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

Carlos Reis, Roberto Borghes, George Kourousias, Roberto Pugliese

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 infrastructure, re-purposed as a data streaming simulation system compatible with ESS-KAKFA framework.

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 configures 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 information 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 utilize.
• **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.
• **Visualize:** The representation to the beamline users of a scientifically meaningful display of the corrected data.

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

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

**DonkiOrchestra**

**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 receives acknowledgments signals
- In order to perform more complex tasks we can set the Players with different priority Levels

**DonkiOrchestra**

**Schematic Architecture**

- 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

**DonkiOrchestra Communication Schema**

**Data Types at ESS**

**Simulated Data Streams Characteristics**

**Simulation Implementation Flow**

- **Scripting**
- **Start DonkiOrchestra**
- **Sending Players**
- **Start Simulation**
- **Results**
- **Stop Simulation**

**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 ESS/WPS data-aggregation architecture

**Acknowledgement**

The authors, members of the Software for Experiment team and the Scientific Computing team, thank the IT Group and the BrightNESS WPS contributors. Among them special thanks for constructive discussions and feedback are due to Atosono Mukai and Tobias Richter.

carlos.reis@elettra.eu

https://brightness.esss.se/