

Exploring polyglot software frameworks in ALICE with FairMQ and fer

Monday, July 9, 2018 2:30 PM (15 minutes)

In order to meet the challenges of the Run-3 data rates and volumes, the ALICE collaboration is merging the online and offline infrastructures into a common framework: ALICE-O2.

O2 is based on FairRoot and FairMQ, a message-based, multi-threaded and multi-process control framework. In FairMQ, processes (possibly on different machines) exchange data via message queues either through 0MQ or nanomsg.

FairMQ allows to build sophisticated topologies of processes: router/dealer, request/reply, publish/subscribe, client/server, etc..

In turn, this enables developers to write their reconstruction or analysis process in whatever language they choose or deem appropriate for the task at hand, as long as that programming language can send and receive data through these message queues.

This paper presents “fer”, a Go-based library compatible and interoperable with FairMQ.

We will start with a brief introduction of the builtin features that make Go a solid choice when dealing with I/O and concurrency.

We will then describe the principal components of fer and how they interact with C++ FairMQ.

Finally, we will report on the performances (CPU, VMem) of fer and conclude with the main figures of merit of fer, in the context of deployment in a distributed computing setup.

Primary authors: Dr BINET, Sebastien (IN2P3/LPC); APHECETCHE, Laurent (Centre National de la Recherche Scientifique (FR))

Presenter: Dr BINET, Sebastien (IN2P3/LPC)

Session Classification: T5 - Software development

Track Classification: Track 5 – Software development