STATUS OF THE FLASH FREE ELECTRON LASER CONTROL SYSTEM

K. Rehlich, DESY, Hamburg, Germany

Abstract

FLASH (Free electron LASer in Hamburg) is the first facility based on the 1.3GHz superconducting cavity technology. It is a test bed for this technology to prepare future accelerators like the XFEL and the ILC. Since 2005 FLASH runs as a reliable FEL source for user experiments. The control system DOOCS (Distributed Object-Oriented Control System) provides the required full bunch resolution of the diagnostics. A fast DAQ (Data AQuisition system) has successfully been integrated to support slow feedback, diagnostics and data recording for both, the linac operation and the user experiments. The control system will be slowly upgraded to implement the further requirements for the XFEL.

INTRODUCTION

Free electron lasers like FLASH [1] require a lot of diagnostics. Operating experience shows that the performance of the accelerator depends on small variations of a lot of parameters. Every electron bunch passes the linac only once and has slightly different parameters and the photon generating process has a statistical behaviour. A well synchronized data recording on a bunch basis is therefore mandatory.

DOOCS ARCHITECTURE

The DOOCS architecture is based on a three tier software model. At the bottom layer are the device servers with connections to the hardware. The middle layer is called the service tier with name servers, automation, DAQ and Web servers. On the top level are the client programs as graphical user interfaces, measurement programs or any other application for operators and experts.

Device Servers

In FLASH device servers are implemented in two categories. The first group is implemented in VME crates with a hardware connection to the timing system. In this set up all crates are synchronized by interrupts and event numbers of the timing system. The second group of device servers are running unsynchronized with the FLASH bunches. The tasks of these servers are to read slow systems like vacuum or to control motors via field busses. Implemented field busses are CAN, ProfiBus, SEDAC (a DESY electronics standard), GPIB, RS232, SNMP, IPMI, and Ethernet.

FLASH has a repetition rate of up to 10Hz with up to 800 bunches per shot. The bunch frequency is 1MHz and the data taking time is 2ms per shot. The complete synchronization of bunches and shots is provided by the timing system. This system is used for the sample clocks of the ADCs and the distribution of programmable hardware and software triggers. In addition event numbers are sent to all timing receivers to allow the software to mark the data. By this means all individual bunches can be identified in the distributed computers.

DOOCS server processes are based on a multi-threaded library. This library implements the entire network I/O, interrupt handling, configurations and the archiving. All tasks with long execution durations run in separate threads. Programmers are freed from the difficult task to implement multi-threading. Long-time archiving a variable is another example that does not have to be programmed or configured by a server designer. Device access and persistence of configuration parameters is a further example.

Camera System

In FLASH about 90 cameras are in use. Almost 30 cameras are operated by the ODCS [2] system for the OTR (optical transition radiation) system. The other cameras are implemented by a new LINUX based environment [3]. Very different camera and hardware connection techniques are managed by an object oriented approach. A standard base class implements the common functions. Various special classes interface to simple USB, FireWire, Ethernet and high performance frame grabber cameras. The high-end cameras provide some functions in hardware (e.g. Region Of Interest) while the software has to implement this for the simple ones in a special class.

From the user and programmer point of view all cameras appear in a similar way. The different features are adapted in the DOOCS device server. The server interface provides full control of the camera parameters and region of interest definition as well as projections and binnings. Images can be read directly by programs, stored in files or send to the data acquisition system (DAQ). Since the camera servers are connected to the timing system a later correlation with a bunch resolution to other data is guarantied.

Fast Data Acquisition System

The fast DAQ system [4] is a novel combination and integration of an accelerator control system and a high performance data collector known from high energy physics experiments. Device servers are configured to send their interrupt triggered data by multicast packets to the network. Collectors in a DAQ server receive this data and sort it into a huge shared memory in a shot by shot order. The DAQ is a middle layer service. Since it provides all machine data of the recent shots in a shared memory it is an ideal place to attach calculation and feed-back processes to it. Orbit correction, phase feed-back and energy measurements are examples of these services implemented in FLASH. A description follows in the next chapter.

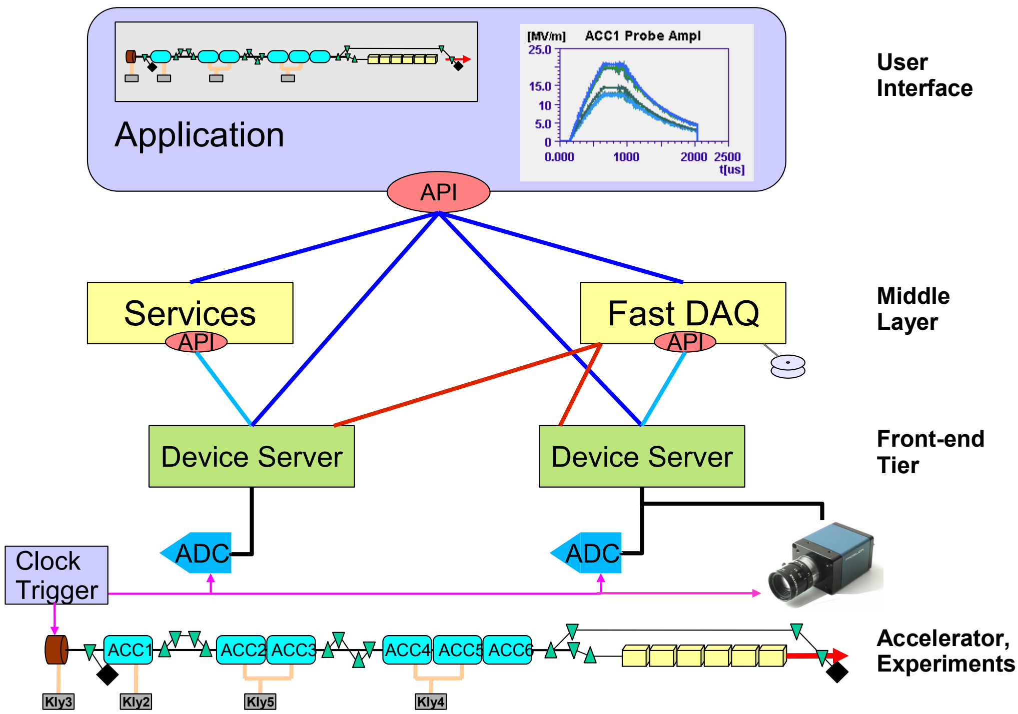

Figure 1: Three tier architecture of the DOOCS control system.

The collected data of the DAQ system is stored in files on a local 24TB disk. This disk can store up to three weeks of FLASH data. Selected data is then transferred to a tape robot in the computer centre. Different streams can be configured by the system. And multiple instances of the DAQ can run in parallel. This allows an independent data handling for photon experiments at FLASH and the data of the accelerator.

Feedback Systems

Often feedbacks are developed as application programs with a high level language like MATLAB. This helps for rapid prototyping. But, when several feedbacks are operated in parallel it is quite hard to inform an operator about their states or to allow him to switch them off. Therefore, feedbacks in FLASH are placed in the service tier of the control system after the prototyping phase. A framework was developed to ease the interfacing of MATLAB or C++ routines to the DAQ.

Communication API

FLASH was designed and build by an international collaboration. A lot of different pieces of hardware and software had to be integrated. A design goal of the control system was to hide the differences and to let the whole system appear in one name space. The application programming interface (API) implements different protocols: RPC, TINE, EPICS and a shared memory access. A name server provides the information of all names in the system including the protocol to use.

APPLICATION DEVELOPMENT

2007 the XFEL project was started. The XFEL is a European free electron laser with a length of about 3 km. It uses the same technology as FLASH. FLASH can therefore be considered as a prototype. Because of the size of the new project, availability concerns are more important and have to be addressed in the hardware and software design. Also the standards being used need to be reviewed.

jddd

12 years ago ddd (DOOCS Data Display) was developed. It is a graphical editor for control system panels that can be used by non programmers. Complex panels including animations can be created. Meanwhile about 1300 panels exist. The editor was developed in C++ with the support of X-Windows and OpenLook libraries. A new project was started to implement all the functions on the modern platform independent language JAVA. In this new design ‘from scratch’ it is planned to implement a lot more functionality. An export of existing panels and display in the JAVA program was already demonstrated.

jDTool

This is a further tool written in JAVA. It allows displaying the address hierarchy of all variables in the control system. A second window shows selected data in a spreadsheet. Columns present properties of devices and the rows a list of devices. Both, rows and columns can be added by drag and drop from the address tree. The data in the table is updated from the control system and modified data can be sent back to the devices.

Alarm Display

The alarm display presents the status of all devices also in a tree view. Icons of the tree nodes represent the actual alarm status of the leaves. A second window shows a ticker with all incoming messages. The third window displays selected histories of alarms of channels or device groups. This program is another JAVA design. Messages are transported from a Web server via JMS (JAVA messaging service). Device servers update the central Web server by XML messages with the alarm information.

HARDWARE DEVELOPMENTS

The FLASH front-end hardware is mainly based on VME. VME is now a quite old standard and support in seven years from now could be difficult. Parallel busses are replaced in modern computer hardware by Gigabit serial links. Improved redundancy and a standardized management environment would be beneficial. ATCA (advanced telecom computing architecture) is a state-of-the-art crate standard that provides a lot the features required for the XFEL. It was also defined as the basis for the ILC design phase.

To verify the performance of the new ATCA and µTCA standards some projects were started to evaluate hardware modules and the crate and module management.

CRATE MANAGEMENT

The ATCA standard defines an IPMI protocol for the management on all levels. Every AMC (advanced mezzanine card) contains a little microcontroller for board supervisory tasks. This includes hot-swap control, power control and measurements of temperatures and voltages etc. The same functions are available on the crate level as well. Bookkeeping by inventory queries is a further feature of the IPMI definition.

A DOOCS server for the IPMI protocol was created. The server reads crates and dynamically creates addresses and variables in the control system with the individual properties of the inserted modules. The server delivers an online inventory of all crates and modules in the system. On ddd or jddd panels an always up-to-date display is presented without reprogramming or redesign.

Modern PCs implement the IPMI protocol as well. The same DOOCS server can be used to control these PCs.

REDUNDANT SERVER

User facilities require high availability. With the increasing number of components this becomes a serious concern. Single points of failures are removed by backup devices or subsystems. The control system has to support automatic failover. A first implementation was successfully tested with the two redundant laser systems of FLASH. Several DOOCS device server are involved in a switch-over from an active laser to the standby laser. Client programs have to use a generic address to communicate with the active laser. The usage of the generic name involves no action in the client program if the active subsystem switches. In addition a program can talk to a specific laser by its specific address. The specific address will never switch over.

This redundancy support was implemented in the communication protocol and in the name server. A management task can check the systems and inform the name server and the subsystem servers to change from the active to the standby system.

CONCLUSIONS

FLASH runs as a FEL user facility since 2005. The control system provides bunch synchronized data recoding by a novel integration of a fast DAQ system. The new XFEL project started this year. Although the control system for FLASH demonstrated a reliable operation with all required features for the new project, it is time to review technology changes for the new project. An evaluation of new crate standards of the telecom industry was started. More reliable operation and integrated management can be expected compared to VME.

Three JAVA based user level programs were developed. With jddd it could be demonstrated that JAVA provides a rich, well developed programmer infrastructure as well as the required features for operators. With jddd it was possible to develop an easy to use editor that allows creating complex operator panels. Further software was developed to gain experience with the new IPMI standard. It could be demonstrated that automatic hardware detection with all available properties of a module can be integrated into the controls system.

REFERENCES

[1] Rossbach et al, “Generation of GW radiation pulses from a VUV free-electron laser operating in the femtosecond regime”*, Phys. Rev. Lett. 88, 104802 (2002.)*

[2] L. Catani, “Integration of a PC based experiment in a Network based Control System: the TTF Optical Diagnostics Control System”, PCaPAC98, Tsukuba (Japan).

[3] G. Grygiel, V. Rybnikov, “DOOCS Camera System”, ICALEPCS 2007, Knoxville,USA.

[4] R. Kammering, et al. “The Data Acquisition System (DAQ) of the FLASH Facility”, ICALEPCS 2007, Knoxville,USA.

[5] E. Sombrowski, “jddd: A JAVA DOOCS Data Display for the XFEL”, ICALEPCS 2007, Knoxville,USA.

[6] O. Hensler et al, “A Proposal for a Central Error server using Web Services”, PCaPAC 2002, Frascati, Italy.