Pilot run of the new DAQ of the COMPASS experiment

doi:10.1088/1742-6596/664/8/082042

Pilot run of the new DAQ of the COMPASS experiment

J Novy[14], **M Bodlak**[5], **V Jary**[1], **I Konorov**[2], **M Virius**[1], **D Steffen**[2], **J Tomsa**[1], **A Kveton**[1], **D Levit**[2], **S Huber**[2] **and V Frolov**[34]

[1]Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University, Brehova 7, 115 19 Prague 1, Czech Repulic
[2]Physik-Department E18, Technische Universitat Munchen, James-Franck-Strae, 85748 Garching, Germany
[3]Joint Institute for Nuclear Research, Joliot-Curie 6, 141980 Dubna, Moscow region, Russia
[4]European Organization for Nuclear Research - CERN, CH-1211, Geneve 23, Switzerland
[5]Department of Low-Temperature Physics, Charles University, V Holesovickach 2, 180 00 Praha 8, Czech Repulic

E-mail: josef.novy@cern.ch

Abstract. This contribution focuses on the deployment and first results of the new data acquisition system (DAQ) of the COMPASS experiment at CERN laboratory. The COMPASS experiment is a fixed target experiment with maximum rate of 1.5 GB/s. The DAQ utilizing FPGA-based event builder is designed to be able to readout data at maximum rate of the experiment. The DAQ is developed under name RCCARS (run control, configuration, and readout system). The RCCARS has been deployed for the pilot run starting from the September 2014 and further developed for long run in 2015. In the paper, we present performance and stability results of the new DAQ architecture; we compare it with the original system in more details.

1. Introduction

The COMPASS [3] is a fixed target high energy physics experiment situated at the SPS particle accelerator at CERN laboratory in Geneva, Switzerland. Its main scientific goal is to study hadron structure and hadron spectroscopy with high intensity muon and hadron beams. Origin of the experiment dates back to September 1998 when the final Memorandum of Understanding was signed. After four years of preparations and commissioning, the physics data taking started in 2002. Since 2002, number of channels increased from 190000 to approximately 300000, trigger rate increased from 5 kHz to 30 kHz; the average event size remained roughly 36 kB. The original DAQ consisted of several layers and was based on software event building paradigm. In order to handle the increased data rates and mainly to cope with aging of the system, it has been decided to develop a new DAQ system during technical shutdown of CERN accelerator in 2013-2014.

2. Original DAQ with software event-building

The first layer of the original DAQ consist of the detector frontend electronics. These electronics continuously preprocess and digitize data in approximately 300000 channels, the data are readout when trigger signal arrives and are concentrated into 250 custom VME modules(CATCHs, HGeSiCAs, GANDALFs). These modules were connected to the event building network using 90 optical links called Slinks [13]. The network consisted of two types of servers: readout buffers and event builders. 30 readout buffers served for data reception and buffering which was used to distribute the load over the entire SPS accelerator cycle. The collected sub-events were transferred over the switched gigabit Ethernet to the 20 event builders that assembled full events. Full events were written into the local disk space and afterwards sent to the central CERN storage facility CASTOR. The system was controlled by adapted ALICE DATE package which implemented run control, event sampling, monitoring, run keeping, and configuration functionality. Maximum in spill data rate of this system was 1.2 GB/s and maximum sustained data rate was 500 MB/s. Full diagram of the original DAQ is shown in Figure 1.

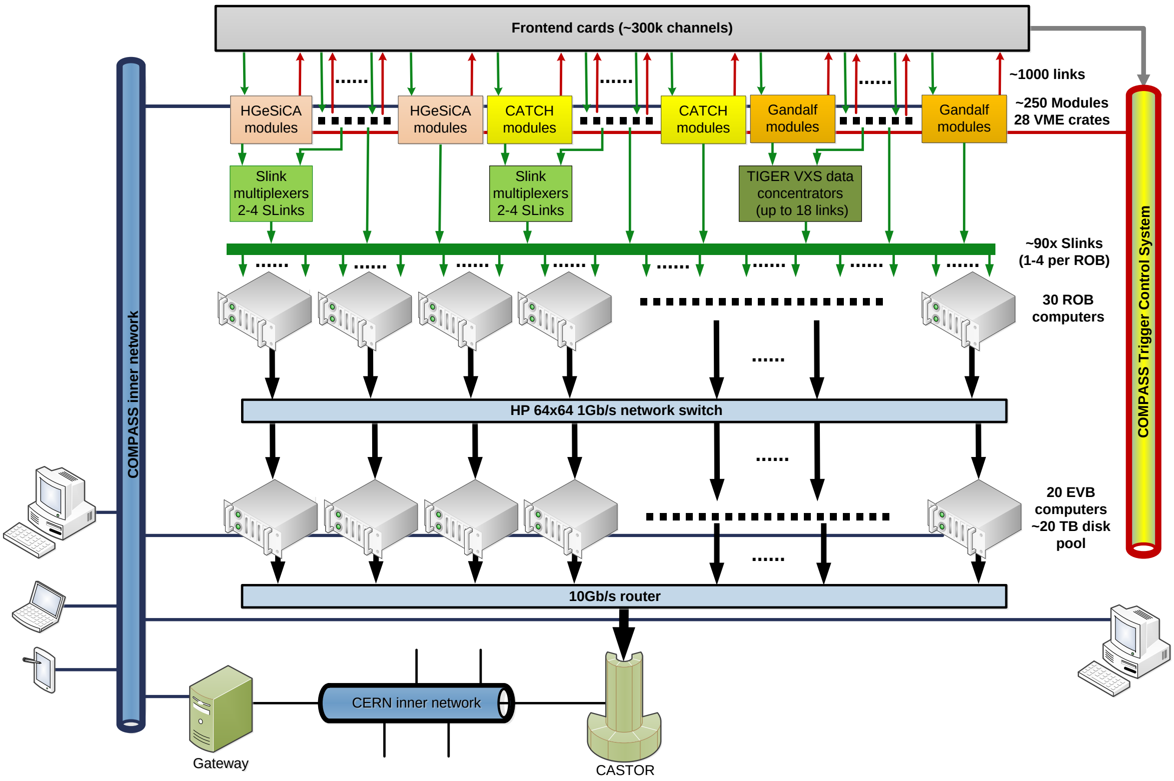

Frontend cards (~300k channels)
~1000 links
~250 Modules
28 VME crates
HGeSiCA modules
HGeSiCA modules
CATCH modules
CATCH modules
Gandalf modules
Gandalf modules
Slink multiplexers 2-4 SLinks
Slink multiplexers 2-4 SLinks
TIGER VXS data concentrators (up to 18 links)
~90x Slinks (1-4 per ROB)
30 ROB computers
HP 64x64 1Gb/s network switch
20 EVB computers ~20 TB disk pool
10Gb/s router
COMPASS inner network
COMPASS Trigger Control System
CERN inner network
Gateway
CASTOR

Figure 1. DAQ with software event-building

3. DAQ with hardware event-building

The new DAQ design [6, 7, 8] was prepared in 2012 as one of proposals for upgrade of the old system. It uses custom FPGA based data handling cards (DHC) for building of full events, which replace the event building network. The cards have been designed in Compact AMC form factor and they feature 16 high speed serial links, 4 GB of DDR3 memory, Gigabit Ethernet connection, and COMPASS Trigger Control System receiver. There are two different versions of firmware: multiplexer and switch. The multiplexer card combines data from up to 15 incoming Slinks and sends them into one outgoing, whereas the switch combines data from up to 8 multiplexers and distributes the full events to one of 8 readout engine servers equipped by spillbuffer PCI-Express cards that receive the data. Spillbuffer cards have 2 GB of DDR memory and together with DHC cards memory are used to store data of more than one spill and to distribute the load over the accelerator cycle period. Readout engine (RE) servers are also used for transformation of events to DATE (Data Acquisition and Test Environment) format [9], for monitoring of data consistency, and for data quality checks. Transformation to DATE format is needed as all monitoring and production programs has been written to work with it. Full diagram of the new DAQ is shown in Figure 2.

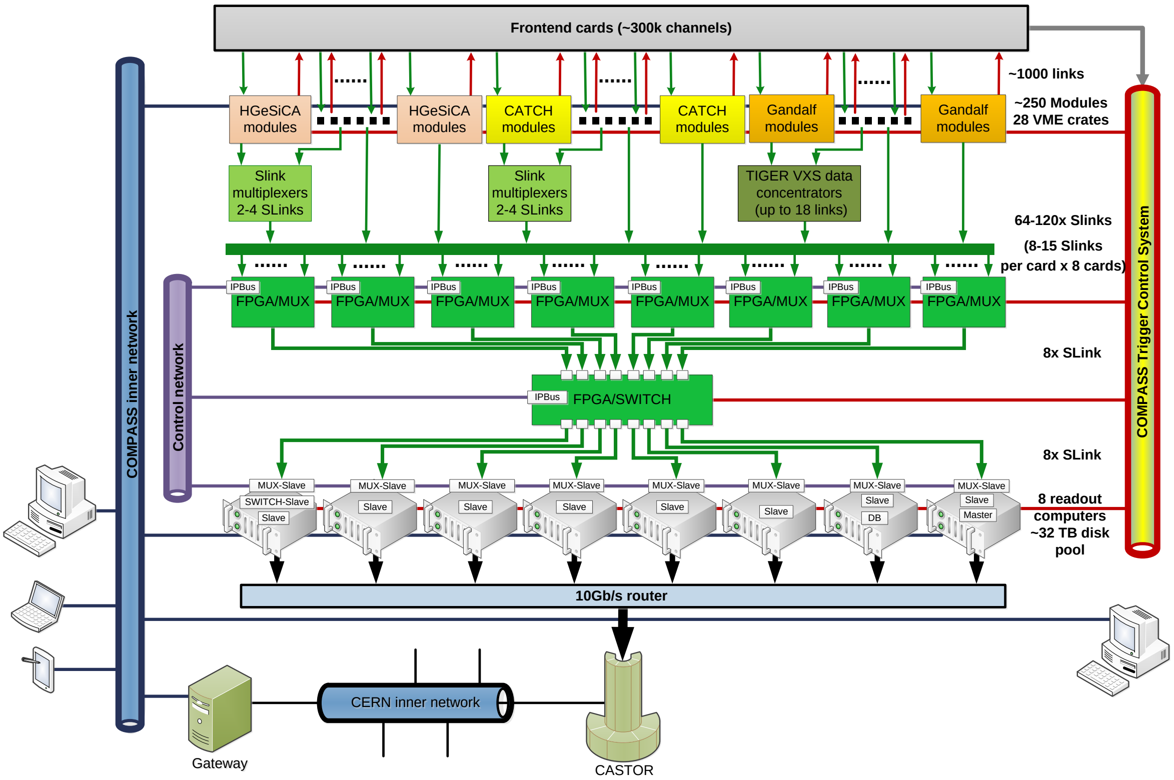


Figure 2. DAQ with hardware event-building

As the DHC cards perform data flow control and event building, the software serves for configuration, run control, monitoring, and readout of full events. For these purposes, we have developed special software package. The main part of the software is implemented in the C++ language with the Qt framework; JavaScript, PHP, TCL, Python languages are used for support tasks. The MySQL relational database has been selected as a storage of system configuration and logs. Communication between processes in the system is implemented using the DIM library [10, 11]. The DIM is a multi-platform library that serves for an asynchronous one to many communication through the Ethernet. It was originally developed for the DEPHI experiment at CERN. Several types of processes are present in the system. The master is the most important process; it exchanges information and commands between user interface and slave processes. Slaves monitor and configure the DHC cards through the IPBus package [12], process data, and provide interface for data access. The IPBus package was developed for the level one trigger update of the CMS experiment. User interface can run either in control or in viewer mode. However, only one instance of user interface may be present in the control mode. Message logger collects messages from all processes involved in the RCCARS and stores them in the database. Message browser is a graphical tool which allows to display and to filter these logs. The RCCARS is configured through the web interface.

4. COMPASS run in 2014

The 2014 run has been chosen as first run of the RCCARS. Goal was to be able to readout data with nominal conditions for Drell-Yan run of 25 kHz with average event size of 25 kB. The RCCARS was deployed in an incomplete state with 5 DHC with MUX firmware, 1 DHC with preliminary SWITCH firmware and 1 RE server in September 2014. New versions of the SWITCH, MUX, and spillbuffer firmware were deployed during run. These updates allowed connection of 2 and later 4 pccores and increased both stability and speed of the system. Figure 3 shows data recorded since start of the run. Initial performance of DAQ was around 1TB per day in dependance on beam availability and needs of experiment. The first major update has been applied 28 days after start of the run and is marked by green line in Figure 3. This update allowed to increase performance up to 5 TB per day by connecting second RE server and improving process of transformation from new data format to DATE format. At first the transformation procedure was executed by single thread, however as more and more equipment was connected, the need for improvement became to be clear visible. Consequently transformation process has been rewritten in following way. Original main thread was kept but the processing of events was delegated to child threads. During run we used up to 10 child threads. The next update of system has been performed 44 days after start of the run and is marked by red line in the figure. The most important improvement in this update was possibility to connect 4 pccores. The DAQ with 4 pccores managed to record up to 11 TB per day and thus started to be able to read the same amount of data as the previous system. These upgrades were done in sync with needs of experiment and they didn't limit performance of physics program.

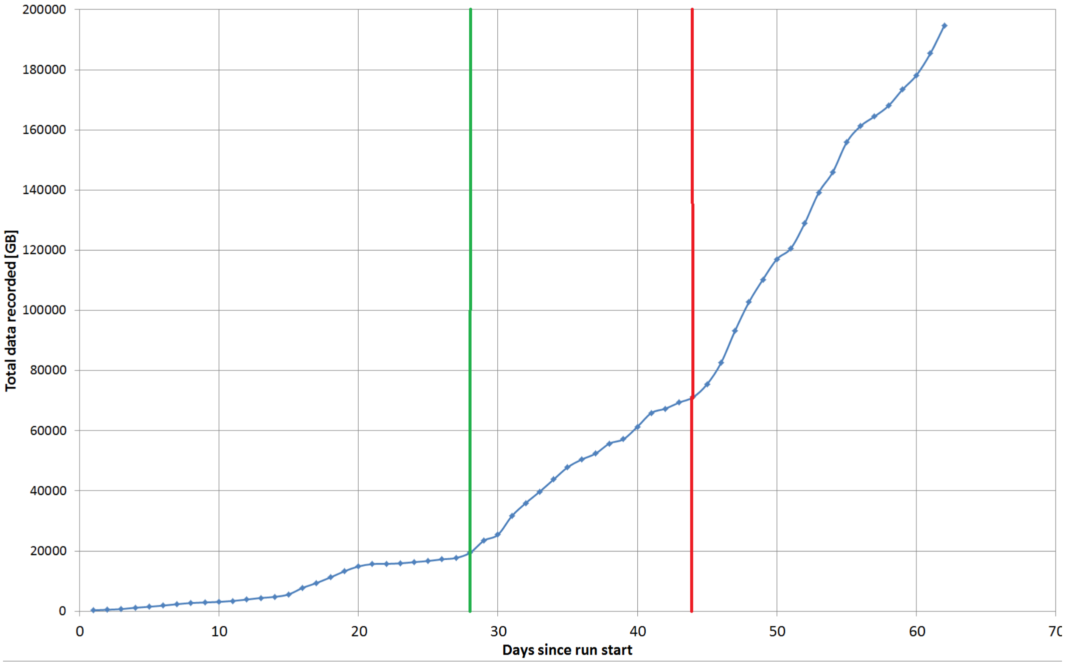
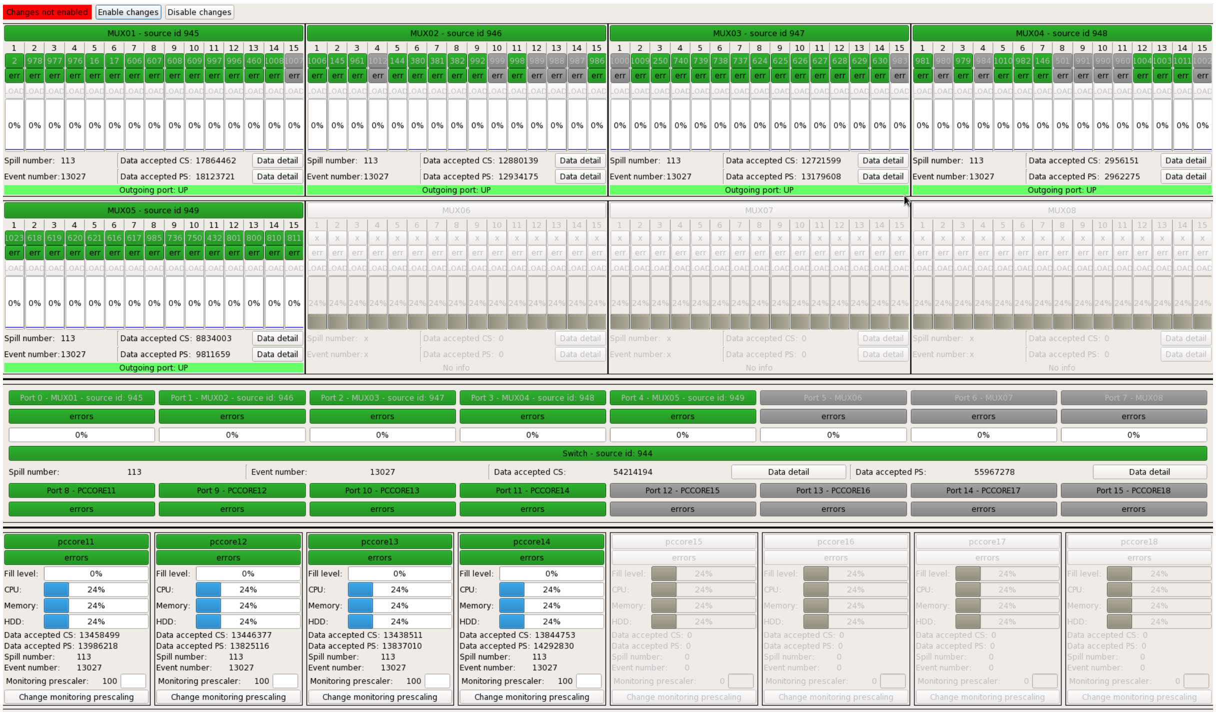


Figure 3. Data recorded during the 2014 run.

The advantages of the new system were visible soon after deployment as the power consumption and space requirements are significantly lower because of the new system compose just from 11 computers and one VME crate; we ceased to be dependant on cooling system of DAQ barracks which failed at the start of the run. Also need to manage less components proved to be really helpful as it was possible to create overview of the whole system in one window. Screen shot of this window is shown in Figure 4. Control of the system is now more understandable and user friendly for both operators and experts, although the complexity and size of RCCARS software package is comparable to the previous system.



MUX01 - source id 945: 2 978 977 976 16 17 606 607 608 609 997 996 460 1008 1007; 0% in all channels
Spill number: 113 | Data accepted CS: 17864462
Event number: 13027 | Data accepted PS: 18123721
Outgoing port: UP

MUX02 - source id 946: 1006 145 961 1012 144 380 381 382 992 999 998 989 988 987 986; 0% in all channels
Spill number: 113 | Data accepted CS: 12880139
Event number: 13027 | Data accepted PS: 12934175
Outgoing port: UP

MUX03 - source id 947: 1000 1009 250 740 739 738 737 624 625 626 627 628 629 630 983; 0% in all channels
Spill number: 113 | Data accepted CS: 12721599
Event number: 13027 | Data accepted PS: 13179608
Outgoing port: UP

MUX04 - source id 948: 981 980 979 984 1010 982 146 501 991 990 960 1004 1003 1011 1002; 0% in all channels
Spill number: 113 | Data accepted CS: 2956151
Event number: 13027 | Data accepted PS: 2962275
Outgoing port: UP

MUX05 - source id 949: 1023 618 619 620 621 616 617 985 736 750 432 801 800 810 811; 0% in all channels
Spill number: 113 | Data accepted CS: 8834003
Event number: 13027 | Data accepted PS: 9811659
Outgoing port: UP

MUX06, MUX07, MUX08: x in all channels; 24% in all channels
Spill number: x | Data accepted CS: 0
Event number: x | Data accepted PS: 0
No info

Ports 0–7: 0%

Switch - source id: 944
Spill number: 113 | Event number: 13027 | Data accepted CS: 54214194 | Data accepted PS: 55967278

pccore11: Fill level: 0% CPU: 24% Memory: 24% HDD: 24%
Data accepted CS: 13458499
Data accepted PS: 13986218
Spill number: 113
Event number: 13027
Monitoring prescaler: 100

pccore12: Fill level: 0% CPU: 24% Memory: 24% HDD: 24%
Data accepted CS: 13446377
Data accepted PS: 13825116
Spill number: 113
Event number: 13027
Monitoring prescaler: 100

pccore13: Fill level: 0% CPU: 24% Memory: 24% HDD: 24%
Data accepted CS: 13438511
Data accepted PS: 13837010
Spill number: 113
Event number: 13027
Monitoring prescaler: 100

pccore14: Fill level: 0% CPU: 24% Memory: 24% HDD: 24%
Data accepted CS: 13844753
Data accepted PS: 14292830
Spill number: 113
Event number: 13027
Monitoring prescaler: 100

pccore15, pccore16, pccore17, pccore18: Fill level: 24% CPU: 24% Memory: 24% HDD: 24%
Data accepted CS: 0
Data accepted PS: 0
Spill number: 0
Event number: 0
Monitoring prescaler: 0

Figure 4. System overview window

System faced also several problems. The first was limitation of 80 MB/s per RE server caused by message exchange between threads. This problem was solved in 2015 (see bellow). Problems with frontend errors handling also appeared, but RCCARS still proved to be more stable then previous DAQ. The last problematic point was implementation of connectivity to older monitoring tools. It proved to be complicated as many tools used different kind of settings with out any extensive documentation.

5. Preparation for run in 2015

Beginning of this year was fully dedicated to improvements of the system based on experiences and comments gathered during run in 2014. The main improvement was rework of the messages generated by readout process that allowed to increase maximum readout speed from 80 MB/s to 150 MB/s which was goal of the RCCARS design. Originally all messages were transferred from all threads to the informer thread over signal-slot mechanics of the Qt framework. Unfortunately this system imposed several restrictions which limited the speed of the system. New design partially keeps signal-slot mechanics, but it uses reallocated message buffers for main processor thread and transformation threads as they are the most critical ones and also they are the largest producers of messages. Each of these threads has its own circular buffer to which it stores messages intended for processing. Informer thread can then asynchronously iterate all buffers and check uniqueness of messages and concentrate messages. The message is sent on the first occurrence of unique message to master process and logger process, but all other occurrences of the same message in the given time interval are just counted and the summary is sent only after set timeout.

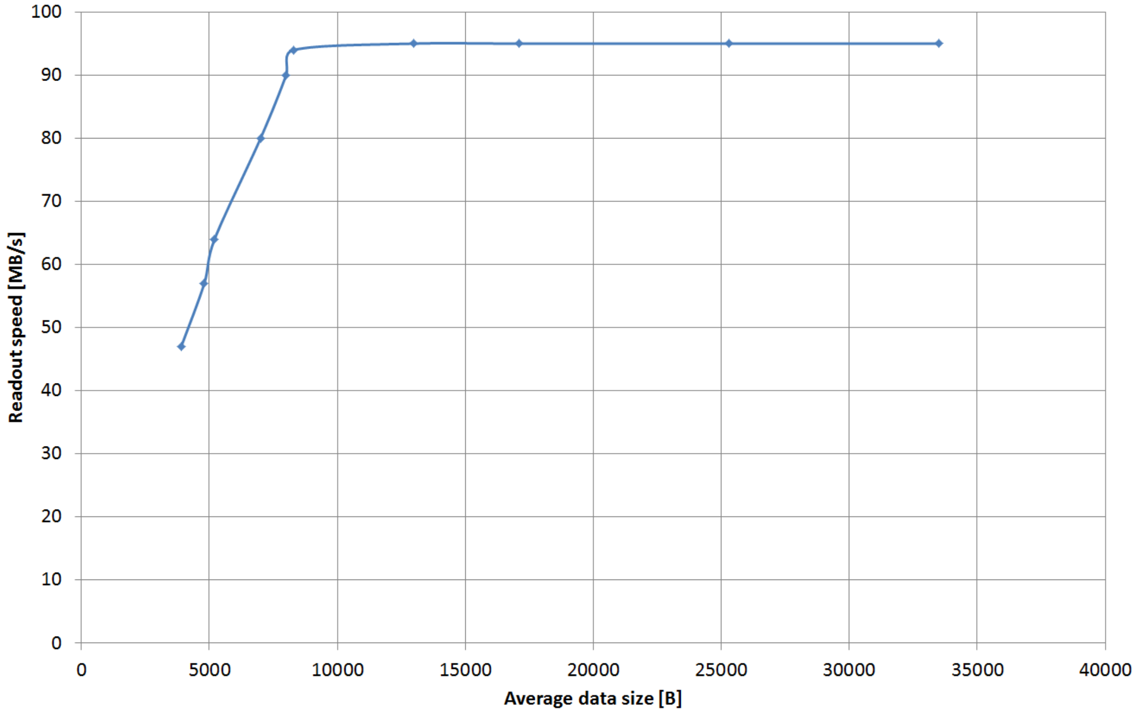


Figure 5. Readout speed test with 4 RE servers

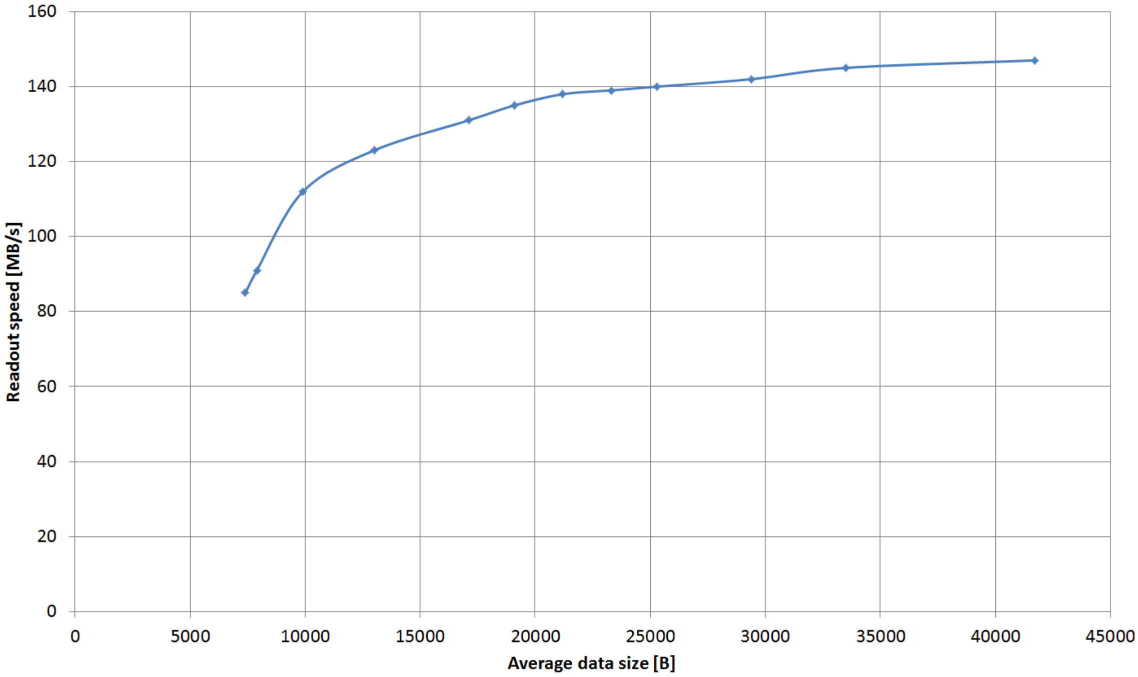


Figure 6. Readout speed test with 2 RE servers

Test of readout speed in dependency on event size has been done with 4 RE servers after these changes to measure performance of the system. Result for one of the four servers is shown in Figure 5. Limit of 95 MB/s at mark of 7.5 KB per event is visible from the picture. This limit was identified as problem of SWITCH firmware prototype which shares some resources between outgoing port couples. Test with 2 RE servers connected to ports with not shared resources has been performed to prove source of limitation. Figure 6 shows results for one server. It is clearly visible that rising trend is not stopped at mark of 7.5 KB per event and speed of 95 MB/s, but continues as expected to values around 150 MB/s.

Other improvements were done to increase flexibility of the system in both software and firmware parts. Firmware update over IPBus functionality was added. The programable buffer sizes in multiplexers were introduced to cope with differences in size of sub-events between detectors. This upgrade was important as differences in size between these sub-events can be even in order of magnitude. It is implemented in the way that each incoming port have reserved part of DDR3 memory of DHC corresponding to value set in registers. These registers are programable by control process through the IPBus.

6. Future development

The future development will introduce full switch with 8 RE connected that should allow to reach the designed total sustainable readout speed of 1.2 GB/s. This allows together with buffering capabilities allows to reach total peak rate of 2.4 GB/s during used duty cycle of 4.8 s on-spill and 4.8 s off-spill.

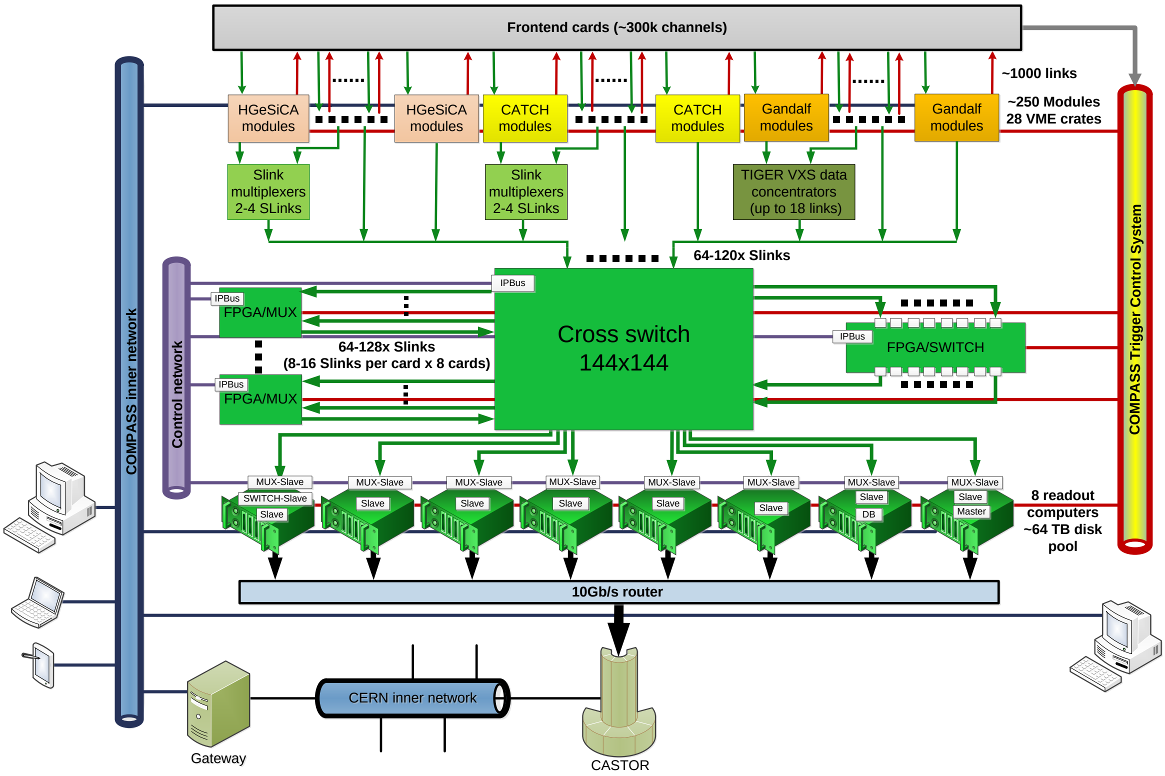


Figure 7. DAQ with hardware event-building and cross switch

Major improvement planed to increase flexibility and reliability is to connect CATCHes, HGeSiCAs, GANDALFs, DHC-MUXs, DHC-SWITCH, and RE servers to cross switch. Interconnection between elements will than be made by configuration of cross switch. This will allow to exchange faulty equipment even during run and also automatize load balancing of MUX modules. Full design with cross switch is shown in Figure 7.

7. Conclusion

The new DAQ system of the COMPASS experiment was successfully deployed during 2014 run. The design of the new DAQ proved to be resistent to errors and easily controllable. RCCARS recorded 195 TB of data in two months of Drell-Yan pilot run and reached goal of readout speed of 150 MB/s per RE server, thus it fulfilled initial demands and its development continues.

Acknowledgments

The work on this project has been supported in part by the following grants: MSMT LA08015 and SGS 11/167. It is also being supported by the Maier-Leibnitz-Labor, Garching and the DFG cluster of excellence Origin and Structure of the Universe.

References

[1] M. Bodlak, et al. *Developing Control and Monitoring Software for the Data Acquisition System of the COMPASS Experiment at CERN.* Acta polytechnica: Scientific Journal of the Czech Technical University in Prague. Prague, CTU, 2013, issue 4. Available at: http://ctn.cvut.cz/ap/

[2] M. Bodlak, et al. *New data acquisition system for the COMPASS experiment.* Journal of Instrumentation. 2013-02-01, vol. 8, issue 02, C02009-C02009. DOI: 10.1088/1748-0221/8/02/C02009. Available at: http://stacks.iop.org/1748-0221/8/i=02/a=C02009?key=crossref.a76044facdf29d0fb21f9eefe3305aa5

[3] P. Abbon, et al.(the COMPASS collaboration): *The COMPASS experiment at CERN.* In: Nucl. Instrum. Methods Phys. Res., A 577, 3 (2007) pp. 455518

[4] M. Bodlak, V. Jary, J. Novy: *Software for the new COMPASS data acquisition system.* In: COMPASS collaboration meeting, Geneva, Switzerland, 18 November 2011

[5] L. Schmitt, et al.: *The DAQ of the COMPASS experiment.* In: 13th IEEE-NPSS Real Time Conference 2003, Montreal, Canada, 1823 May 2003, pp. 439444

[6] V. Jary: *Analysis and proposal of the new architecture of the selected parts of the software support of the COMPASS experiment* Prague, 2012, Doctoral thesis, Czech Technical University in Prague

[7] M. Bodlak: *COMPASS DAQ Database Architecture and Support Utilities* Prague, 2012, Master thesis, Czech Technical University in Prague

[8] J. Novy: *COMPASS DAQ - Basic Control System* Prague, 2012, Master thesis, Czech Technical University in Prague

[9] T. Anticic, et al. (ALICE DAQ Project): *ALICE DAQ and ECS User's Guide* CERN, EDMS 616039, January 2006.

[10] Gaspar C., Donszelmann: *DIM - A Distributed Information Management System for the DELPHI Experiment at CERN*, Proceedings of the 8th Conference on Real-Time Computer applications in Nuclear, Particle and Plasma Physics, Vancouver, Canada, June 1993.

[11] Gaspar C., Donszelmann: *DIM, a Portable, LightWeight Package for Information Publishing, Data Transfer and Inter-process Communication*, Presented at: International Conference on Computing in High Energy and Nuclear Physics (Padova, Italy, February 2000)

[12] C. Ghabrous Larrea, et al.: *IPbus: a flexible Ethernet-based control system for xTCA hardware*, 2015 JINST 10 C02019. doi:10.1088/1748-0221/10/02/C02019

[13] S-Link - High Speed Interconnect [online]. 17.6.2003 [cit. 2012-04-18]. At: http://hsi.web.cern.ch/HSI/s-link/