ACEOLE).

References

[1] The LHCb Collaboration A A J *et al.* 2008 *JINST*
[2] Bay A, Gong A, Gong H, Haefeli G, Muecke M, Neufeld N and Schneider O 2006 *Nuclear Instruments and Methods* **A560** 494–502

[3] LHCb HLT homepage http://lhcb-trig.web.cern.ch/lhcb-trig/HLT
[4] Jost B and Neufeld N Raw-data Transport Format Tech. Rep. EDMS 499933
[5] OpenFabrics Enterprise Distribution http://www.openfabrics.org/OFED-Overview.html
[6] Internet Protocol over InfiniBand http://infiniband.sourceforge.net/NW/IPoIB/index.htm
[7] Socket Direct Protocol http://www.openfabrics.org/archives/aug2005datacenter/das_SDP_Linux.pdf
[8] Reliable Datagram Socket http://oss.oracle.com/projects/rds/
[9] MPI over InfiniBand http://mvapich.cse.ohio-state.edu/
[10] Mellanox Technologies, Inc *Mellanox IB-Verbs API (VAPI)* http://www-linux.gsi.de/ mbs/fair/docs/MellanoxVerbsAPI.pdf
[11] Adamczewski J, Essel H G, Kurz N and Linev S 2007 *Proc. International Conference on Computing in High Energy and Nuclear Physics* (Victoria, BC)
[12] High performance data acquisition with InfiniBand http://dabc.gsi.de/dabc/doc/HK5.6-Linev-IB.pdf
[13] Galli D *et al.* 2010 *Proc. 17th IEEE NPSS Real Time Conference* (Lisboa, Portugal)