

# Applications of Message Queues with GPUs

## *Using the FairMQ framework*

2014-12-04 | Ludovico Bianchi

# Challenges in PANDA

- PANDA's computational goals require a broad variety of hardware cooperating in the data acquisition/processing chain
  - Inclusion of heterogeneous systems (co-processors, GPUs, FPGAs, ...) more and more relevant in offline/analysis tasks
- ⇒ Interface layer for flexible, high-performance, device-agnostic communication

# Message Queues

- Asynchronous communication between 2+ nodes
- Interaction of nodes through shared queue(s)
- Transport of control and data
- Many types and implementations for different contexts (application/framework, language, OS, ...)

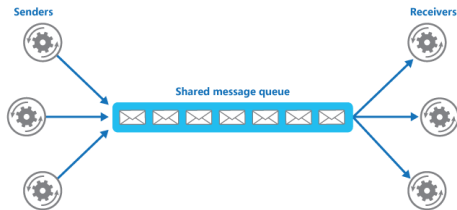
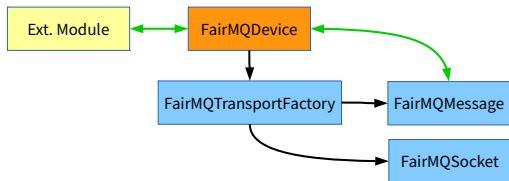


Image credit: microsoft.com

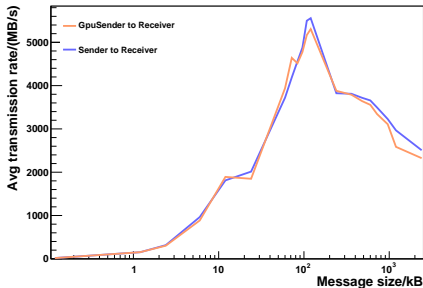
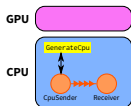
- Message-based system for data transmission in FairRoot
- Abstraction front-end from messaging libraries (ØMQ, nanomsg, ...)
- Versatile and scalable interface: easy to change/extend architectures and communication channels
- Active development, increasingly tested and adopted (ALICE)
- Code on [GitHub](#)

- Main user class is FairMQDevice
- FairMQMessages are created and received
- Transport managed by FairMQTransportFactory

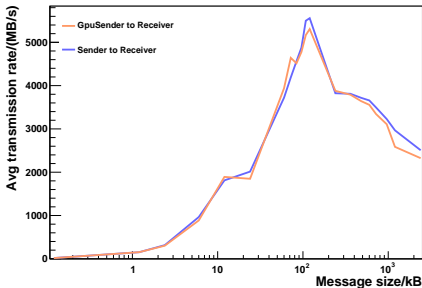
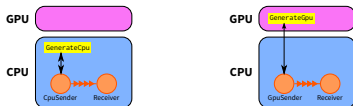


- Messaging patterns available
  - push/pull (parallel task/data distribution)
  - pub/sub (data is broadcasted from one publisher to  $n$  subscribers)
  - req/rep (request confirmation of transmission)

- Test FairMQ transmission rate (no overhead from computation)

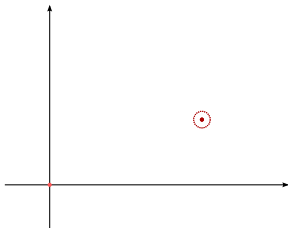


- Test FairMQ transmission rate (no overhead from computation)



# Circle Hough algorithm

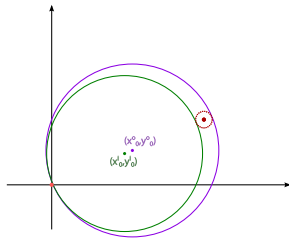
- Hough transform: parametrize image features (tracks), sample parameter space, collect votes
  - For Circle Hough: circles passing through IP and tangent to isochrone of radius  $R_h$  at hit point  $(x_h, y_h)$
- 👉 Andreas Herten's talk in the Pattern Recognition session





# Circle Hough algorithm

- Hough transform: parametrize image features (tracks), sample parameter space, collect votes
  - For Circle Hough: circles passing through IP and tangent to isochrone of radius  $R_h$  at hit point  $(x_h, y_h)$
- 👉 Andreas Herten's talk in the Pattern Recognition session

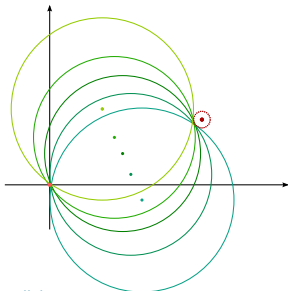


# Circle Hough algorithm

- Hough transform: parametrize image features (tracks), sample parameter space, collect votes
- For Circle Hough: circles passing through IP and tangent to isochrone of radius  $R_h$  at hit point  $(x_h, y_h)$

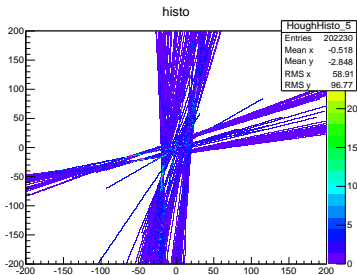


Andreas Herten's talk in the Pattern Recognition session



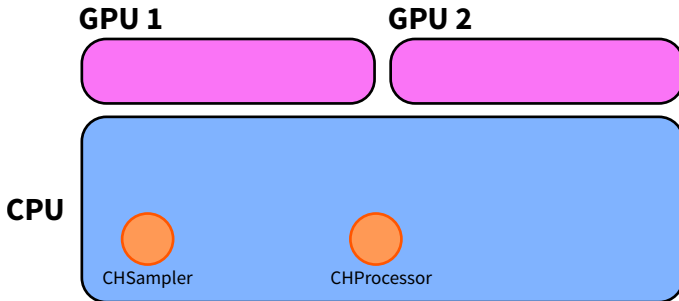
# Circle Hough algorithm —Implementation

- Angle sampling: CircleHoughCpu (CPU, C++), CircleHoughGpu (GPU, CUDA C/C++)
- CreateHisto: collects votes in histograms
- ReadFile: reads input hit data from ASCII file
- Each module compiled separately as a shared library



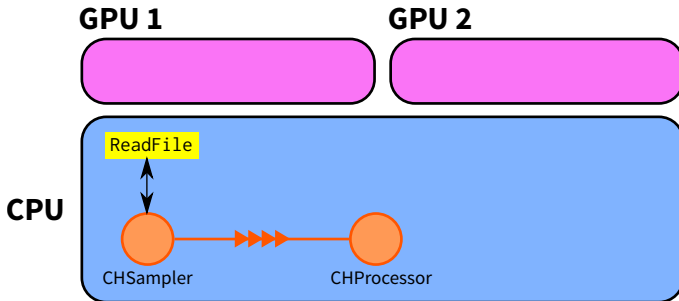
# CH—FairMQ prototype

Single node



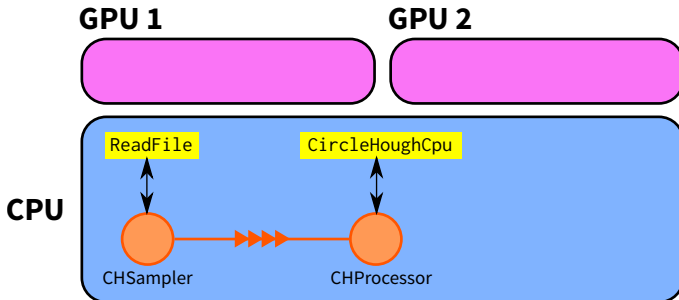
# CH—FairMQ prototype

Single node



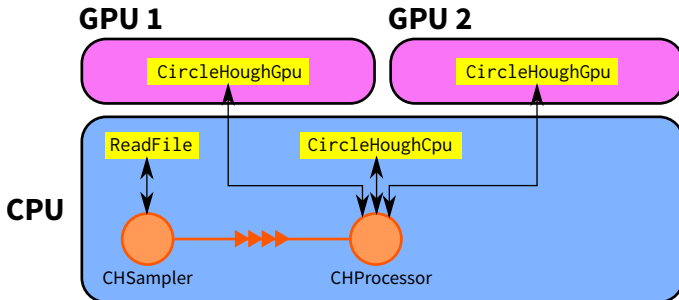
# CH—FairMQ prototype

Single node



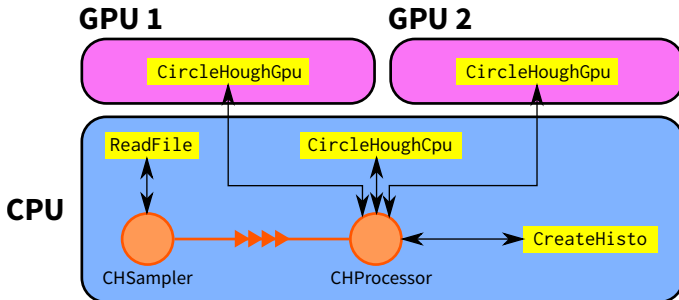
# CH—FairMQ prototype

Single node



# CH—FairMQ prototype

Single node

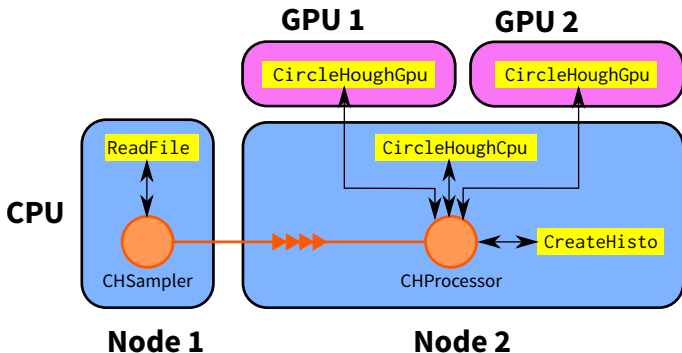




# CH—FairMQ prototype

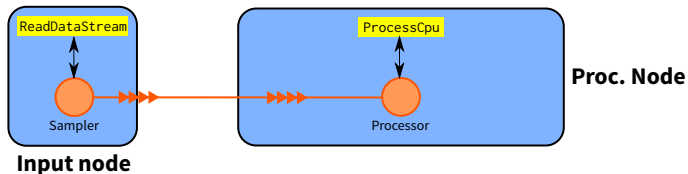
Two nodes, over network

- Modification for changing configurations are minimal



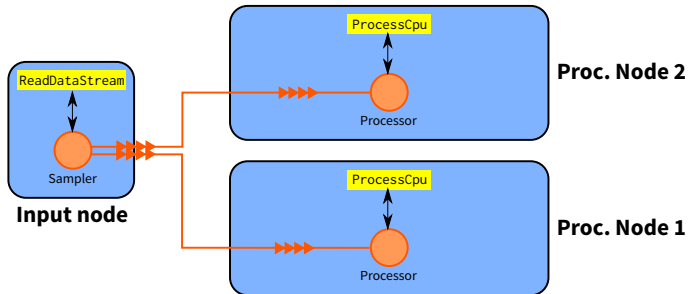
# CH—FairMQ

*Example of extension*



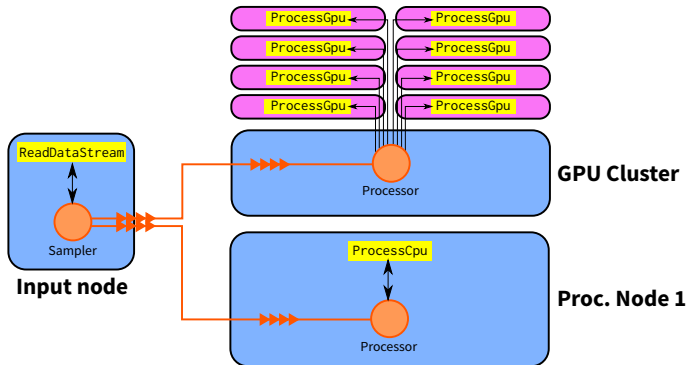
# CH—FairMQ

*Example of extension*

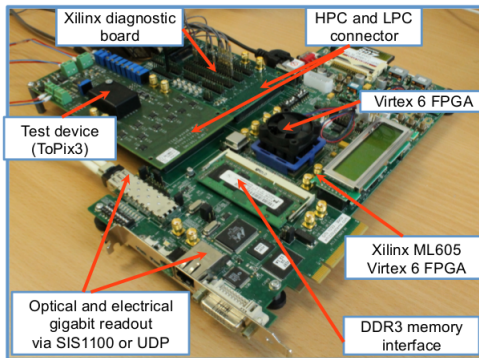


# CH—FairMQ

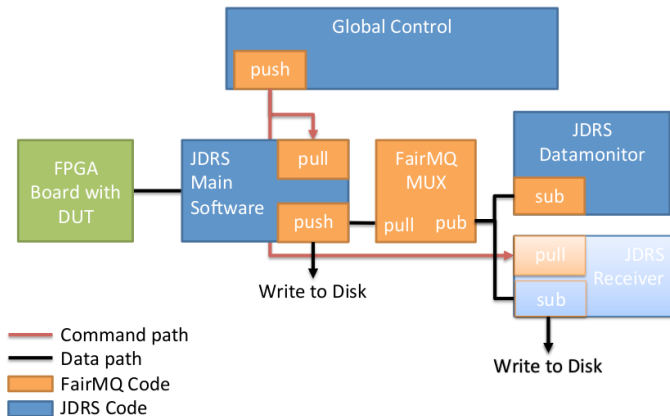
Example of extension



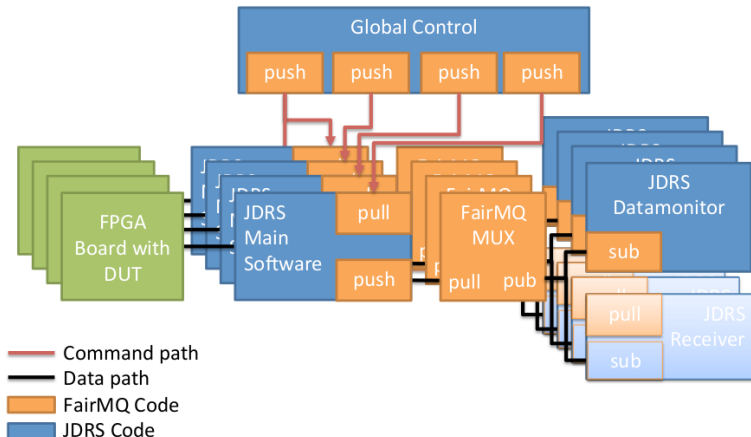
- Readout system for front end ASIC development
- One system to characterize different ASICs
- Requirements:
  - Fast for high data rates,
  - flexible for different kinds of ASICs
  - easy adoptable



# JDRS test beam setup



# JDRS test beam setup



- Stand-alone prototype of a CPU-GPU integrated implementation of a tracking algorithm using FairMQ as communication layer
  - CPU and GPU code interchangeable
  - Flexible, easily extended configuration
  - Multi-GPU support
  - **Future developments:** integration with PandaRoot, exploring additional FairMQ functionality, configuration and performance optimizations
- DAQ control and real-time monitoring for ToPix test beam
  - Results are positive: reliable, easily scalable and adaptable
  - Active contact with developers



# The End

Thanks for your attention!



Image credit: deviantart.com

[l.bianchi@fz-juelich.de](mailto:l.bianchi@fz-juelich.de) | [s.esch@fz-juelich.de](mailto:s.esch@fz-juelich.de)