THE DATA ACQUISITION SYSTEM (DAQ) OF THE FLASH FACILITY

A. Agababyan, G. Grygiel, B. Fominykh, O. Hensler, R. Kammering, K. Rehlich, V. Rybnikov
Deutsches Elektronen-Synchrotron, DESY, Hamburg, Germany
L. Petrosyan, YerPhi, Yerevan Physics Institute, Armenia, currently at DESY
G. Asova, G. Dimitrov, G. Trowitzsch, M. Winde
Deutsches Elektronen-Synchrotron, DESY, Zeuthen, Germany
T. Wilksen, LEPP Cornell University, Ithaca, NY, USA

Abstract

Nowadays the photon science experiments and the machines providing these photon beams, produce enormous amounts of data. To capture the data from the photon science experiments and from the machine itself we developed a novel Data AcQusition (DAQ) system for the FLASH (Free electron LASer in Hamburg) facility.

Meanwhile the system is not only fully integrated into the DOOCS control system, but is also the core for a number of essential machine related feedback loops and monitoring tasks.

A central DAQ server records and stores the data of more than 900 channels with 1MHz up to 2GHz sampling and several images from the photon science experiments with a typical frame rate of 5Hz.

On this server all data is synchronized on a bunch basis which makes this the perfect location to attach e.g. high level feedbacks and calculations. An overview of the architecture of the DAQ system and it's interconnections within the complex of the FLASH facility together with the status of the DAQ system and possible future extensions/applications will be given.

INTRODUCTION

The FLASH is not only a permanent user facility providing laser like X-ray beams in a before unmatched wavelength regime, but also serves as a R&D study for exploring the superconducting cavity technologies to be used at future linear accelerators.

The requirements for both of these efforts demand a very high level of diagnostic and electronic instrumentation.

To get a deep and clear understanding of the machine, we developed a data acquisition system capable to record all relevant data from about 900 ADC channels distributed over tenths of VME crates with the full machine repetition rate (5 Hz with up to 800 bunches per cycle).

THE FLASH DAQ SYSTEM

The core of the DAQ system is formed of a multiprocessor SUN Fire E2900 hosting 32 GB of memory (the DAQ server) and a SUN Fire X4500 providing 22 TB RAID file space using SUNs Zeta byte File System (ZFS) (Fig. 1).

For the necessary network connections the existing infrastructure of the FLASH accelerator (1 Gigabit Ethernet) is used.

Figure 1: The DAQ core system is build up of two SUN enterprise machines – one hosting 32 GB main memory, the so called buffer manager - the second (used for storing the data of the last weeks) hosting 22 TB disc space.

THE ARCHITECTURE

Figure 2 shows all components the DAQ system. The central component is the buffer manager receiving the data from all participating clients. Here two groups of components can be identified: data providers and data consumers. All of these need to register at the buffer manager before they can access the buffer manager main memory.

Middle layer servers attached to the buffer manager can access all machine data here synchronized on machine shoot basis. The data to be stored is pushed from the distributor process to the event builder which finally writes the data to the main DAQ storage machine. Here the data is kept for approximately one month and depending on its nature and importance, either overwritten or finally moved to a long term tape storage architecture called dCache [3].

The components of the DAQ are in detail:

The Front-Ends

At the FLASH most of the fast monitoring hardware is readout and digitized by VME crates controlled by SPARC CPUs running Solaris OS. To ship the data efficiently to the main DAQ server machine the multicast UDP protocol is used. This allows to scale the whole system to run using multiple DAQ servers which could all *see* the data coming from the front ends without additional load on the front end.

Beside the continuous stream of digitized ADC data a number of camera systems installed at FLASH can send their data to the DAQ server. The camera images are transferred the same way as the ADC data, but make up a very big fraction of the total amount of data running through the main memory. Because of this the camera system are in normal operation not stored to tape but only for certain photon experiments (see figure 3).

Data Collectors

There are two sorts of collector processes: A *slow collector* for pulling slowly varying data from systems like PLC, etc. (e.g. vacuum or magnets), and multiple instances of *fast collector* processes listening to the multicast streams send by the fast front-ends.

Buffer Manager

The core of the DAQ is the buffer manager. It controls all read write operations to and from the main memory. Main ideas and concepts for this part of the DAQ system are borrowed from high energy physics experiments.

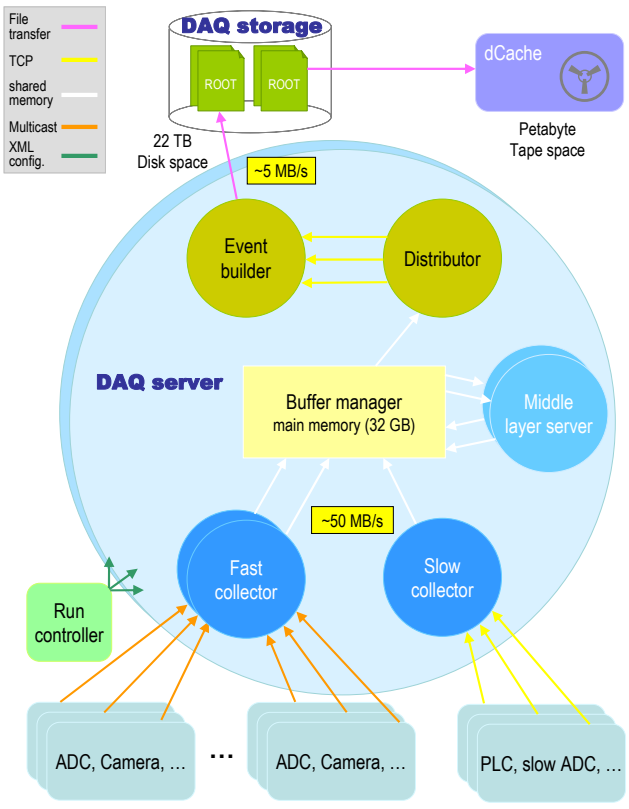

Figure 2: The architecture of the DAQ system. The blue circle is the domain of the main DAQ machine. On the top there are the storage facilities – the temporary disk storage and the long term storage tape archive.

Middle Layer Processes

Since all beam relevant data is passing through the buffer manager's main memory, this is the perfect place for attaching slow feedbacks and monitoring processes. A number of high level control and monitoring programs have been implemented this way already.

Distributor, Event Builder Processes

To have maximum flexibility in control of the data passed to the final storage destinations, a *distributor process* passes a set of streams to an *event builder* process which actually writes the data to discs. A fine graded control of the data streams allows to select down to the property level, which data is going to the log term storage and which data will be overwritten after a few weeks.

DAQ Storage

The event builder writes the data to be stored to the DAQ storage machine. Here all the beam relevant data of the last ~ 30 days is kept, for having good performance on retrieval for this data. By default the data is written in ROOT [4] file format.

dCache

Finally data older than 30 days (or earlier in case of unusual big data output stream sizes) is moved to the central tape storage, hosting a sophisticated long term archive architecture called *dCache* [3].

Run Controller

All clients participating in a *DAQ run* get the configuration (repetition rates, number of bunches, …) transmitted by XML strings. A finite state machine implemented in **all** clients, used for synchronization, is supervised by the run controller.

All of the describe components are implementing the DOOCS API so that they can be accessed and controlled by using standard DOOCS communication and tools. This way the system seamlessly integrates into the existing control system infrastructure already present at FLASH.

USAGE EXAMPLES

Different from the original expectations, it showed that not the machine physicists but instead the photon experiments done at FLASH are heavily making use of this new possibility for mass data storage. Here the huge data volume produced by various camera systems and some gigasample ADCs need to be stored and methods for comfortable data retrieval must be provided.

Most experimenters are just doing one experiment at this facility, since up till now beam time in this new wavelength regime is very limited, especially since the photon beam FLASH produces is up till now unique in the world. So experiments are not running that routinely like at synchrotron facilities and are mostly not prepared for the huge data rates they accumulate within short beam times at the FLASH. Therefore it is of great benefit for the photon experiments if the necessary infrastructure for storing hundreds gigabytes is simply available on site and can directly be used.

Since CERN's ROOT file format is relatively uncommon in the photon sciences, we need to extend the capabilities for easy data retrieval, evaluation and interpretation.

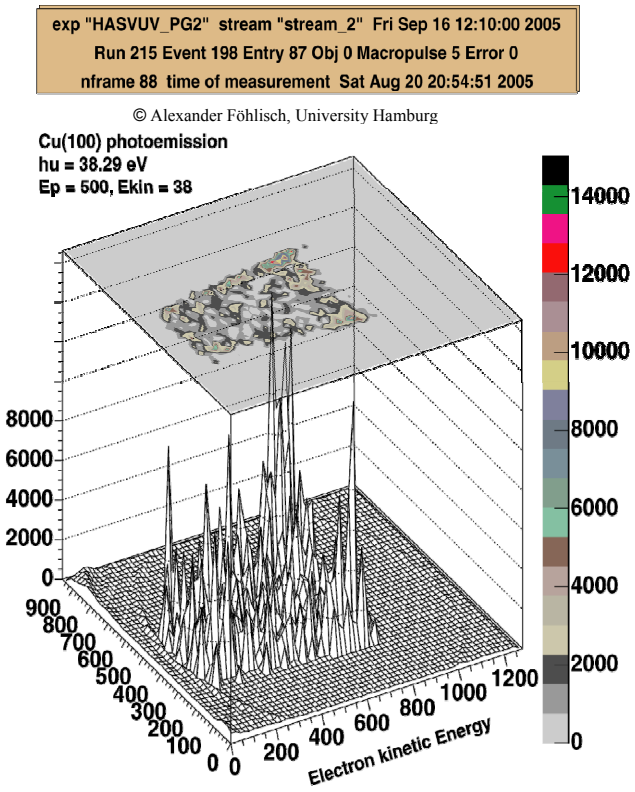

Figure 3: The picture shows the photo emission spectra of Cu(100) recorded by the DAQ system during the first FLASH user run.

The second major impact the DAQ system has for the machine operation are the new opportunities to access the full data of the last machine shoots in a synchronized way at the stage of the buffer managers shared memory. A number of processes needing lots of data from the machine with full resolution and repetition rate could only be implemented due to this architecture. Some of these are now already part of the standard machine operation. Here an API to easily attach C++ and MATLAB based applications has been developed and is already in use since more than a year thereby proving its practicability and reliability.

Figure 4 shows two displays of the energy monitoring server and the architectural layout for this middle layer software as an exsample. This server is written in a very generic way thereby allowing to easily adopt it to different purposes [5].

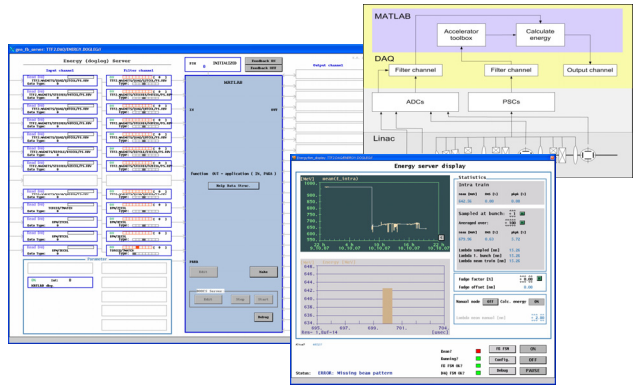

Figure 4: One example of a DAQ based middle layer server is the energy monitor used at FLASH now already routinely.

CONCLUSIONS

The DAQ system developed and now already running more than one year reliable at the FLASH facility, has extended the possibilities for understanding and controlling a complex linear accelerator like this.

The novel combination of high energy physics (HEP) techniques and an accelerator control system successfully shows that such an integrated framework can solve the requirements demanded by the high data rates and huge number of components.

After this stage of implementation and commissioning of the DAQ system we need to extend the efforts to ease the access to the stored data. Here it turns out that every specific user group as its own favorite data analysis tool (machine physicists love MATLAB whereby the photon experimenters are more using tools like *Igor* or *Origin*).

The FLASH DAQ system was developed in a collaboration of three institutes: DESY Zeuthen, DESY Hamburg and LEPP Cornell University, Ithaca, NY, USA

REFERENCES

[1] http://flash.desy.de
[2] http://doocs.desy.de
[3] http://dCache.desy.de
[4] For ROOT see: http://root.cern.ch
[5] PCaPAC 2006, DAQ based high level software applications using MATLAB