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 Proton Annihilation at Darmstadt
- 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

Triplet Finder algorithm on single GPU: 6×10⁶ hits/s
PANDA requirements: 10⁹ hits/s
→ Online tracking with multiGPU system in a heterogeneous event building stage

Offline storage: 3 PB/year
→ Data reduction factor 1/1000

Test system with CUDA and FairMQ

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

- 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

- 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)