A Simulation System For The European Spallation Source (ESS) Distributed Data Streaming

Carlos Reis, Roberto Borghes, George Kourousias, Roberto Pugliese

Elettra Sincrotrone Trieste
Scientific Computing

brightness

European Spallation Source (ESS), the next-generation **neutron** source facility, is going to produce an immense amount of **data**. Various working groups mostly associated with the EU project: *Building a Research Infrastructure and Synergies for Highest Scientific Impact in ESS (BrightnESS)* aim at developing solutions for its data-intensive challenges. The real-time data management and **aggregation** is among the top priorities. The Apache **Kafka** framework will be the base for ESS real-time distributed data **streaming**. One of the major challenges is the **simulation** of data streams from experimental data generation to data analysis and storage. We present a simulation approach based on the **DonkiOrchestra** data acquisition and experiment control framework, re-purposed as a data streaming simulation system compatible with ESS-KAKFA infrastructure.

BrightnESS is a European Union-funded project within the European Commission's Horizon 2020 Research and Innovation programme.
The BrightnESS programme is designed to ensure that key challenges are met in order to build an ESS that can deliver high-impact scientific and technological knowledge.

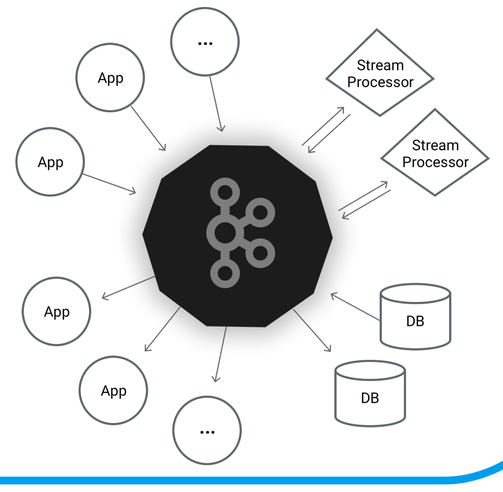
Expected Data Flow for a Neutron Experiment

The following list provides a typical data flow for a neutron scattering experiment.

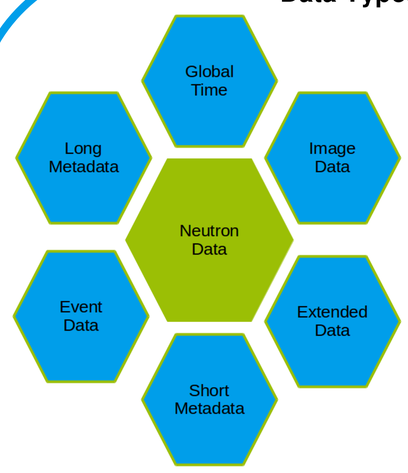
- **Experiment Control:** The team of users configure the components of the instrument and sample environment using an experiment control system that interfaces with the neutron instrument components.
- **Stream:** Data are taken in event mode whereby the individual detector counts are tagged with useful experimental metadata to create a dataset. The list of event and metadata are aggregated in software and broadcast over a network in a continuous stream of data that external softwares systems can utilise.
- **Reduce:** The raw data are transformed and corrected from the base unit of the instrument to a data type that is scientifically useful and valid.
- **Visualise:** The representation to the beamline users of a scientifically meaningful display of the corrected data.
- **Analyse:** A scientific model is generated in order to scientifically interpret the experimental data.

APACHE kafka™
A distributed streaming platform

Kafka is used for building real-time data pipelines and streaming apps. It is horizontally scalable, fault-tolerant, wicked fast, and runs in production in thousands of companies.



Data Types at ESS



- Proton pulse repetition rate = 14Hz
- 1e6 events per pulse
- Expected data rate = 160Mbit/s

Simulated Data Streams Characteristics

Data Stream	Frequency	Data Type
A	1 Hz	scalar
B	50 Hz	2D image
C	1 Hz – 1 kHz	1D spectrum
D	1 kHz	3D array
E	10 Hz	mixed

DonkiOrchestra

A specialized framework that offers advanced **WORKFLOW** *control for beamline end-station software, re-purposed to be used with* **KAFKA-ESS** *technologies aiming at providing rapid* **SCRIPTING** *of complex scenarios of* **DATA STREAM** *generation and processing for* **SIMULATION** *purposes.*

DonkiOrchestra at a Glance

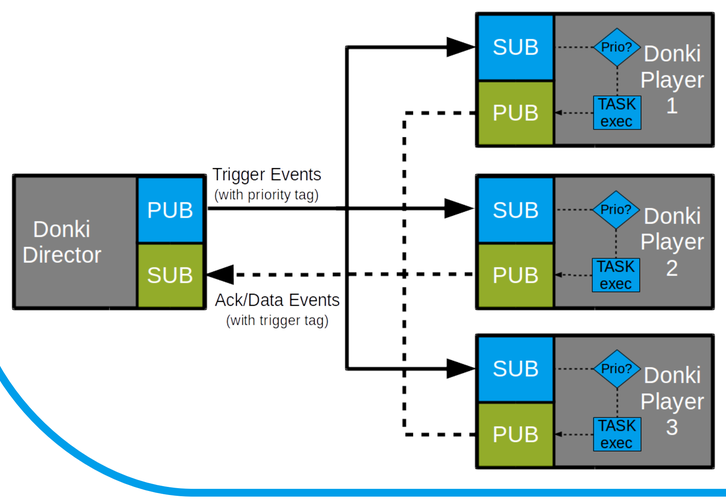
✓ It works as a **scheduler** tool that is able to manage tasks in **parallel**, deal with concurrency and complex workflow

✓ DonkiOrchestra has two principal elements: **Director** and **Player**

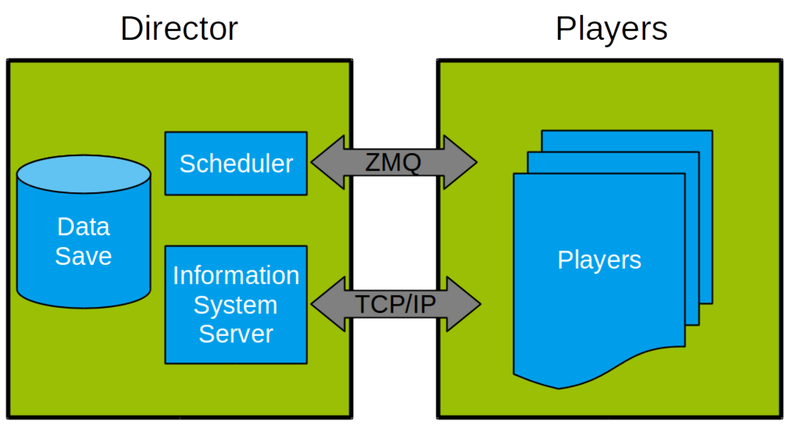
✓ The Director communicates with the Players by sending a sequence of *triggers* and receiving acknowledgments signals

✓ In order to perform more complex tasks we can set the Players with different **priority Levels**

DonkiOrchestra Schematic Architecture



DonkiOrchestra Communication Schema



DonkiOrchestra – Structure & Evolution

✓ The **Director** and **Player** of DonkiOrchestra are independent software components (i.e., servers, scripts or any Python object) distributed on different computers connected through an Ethernet network

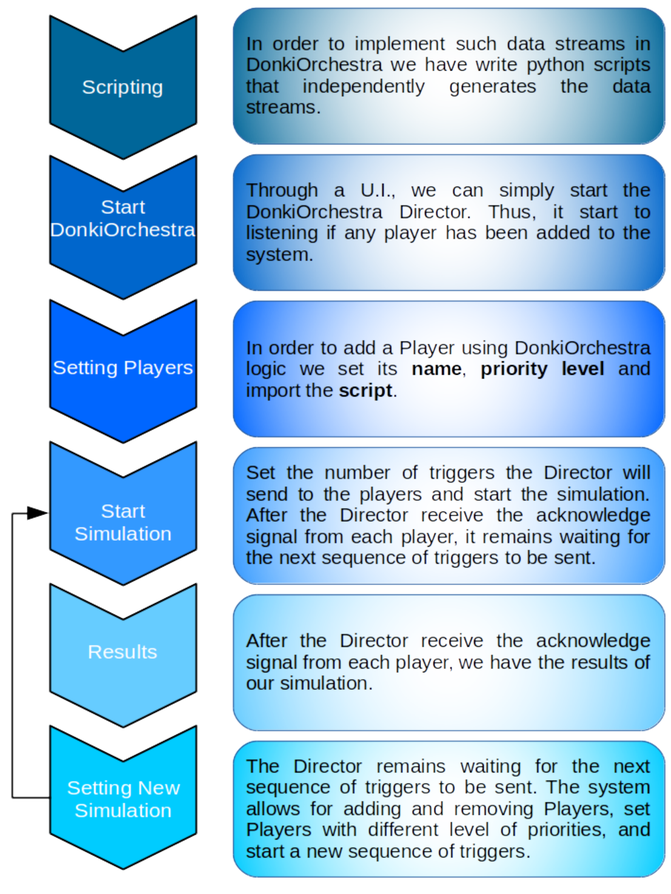
✓ The framework, **re-purposed**, became TANGO independent

✓ ZeroMQ is the messaging system for the **Scheduler**: it fits the need of having a scalable distributed application and also maximizes the opportunity of performing parallel tasks

✓ TCP/IP protocols establish the communication between the Director and Players as an **Information System Server**

✓ A **Data Save** system is being develop for future implementation

Simulation Implementation Flow



✓ **What have we done...**
- On the previous months we have been working on separating/splitting DonkiOrchestra from its current architecture in order to integrate it with ESS-Kafka
- Open sourcing

✓ **What are we going to do...**
- Develop data save system
- Integration with current WP5 data-aggregation architecture

[1] BrightnESS, https://brightness.esss.se/about
[2] Work Package 5: Real-Time Management of ESS Data, https://brightness.esss.se/about/work-packages/work-package-5-real-time-management-ess-data
[3] Tango Controls, tango-controls.readthedocs.io/en/latest/contents.html
[4] R. Borghes and G. Kourousias, "DonkiOrchestra: a scalable system for data collection and experiment management based on ZeroMQ distributed messaging", in *Proc. 11th New Oportunities for Better User Group Software Conf. (NOBUGS'11)*, Copenhagen, Denmark, October 2016, paper 10.17199/NOBUGS2016.36, pp. 41-46.
[5] R. Borghes, G. Kourousias, V. Chenda, A. Gianoncelli "Evolving a Labview End-Station Software to a Tango-Based Solution at the Twinmic Elettra Beamline", presented at 16th Int. *Conf. On Accelerator and Large Experimental Control Systems (ICALEPCS'16)*, Barcelona, Spain, October 2017, paper TUPHA208, this conference.
[6] A. Mukai *et al.*, "Development, testing and deployment of the ESS data aggregation and streaming software", poster presented in *11th New Oportunities for Better User Group Software Conf. (NOBUGS'11)*, Copenhagen, Denmark, October 2016.
[7] ZeroMQ, http://zeromq.org
[8] C. Aggarwal, Data Streams: Models and Algorithms. IBM T. J. Watson Research Center, NY, USA: Springer US, 2007.
[9] ESS, https://europeanspallationsource.se/data-management
[10] ESS, https://europeanspallationsource.se/sites/default/files/dmsc_workflow_ar2015_pdf_0.pdf

Acknowledgement

The authors, members of the *Software for Experiments* team and the *Scientific Computing* team, thank the IT Group and the BrightnESS WP5 contributors. Among them special thanks for constructive discussions and feedback are due to Afonso Mukai and Tobias Richter.

carlos.reis@elettra.eu

https://brightness.esss.se/