Processing High-Bandwith Bunch-by-Bunch Observation Data from the RF and Transerve Damper Systems of the LHC

M. Ojeda[1], P. Baudrenghien[1], A.C. Butterworth[1], J. Galindo[1], W. Hofle[1], T.E. Levens[1], J.C. Molendijk[1], D. Valuch[1], F. Vaga[2]

[1]CERN, Geneva, Switzerland [2]University of Pavia, Pavia, Italy

Introduction

The **radiofrequency** and **transverse damper** feedback systems of the Large Hadron Collider (**LHC**) digitize beam phase and position measurements at the bunch repetition rate of 40 MHz. Embedded memory buffers allow a few milliseconds of full rate bunch-by-bunch data to be retrieved over the VME bus for diagnostic purposes, but experience during LHC Run I has shown that for beam studies much **longer data records** are desirable.

A new "observation box" (**ObsBox**) diagnostic system is being developed which parasitically captures data streamed directly out of the feedback hardware into a **Linux** server through an optical **fiber** link, and permits processing and buffering of **full rate** data for several **minutes**. The system is connected to an LHC-wide trigger network for detection of beam instabilities, which allows efficient capture of signals from the onset of beam instability events. The data is made available for analysis by client applications through interfaces which are exposed as standard equipment devices within CERN's controls framework.

It is also foreseen to perform **online** Fourier **analysis** of transverse position data inside the observation box using GPUs with the aim of extracting betatron tune signals.

Transmitter (VME module)

The *transmitters* are typically custom-built *VME* modules in the beam feedback systems with embedded *FPGA*s that sample different aspects of the accelerator's beam at a high frequency. In many cases, the acquisition of these data streams for diagnostic purposes can be done through the VME bus into the Linux host residing in the crate. However, some analyses of the beam data can really benefit from a substantially higher data bandwidth than the VME bus can provide.

Typically, the transmitters are *close to the beam*, in the underground caverns, which implies *costly* and/or complicated *access* requirements for development and maintenance. Therefore, moving part of the system to the surface server rooms is an advantage. Due to *distance* and *noise* requirements, a good candidate for transporting the signal is a *fiber link* from the underground to the surface, where the receiver, a Linux server, is running.

Receiver (Linux server)

The new transmitters designed for the ObsBox system stream the data out through the fiber links. At the server end, for each fiber, a *CERN-designed PCI Express v1 module* is used to act as the receiver: the Simple PCI Express Carrier (*SPEC*) module.

The ObsBox server runs a *customized Linux kernel* intended for hosting *real-time software*. A *tailored* Linux kernel device *driver* is used to control the SPEC modules over the PCIe bus and is able to stream the data to a user-space process through syscalls and a *sysfs* interface.

User-space process

The *user-space process* is responsible for reading the *driver* streamed data regularly and quickly enough and storing it into very large memory buffers, where the data waits to be requested eventually by clients. Clients connect to the ObsBox server over the CERN's technical network, a standard *Ethernet* network. They may request any recent data at any moment, with different possible filtering criteria.

In order to satisfy the *real-time* requirements and provide all the *features* required by users while keeping the software as *simple* as possible, the user-space process takes advantage of the Front-End Software Architecture (*FESA*) framework.

The user-space process is also foreseen to perform *online data analysis* if the network bandwidth is not enough to stream it out to clients. Since at this stage the data is local, using a *coprocessor* (like a GPU) to perform the analysis is a viable option, which in turn opens the possibility of running other kinds of analysis for which the CPU may not be performant enough.

References

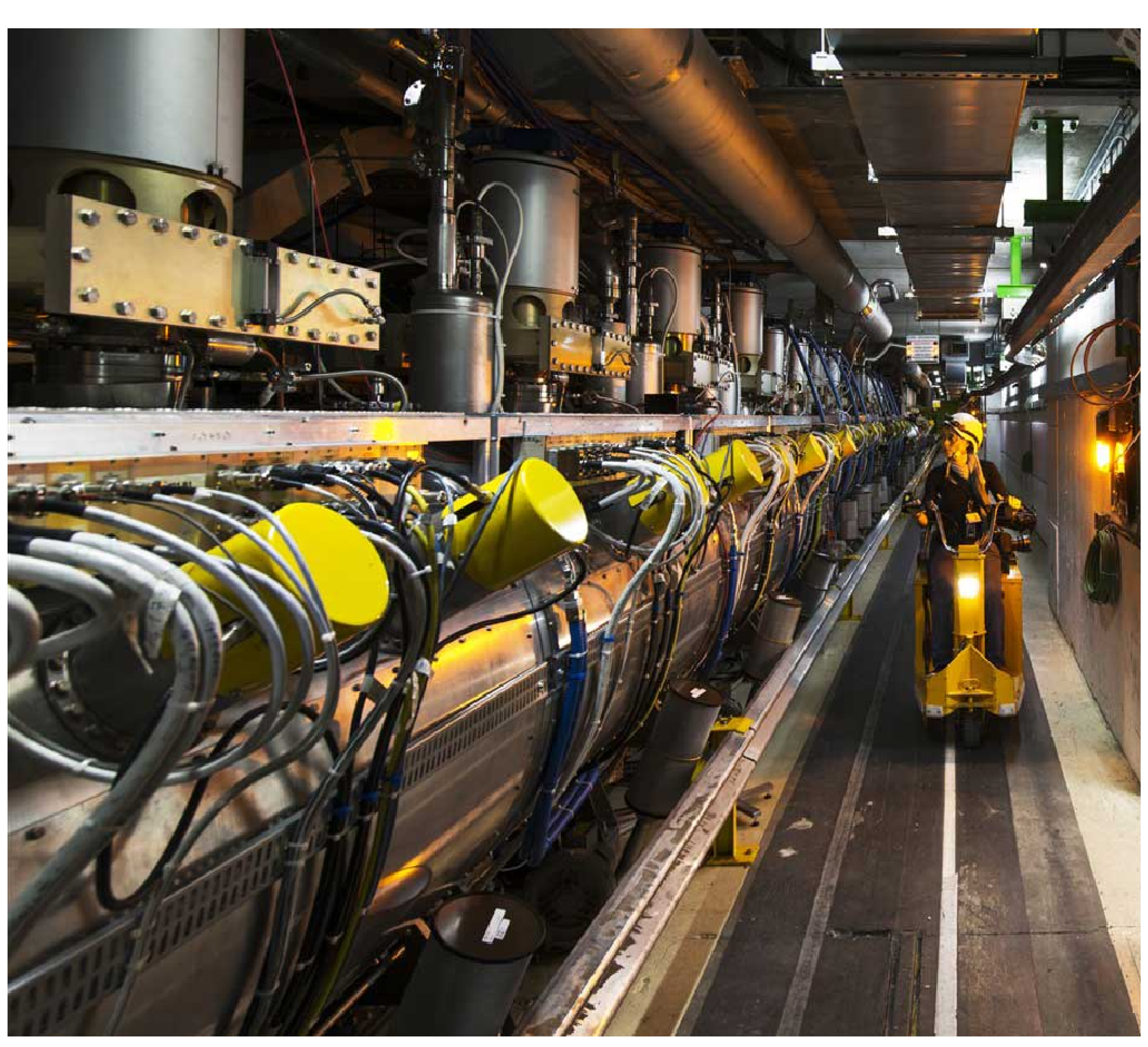
[1] P. Baudrenghien et al., "The LHC Low Level RF", EPAC'06, Edinburgh, Scotland, June 2006, TUPCH195 (2006), `http://www.JACoW.org`

[2] Simple PCIe FMC carrier (SPEC) website: `http://www.ohwr.org/projects/spec`

[3] M. Arruat et al., "Front-End Software Architecture", ICALEPCS'07, Tennessee, USA, October 2007, WOPA04 (2007), `http://www.JACoW.org`

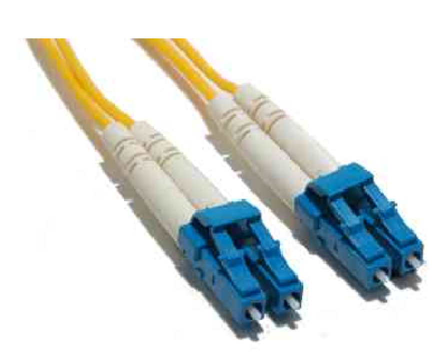
ICALEPCS 2015

RF cavities

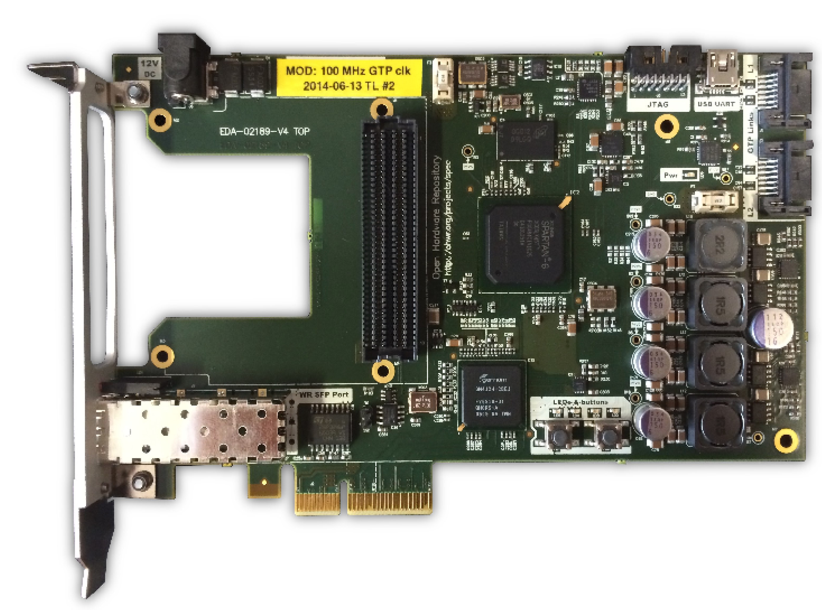


RF VME modules

Fiber link

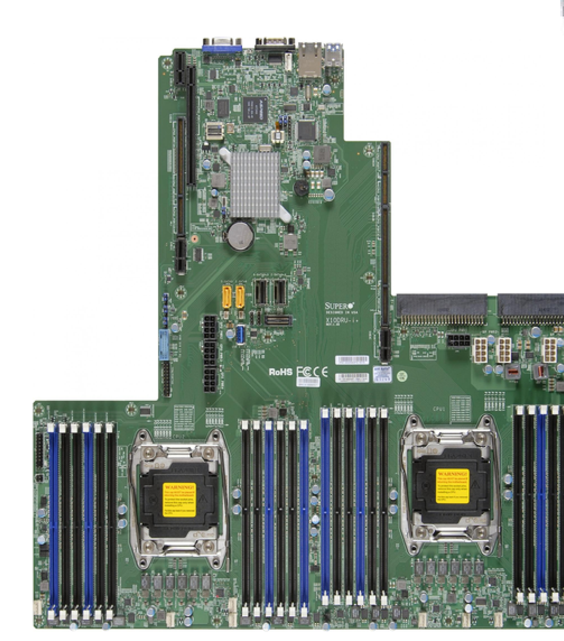


Simple PCIe FMC carrier (SPEC)



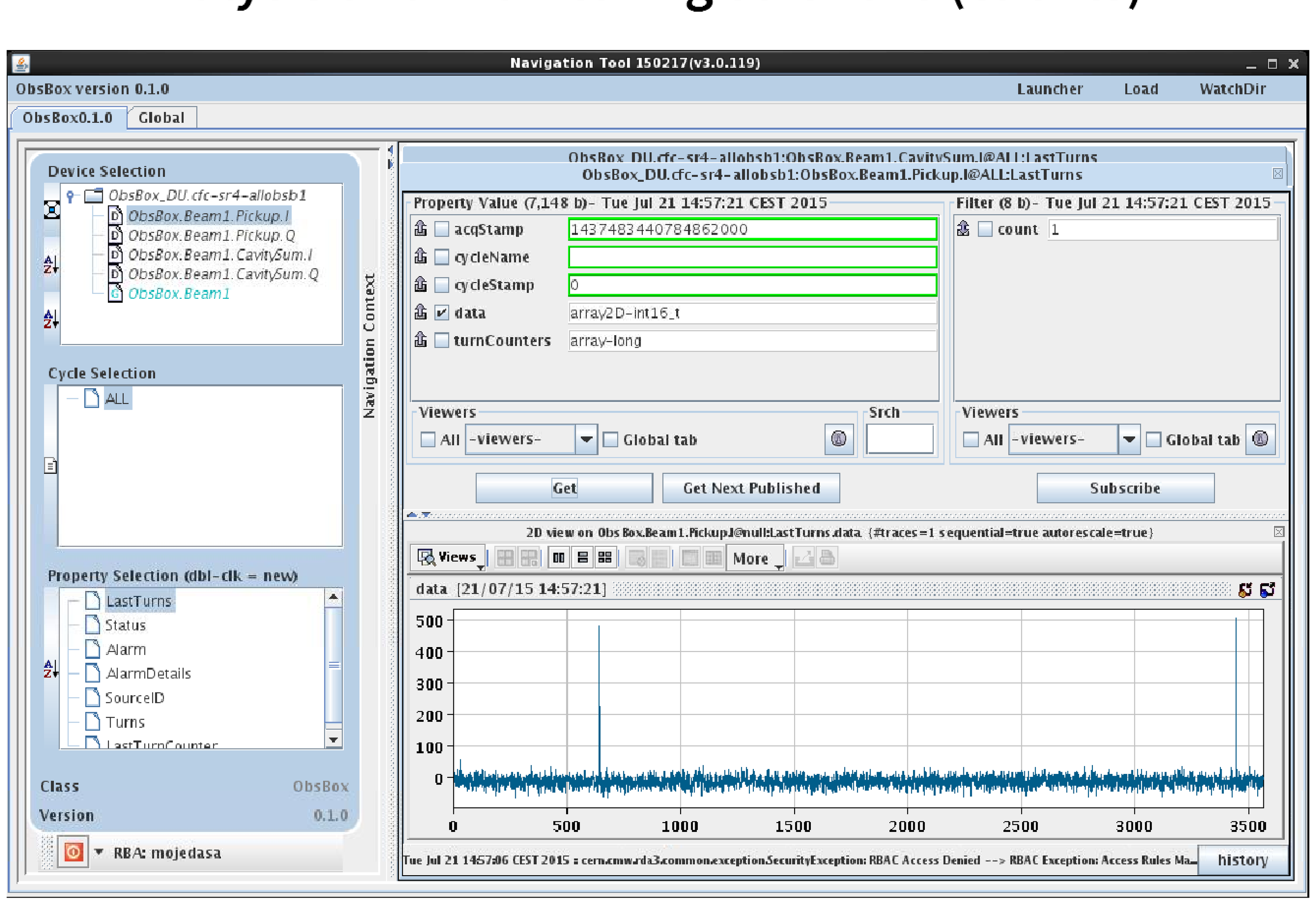
SuperMicro chassis

ObsBox

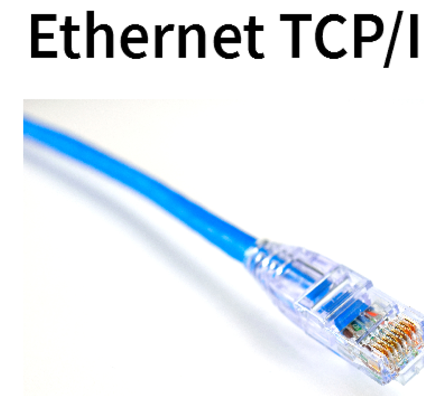


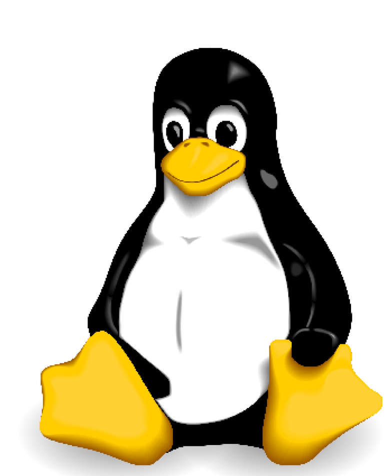
SuperMicro motherboard

Analysis and monitoring software (clients)



Ethernet TCP/IP





Custom Linux kernel driver

The Ultimate I/O framework (ZIO)

Front-End Software Architecture (FESA)