The data-acquisition system of the CMS experiment at the LHC

G Bauer\textsuperscript{7}, B Beccati\textsuperscript{2}, U Behrens\textsuperscript{1}, K Biery\textsuperscript{6}, O Bouffet\textsuperscript{2}, J Branson\textsuperscript{5}, S Bukowiec\textsuperscript{2}, E Cano\textsuperscript{2}, H Cheung\textsuperscript{6}, M Ciganek\textsuperscript{2}, S Cittolin\textsuperscript{a}, J A Coarasa\textsuperscript{2}, C Deldicque\textsuperscript{2}, A Dupont\textsuperscript{2}, S Erhan\textsuperscript{4}, D Gigi\textsuperscript{2}, F Glege\textsuperscript{2}, R Gomez-Reino\textsuperscript{2}, D Hatton\textsuperscript{1}, A Holzner\textsuperscript{5}, Y L Hwong\textsuperscript{2}, L Masetti\textsuperscript{2}, F Meijers\textsuperscript{2}, E Meschi\textsuperscript{2}, R K Mommsen\textsuperscript{6}, R Moser\textsuperscript{2}, V O’Dell\textsuperscript{6}, L Orsini\textsuperscript{2}, C Paus\textsuperscript{7}, A Petrucci\textsuperscript{2}, M Pieri\textsuperscript{5}, A Racz\textsuperscript{2}, O Raginel\textsuperscript{7}, H Sakulin\textsuperscript{2}, M Sani\textsuperscript{3}, P Schieferdecker\textsuperscript{2b}, C Schwick\textsuperscript{2}, D Shpakov\textsuperscript{6}, M Simon\textsuperscript{2} and K Sumorok\textsuperscript{7}

\textsuperscript{1} DESY, Hamburg, Germany
\textsuperscript{2} CERN, Geneva, Switzerland
\textsuperscript{3} Eidgenössische Technische Hochschule, Zürich, Switzerland
\textsuperscript{4} University of California, Los Angeles, California, USA
\textsuperscript{5} University of California, San Diego, California, USA
\textsuperscript{6} FNAL, Chicago, Illinois, USA
\textsuperscript{7} Massachusetts Institute of Technology, Cambridge, Massachusetts, USA

\textsuperscript{a} Also at University of California, San Diego, California, USA
\textsuperscript{b} Now at University of Karlsruhe, Karlsruhe, Germany

E-mail: remigius.mommsen@cern.ch

Abstract. The data-acquisition system of the CMS experiment at the LHC performs the read-out and assembly of events accepted by the first level hardware trigger. Assembled events are made available to the high-level trigger which selects interesting events for offline storage and analysis. The system is designed to handle a maximum input rate of 100 kHz and an aggregated throughput of 100 GB/s originating from approximately 500 sources. An overview of the architecture and design of the hardware and software of the DAQ system is given. We discuss the performance and operational experience from the first months of LHC physics data taking.

1. Introduction
The Compact Muon Solenoid (CMS) experiment at CERN, Switzerland, is one of the 2 general purpose experiments built at the LHC. The central feature is a super-conducting solenoid, of 6 m internal diameter, providing a field of 3.8 T. Within the field volume are the silicon pixel and strip tracker, the crystal electromagnetic calorimeter (ECAL), the preshower detector (ES) and the brass/scintillator hadron calorimeter (HCAL). Muons are measured in gas-ionization detectors (CSC, DT, RPC) embedded in the steel return yoke. In addition to the barrel and endcap detectors, CMS has extensive forward calorimetry (HF, Castor). In total, there are \( \sim 10^8 \) readout channels. A much more detailed description of CMS can be found in reference [1].
Figure 1. Average event-fragment sizes for the different sub-systems for proton-proton collisions at an instantaneous luminosity of $150\,\mu\text{b}^{-1}\,\text{s}^{-1}$ (left) and for heavy-ion running (right). Each wedge corresponds to an individual FED, color-coded for the different sub-systems. The radial coordinate of the wedge gives the average event-fragment size on a logarithmic scale.

The first level (L1) of the CMS trigger system, composed of custom hardware processors, runs synchronously with the LHC bunch crossing frequency of 40 MHz. It uses information from the calorimeters and muon detectors to select in $3.2\,\mu\text{s}$, the most interesting events. The maximal allowed accept rate is 100 kHz. Accepted events are assembled through a 2-stage event-building system with a maximum throughput of 100 GB/s. Complete events are fed into the High Level Trigger (HLT) processor farm which reduces the event rate to $O(100\,\text{Hz})$. These events are written to a temporary disk buffer before being transferred to the computing center (Tier 0) at CERN for offline processing.

2. Detector readout and event sizes
The individual detector sub-systems use different custom hardware to read out their detectors after a L1 accept. A common sender mezzanine board plugged onto the front-end driver (FED) board, is used to interface the sub-system readout with the DAQ and trigger. The number of FEDs per sub-system varies between 1 and 438.

Figure 1 shows the average event-fragment sizes per FED for the different sub-systems. The fragment sizes depend on the instantaneous luminosity (occupancy) and on the level of zero suppression. The average event size is $\sim 380\,\text{kB}$ at the current instantaneous luminosity of $150\,\mu\text{b}^{-1}\,\text{s}^{-1}$. The system is designed for an average event size of 1 MB.

For the first heavy-ion (Pb-Pb) run taking place in November 2010, the silicon-strip tracker is read out without zero suppression. The aim is to understand effects of the high occupancy on the zero-suppression algorithm offline without jeopardizing the physics reach. The side effect of this readout scheme is that the event size of $\sim 20\,\text{MB}$ will be mostly independent of the particle multiplicity.

3. Event-data flow
The assembly of event fragments into complete events takes place at the L1 accept rate of 100 kHz. In order to cope with the aggregated throughput of 100 GB/s and to provide a scalable system, a 2-tier approach for the event building was chosen (see figure 2): an initial pre-assembly
Figure 2. The 2-stage event builder assembles event fragments from typically 8 front-ends located underground (USC) into one super-fragment which is then fed into one of the 8 independent readout slices on the surface (SCX) where the complete event is built.

by the FED builder on the first stage, and a final assembly and event selection by 8 independent readout slices on the second stage.

Each FED builder assembles data from typically 8 FEDs into one super-fragment using Myrinet [2] switches. The super-fragment is delivered to one of the 8 independent readout slices on the surface, where it is buffered in readout units (RUs) running on commodity PCs. For a detailed discussion of the FED builder, see reference [3].

The readout slices are mutually independent systems, which are in charge of building full events out of super-fragments, performing the physics selections, and forwarding the selected events to mass storage.

Each readout unit (RU) is connected via a 540-ports switch to builder units (BUs) using the TCP/IP protocol over Gigabit Ethernet [4]. The event building in each slice is controlled by one event manager (EVM), which receives the L1 trigger information. The RUs, BUs, and the EVM make up the RU-builder [5]. The BUs store the complete events until the filter units (FUs) running the HLT algorithms [6] either reject or accept the events. Data of accepted events is compressed and sent to the storage manager [5], which writes it to disk.

The RU-builder and HLT farm consist of 640 Dell PE 2850 dual dual-core, 2 GHz, 4 GB memory PCs used as RUs and 720 Dell PE 1950 dual quad-core, 2.6 GHz, 16 GB memory PCs used as BUs and filter units. The RU-builder switching-fabric consists of 8 Force-10 E1200 switches [7] (one for each DAQ slice).

The currently installed CPU power allows an average of 50 ms to be spent on each event at a level-1 accept rate of 100 kHz. This CPU budget was sufficient for the initial running period. With increasing luminosity and additional sophistication of the L1-trigger selection, more CPU power will be needed. Therefore, additional machines will be installed in the near future.

Each slice has one or two independent storage managers (SMs) connected to the HLT farm via a Force 10 Gigabit Ethernet switch. 4 Fiber Channel switches (QLogic SanBox 5600) provide redundant pathways with automatic fall-over to a total of 8 NexSan SATABeasts (RAID-6 disk arrays). The latter provide a data buffer of $\sim 225$ TB that allows data to be taken for several
days in case the data cannot be transferred off-site. Under normal conditions, the data is copied to the Tier-0 computing center, where it is reconstructed [8]. Part of the data is also served by the SMs to clients via HTTP. These clients process the events in real-time for data quality monitoring purposes [9].

3.1. Performance

Figure 3 shows the performance of the event builder for different FED event-fragment size. The measurement is done by injecting fake data either into the FED builder and dropping it at the RU, or injecting into the RU builder. The combined performance is limited by the FED builder. The drop-off seen for a fragment size greater than 3 kB is not yet fully understood. The combined system has about 25% headroom over the required throughput for the nominal fragment size of 2 kB.

Figure 4 shows the performance of the SM disk arrays. The full system supports writing up to 1000 MB/s with concurrent transfers to Tier 0 at the same rate. With the current compressed event size of ~200 kB and data-streams definitions, roughly 200 MB/s are written. Therefore, during proton running, only half of the system is active while the other half serves as a hot spare.

During the heavy-ion run, where the compressed event size inflates to ~11 MB, the data transfer rate using the full system is ~1.8 GB/s. If the transfers to Tier 0 are switched off, data write rates of up to 2.8 GB/s can be achieved.

4. Online software framework

All data-flow applications are written in C++ and are based on the XDAQ framework [10]. XDAQ is middleware that eases the development of distributed data acquisition systems. It provides services for data transport, configuration, monitoring, and error reporting. The framework has been developed at CERN and builds upon industrial standards, open protocols and libraries.

The monitoring of the data-acquisition system is based on a 3-tier structure. Any of the ~20,000 applications distributed over ~2,000 PCs can publish monitoring tuples or exceptional conditions (warnings, errors, etc). This information is collected and aggregated by a scalable pool of data collection and load balancing services. The aggregated data can then be accessed.
by visualization applications, error reporting GUIs, and expert systems. The latency of the data is $O(1\text{s})$. A much more detailed description of the monitoring system can be found in [11].

5. Run Control
The $\sim 10,000$ data-flow applications are configured and controlled by the CMS run-control system [12]. The run-control framework is written in Java and makes use of various web technologies. It consists of an hierarchy of function managers. Each function manager controls a set of child resources, which are either other function managers or a set of XDAQ applications belonging to the same domain. Function managers typically define a state machine and a set of parameters that can be set or read by the parent. The domain specific knowledge is encapsulated in handlers reacting to state transitions, user input and asynchronous notifications from the child resources.

The top level function manager (FM) is called Level-0. It defines the global state machine that includes all the states of the sub-detector, trigger and central DAQ state machines. As the other function managers, it runs as a web application in an Apache Tomcat servlet container. The user interface runs in a web browser. The Level-0 FM allows the operator to step the global state machine through the various states to configure and start the system. It also allows to exclude entire sub-systems, sub-system partitions or individual FEDs in case of problems or local running of some sub-systems. In addition, the L1 and HLT trigger configuration and the clock source (local or LHC) can be selected. Individual sub-system configurations, e.g. the level of zero suppression, can be selected and their state machine can be stepped through individually. This feature allows for rapid recovery from sub-system errors. It also reports status and error messages from sub-system function managers. Furthermore, the Level-0 FM is aware of LHC states, e.g. ‘stable beams’. It also helps the shifter to deal with the many dependencies between the systems, e.g. when the clock source changes due to LHC state changes, or when excluding a sub-system affects the L1 trigger, by displaying badges next to suggested actions or pop-ups warning about inconsistencies.

6. Experiences from data-taking
Figure 5 shows the delivered and recorded luminosity since the start of the LHC physics program in March 2010. CMS has achieved an already high data-taking efficiency of $\sim 92\%$. Central DAQ was unavailable for only $\sim 0.5\%$ or $\sim 3\text{h}$ during stable beam periods. The dead time accrued is split roughly in half between software bugs and hardware problems. Problems due to failing computers can be quickly dealt with by masking one of the readout slices while the problem is being fixed. In addition, essential services are redundant with automatic fail-over. However, mainly on the front-end and FED-builder side, several single points of failures exist. Possibilities to mitigate the impact of such failures on the DAQ availability will be investigated.

Despite the already high availability of the DAQ system, several areas have been identified where improvements could be made:

- Corrupted data from FEDs blocks the run, especially out-of-order event fragments. While it is often necessary to stop the run to resynchronize the front-ends, fault tolerance against certain cases is being considered.
- The current static round-robin scheme used to assigning events to readout slices assumes an equal performance of all readout slices. Thus, the slowest slice dictates the maximal trigger rate. Studies how to employ dynamic load balancing between the slices have started [13].
- To keep the data-taking efficiency high with less experienced shifters, further automation, simplified diagnostic of failures, and automatic error recovery procedures are being studied.

In addition to failures of the central DAQ, runs need to be restarted to recover from many problems due to sub-system failures. A typical stop/start cycle takes $\sim 128\text{s}$. In total about
15.5 h or 2% of stable beam time was lost due to 490 run starts in 2010. Further work is foreseen to reduce both the number of required run starts and the time it takes for each cycle.

7. Summary
The data-acquisition (DAQ) system of the CMS experiment at the LHC is very flexible. It can be easily configured for high L1-trigger rates in proton physics or large events at moderate trigger rates during heavy-ion periods. There is contingency on all stages to deal with unexpected fluctuations. The high-level trigger CPU farm can be expanded to accommodate more sophisticated selection algorithms. The strongest point about the CMS DAQ is its capability to build events at the L1-trigger rate of 100 kHz, which allows employment of advanced algorithms on all data accepted by L1. The availability of the central DAQ system during the 2010 proton run was > 99%.

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

References
[1] Adolphi R et al. (CMS Collaboration) 2008 JINST 3 S08004
[2] Myricom, see http://www.myri.com
[3] Sakulin H et al. 2008 IEEE Trans. Nucl. Sci. 55 190–197
[4] Bauer G et al. 2008 IEEE Trans. Nucl. Sci. 55 198–202
[5] Bauer G et al. 2010 J.Phys.Conf.Ser. 219 022038
[6] Bocci A et al. 2010 in these proceedings
[7] Force10, see http://www.force10networks.com
[8] Hufnagel D et al. 2010 in these proceedings
[9] Morovic S et al. 2010 in these proceedings
[10] Bauer G et al. 2010 J.Phys.Conf.Ser. 219 022011
[11] Bauer G et al. 2010 J.Phys.Conf.Ser. 219 022042
[12] Bauer G et al. 2010 to appear in proceedings for RealTime 2010
[13] Simon M et al. 2010 in these proceedings