A Bunch-Synchronized Data Acquisition System for the European XFEL Accelerator.

Tim Wilksen, Arthur Aghababyan, Lars Fröhlich, Olaf Hensler, Raimund Kammering, Kay Rehlich, Vladimir Rybnikov (DESY, Hamburg)

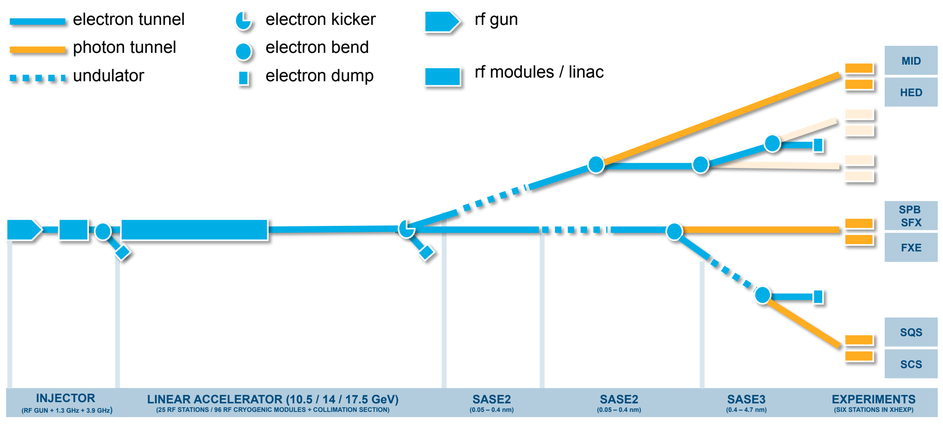
Abstract

The linear, super-conducting accelerator at the new European XFEL facility will be able to produce up to 2700 electron bunches for each shot at a repetition rate of 10 Hz. The bunch repetition rate might vary initially between 100 kHz and 4.5 MHz to accommodate the various needs of experiments at three different SASE beam lines.

A solution, which is able to provide bunch-resolved data of multiple data sources together in one place for each shot, has been implemented at the E-XFEL as an integral part of the accelerator control system. This will serve as a framework for high-level control applications, including online monitoring and slow feedback services.

A similar system has been successfully run at the FLASH facility at DESY for more than a decade now. This paper presents design, implementation and first experiences from commissioning the XFEL control system data acquisition.

The European XFEL Accelerator – Beam Line Layout

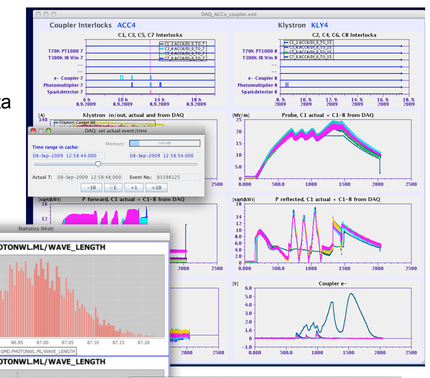


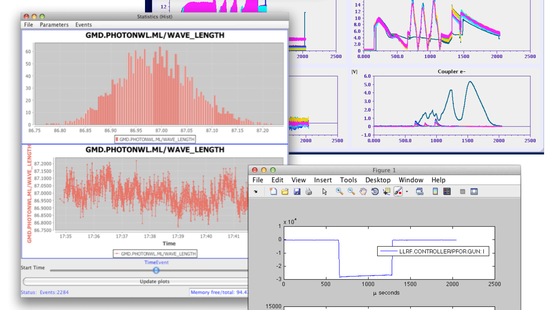
Specifications
Superconducting linear accelerator with beam energies 10.5 GeV / 14 GeV / **17.5 GeV**
Pulse repetition rate at **10 Hz** with **27000 bunches/s**
3 Photon beam lines with 6 experiment stations
Photon energy 0.3 – 24 keV
Photon pulse length 10 – 100 fs
Photon Pulse energy ~ mJ

Parameter	
Electron beam energy	10.5 GeV / 14 GeV / 17.5 GeV
Bunch charge	0.02 – 1 nC
Peak current	2 – 5 kA
Slice emittance	0.4 – 1.0 mm mrad
Slice energy spread	4 – 2 MeV
Shortest SASE wavelength	0.04 nm
Pulse repetition rate	10 Hz
Bunches per pulse	2700
RF-Pulse length (flat top)	600 µs

Applications and Tools

JDDD – Java DOOCS Data Display can display DAQ data via daq data server



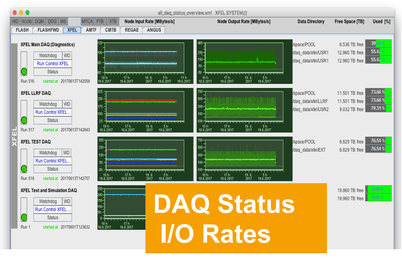


DAQ Data GUI – Java Application for displaying data via DAQ data server with built-in analysis functionality

MATLAB – MEX Functions for retrieving data via DAQ data server

Data Storage and Retrieval Tools

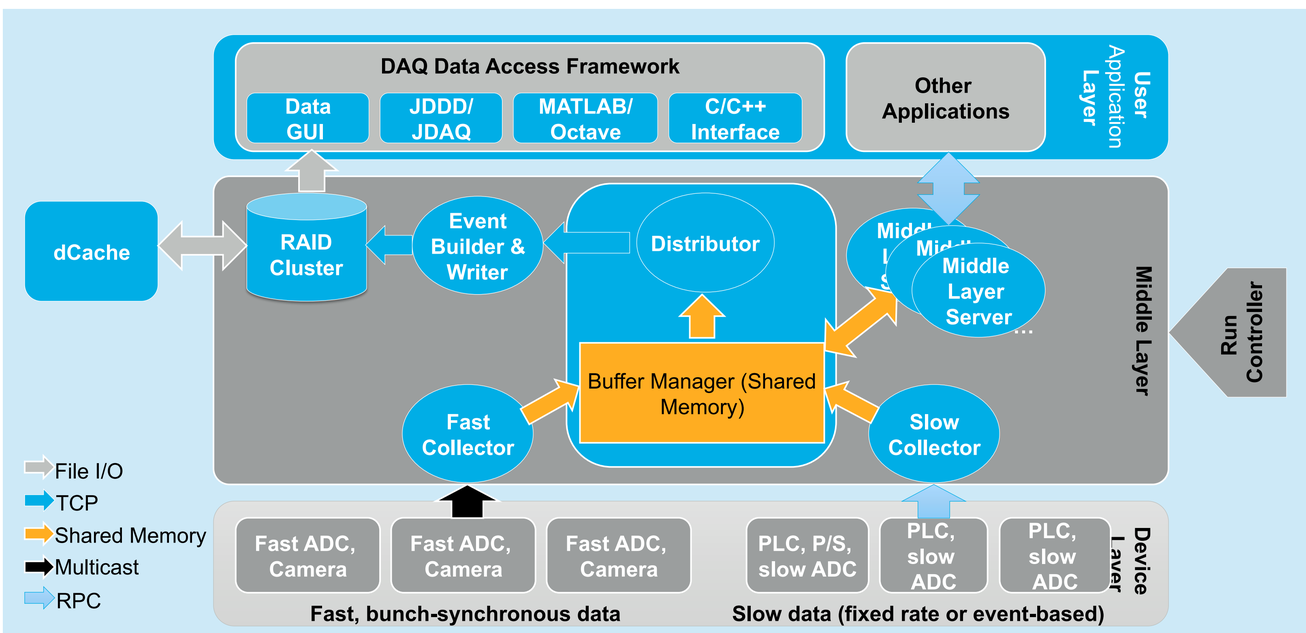
- C/C++ Interface - API
- Java Interface API
- MATLAB MEX Functions
- Python API in preparation
- DAQ data GUI and file manager for browsing stored and taped files
- DAQ data extractor to select specific channel data from files
- Backend used is dCache (IT)



Data Acquisition System Statistics

- Running since first day of RF-gun operations
- Four instances for diagnostics, RF/ LLRF
- About 1.5 GB/s sustained input rate
- More than 13 k complex DOOCS channels
- Compressed data rate to disk up to 30 TB/day

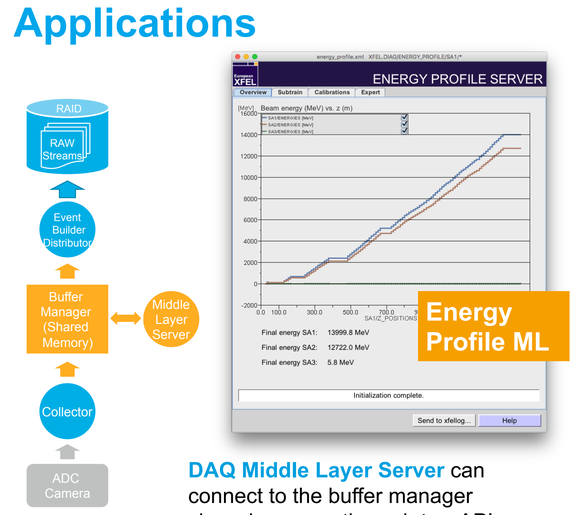
Accelerator Data Acquisition Layout



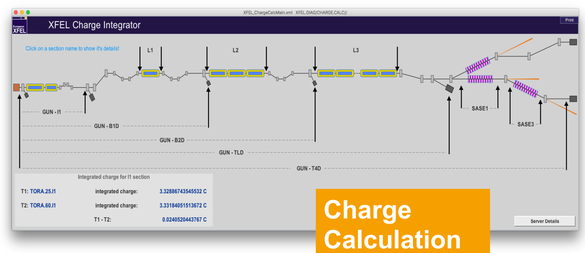
Shot-Synchronous, Bunch-Resolved Data Acquisition

- Collects from shot-synchronized, triggered data source (e.g. MicroTCA ADC modules) and from non-triggered sources via DOOCS calls slow data.
- Combines all data into one event record per shot in shared memory and offers it to middle layer server connected to the buffer manager for further computation.
- Completed event record is written via event builder and writer processes into dedicated streams and files to disk and optionally to tape for offline analysis.
- All recorded data is therefore available in a single data structure simplifying analysis tasks like correlations e.g.

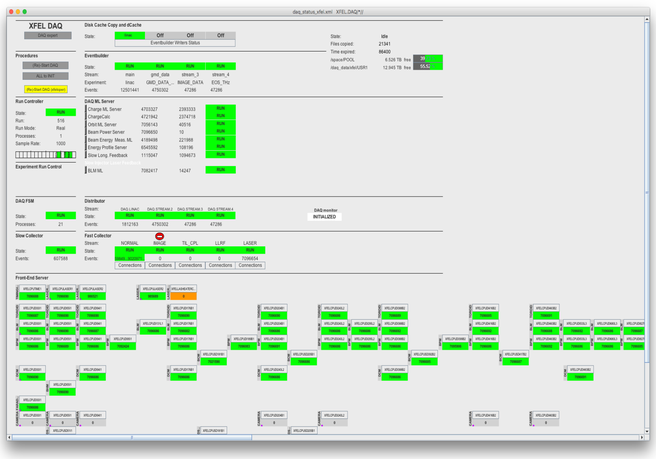
High-Level Physics Applications



DAQ Middle Layer Server can connect to the buffer manager shared memory through two API and are able to read and write to the current shot data record.



Status Overview



Data Acquisition System Overview

- Represents the layout of the DAQ
- Shows status of all sender (== data sources)
- Shows status of collector instances
- Displays all middle layer status and it run state
- Displays event writer status and disk states (dCache)
- Configuration and state information on DAQ

Front-End Server And Data Sources

Virtual XFEL Instance

The Virtual XFEL is a separate DAQ instance with its own timing system and real and simulated data sources. It allows for testing primarily middle layer server with API connection to the buffer manager but also front end server w/o hardware access.