ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-THBPP02

DONKIORCHESTRA: A SOFTWARE TRIGGER-DRIVEN FRAMEWORK FOR DATA COLLECTION AND EXPERIMENT MANAGEMENT BASED ON ZeroMQ DISTRIBUTED MESSAGING

R. Borghes, F. Billè, V. Chenda, G. Kourousias, M. Prica
Elettra-Sincrotrone Trieste, Basovizza, Italy

Abstract

Synchrotron end-stations consist of a complex network of devices. The setup is not static and is often upgraded. The data acquisition systems are constantly challenged by such changes and upgrades, so scalability and flexibility are crucial skills. DonkiOrchestra is a ZeroMQ-based framework for data acquisition and experiment control based on an advanced software trigger-driven paradigm. In the DonkiOrchestra approach a software device, referred to as Director, provides the logical organization of the experiment as a sequential workflow relying on triggers. Each software trigger activates a set of Actor devices that can be hierarchically organized according to different priority levels. Data acquired by the Actors is tagged with the trigger number and stored in HDF5 archives. The intrinsic asynchronicity of ZeroMQ maximizes the opportunity of performing parallel operations and sensor readouts. This paper describes the software architecture behind DonkiOrchestra, which is fully configurable and scalable, so it can be reused on multiple endstations and facilities. Furthermore, experimental applications at Elettra beamlines and future developments are presented and discussed.

INTRODUCTION

Elettra Sincrotrone Trieste is a multidisciplinary international research center located on the outskirts of Trieste that hosts two advanced particle accelerators: Elettra and FERMI. It provides state-of-the-art techniques to lead experiments in physics, chemistry, biology, life sciences, environmental science, medicine and cultural heritage. Elettra is a third-generation synchrotron light source that delivers more than 5000 hours/year of synchrotron light from IR to soft x-rays to 28 beam lines. A substantial facility upgrade, named Elettra 2.0, is currently planned [1]. FERMI is a seeded free electron laser facility that provides fully coherent ultrashort 10-100 femtosecond pulses with a peak brightness ten billion times higher than that made available by third-generation light sources. Currently 6 versatile experimental stations carry out outstanding research with photon radiation coming from FERMI.

The Software For Experiments Team manages a set of core services for the beamlines of the two facilities, spanning from the instruments integration to the data acquisition and experiment management. The range of applications is wide and heterogeneous, it is crucial for the efficiency of the team to adopt common and reusable frameworks. For this reason in the recent years we designed and developed a set of software tools highly configurable and scalable that can be adapted and used on several beamlines.

The next sections provide the reader with a technical overview of DonkiOrchestra, a modular framework focused on data acquisition and experiment management. DonkiOrchestra represents a novel approach, a software trigger-driven architecture that solves efficiently the intrinsic problem of non-synchronicity of any heterogeneous and distributed experimental system. Thanks to the modular and configurable structure of DonkiOrchestra it was possible to use it on different Elettra beamlines. Two sections of the paper are dedicated to the presentation of the experimental applications of DonkiOrchestra at TwinMic and XRF beamlines. Lastly future developments are presented and discussed.

CONCEPTUAL DESIGN

Synchrotron and Free Electron Laser beamlines consist of a complex distributed network of devices like sensors, detectors, motors, but also computational resources. DonkiOrchestra draws on years of experience and has been designed in order to solve a set of critical challenges:

- *Scalability and Customizability*: each scientific setup is unique and designed for a specific technique but needs frequent adaptations for the integration of new instrumentats or for carrying out specific user experiments.
- *Parallelism and Synchronization*: sensors, motors and cameras in a distributed control system act independently but often do not share a common time base. Most of the experiments require the simultaneous acquisition of different data sources and a high level of synchronicity is essential for the data analysis phase.
- *Communication Efficiency*: data communication overheads directly affect the duration of the experiment and the quality of the data synchronization. The choice of an efficient, reliable and modern communication system is critical for the performance of the entire framework.

One of the most important aspects in a beamline experiment is how the sensors and actuators are best synchronized. Often it is fundamental to know not only what happened, but also when it happened. The use of a wired trigger signal is the quickest, easiest and most accurate way to synchronize events in different places. Nevertheless, in the heterogeneous environment of a photon source facility, this solution is not always adoptable. It could be due to the presence of non-triggerable instrumentation or the need of having an

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-THBPP02

extremely dynamic configuration that does not always fit into a static hardware solution.

The design of DonkiOrchestra is based on the idea of not using a hardware signal as synchronization trigger but a software trigger event shared through the instrumentation network. Any experiment is organized as a train of independent phases uniquely identified by an event index. Control and acquisition actions start upon the arrival of the software trigger and the acquired data can be labelled with the trigger index. Such approach forces the intrinsic parallelism of a distributed control system and introduces an evident correlation degree between the acquired data through the use of the trigger index. More than this, DonkiOrchestra introduces the concept of *structured trigger*: each event contains a structure with an incremental and a priority index. The first value uniquely identifies each phase of the experiment, while the other is used to organize the individual actions in priority levels within each phase.

The core of DonkiOrchestra is a Director module that coordinates several distributed Player modules that are coded for executing a specific action. Each Player can be dynamically assigned to a Priority level. The Director conducts the experimental sequence by sending a train of software triggers to the Players. For each phase of the experimental sequence, a trigger structure with Priority 0 is sent to all the Players, followed by a trigger structure with Priority 1 and so on. Each Player executes its task upon the arrival of the trigger structure with the right priority value and sends back to the Director an acknowledge event. A Player associated to a sensor acquires the data, tags it with the trigger index and sends it back to the Director for storing it in suitably structured archives. The described architecture is schematically shown in Figure 1.

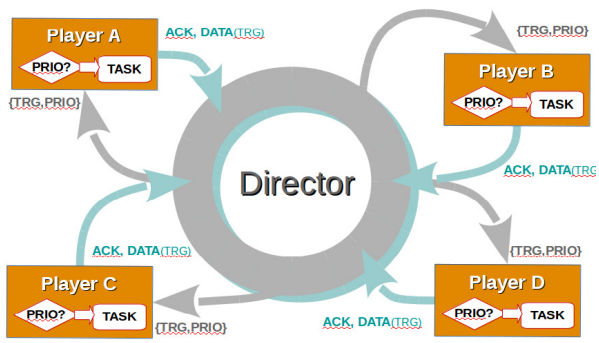

Figure 1: DonkiOrchestra schematic design.

As an example, a standard raster scan experiment could be implemented using one or more motion Players and several acquisition Players. Motion players should be configured at priority 0 (highest), while acquisition players should be configured at priority 1. Every phase of the experiment will consist in a movement of the motors followed by the synchronous acquisition of the configured sensors. Positions and sensor's data will be coherently labelled with the same trigger index in order to be easily handled by the data processing software.

TECHNICAL CHOICES

Since the new framework should be robust, efficient, highly dynamic and portable on different operating systems we decided to adopt Python as core programming language.

A fundamental brick of the framework is the messaging system used to distribute trigger events and share the collected data. The ZeroMQ [2] messaging system has been chosen for its asynchronous I/O model that perfectly fits the need of having a scalable distributed application. It provides the benefits of reliability and persistence, has a score of language APIs and runs on most operating systems. We decided to adopt the classic TCP *Publish/Subscribe* pattern, where messages are published without knowledge of which subscribers there may be.

One of the key aspects of the planned architecture is the overhead related to the communication protocol, so we performed some preliminary tests for evaluating the ZeroMQ performance. As an example, we evaluated the time spent by the Director device for sending a trigger event and receiving back the acknowledge message, parameter that directly influences the time resolution of the entire system. It was measured an average delay of 0.46 ms with a standard deviation of 0.15 ms. Figure 2 shows the results of the test performed using Python2.7 on two x64 Linux machines connected over the public 1Gb company network.

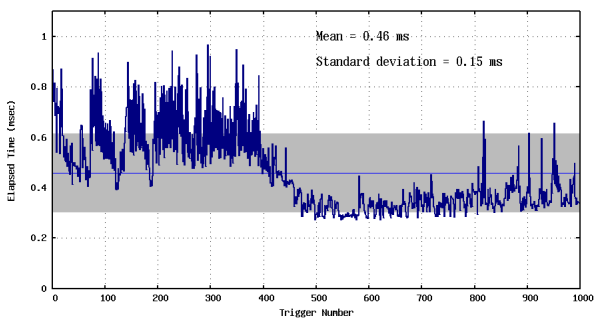

Figure 2: Measurements of the elapsed time from the dispatchment of the ZeroMQ trigger to the arrival of the acknowledge.

In the planned architecture experimental data will be shared through ZeroMQ messages, it is crucial to measure and evaluate the protocol performance at the increasing of the data block size. ZeroMQ uses a *"fire and forget"* architecture: sent messages are internally stored in an in-memory queue without affecting too much the publishing application. One of the benchmark tests focused on the measurament of the overhead time due to a ZeroMQ *publish()* operation after acquiring the data. Figure 3 shows the results of the test compared to an equivalent Python in-memory copy operation (*copy.deepcopy()*). The blue curve has been obtained by measuring the performance of a single publisher application with the presence of three subscribers, the green one refers to zero subscribers and the

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-THBPP02

red one refers to the in-memory copy operation. The graph clearly indicates that the overhead introduced by a data publishing operation substantially corresponds to the delay due to a Python in-memory copy operation and basically does not depend on the number of subscribers.

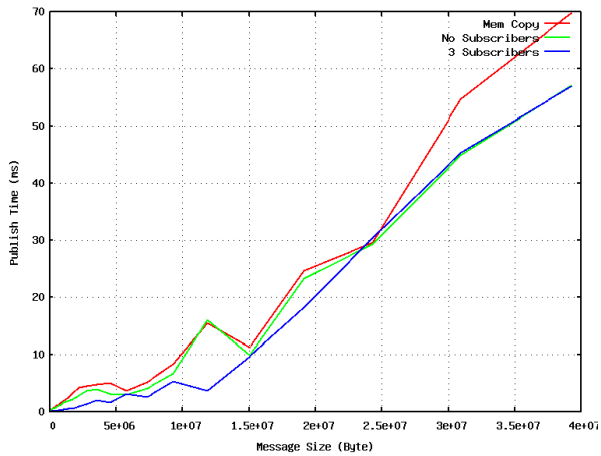

Figure 3: Measurements of the overhead time due to a ZeroMQ publish() operation vs data message size.

As scientific storage data format we adopted HDF5 [3], a versatile and widely used model capable of organizing complex and large scientific data collections. It has been chosen for its portability, its software library that runs on a range of different platforms and a set of integrated performance features that allow for access time and storage space optimizations.

TECHNICAL DESIGN

The structure of DonkiOrchestra is composed by a core python device called Director that schedules the parallel execution of multiple actions performed by Player devices. The Director and Player python classes can be easily integrated in any software environment because there are really few external dependencies: the h5py and numpy modules for the data storage phase, the zmq module for the communication library. At Elettra, since the control system of many beamlines is based on the TANGO [4] framework, DonkiOrchestra has been integrated as a set of TANGO device servers.

The Player class architecture is extremely generic and customizable: its core implements the communication actions while the interaction with the instrumentation is delegated to an external python script that can be adapted to each specific situation.

On the other side the central Director device provides a set of fundamental services:

- a central *Information Service*;
- a *Scheduler Machine* that prepares and manages the train of trigger events for the experimental sequence;
- a *Data Storage Service* for collecting data and metadata from the Players and storing it in HDF5 archives.

The Information Service is of fundamental importance for setting up the ZeroMQ connections between the Director and the Players. Since all the devices instantiate a publishing socket on a random TCP port, it becomes crucial to use an Information service for sharing the connection parameters. At start-up each Player connects to the Information Service, retrieves the Director ZeroMQ coordinates and notifies its own ZeroMQ coordinates. From this moment onwards the communication between the Director and the Player proceeds using exclusively ZeroMQ messages. The Information Service has been implemented as an ASCII TCP server using a back-end database for persistency. Figure 4 shows a representation of the general architecture of DonkiOrchestra.

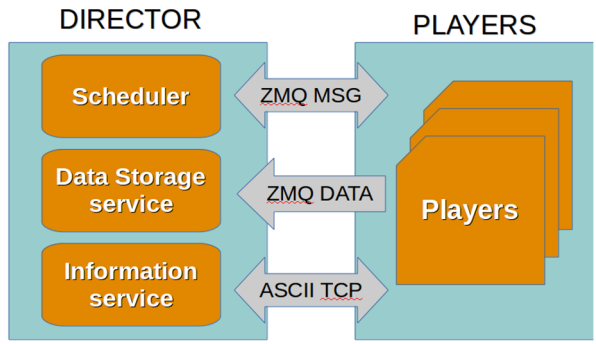

Figure 4: DonkiOrchestra general architecture.

The DonkiOrchestra ZeroMQ communication protocol is based on a set of five message categories:

- *INFO_MSG*: used by the Director to retrieve information from a Player (name, data sources, metadata sources);
- *CMD_MSG*: used by the Director to send commands to a Player (start, stop, change priority,...);
- *TRG_MSG*: used by the Director to send trigger events to the Players;
- *DATA_MSG*: used by the Players to share data and metadata with the Director or any other subscriber;
- *LOG_MSG*: used by the Players to share log messages with the Director.

Any experimental sequence is organized and managed by the Director in three consequent phases:

- *PLANNING Phase*: the Director sets the Player's priority levels, enables or disables data and metadata sources and sends a *Start* command to all the Players for pre-operation actions. During this preliminary procedure each Player prepares the instrumentation or the sequence of operations to be performed upon the arrival of the event triggers, e.g. a motion player may prepare the sequence of target positions for a scan.
- *EXPERIMENT Phase*: the sequence of trigger messages structured as *{index,priority}* is sent to the Players. On the base of its priority, each of them executes a specific task. Any acquisition Player publishes a data message, tagged with the trigger index, that is collected by the Director and stored in a HDF5 archive.

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-THBPP02

- *CLOSEOUT Phase*: the Director sends a Stop command to all the Players for closeout operations.

As shown in Figure 5, during the EXPERIMENT phase Player devices react to the arrival of a trigger event with the publication of a simple acknowledge message or even with a message containing data. Such messages are collected by the Director that stores the incoming data in HDF5 archives having a standard structure. Data relative to different sensors are stored in different dataset arrays: each row of such dataset array corresponds to a trigger number, so data belonging to different datasets may be easily correlated by the post-processing analysis software. In case of long sequences, acquired data can be subdivided in multiple HDF5 archives. DonkiOrchestra allows to define also metadata sources that are periodically acquired during the experimental sequence.

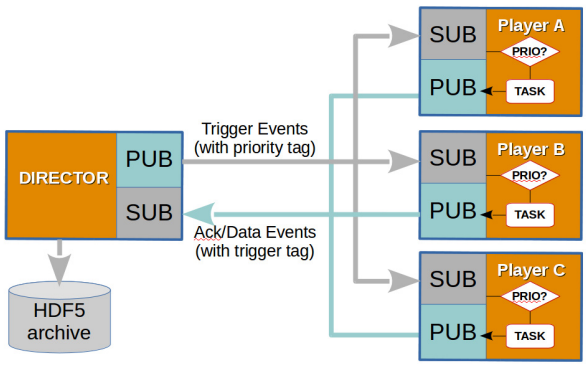

Figure 5: ZeroMQ communication flow during the experimental sequence.

TWINMIC@ELETTRA

The TwinMic beamline at the Elettra integrates the advantages of complementary scanning and full-field imaging modes into a single instrument. Using simultaneously the x-ray transmission and emission detection systems it is possible to perform elemental, absorption and phase contrast imaging that provides complementary chemical and morphological information for the sample under investigation. The extreme versatility of the Twinmic beamline allows to plan a wide range of different scientific experiments. It is a perfect application field for DonkiOrchestra, where it has been installed and adapted in order to acquire and control a set of scientific instruments:

- a Physics Instruments piezoelectric sample stage;
- a Princeton X-ray CCD 1300x1340;
- an Andor IXON DV860 camera;
- a XGLab low-energy X-ray fluorescence detector.

DonkiOrchestra has been used for managing various 2D scanning experiments. As an example Figure 6 shows the phase reconstruction of a soft X-ray Ptychography experiment relative to a sub-micrometric imaging of human lung cells with asbestos fibers. Such experiment required special data acquisition strategies that DonkiOrchestra managed to implement efficiently and relatively fast. Moreover DonkiOrchestra has been integrated into a scripting framework that allowed to schedule multiple scans, an extremely efficient solution that led to a considerable time saving.

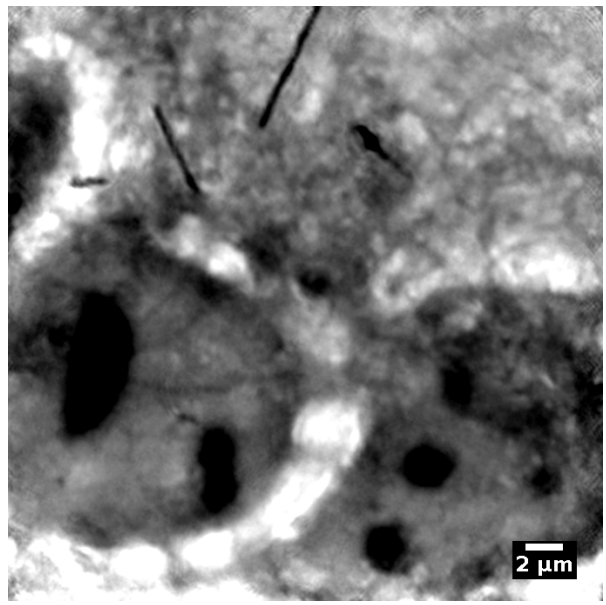

Figure 6: Phase reconstruction of a soft X-ray Ptychography experiment in TwinMic@ELETTRA controlled by DonkiOrchestra.

XRF@ELETTRA

XRF@ELETTRA is a highly versatile beamline that presently hosts an ultra-high vacuum chamber operated in partnership with the IAEA Agency [5].

X-Ray Fluorescence (XRF) is a well-established and versatile analytical technique for studying the elemental composition of different kind of materials. The analytical capabilities of XRF are considerably improved in terms of elemental sensitivity and spatial resolution by using Synchrotron radiation for excitation. Advanced sample manipulator stages make possible to perform near surface, angular dependent and 2D scanning or transmission measurements. In addition, the high resolving power of the crystal monochomators installed at the XRF beamline allows to perform X-ray Absorption Near-Edge Structure (XANES) measurements and to gather valuable information on the speciation of the detected elements.

DonkiOrchestra has been recently installed and adapted in order to manage XANES measurements by means of controlling a set of scientific instruments:

- a double-crystal Si(111) monochromator providing an energy range for the incident beam from about 3.7 to 14 keV;
- a XRF silicon drift detector Bruker Nano GmbH XFlash 5030;
- a X-ray beam intensity detector sampled by a Keithley 6485 Picoammeter;
- a four-channel solid-state beam monitoring system interfaced via an Elettra AH501B picoammeter.

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-THBPP02

The XANES experiment has been organized as a 1D step scan in which the fluorescent yield and various diagnostic sensors are simultaneously measured as a function of the monochromator wavelength. The DonkiOrchestra setup allows to divide the energy range in multiple regions of interest characterized by a different scanning step size. DonkiOrchestra Players have been organized in priority levels as follows:

- *priority0*: Energy Control Player;
- *priority1*: Acquisition Players associated to Fluorescence Detector, Photon Beam Intensity and Position monitors;
- *priority2*: Calculation Player for aggregate data processing.

Associated to this DonkiOrchestra application has been developed a specific graphical interface that exploits the event-based infrastructure for the visualization of the results. The interface, shown in Figure 7, receives ZeroMQ data messages from the Players: the monochromator energy and the sensor's values are correlated through the trigger index for updating the scan plot.

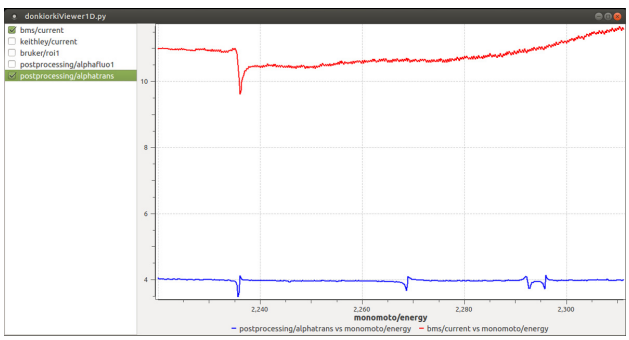

Figure 7: DonkiOrchestra graphical interface for XANES experiments at XRF@ELETTRA beamline.

CONCLUSIONS AND FUTURE DEVELOPMENTS

The present paper describes the software architecture behind DonkiOrchestra, a framework which supports both data acquisition and experiment control. The DonkiOrchestra approach is fully configurable and scalable, it can be reused on different endstations and facilities. The strength of DonkiOrchestra resides in its design choices: a powerful messaging system like ZeroMQ that maximizes the opportunity of performing parallel operations and sensor readouts, a dynamic and portable language like Python and a fully configurable structure that permits a high degree of customization.

The use of DonkiOrchestra on different endstations demonstrated the potential of the tool and the simplicity of customizing it. In the near future we're planning to install it on a larger number of beamlines, the planned Elettra Synchrotron upgrade represents a great opportunity. Future developments will focus on strengthening the DonkiOrchestra logging system, the error management and the graphical interface applications.

ACKNOWLEDGEMENT

The authors, members of the *Software For Experiments* team and the *Scientific Computing* team of Elettra, thank the IT Group, the XRF and Twinmic beamline staff of Elettra Sincrotrone Trieste. Moreover they acknowledge the importance of advanced technologies and open source software like TANGO, NumPy, HDF5 and ZeroMQ.

REFERENCES

[1] E. Karantzoulis, A. Carniel, R. DeMonte, S. Krecic and C. Pasotti, "Status of Elettra and Future Upgrades", in *Proc. 9th Int. Particle Accelerator Conf. (IPAC'18)*, Vancouver, Canada, April 2018, pp. 4054-4056, doi:10.18429/JACoW-IPAC2018-THPMF010

[2] ZeroMQ, http://zeromq.org

[3] HDF5, http://www.hdfgroup.org/HDF5

[4] Tango Controls, https://www.tango-controls.org

[5] IAEA, https://www.iaea.org