THE VIRTUAL EUROPEAN XFEL ACCELERATOR

R. Kammering, W. Decking, L. Froehlich, O. Hensler, T. Limberg, S. Meykopff, K. Rehlich, V. Rybnikov, J. Wilgen, T. Wilksen, DESY, Hamburg, Germany

Abstract

The ambitious commissioning plans for the European XFEL require that many of the high-level controls are ready from the beginning. The idea arose to create a virtual environment to carry out such developments and tests in advance, to test interfaces, software in general and the visualisation of the variety of components. Based on the experiences and on the systems that are already in operation at the FLASH facility for several years, such a virtual environment is being created. The system can already simulate most of the key components of the upcoming accelerator. Core of the system is an event synchronized data acquisition system (DAQ). The interfaces of the DAQ system towards the device level, as well as to the high-level side is utilising the same software stack as the production system does. Thus, the software can be developed and used interchangeably between the virtual and the real machine. This allows to test concepts, interfaces and identify problems and errors at an early stage. In this paper the opportunities arising from the operation of such a virtual machine will be presented. The limits in terms of the resulting complexity and physical relationships will also be shown.

THE IDEA

Lessons learned from the fast successful start up of the Linac Coherent Light Source (LCLS) at the SLAC National Accelerator Laboratory showed that one needs to have not only all hardware related software ready and checked from the first hours on, but also all foreseen high level software.

Since most high level software is not directly acting on the hardware layer itself but needs a vast amount of infrastructure to be working properly, testing high level software can only be done with this infrastructure up and running. However, having the high level software ready and tested prior to having the first beam is in contradiction. To still be able to accomplish this task, the idea of having a test environment for this, came up within our group which is in charge of creating high level controls and applications. For a general overview of the concepts and architecture of the foreseen high level software for the European XFEL see [1].

FROM FLASH TO XFEL

The FLASH facility can be seen as the *small brother* of the European XFEL. On the one hand due to the use of the same hardware technologies, but also in the sense of layout, beam physics and creation of the FEL pulses.

Similarities

FLASH has always been used for prototyping and testing hardware to be later be used at the European XFEL. To name some examples of the major hardware systems, there are the:

- µTCA crate standard
- Timing System
- Machine Protection System

Thus FLASH serves already since several years as the test bed for the interplay between the new hardware systems and the software layers for processing and providing this data to the operators and experts. But the European XFEL has, roughly speaking, ten times the extend compared to the FLASH facility. Table 1 shows a comparison of some hardware and characteristics for both machines.

Table 1: FLASH vs. XFEL

System	FLASH	XFEL
Crates	~ 30 VME, µTCA crates	~ 200 µTCA crates
BPMs	~ 40	~ 460
Data rate to DAQ	< 100 Mbytes/s	>> 100 Mbytes/s
Length	~ 300 m	~ 3000 m

The shear number of devices to be installed in the European XFEL and the thereby resulting data rates require a strong data reduction and pre-processing already on the lower hard- and software layers. But beside the need for data concentration at an early stage, the operation of such a large scale facility as the European XFEL require precise synchronization of many different hardware devices.

Further did lessons learned from the operation of the FLASH facility show that one needs to work more in terms of physical meaningful values instead of reading *raw* data directly from the hardware devices.

Even though many ideas and concepts for high level applications can be lend from the FLASH facilty, one needs to take the increased data rate into account. A central concept to overcome this, is to use the, at the FLASH facility already since years used, data acquisition system.

ISBN 978-3-95450-148-9

THE DAQ SYSTEM AS THE CORE OF THE CONTROL SYSTEM

The overall structure and concepts of the control system will not be discussed here but can be found at e.g. [2]. For details about the status of the controls of the European XFEL see [3].

The data acquisition system of the European XFEL's control system is of major importance for the foreseen high level applications. The overall architecture and concepts of the DAQ system itself have been discussed in e.g. [4]. To enlighten the central role of the DAQ system within the scope of the virtual XFEL these aspects will be discussed here in more detail.

The big amount of data produced by the front-ends require data reduction before the data can be made available to the higher levels and finally to the display level. But beside the pure need for data reduction also the data from various different sources needs to be synchronized. Further more did lessons learned from the FLASH facility have shown that one needs to think much more in terms of *physics entities* instead of passing the digitized data simply up to the user or operator level.

To accomplish this all data from the various front-ends is pushed up to the DAQ system using multicast transmission. This allows for having multiple instances of the DAQ system without the need to re-send data multiple times over the network.

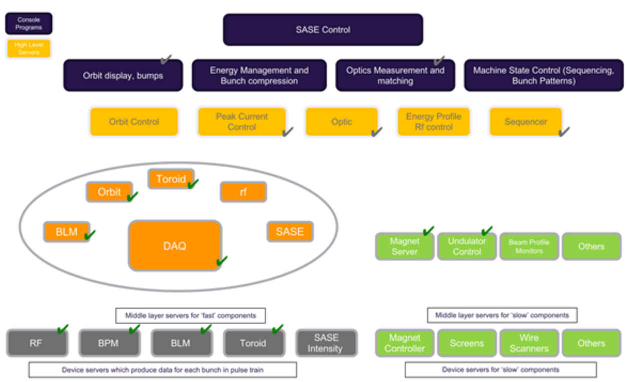

Figure 1: Overview of the XFEL control system architecture (with checkmarks marking *ready* components as to mid 2014).

The overall structure of the control system follows the typical three layered architecture as shown in figure 1.

The DAQ system serves in this sense as the place to attach middle layer processes to do data concentration and pre-processing. As shown in figure 1 are there multiple of these middle layer services attaching to the DAQ system.

THE VIRTUAL XFEL

With having already numerous server processes ready, as depicted in figure 1, we have been able to produce a closed loop through the control system architecture simulating a realistic data flow for the beam position monitors (BPMs) as shown in figure 2.

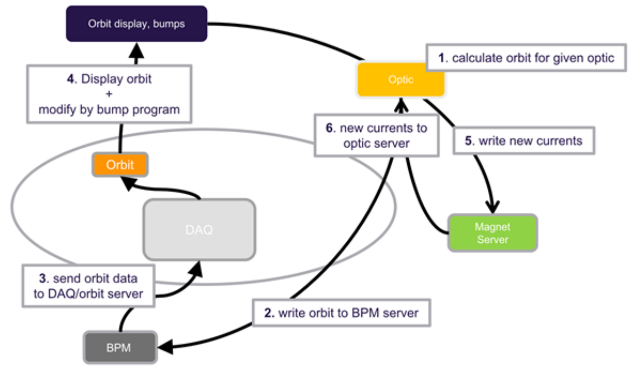

Figure 2: Simulated closed loop data flow for the case of the beam position monitors.

Starting with the test for the BPMs we where able to test not only the network infrastructure but also the processing speed of the attached middle layer servers (in this case the orbit server). After proper configuration and adjustments, a continuous data throughput of roughly 160 Mb/sec had been archived allowing to run with the foreseen nominal repetition rate of the European XFEL of 10 Hz. Figure 3 shows a betatron oscillation within the horizontal plane, introduced by a disturbance (modification of a magnet current) in the injector region of the European XFEL.

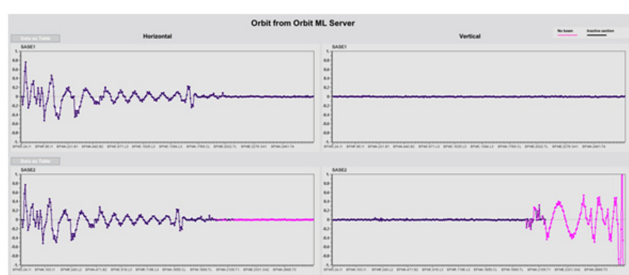

Figure 3: Betatron oscillation introduced within the data flow of the beam position monitors simulating the closed loop.

After accomplishing this first test, which had been the case of the original idea, we started to extend the system step by step by further front-end server processes and corresponding middle layer processes. Figure 4 shows a jddd [5] panel showing all available processes of the virtual XFEL as to date of mid 2015.

ISBN 978-3-95450-148-9

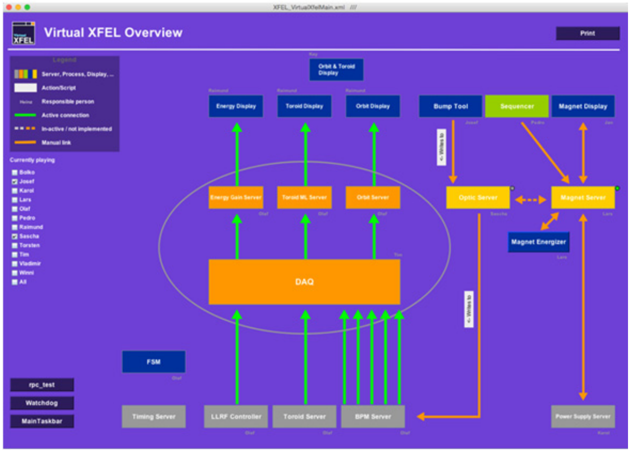

Figure 4: The jddd based control panel of the virtual XFEL (status mid 2015).

Up to date the following hardware systems have been integrated in the simulation of the European XFEL:

- Beam position monitors (BPMs)
- Toroids
- LLRF partial vector sums
- Beam loss monitors (BPMs)
- Magnets

SUMMARY

Starting as a pure test for simulating the data throughput, it soon showed up that the virtual XFEL can be used for much more than this. Especially the test and debugging of all processes in need of synchronized shoot data could be done within this simulation, thus allowing to test handling of complex bunch patterns foreseen to be run at the European XFEL.

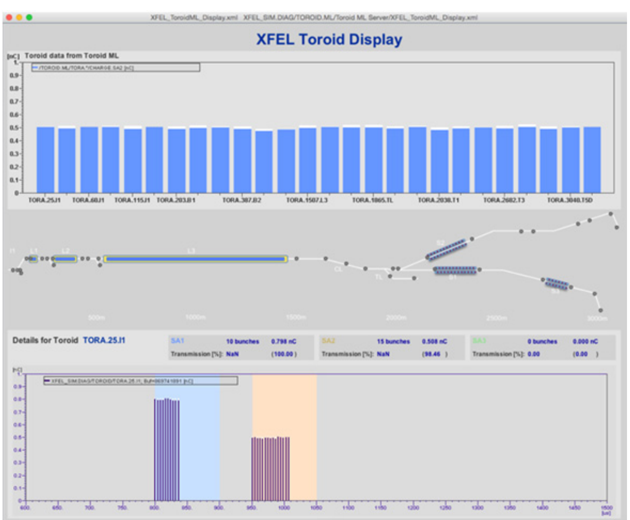

Figure 5: Transmission display developed at the virtual XFEL and nowadays used in standard operation at the FLASH facility.

But even further it showed up that this environment is perfectly suited to do development, testing and debugging of complex graphical user interfaces (see figure 5).

Beside using the virtual XFEL for testing software, configurations created here allow to port the software unchanged from the virtual to the real machine. This is archived due to the reason that the software above the front-end level is completely unchanged and thus only minimal modification of the naming is needed to do the porting.

CONCLUSIONS

The virtual XFEL showed to be of much more use than original intended. Nearly no extra work needed to be done to get this simulation working. Higher level software developed here can be used unchanged later on at the real machine and even configurations can be ported one to one with only minimal modifications.

Thus in contrast to the original idea, it has been decided to keep the hardware for the virtual XFEL dedicated for the simulation and instead build up *a copy* of the system for the real machine.

Working now already for more than a year with this simulated accelerator it showed up that beside all benefits of the virtual XFEL, it also behaves like a real machine. Keeping this virtual machine up and running is also causing some extra work (somewhat it really *feels like a real machine*). Thus a good balance of what is the simulation to be used for and where the limits are being needed here. Thus did it turned out to be in strong contradiction to do basic software developments while someone else is working on optics simulations in parallel.

Nonetheless did the tremendous success of this project showed that setting up an accelerator simulation can sometimes be done without any big extra effort, while providing an enormous profit and thus allowing to be well prepared much ahead of time before the real hardware is in place.

REFERENCES

[1] B. Beutner et al, "High level software for the European XFEL", ICALEPCS'15, Melbourne, Australia, 2015

[2] G. Grygiel, O. Hensler, K. Rehlich, "DOOCS: a Distributed Object Oriented Control System on PC's and Workstations", PCaPAC96, DESY, Hamburg, October 1996

[3] A. Aghababyan et al, "The large scale European XFEL control system: Overview and status of the commissioning", ICALEPCS'15, Melbourne, Australia, 2015

[4] K. Rehlich et al., "*Multi-Processor Based Fast Data Acquisition for a Free Electron Laser and Experiments*", IEEE Transactions on Nuclear Science, vol. 55, no. 1, February 2008, pp. 256-260

[5] E. Sombrowski, A. Petrosyan, K. Rehlich, P. Tege, "jddd: A Java Doocs Data Display for the XFEL", ICALEPCS'07, Knoxville, Tennessee, USA, 2007

ISBN 978-3-95450-148-9