Evolution and use cases of FairMQ

A. Rybalchenko[1], *M. Al-Turany*[1], *and N. Winckler*[1]
[1]GSI, Darmstadt, Germany

The FairRoot framework [1] is a framework for simulation, reconstruction and data analysis of particle experiments. In the past, FairRoot has been extended with FairMQ, that enables reconstruction and analysis of free streaming data [2] [3]. FairMQ allows to use and extend components called devices to execute user tasks in a multiprocess, multithreaded and/or network environment, connecting them together efficiently.

General Developments

To make FairMQ devices more flexible, a concept of a command channel has been introduced, that allows the devices to change their internal state and configuration in response to external events. The concept has been successfully tested during the PANDA test beam [**?**].

To improve static configuration of FairMQ processes, a flexible system for command line options has been added. Utilizing boost::program_options library, the new options parameters provide support for default, optional and repeated parameters and also enable reading of parameters from external files.

For the dynamic deployment, FairMQ has added a number of features to allow more flexible configuration, such as automatic port finding from a given port range or more granular state transitions to allow access to configured properties. Making use of these features is the Dynamic Deployment System that is currently in development at GSI, which allows to dynamically distribute a user defined set of tasks over a number of nodes using any resource management system [**?**].

In addition to the original binary data transport format, a serialization using either Google Protocol Buffers, Root TMessage or Boost Serialization have been introduced, together with implementation examples. Furthermore, a generic device implementation based on Policy-based Design has been introduced to decouple device functionality from transport format [4].

FairMQ Devices for ALICE experiment

After the Long Shutdown 2, the upgraded ALICE detector will produce over 1 TB of data per second, which is to be distributed from about 250 First Level Processors (FLPs) to O(1000) Event Processing Nodes (EPNs), where each EPN collects sub-timeframes from every FLPs to build a full timeframe as is shown in Fig. 1. In the Alice O^2 software prototype two FairMQ devices have been implemented for transport and distribution of the data from FLPs to EPNs. Both FLP and EPN devices make use of ”zero-copy” and multipart messages of ZeroMQ to ensure maximum performance and efficient packaging. The system has been shown to scale to a large number of nodes. Additionally, an algorithm has been implemented on the FLP devices to optimize the usage of the network and to minimize contention when several FLPs are sending to the same EPN. The results of the algorithm can be seen in a histogram in Fig. 2, which shows time intervals between receiving of data from the the same FLP on a given EPN. The application of the algorithm results in a balanced network usage and predictable traffic pattern.

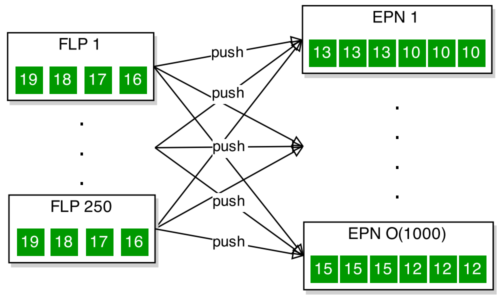

Figure 1: Alice O2 example topology

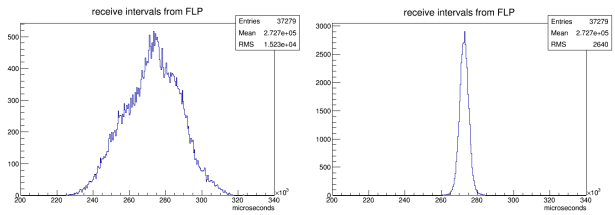

Figure 2: left: without traffic shaping, right: with traffic shaping

References

[1] FairRoot: http://fairroot.gsi.de.

[2] A. Rybalchenko, M. Al-Turany: GSI Scientific Report: Streaming data processing with FairMQ (2014)

[3] M. Al-Turany, D. Klein, A. Manafov, A. Rybalchenko, F. Uhlig: Extending the FairRoot framework to allow for simulation and reconstruction of free streaming data. accepted for publication by, Journal of Physics: Conference Series (2013).

[4] N. Winckler et al.: Generic and modular design for FairMQ devices (2015)

DOI:10.15120/GR-2015-1-INFRASTRUCTURE-IT-01