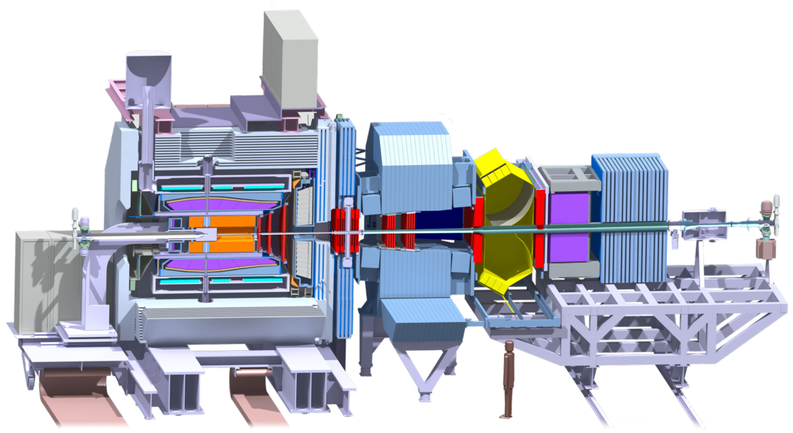
Combination Of Message Queues And GPUs For The Event Building of the PANDA Experiment

Ludovico Bianchi, Forschungszentrum Jülich, Germany

Online Tracking with GPUs at PANDA

PANDA Experiment

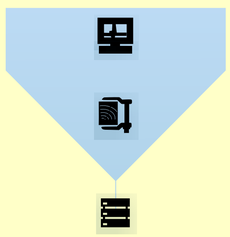


Anti**P**roton **An**nihilation at **Da**rmstadt
- Fixed proton target
- 1.5–15 GeV/*c* antiproton beam

- No hardware trigger
- Online event reconstruction and event selection

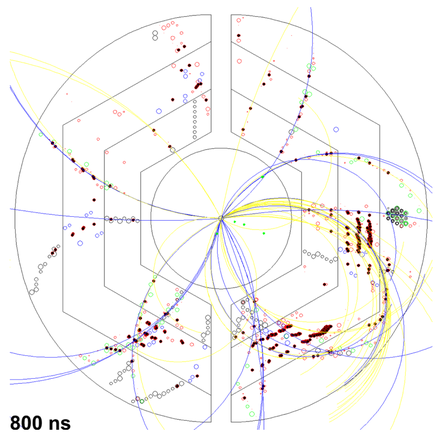
➔ **Online tracking**

Incoming data rate: **200 GB/s**



Offline storage: **3 PB/year**

➔ Data reduction factor **1/1000**



- Triplet Finder algorithm on single GPU: 6×10^6 hits/s
- PANDA requirements: 10^9 hits/s

➔ Online tracking with multiGPU system in a heterogeneous event building stage

FairMQ

- Data transport layer in FairRoot framework, based on message queues
- Abstraction front-end for transport libraries (ZeroMQ, nanomsg ...)
- Message-based system, inherently supporting parallel computing models

```
FairMQMessage* msg = fTransportFactory->CreateMessage(sizeof(Content));
memcpy(msg->GetData(), payload, sizeof(Content));
fPayloadOutputs->at(0)->Send(msg);
```

- Support for many programming languages and hardware architectures
- Versatile: capable of inter-thread, inter-process or inter-node communication
- Scalable to large, complex systems

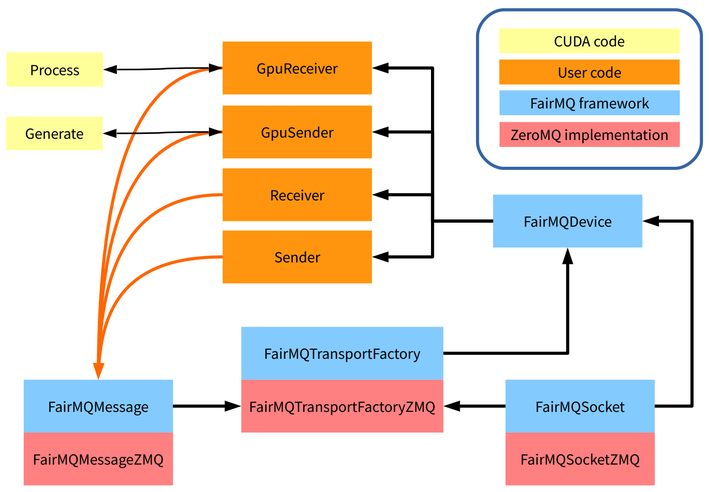
Test system with CUDA and FairMQ

Cross-compilation of CUDA C and FairRoot C++ using CMake

```
__global__ void kernel();
extern cudaWrapper() { kernel();}
```

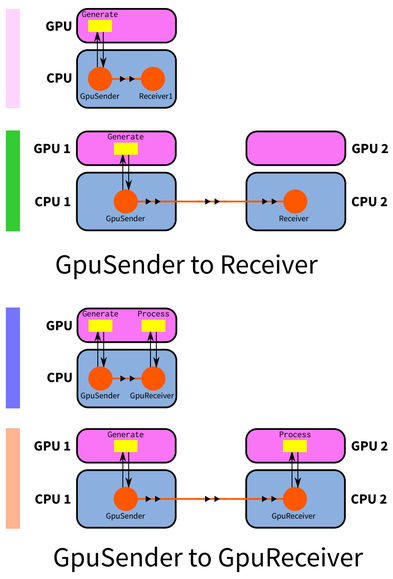
```
extern "C" cudaWrapper();
void cpuCaller() {cudaWrapper();};
```

```
CUDA_ADD_LIBRARY(fmqcuda cudaCode.cu)
set(DEPENDENCIES fmqcuda FairMQ)
```



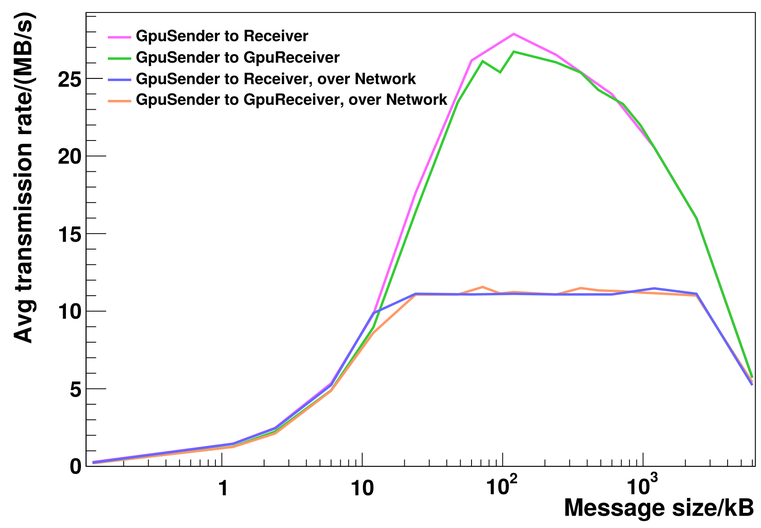
- **Sender/Receiver**: create/read message over transport factory
- **GpuSender**: creates message, calls Generate on GPU to create test data payload
- **GpuReceiver**: reads message, calls Process on GPU to perform operations on test data payload
- **Example GPU** code for the test system
  Generate: Random number generator (cuRAND)
  Process: Parallel data manipulation

Test system response



GpuSender to Receiver

GpuSender to GpuReceiver



- Study response of data generation and transmission chain: compare transmission rate in the test system as a function of the number of events for different configurations
- Limited transmission rate depends on test system, and available network bandwidth:
  FairMQ has no maximum throughput limit

Conclusions & Outlook

- First basic test system integrating CUDA GPU code and FairMQ

Future developments:

- More realistic simulation of PANDA DAQ chain
  - Implement simulated events, tracking algorithms
  - Integration with FPGA pre-processing stage
  - MultiGPU setup
- Performance study and optimization
- Deeper GPU integration
  - Device access to FairMQ classes
  - GPU-enabled transport libraries (nanomsg API)

JÜLICH FORSCHUNGSZENTRUM

panda