COMPUTING INFRASTRUCTURE FOR ONLINE MONITORING AND CONTROL OF HIGH-THROUGHPUT DAQ ELECTRONICS

S. Chilingaryan, M. Caselle, T. Dritschler, T. Farago, A. Kopmann, U. Stevanovic, M. Vogelgesang,
Karlsruhe Institute of Technology, P.O. Box 3640, 76021 Karlsruhe, Germany

Abstract

New imaging stations with high-resolution pixel detectors and other synchrotron instrumentation have ever increasing sampling rates and put strong demands on the complete signal processing chain. Key to successful systems is high-throughput computing platform consisting of DAQ electronics, PC hardware components, communication layer and system and data processing software components. Based on our experience building a high-throughput platform for real-time control of X-ray imaging experiments, we have designed a generalized architecture enabling rapid deployment of data acquisition system. We have evaluated various technologies and come up with solution which can be easily scaled up to several gigabytes-per-second of aggregated bandwidth while utilizing reasonably priced mass-market products. The core components of our system are an FPGA platform for ultra-fast data acquisition, Infiniband interconnects and GPU computing units. The presentation will give an overview on the hardware, interconnects, and the system level software serving as foundation for this high-throughput DAQ platform. This infrastructure is already successfully used at KIT's synchrotron ANKA.

INTRODUCTION

There are several challenging tasks to be solved while building the computing infrastructure for high-throughput DAQ electronics. The Linux Kernel Driver Interface is volatile and the kernel drivers are hard to develop and maintain. Additional complexity is added by the necessity to synchronize the development of detector hardware and the required readout software. Due to limited bandwidth of system memory, effective streaming of data requires a very efficient realization of the DMA protocol. To reduce the impact of memory subsystem, the vendors of HPC hardware have developed special techniques, like *GPUDirect* [1]. If feedback loops are desired, large computing resources must be available to process dense streams of information. This is often solved using various accelerator cards with heavily parallel architectures. To reduce the latencies of feedback loops also techniques like *GPUDirect* may be applied. Another important aspect is the storage and preservation of the produced data. The storage subsystem should be able to handle several gigabytes of data per second for a long time. Finally, the question of scalability often arises. Systematically solving these questions is a difficult challenge and requires expertise in different areas of hardware, system, and software engineering.

Based on our experience building a high-throughput platform for real-time control of X-ray imaging experiments [2-5], we started to develop a hardware infrastructure and a software middleware with the goal to rapidly deploy data acquisition systems for different types of high-throughput detectors with the data rates up to 8 GB/s. We have evaluated various technologies and designed a scalable architecture based on Infiniband interconnects and GPU computing units. The core components of our system are an FPGA platform for ultra-fast data acquisition, the GPU-based Image Processing Framework "*UFO*", and the fast control system "*Concert*" [6-7]. A lot of work was put in the system services to simplify interaction of the detector, hardware, and our software components. In this work we will focus on the hardware, interconnects, and the system level software serving as foundation for our platform.

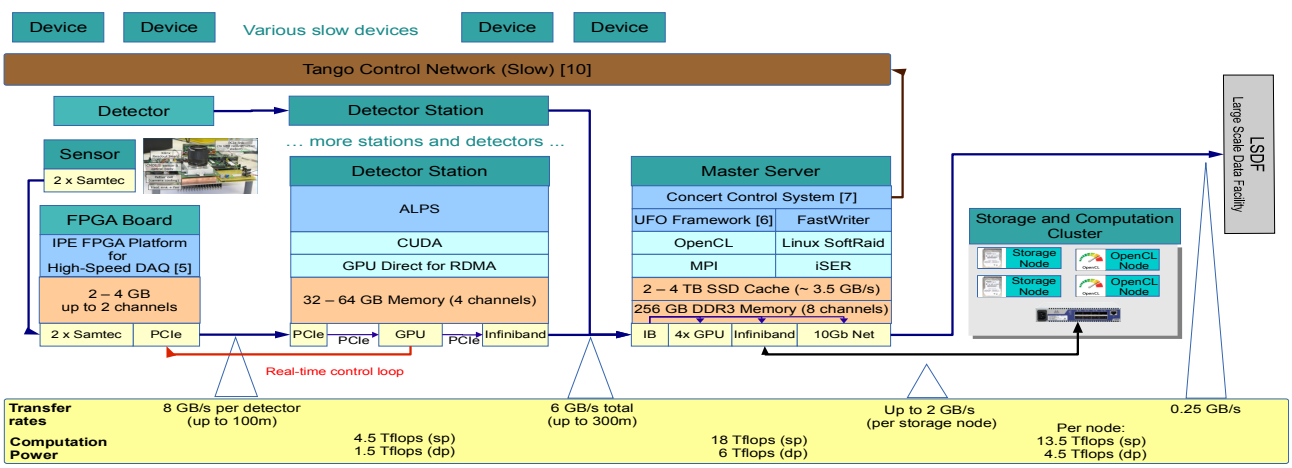

Figure 1: Architecture of control network.

ISBN 978-3-95450-146-5

ARCHITECTURE

The components of the platform are presented on the Figure 1. The detector electronics is built based on high-speed FPGA platform developed in-house. However, we can integrate 3rd party hardware as well. It is connected to detector station using external PCIe interface with electrical or optical cables. The optical connection induces a slightly higher latency but provides the option to place the computing infrastructure up to several hundred meters away from the detector (300 hundred with bandwidth limited to PCIe gen2). The detector station is selected mainly based on the memory throughput and availability of IPMI-type remote control. Currently we are using Asus Z9PA-U8 motherboard with Xeon E5-1620 v.2 processor. The station is equipped with 1-2 GPU cards for data preprocessing and to provide occasional feedback to the detector. From the detector station, the data is streamed to the central server equipped with large amount of memory and high-speed SSD Raid for caching overflow data. To ensure the high speed communication, a Infiniband link is used. The distributed setups are served with the optical cable allowing distances of up to 300 meters. The server is equipped with highly parallel computation units to allow on-line processing of complete data stream. Current configurations are based on Supermicro 7047GR-TPRF platform and equipped with 4x NVIDIA GTX Titan cards which provide reasonable compromise between good performance in both single- and double-precision computations and an affordable price. The system can be easily scaled up with PCIe expansion boxes. Such boxes usually contain 4 or more GPU cards and connected to the host system using a single x16 external PCIe interface. All GPUs in the box are commuted using PLX PCIe switch and transparently accessible to the host system as local devices. Further scalability can be achieved with commodity GPU-nodes integrated using the Infiniband fabric. The computational tasks are efficiently managed by *UFO Framework* and scheduled on cluster nodes with *MPI* and GPUs with *OpenCL* [6,8-9]. The *Concert* controls all experiment devices with Tango control system, but manages memory buffers for direct streaming from high-bandwidth detectors using Infiniband RDMA protocol [7,10]. This is released using KIRO (KIT Infiniband Remote cOmmunication) which extends TANGO protocol with secondary high-speed communication channel while still retaining standard control schemes over TANGO with high-throughput and very low latencies [11]. The received data is then passed to *UFO framework* for online processing. Finally, after analysis the selected data is moved for long-term preservation to large-scale storage facilities at computing center.

The high-speed storage is provided using a storage cluster. After evaluation of various cluster file systems like *GlusterFS*, *FhGFS* we found that a relatively high number of nodes are required to deterministically provide the desired streaming performance. To keep costs low we designed a storage work-flow consisting of 2 stages: real-time streaming and long-term storage. It is presented on the Figure 2. During the experiment, the data is streamed to the smaller but faster real-time storage (RTS). In the pauses between data acquisition sessions, the data is moved from the RTS to the long-term storage. Each storage server in the cluster splits its disk space into the 2 parts. A small and fast space located in the outer edge of the magnetic disks is allocated for RTS. This storage is mapped directly to the Master server using the *XFS* file system over *iSER* (iSCSI over Infiniband) protocol [12-13]. With this solution we avoid performance drops due to network interlocking and may use the remote storage as a local block device. Multiple storage nodes are combined using *Linux Software Raid*. The bigger part of the storage is exposed to the clients and cluster computers using GlusterFS. It can't sustain high streaming rates, but provides a good overall performance when multiple clients are analysing the stored data. To avoid performance penalties by standard *POSIX* stack, we have developed an own data streaming library "*FastWriter*" based on *Linux Kernel AIO* [14]. Using just 2 storage nodes with 16 disks each we were able to achieve a stable streaming read-and-write performance of up to 2.5 GB/s on the RTS partition.

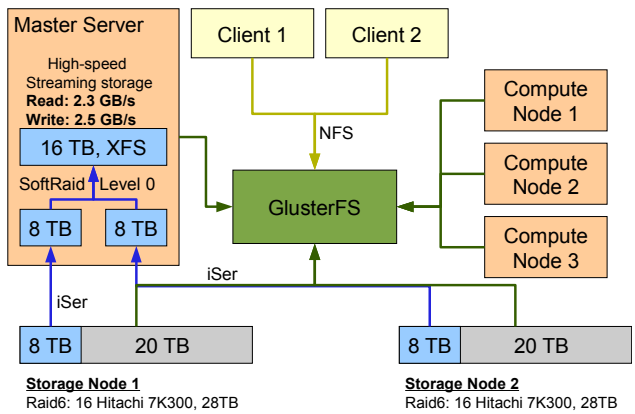

Figure 2: Architecture of the storage subsystem.

ALPS

The central software component is Advanced Linux PCI Services (*ALPS*) allowing us to rapidly implement software support for various PCI-based DAQ electronics. *ALPS* has a modular design and can easily be adopted to new hardware revisions, new DMA protocols, and even new detectors. It provides a flexible scripting interface that enable our hardware engineers to debug the developed electronics. The integration with commercial control software is planned in the next version through Web-Service interface.

The architecture of *ALPS* is presented on the Figure 3. To simplify maintenance we try to keep the Linux driver part as small as possible and move most of the functionality including the implementation of DMA protocol to the user-space library. Basically, the driver is only responsible for device configuration, interrupt handling, and management of DMA buffers. The library provides several API layers to work with the electronics. The PCI Memory layer provides unrestricted access to the device memory, mainly for debugging purposes. The register model is defined in XML and provides an easy

ISBN 978-3-95450-146-5

way to configure the device operation. To support more sophisticated hardware, additional registers can be defined in run-time. The high-speed data communication is carried out by DMA engine. The specific DMA protocols are implemented using plugins. The event engine defines an event-based model to integrate device-specific functionality. Each device can define multiple events and for each event several data types. The events will be triggered in hardware or requested by software. The client application may subscribe to get event notifications. Upon event notification, the application can request the desired type of data. The functionality of the library is fully exposed using command line interface.

Despite of user-space implementation, we are able to sustain the data rates produced by currently operating electronics (up to 2 GB/s) and there is high potential to scale for higher loads.

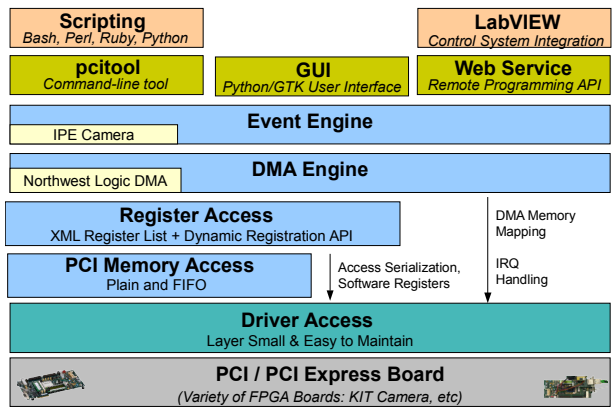

Figure 3: Advanced Linux PCI Services.

OUTLOOK

ALPS was developed having direct PCIe transfers in mind. Currently, we are working to implement the GPUDirect for RDMA and allow direct transfers from the detectors to GPU [1]. This will allow us to process a high data throughput in real-time and create feedback control loops with low latency. Considering that direct GPU-to-Infiniband transfers are already possible with Mellanox drivers, it is possible to configure a direct data flow from the camera to the main processing server bypassing the memory of readout PC at all. This makes high-performance detector readout superfluous and allows us to use small and energy efficient hardware instead. Hence, we get the opportunity to ingrate detector station and electronics. The integrated solution will have Infiniband interface instead of external PCI express which will made it plug-n-play and significantly easier to use. Development of this scheme is in progress using ARM-based SECO GPU Development Kit on the NVIDIA Kayla platform. Restriction to the proprietary NVIDIA *CUDA* technology applied by *GPUDirect* is a single drawback of the approach. However, the restriction only applies to detector station. Further processing at the computing cluster is possible with *OpenCL* which is integral part of our image processing framework and allows more flexibility in selection of parallel hardware.

REFERENCES

[1] G. Shainer, A. Ayoub, T. Liu, M. Kagan, C. Trott, G. Scantlen, P. Crozier, "The development of Mellanox/NVIDIA GPUDirect over InfiniBand—a new model for GPU to GPU communications", Computer Science - Research and Development, vol 26, issue 3-4, pp. 267-273, 2011

[2] S. Chilingaryan, A. Mirone, A. Hammersley, C. Ferrero, L. Helfen, A. Kopmann, T. dos Santos Rolo, P. Vagovic, "A GPU-Based Architecture for Real-Time Data Assessment at Synchrotron Experiments," IEEE Transactions on Nuclear Science, Volume 58, Issue 4, pp. 1447-1455, 2011

[3] D. Haas et al. "Status of the Ultra Fast Tomography Experiments Control at ANKA." Proceedings of the PCaPAC. 2012

[4] M. Caselle, S. Chilingaryan, A. Herth, A. Kopmann, U. Stevanovic, M. Vogelgesang, M. Balzer, M. Weber, "Ultrafast Streaming Camera Platform for Scientific Applications," IEEE Transactions on Nuclear Science, Volume 60, Issue 5, pp. 3669-3677, 2013

[5] M. Caselle, M. Balzer, S. Chilingaryan, M. Hofherr, V. Judin, A. Kopmann, N. Smale, P. Thoma, S. Wuensch, A. Müller, M. Siegel, M. Weber, "An ultra-fast data acquisition system for coherent synchrotron radiation with terahertz detectors, " Journal of Instrumentation, Volume 9, 2014

[6] M. Vogelgesang, S. Chilingaryan, T. dos Santos Rolo, A. Kopmann, "UFO: A Scalable GPU-based Image Processing Framework for On-line Monitoring," Proceedings of HPCC-ICESS, pp. 824-829, 2012

[7] M. Vogelgesang, A. Kopmann, T. Farago, T. dos Santos Rolo, T. Baumbach. "When Hardware and Software Work in Concert." Proc. of the 14th Intl. Conf. on Accelerator and Large Experiment Physics Control Systems, 2013

[8] MPI: A Message-Passing Interface Standard. Available from: http://www.mpi-forum.org/docs/docs.html

[9] OpenCL - The open standard for parallel programming of heterogeneous systems. Available from: http://www.khronos.org/opencl/

[10] A. Gotz, E. Taurel, J. Pons, P. Verdier, J. Chaize, J. Meyer, F. Poncet, G. Heunen, E. Gotz, A. Buteau, et al., "Tango a corba based control system," in ICALEPCS, 2003

[11] T. Dritschler, S. Chilingaryan, T. Farago, A. Kopmann, M. Vogelgesang, "Infiniband Interconnects for High-Throughput Data Acquisition in Tango Environment," In proc. of the PCaPAC. 2014

[12] XFS User Guide. Available from: http://xfs.org

[13] M. Ko, J. Hufferd, M. Chadalapaka, U. Elzur, H. Shah, P. Thaler, "iSCSI Extensions for RDMA Specification," 2013. Available from: http://www.rdmaconsortium.org

[14] M. Jones, "Boost application performance using asynchronous I/O," 2006. Available from: http://www.ibm.com/developerworks/library/l-async/

ISBN 978-3-95450-146-5