FERMILAB-CONF-15-206-E

# A New Event Builder for CMS Run II

**K Albertsson[2], J-M Andre[5], A Andronidis[2], U Behrens[1], J Branson[4], O Chaze[2], S Cittolin[4], G-L Darlea[6], C Deldicque[2], M Dobson[2], A Dupont[2], S Erhan[3], D Gigi[2], F Glege[2], G Gomez-Ceballos[6], J Hegeman[2], A Holzner[4], R Jimenez-Estupiñán[2], L Masetti[2], F Meijers[2], E Meschi[2], R K Mommsen[5], S Morovic[2], C Nunez-Barranco-Fernandez[2], V O'Dell[5], L Orsini[2], C Paus[6], A Petrucci[2], M Pieri[4], A Racz[2], P Roberts[2], H Sakulin[2], C Schwick[2], B Stieger[2], K Sumorok[6], J Veverka[6], S Zaza[2] and P Zejdl[2]**

[1] DESY, Hamburg, Germany
[2] CERN, Geneva, Switzerland
[3] University of California, Los Angeles, California, USA
[4] University of California, San Diego, California, USA
[5] FNAL, Chicago, Illinois, USA
[6] Massachusetts Institute of Technology, Cambridge, Massachusetts, USA

E-mail: `remigius.mommsen@cern.ch`

**Abstract.** The data acquisition system (DAQ) of the CMS experiment at the CERN Large Hadron Collider (LHC) assembles events at a rate of 100 kHz, transporting event data at an aggregate throughput of 100 GB/s to the high-level trigger (HLT) farm. The DAQ system has been redesigned during the LHC shutdown in 2013/14. The new DAQ architecture is based on state-of-the-art network technologies for the event building. For the data concentration, 10/40 Gbps Ethernet technologies are used together with a reduced TCP/IP protocol implemented in FPGA for a reliable transport between custom electronics and commercial computing hardware. A 56 Gbps Infiniband FDR CLOS network has been chosen for the event builder. This paper discusses the software design, protocols, and optimizations for exploiting the hardware capabilities. We present performance measurements from small-scale prototypes and from the full-scale production system.

## 1. Introduction

The Compact Muon Solenoid (CMS) experiment at CERN is one of the two general purpose experiments located at the LHC. CMS is designed to study both proton-proton and heavy ion collisions at the TeV scale [1]. The detector comprises about 55 million readout channels. The online event-selection is performed using two trigger levels: a hardware-based first-level trigger accepting up to 100 kHz of events and a software-based high-level trigger (HLT) selecting $\mathcal{O}(1\%)$ of these events.

The CMS data-acquisition system from run 1 (DAQ 1) [2] was upgraded during the first long-shutdown of the LHC (2013/14). The main motivation for the upgrade is the aging of the existing hardware (both PCs and network equipment are more than 5 years old), and the need to accommodate sub-detectors with upgraded off-detector electronics that exceed the original

Operated by Fermi Research Alliance, LLC under Contract No. De-AC02-07CH11359 with the United States Department of Energy.

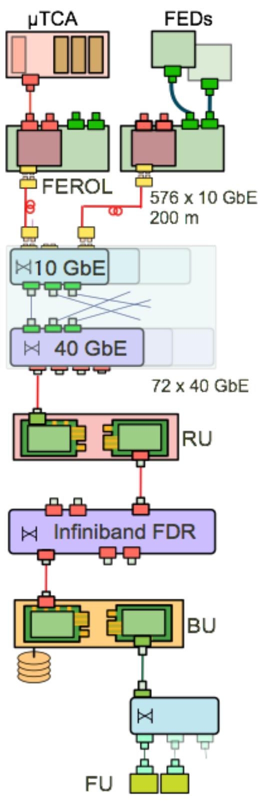

**Figure 1.** Schematic of the DAQ 2 system (see text for explanation.)

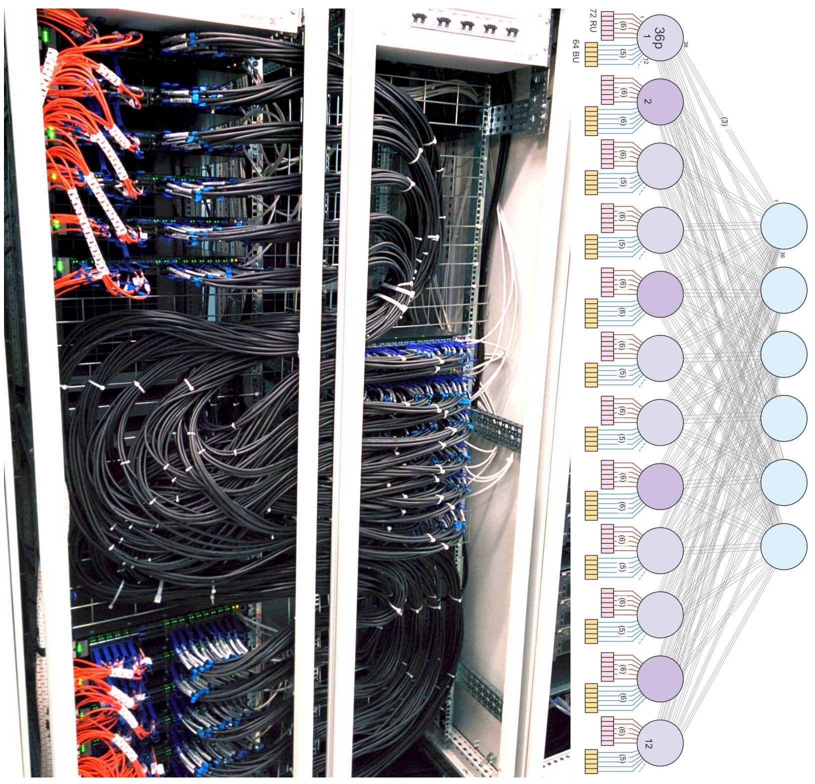

**Figure 2.** The Inifiniband Clos network based on 2×6 leaf switches (on the left) and 6 spine switches (on the right). All switches have 36 ports FDR. The optical cables (orange) connect to the readout-unit (RU) and builder-unit (BU) machines, while the black copper links interconnect the switches.

specification of the DAQ system. The new DAQ architecture (DAQ 2) [3, 4, 5] uses state-of-the-art network technologies for the event building.

## 2. DAQ 2

Figure 1 shows a schematic of the DAQ 2 system. The DAQ collects data from more than 600 custom detector Front-End Drivers (FEDs). In run 1, all FEDs were read out through a custom point-to-point copper link (SLINK) to interface boards called Front-end Readout Link (FRL). The FRL transferred the data to a commercial Myrinet [6] network based on 2.5 Gbps optical links. A new custom Front-End Readout Optical Link (FEROL) [7, 8] has been developed to replace the Myrinet card. The FEROL interfaces to the legacy FEDs through the FRL. These FEDs send fragments of 1–2 kB. Depending on the fragment size, one or two FEDs are assigned to one FEROL. Some sub-systems upgrade the readout electronics for or during run 2 based on the $\mu$TCA standard [9]. Those FEDs are connected to the FEROL through a custom optical point-to-point readout link running at up to 10 Gbps. The fragment size of the new FEDs is expected to be ~8 kB. The FEROL sends the data from each FED as a TCP/IP stream over optical 10 GbE to one of the 72 readout unit (RU) computers. A series of switches is used to concentrate the data onto 40 GbE and transport the data to the surface. Each RU assembles the data from a pre-defined set of FEROLs into super-fragments. The super-fragments are then send over the event-builder switch to one of the 62 builder unit (BU) machines. The event-builder

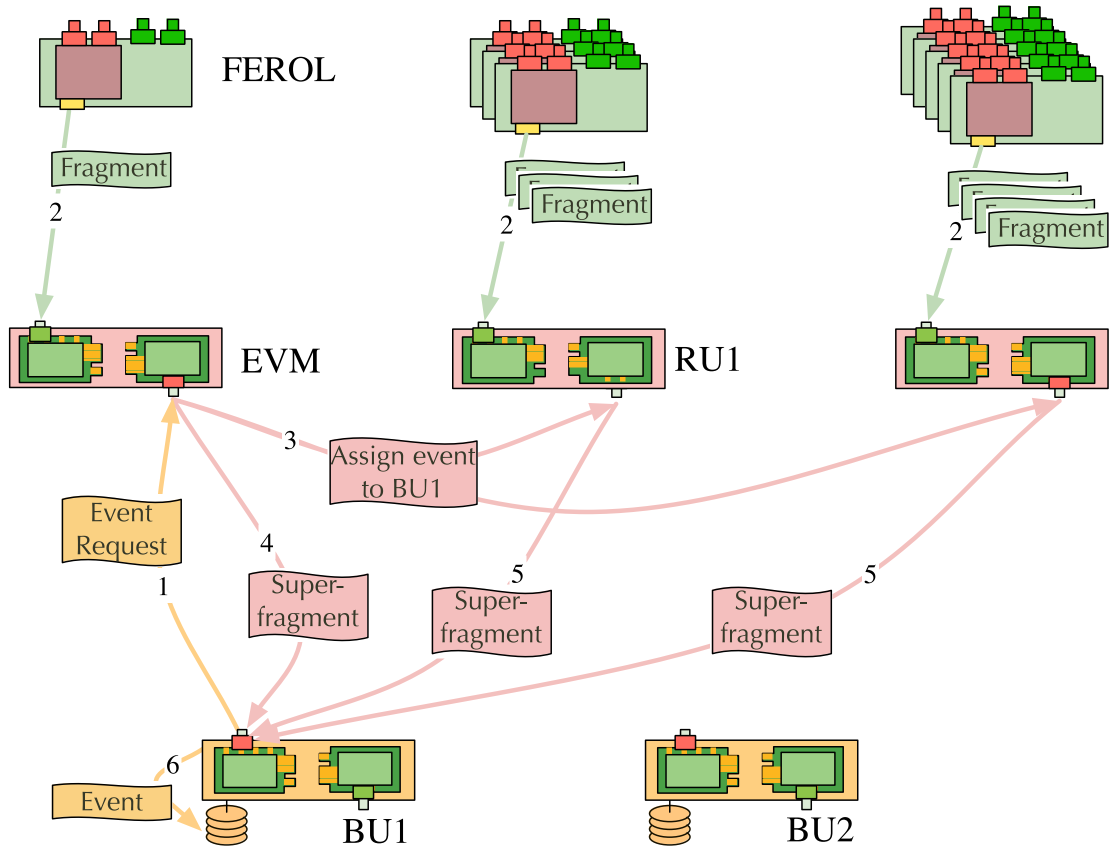

**Figure 3.** Simplified event-building protocol for run 2. A number of Front-End Readout Optical Link (FEROL) boards is assigned to a given readout unit (RU). The event manager (EVM) gets only the trigger data from one FEROL. A number of builder units (BUs) are responsible for the event building. See text for detailed explanation.

switch uses Infiniband [10] FDR at 56 Gbps. It employs a Clos network structure with 12 leaf and 6 spine switches (see figure 2.)

The BU assembles the super-fragments into complete events. The events are checked for consistency by checking the payload of each FED. Finally, the BU writes the event to files residing on a local RAM disk. The RAM disk is exported via NFS v4 to 10–20 filter units (FUs) which run the high-level trigger code. The event selection is using a standard CMSSW [11] process reading the events from files [12].

## 3. Event-Building Protocol and Implementation

Figure 3 shows a simplified schematic of the event-building protocol. (1) Any builder unit (BU) which has resources to build events sends a request for a predefined number of events to the event manager (EVM). (2) The EVM receives asynchronously the event fragment from the FEROL connected to the trigger readout. (3) Once the EVM received enough event fragments from the trigger to satisfy the number of requested events, or if a timeout is reached, the EVM sends a message to all readout units (RUs) participating in the event building. This message contains the level-1 trigger numbers of the events to be sent to the BU which requested the events. (4) At the same time, the EVM packs the event fragments corresponding to the requested events into

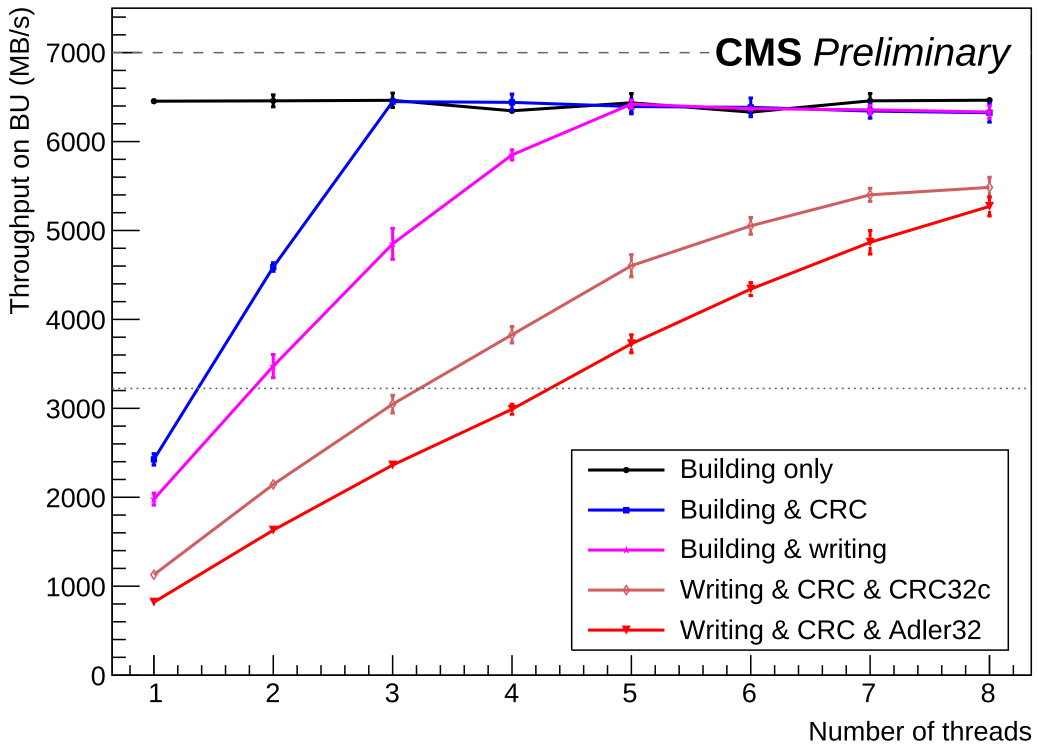

**Figure 4.** The throughput of the builder unit (BU) as function of number of threads when building 1 MB events. Dummy data generated on 3 RUs and one EVM is sent to one BU. Building events only (black) can be done with a single thread at nearly line speed (dashed line). Verifying the CRC of the payload for each FED fragment (blue) needs more resources. Building and writing events to disk (purple) without any checks achieves full performance with 5 threads. Building and writing including all checks (red) requires at least 5 threads to meet the requirement (dotted line). With the CRC32-c algorithm available in recent Intel CPUs (SSE4.2) we gain about 25% in throughput (brown).

an Intelligent Input/Output (I2O)[13] message and sends it to the BU. (5) When the RU receives the event assignment from the EVM, it combines all FEROL fragments into a super-fragment. Each RU knows which FEROLs participate in the readout and waits until it has received all fragments. The super-fragments for all events assigned to the given BU are packed into one or several I2O messages with a fixed size of 131,072 Bytes. They are then send to the BU. (6) Once the BU has received the super-fragments from all RUs, it builds the event. Each super-fragment message contains the identifiers of all RUs which participate in the event building. Therefore, the BU knows on an event-by-event basis if it got the super-fragments from all RUs. The BU checks the consistency of the event by verifying the correct structure of the FED data, the trigger number embedded in each FED fragment and the CRC of the FED payload. Finally, the event is written to a file residing on the RAM disk. Optionally, another checksum over the payload of all FEDs is calculated. This checksum is either the Adler32 checksum, or a CRC32 using the hardware implementation from the SSE4.2 instruction set available in recent Intel CPUs [14].

The event building applications are built upon the XDAQ framework [15]. XDAQ is middleware that eases the development of distributed data acquisition systems. The framework has been developed at CERN and builds upon industrial standards, open protocols and libraries. It provides services for data transport, configuration, monitoring, and error reporting. In particular, it also provides the zero-copy architecture for Infiniband [16].

In order to achieve the full performance, multiple threads are working in parallel. In the EVM and RUs typically 4 threads are handling the building and sending of the super-fragments in parallel. In the BUs, typically 5 threads request, build, and write events independently. Figure 4 shows the performance of the event building on the BU as function of number of threads. Each thread on the BU writes to a separate file in order to avoid congestion.

In addition, the number of memory copies needs to be kept to a minimum. The building of super-fragments and events is done with pointers while keeping the data in the input buffers. The FED fragments are kept in the TCP socket on the RU until a super-fragment is built. The super-fragments are packed directly into the RDMA buffer of the Infiniband network interface cards (NICs). On the BU, the super-fragments are moved from the Infiniband remote direct memory access (RDMA) buffer into files without leaving the kernel space.

In order to cope with the non-uniform memory architecture (NUMA) of the computers used as RUs and BUs, we had to define which work is done on which CPU, depending on the location of the corresponding NICs. Therefore, each thread is bound to a core and all memory structures are allocated on a pre-defined CPU socket. In addition, the interrupts from the NICs are restricted to certain cores that are not used for any data handling.

Finally, the Linux TCP stack had to be tuned to achieve maximum performance when receiving data from the FEROLs.

## 4. Measurements

The performance of the new event-building system is evaluated on a small-scale test bed and on the full-scale production system. The hardware on both systems is identical. The readout unit (RU) is a Dell PowerEdge R620 machine with dual 8 core Xeon CPU (E5-2670 0) running at 2.6 GHz and with 32 GB of memory. The builder unit (BU) is a Dell PowerEdge R720 machine with the same CPUs. The BU uses an asymmetric memory configuration with 32 GB attached to CPU 0 and 256 GB attached to CPU 1. 240 GB on CPU 1 is used for the RAM disk. The network interface cards for 40 GbE and 56 Gb Infiniband are Mellanox Technologies MT27500 Family [ConnectX-3]. We use Mellanox SX1024 & SX1036 for 10 and 40 GbE switches and Mellanox SX6036 for the Infiniband Clos network.

All measurements were carried out by generating dummy event fragments on the FEROLs. The EVM always gets one fixed-size fragment of 1 kB, which roughly corresponds to the trigger fragment. The fragment size of the other FEROLs is set to a fixed size varied between 256 Bytes and 16 kB. Tests with log-normal distributed fragment sizes have not shown any difference.

### *4.1. Data Concentrator*

To measure the performance of the data concentrator, we use one EVM and one RU sending data to two BUs. Between 4 and 16 FEROLs are assigned to the RU. The BUs only assemble events, but do not write them to files. Figure 5 shows the throughput on the RU as function of the generated fragment size in each FEROL. The throughput is independent of the number of FEROLs and TCP streams. It reaches the 40 GbE line speed. The throughput falls off for super-fragments of $\gtrsim$100 kB due to limitation of the memory copy. The 4-FEROL case is limited by the message rate of the individual FEROL. To achieve the required trigger rate of 100 kHz, one can safely merge 12 or 16 FEROLs for the legacy FEDs and 4 FEROLs for the new $\mu$TCA FEDs.

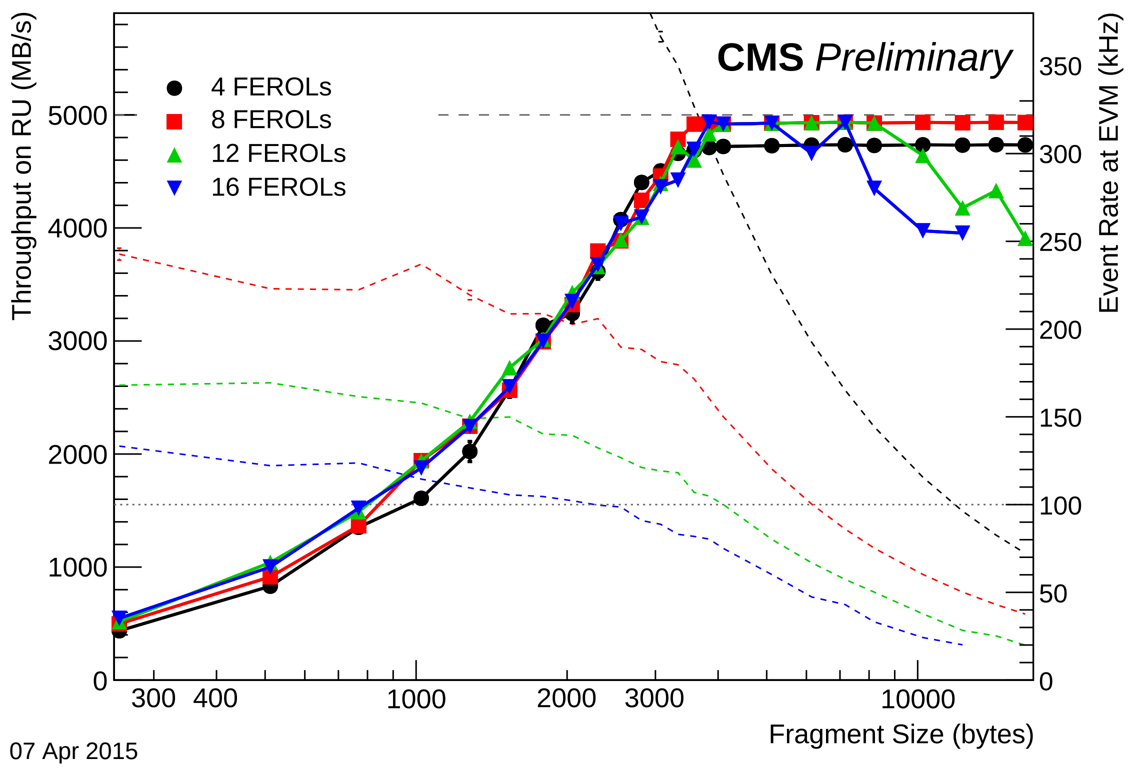

**Figure 5.** Performance measurement of the data concentrator: the thick lines show the throughput on the RU as function of the fragment size for different numbers of FEROLs connected to the RU. For fragment sizes of $\gtrsim$3 kB, the 40 Gb line speed is reached (horizontal dashed line). The dashed lines show the equivalent event rate in kHz (right axis). The point where they go below the requirement of 100 kHz (dotted line) indicates the maximum fragment size which can be handled by the given setup.

## *4.2. Performance of the Builder Unit (BU)*

The performance of the builder unit is evaluated by using 44 RUs each with 8 FEROLs sending the data to one BU. The BU uses 5 threads for building and writing the events. The fragment size on the FEROL is changed to yield different event sizes. Figure 6 shows the throughput on the BU as function of the event size. Compared to figure 4, the throughput when only building events is reduced by ~15% because the BU has to handle more but smaller fragments, originating from 44 RUs instead of 3 RUs. The throughput to disk is the same as in the 3-RUs case when using the Adler32 checksum. The requirement of handling event sizes of $\gtrsim$2 MB at 100 kHz/62 BUs $\approx$ 1.6 kHz can be met in any case.

## *4.3. Scaling behaviour*

The scaling behavior of the system is evaluated by changing the system size from 1 RU sending data to 1 BU to 44 RUs sending data to 44 BUs. Each RU has 8 FEROLs attached. Figure 7 shows the throughput for 4 differently sized systems. While the 1 $\times$ 1 system reaches the full 40 GbE line speed, the performance does not scale to larger systems. In all cases the requirement of 100 kHz is met for fragment sizes of $\gtrsim$1 kB. However, this non-scaling behavior needs further

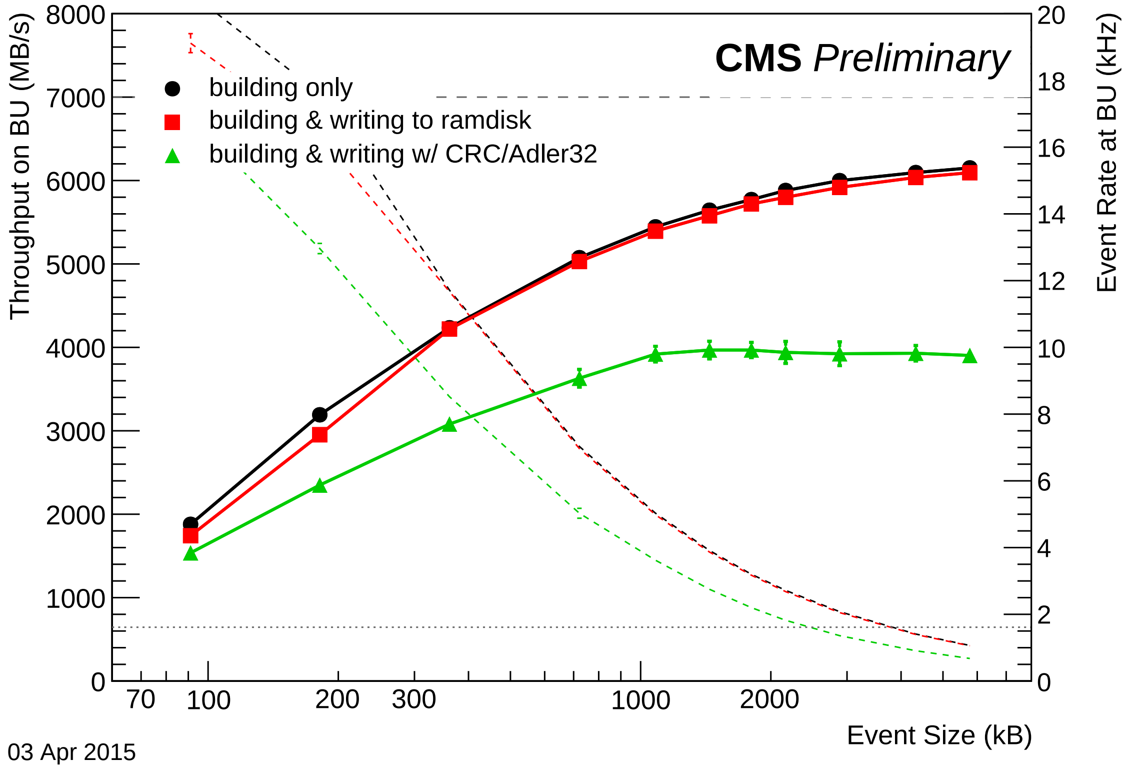

**Figure 6.** Performance measurement of the builder unit: the thick lines show the throughput on the BU as function of the event size. The dashed lines show the equivalent event rate (right axis). The requirement of 1.6 kHz is shown as dotted line.

investigation. Initial studies point to an issue in the routing of the data through the Infiniband Clos network. We believe that the problem is due to the default routing algorithm which assumes a uniform and random traffic pattern. This is clearly not the case in the event building where many sources (RUs) send to one sink (BU) at a given time. We think that this can be improved by using a custom routing which takes into account this particular traffic pattern.

## 5. Summary
CMS has built and commissioned a completely new DAQ system for LHC run II. The new DAQ 2 can handle larger events originating from legacy and new detector front-end electronics. It uses state-of-the-art network technology, which allows to shrink the size of the system by an order of magnitude compared to the previous DAQ. However, the use of high-end hardware requires a more efficient event-builder protocol and new software to exploit the hardware capabilities. In addition, a lot of fine tuning is needed to achieve the full performance capability of the hardware. The scaling behavior of the event builder is not yet fully understood. We believe that it can be ameliorated by using a routing scheme adapted to the event-building traffic pattern. Nevertheless, the DAQ 2 system is ready for the first physics data from LHC run II.

**Acknowledgments**
This work was supported in part by the DOE and NSF (USA).

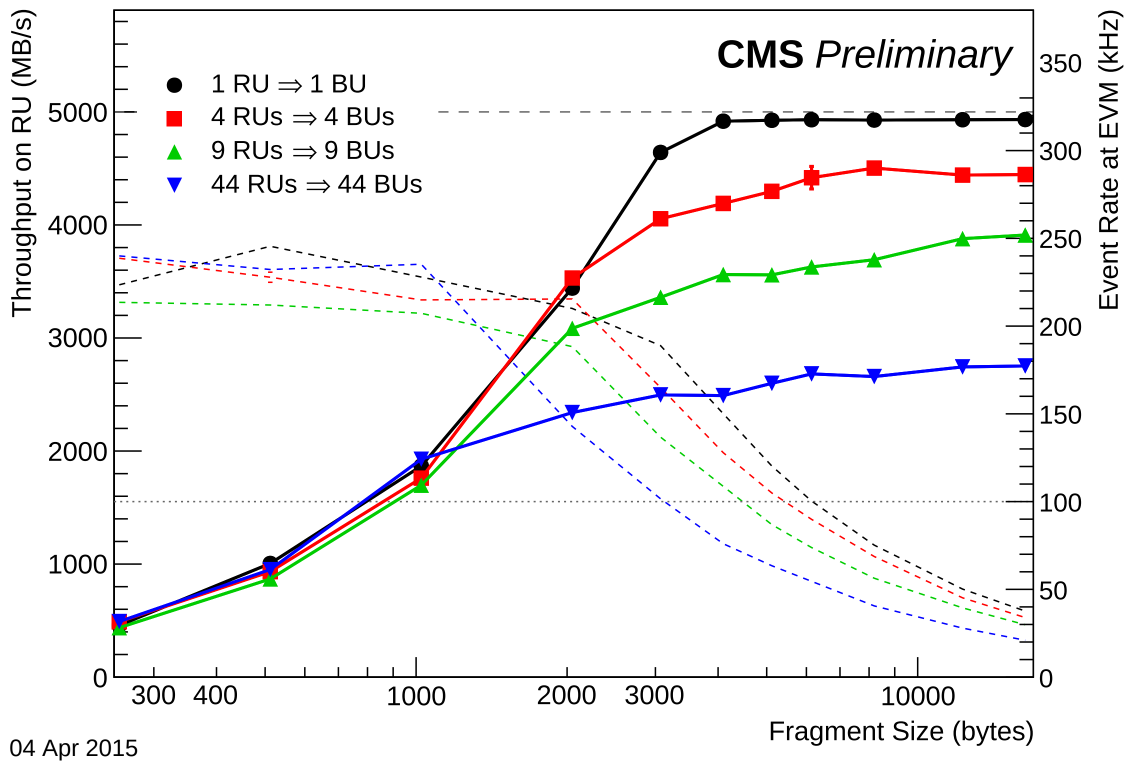

**Figure 7.** The scaling behavior of the system: the thick lines show the average throughput on the RUs as function of the fragment size for different sizes of the system. The dashed lines show the equivalent event rate (right axis). The requirement of 100 kHz is shown as dotted line.

## References

[1] Adolphi R *et al.* (CMS Collaboration) 2008 *JINST* **3** S08004
[2] Mommsen R K *et al.* 2011 *J.Phys.Conf.Ser.* **331** 022021
[3] Petrucci A *et al.* 2012 *J.Phys.Conf.Ser.* **396** 012039
[4] Holzner A *et al.* 2014 *J.Phys.Conf.Ser.* **513** 012014
[5] Sakulin H *et al.* 2015 *to appear in the proceedings of IEEE Real Time conference 2014*
[6] Myricom, see http://www.myri.com
[7] Zejdl P *et al.* 2014 *J.Phys.Conf.Ser.* **513** 012042
[8] Zejdl P *et al.* 2013 *JINST* **8** C12039
[9] Hazen E *et al.* 2013 *JINST* **8** C12036
[10] InfiniBand Trade Association, see http://www.infinibandta.com
[11] Lange D J (CMS Collaboration) 2011 *J.Phys.Conf.Ser.* **331** 032020
[12] Meschi E *et al.* 2015 *in these proceedings*
[13] I2O Special Interest Group, 1999 Intelligent I/O (I2O) Architecture Specification v2.0
[14] Intel SSE4 Programming Reference https://software.intel.com/sites/default/files/m/8/b/8/D9156103.pdf
[15] Gutleber J *et al.* 2010 *J.Phys.Conf.Ser.* **219** 022011
[16] Petrucci A *et al.* 2013 *IEEE Trans.Nucl.Sci.* **60** 4595–4602

Operated by Fermi Research Alliance, LLC under Contract No. De-AC02-07CH11359 with the United States Department of Energy.