STATUS OF THE FLUTE CONTROL SYSTEM

S. Marsching*, N. Hiller, V. Judin, A.-S. Müller, M.J. Nasse. R. Ruprecht, M. Schuh
Karlsruhe Institute of Technology, Karlsruhe, Germany

Abstract

The accelerator test facility FLUTE (Ferninfrarot, Linac-Und Test-Experiment) is being under construction nearby ANKA at the Karlsruhe Institute of Technology (KIT). FLUTE is a linac-based accelerator facility for generating coherent THz radiation. One of the goals of the FLUTE project is the development and fundamental examination of new concepts and technologies for the generation of intensive and ultra-broad-band THz pulses fed by femtosecond electron-bunches. In order to study the various mechanisms influencing the final THz pulses, data-acquisition and storage systems are required that allow for the correlation of beam parameters on a per-pulse basis. In parallel to the construction of the accelerator and the THz beam-line, a modern, EPICS-based control system is being developed. This control system combines well-established techniques (like S7 PLCs, Ethernet, and EPICS) with rather new components (like MicroTCA, Control System Studio, and NoSQL databases) in order to provide a robust, stable system, that meets the performance requirements. We present the design concept behind the FLUTE control system and report on the status of the commissioning process.

INTRODUCTION

FLUTE (Ferninfrarot, Linac- Und Test-Experiment) is a new accelerator facility [1,2] being under construction at the Karlsruhe Institute of Technology (KIT). FLUTE is designed as an accelerator test facility, aiming at studying and improving techniques for producing very short electron bunches and studying the mechanisms involved in the generation of THz radiation from these electron bunches.

In the first stage, the accelerator will consist of a 7 MeV photo injector followed by a 40 MeV linac and a magnetic chicane used as a bunch compressor, as depicted in Fig. 1. Finally, synchrotron radiation generated in the final magnet of the bunch compressor or by an optionally inserted foil after the magnet is coupled out into a THz beamline for use by experiments. This THz radiation can then be analyzed in order to investigate properties of the electron bunch. However, it can also be used for independent experiments.

In order to make systematic studies of the parameters and mechanisms affecting the bunch compression and the generation of THz radiation, a pulse-synchronous data-acquisition system is needed. Such a system has to record key parameters (e.g. accelerator settings, RF pulses, laser pulses, beam charge and profile) for every single electron bunch, so that they can be correlated. At the same time, the control system has to provide live information, that is needed in order to optimize the accelerator operation.

* sebastian<dot>marsching<at>partner.kit.edu

CONTROL SYSTEM DESIGN

The design of the FLUTE control system was driven by the aforementioned demands as well as by the experience with the control system used for the ANKA accelerators.

We chose EPICS [3] as the core control-system framework for several reasons: First, EPICS is already used at many accelerators and thus there is a large number of already existing device drivers, reducing the development costs. Second, the underlying concepts are simple and thus easy to understand. This allows for a short training of new team members, so that they can quickly start working on control-system related tasks. Finally, EPICS is already in use at ANKA, so that we can benefit from our experience.

We use standard off-the-shelf components whereever feasible. This means that most computers are x86_64 systems running a Linux operating system. All systems are connected to a private IP/Ethernet network. The individual network connections use Gigabit Ethernet, however the backbone is already designed for 10 Gigabit Ethernet, allowing for higher data rates in the future.

We use the MicroTCA [4] platform for fast data-acquisition and feedback systems (e.g. the low-level RF (LLRF) system and beam-position monitor readout). This allows us to use the x86_64 platform while having the input/output (I/O) capabilities required for those applications.

For slow control tasks, we use devices with embedded IP/Ethernet controllers or serial interfaces where possible. Other devices are integrated using SIEMENS S7 programmable-logic controllers (PLCs) [5]. We use GigE Vision [6] cameras, allowing for a direct connection to the control system network.

We use Control System Studio (CSS) [7] as the main operator's interface to the control system. CSS is already in use at ANKA and provides an integrated user interface with tools for designing operator's panels, plotting archived data and displaying the alarm status.

DATA ACQUISITION

While the accelerator will initially operate with a repetion rate not exceeding 10 Hz, the data acquisition system is designed for repetition rates of up to 50 Hz to allow for upgrades in the future. This does not mean, that all diagnostics components can operate at this frequency, however the data-acquisition framework is supposed to handle this rate.

In EPICS, each process-variable (PV) sample has a time stamp with nanosecond precision. However, it is hard to synchronize the clocks of different components so accurately, that the time stamp can be used to correlate samples. There are systems like White Rabbit [8] that can provide a sufficiently synchronized clock across different computer

ISBN 978-3-95450-146-5

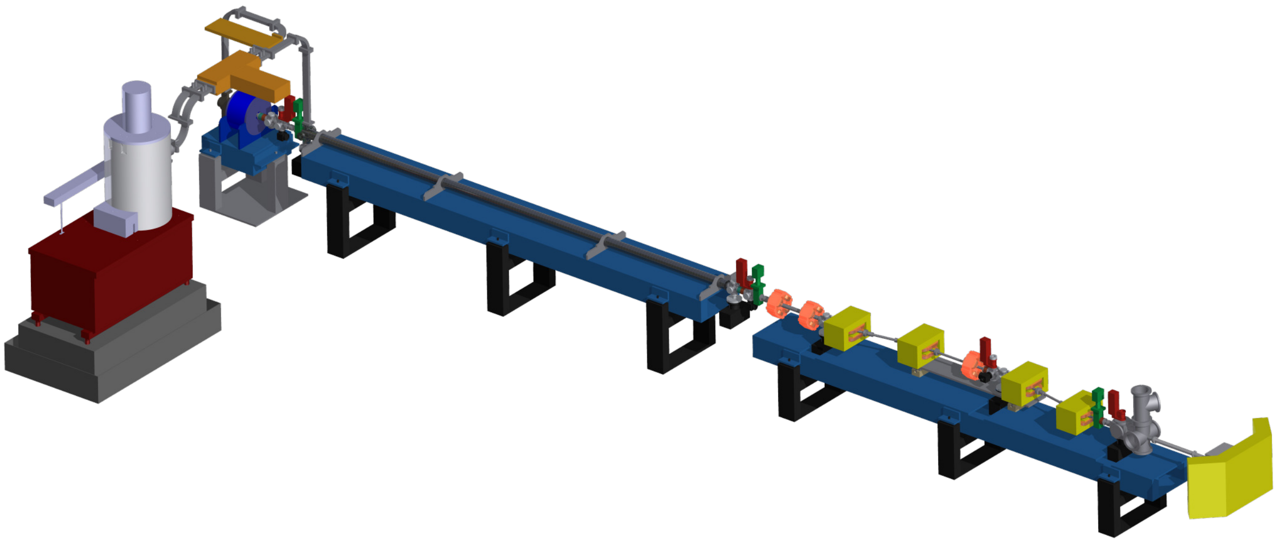

Figure 1: Layout of the FLUTE accelerator: Electron bunches are generated in a photo-RF injector (left), accelerated in an S-band Linac strucuture, and longitudinally compressed in a magnetic chicane, before being used to generate THz radiation. Finally, the electron bunches are destroyed in a beam dump (right). The injector and the linac are both powered by the same klystron (lower left).

systems, however these systems typically rely on special hardware. This would undermine the objective of using standard off-the-shelf hardware so that are different scheme is used.

Instead of trying to use consistent time stamps across the whole control system, the consistency of time stamps is only guaranteed within a single component. In addition to the PVs representing the actual measured quantities, each component has an artificial PV that represents the pulse counter. The time stamp of the pulse counter PV is synchronized with the time stamps of the measured PVs. Each time new measured data arrives, the pulse counter is incremented. This way, each PV sample can be correlated with a pulse sequence number. The measurement equipment can be synchronized using an external trigger signal that is independent of the control system. This solution has the advantage that it works with any measurement hardware (even hardware that is connected using an asynchronous interface like RS-232) as long as the measured data is transferred using a FIFO (so that no samples are lost) and the hardware supports some kind of external trigger.

ARCHIVING AND PROCESSING

We use an archiving solution [9, 10] based on Apache Cassandra [11] for archiving the PV values. The data is automatically distributed on a cluster of servers (see Fig. 2) allowing for both reliability and scalability. The performance scales linearly [12] with the number of cluster nodes, thus we can easily match future demands by just adding more computers.

The use of Apache Cassandra also allows us to use a MapReduce [13] based algorithm for agregating the PV values belonging to the same pulse and running further analysis. This way, we can benefit from the fact that the data is distributed in the cluster and run the analysis code on the same node that stores the corresponding data, thus improving the performance. We are currently evaluating Apache Spark [14] as the software framework for running this analysis code.

PROJECT STATUS

The development and commissioning of the control system is ongoing. Two virtual machine hosts, each running several virtual machines (e.g. for DHCP and DNS servers), have been prepared and are running in a test lab together with a PC designated as the operator's console. Other control-system components (e.g. network switches) have already been delivered and have been tested.

The software development is progressing. We have already created device drivers for some components (e.g. motor controllers). In close collaboration with DESY, we are working on integrating the LLRF electronics (developed and supplied by DESY) into our EPICS control system [15].

We expect the building infrastructure for FLUTE to be finished in early 2015, so that we can move the equipment from the test lab to its final location until spring 2015 and subsequently connect everything to the actual accelerator components. Thus, we expect to have first beam operation in 2015.

SUMMARY

For FLUTE, an EPICS-based control system has been designed and is currently under development. The control system uses standard off-the-shelf components where possible and leverages modern NoSQL-database technologies for data archiving and analysis. The control system is due to be

ISBN 978-3-95450-146-5

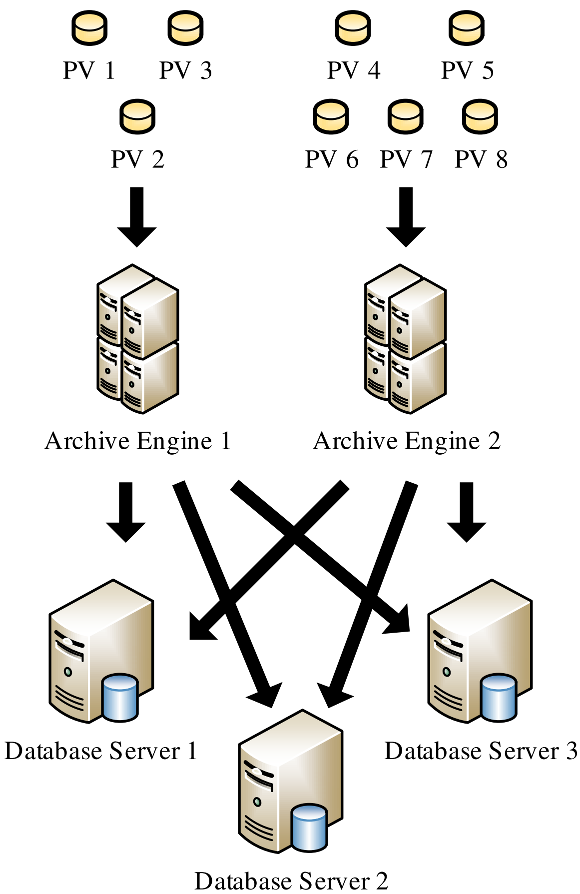

Figure 2: Different process variables can be handled by an indefinite number of archive engines. The archive engines automatically distribute the data on a cluster that can range from a few to a few hundred nodes and can be extended while running.

commissioned in early 2015 or and first beam operation is expected in 2015.

REFERENCES

[1] M.J. Nasse et al., "FLUTE: A versatile linac-based THz source", Rev. Sci. Instrum. 84, 022705 (2013).

[2] M. Schuh et al., "Status of FLUTE", IPAC'14, Dresden, June 2014; http://www.jacow.org/

[3] "EPICS - Experimental Physics and Industrial Control System" website: http://www.aps.anl.gov/epics/

[4] S. Jamieson, "Micro Telecommunications Computing Architecture Short Form Specification", PICMG, September 2006; http://www.picmg.org/pdf/MicroTCA_Short_Form_Sept_2006.pdf

[5] Siemens, "Modular PLC controllers SIMATIC S7" webiste: http://www.automation.siemens.com/mcms/programmable-logic-controller/en/simatic-s7-controller/

[6] AIA, "GigE Vision - Video Streaming and Device Control Over Ethernet Standard - version 2.0", Ann Arbor, November 2011; http://www.visiononline.org/

[7] K. Kasemir, "Control System Studio Applications", ICALEPCS'07, Knoxville, October 2007; http://www.jacow.org/

[8] J. Serrano et al., "The White Rabbit Project", IBIC'13, Oxford, September 2013; http://www.jacow.org/

[9] S. Marsching, "Scalable Archiving with the Cassandra Archiver for CSS", ICALEPCS'13, San Francisco, October 2013; http://www.jacow.org/

[10] aquenos GmbH, "Cassandra Archiver for CSS" website: http://oss.aquenos.com/epics/cassandra-archiver/

[11] "Apache Cassandra" website: http://cassandra.apache.org/

[12] J. Ellis, "2012 in review: Performance", DataStax Developer Blog, January 2013; http://www.datastax.com/dev/blog/2012-in-review-performance

[13] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters", OSDI'04, December 2004; http://usenix.org/publications/library/proceedings/osdi04/tech/full_papers/dean/dean.pdf

[14] "Apache Spark - Lightning-Fast Cluster Computing" website: http://spark.apache.org/

[15] M. Killenberg et al., "Drivers and Software for MicroTCA.4", these proceedings, PCaPAC'14, Karlsruhe, October 2014; http://www.jacow.org/

ISBN 978-3-95450-146-5