SYCL-based online data processing framework concept for PANDA

CHEP 2024 | 23.10.2024

Bartosz Soból

Outline

- PANDA experiment
- PandaR2 framework concept
- Development and roadmap

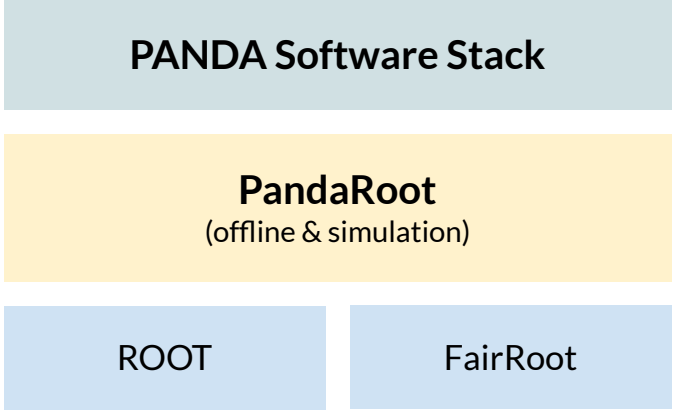
PANDA experiment

- One of major experiments at FAIR
- DAQ system with software trigger
- Up to 300 GB/s raw data in final setup
  - From multiple subsystems
  - To be processed online

    with diverse algorithms

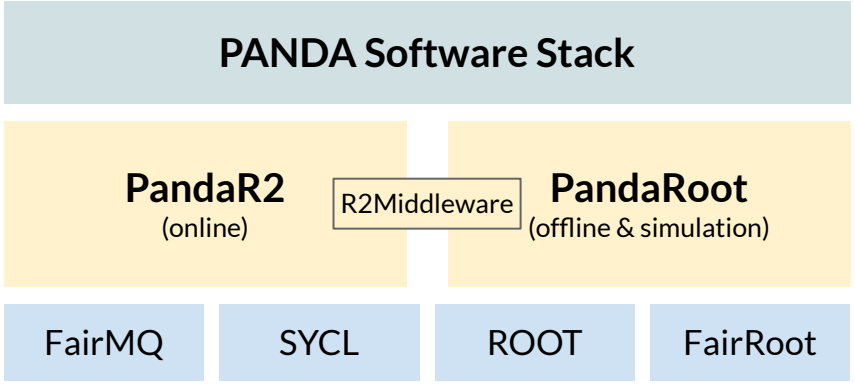
Existing PANDA software

- PandaRoot framework for offline simulation and analysis
  - Not applicable for online execution
- Based on FairRoot
  - Also a foundation of simulation stacks in CBM, ALICE and more



PandaR2 - the concept

- Online data processing framework concept
  - New codebase (modern CMake, C++20)
  - No ROOT dependency in *compute* part
- Designed around SYCL
  - For parallel CPU and GPU compute
- FairMQ used for messaging
- *Middleware* between PandaRoot and PandaR2
  - R2 tasks (algorithms) available from PandaRoot
- Connection to the ROOT world
  - Dictionaries for data classes (Digi, Hit, Track, etc.)
  - ROOT file I/O

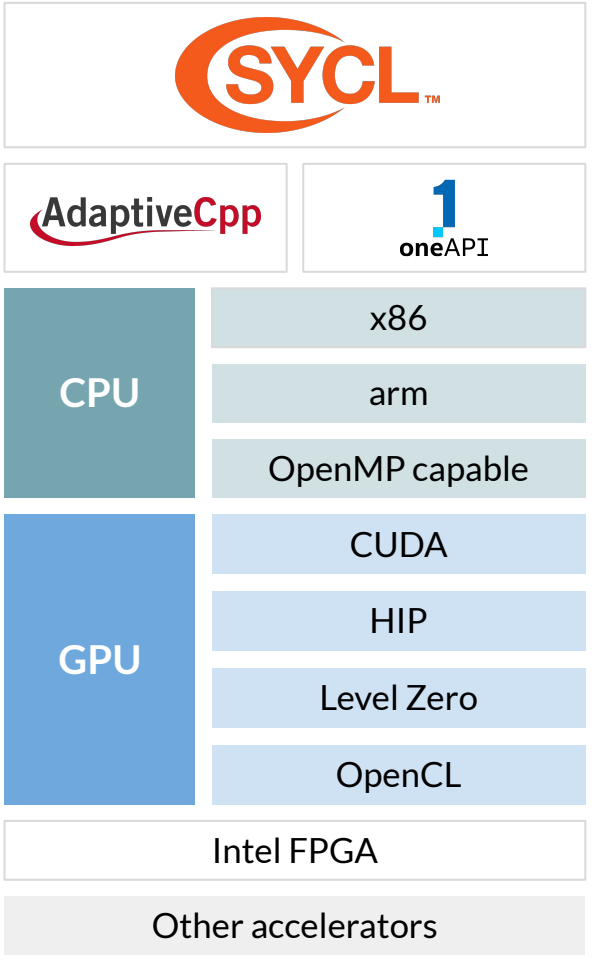


What is SYCL?

- Open standard, higher-level heterogeneous programming model – CPU, GPU, FPGA, ...
- Based on standard C++17 – without language extensions
  - Tools for C++ work with SYCL (IDEs, static analysis, linters, formatters, …)
- Single source for host and kernel/device code
  - Kernel == any callable (function, lambda, function object)*
  - C++ functions called by kernel are also compiled as a part of device code
  - Implicit device-host separation
- Implicit memory management and task scheduling
- Living ecosystem
  - Evolving standard, growing community, open-source and stable compilers

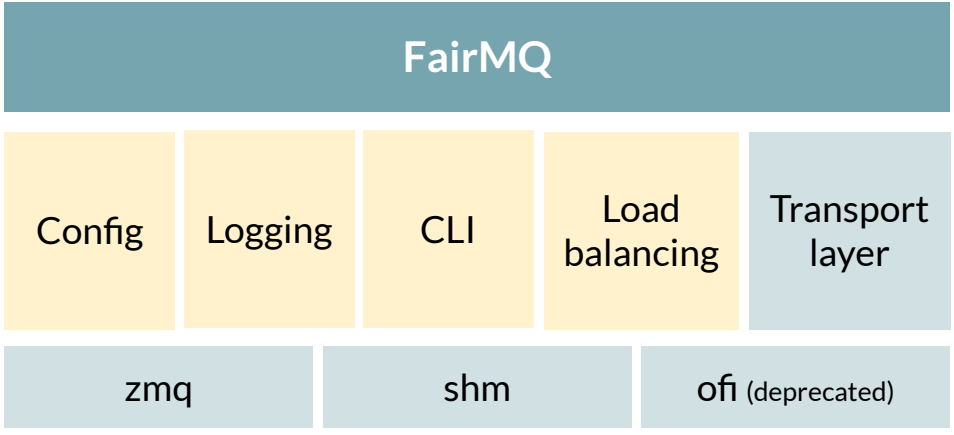
PandaR2 and SYCL

- Supports AdaptiveCpp (default) and Intel DPC++ compilers
- Wide range of supported platforms
- Possibility to fall-back from accelerator execution to CPU
  - Easier development
  - Easy to test CPU vs GPU performance
- Can also incorporate other programming models (OpenMP)



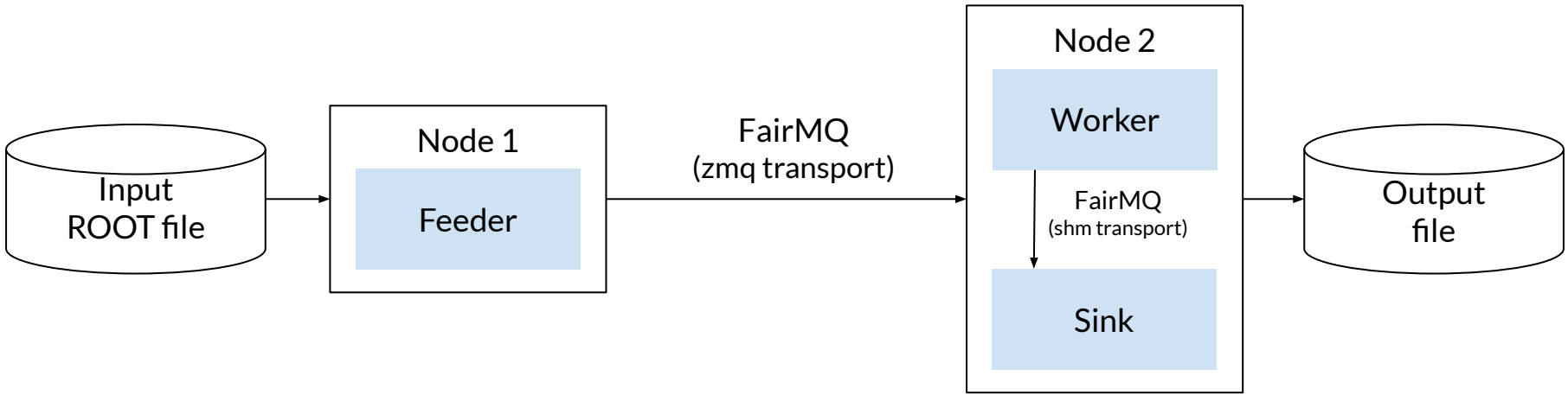
FairMQ

- Message Queuing Library and Framework
  - Developed at GSI/FAIR
- Large-scale data movement in distributed environments
- Abstract transport layer
- Provides tools, utilities and primitives
  - Ability to build a complete pipeline of connected processes (devices)
- Well tested on the large scale in ALICE



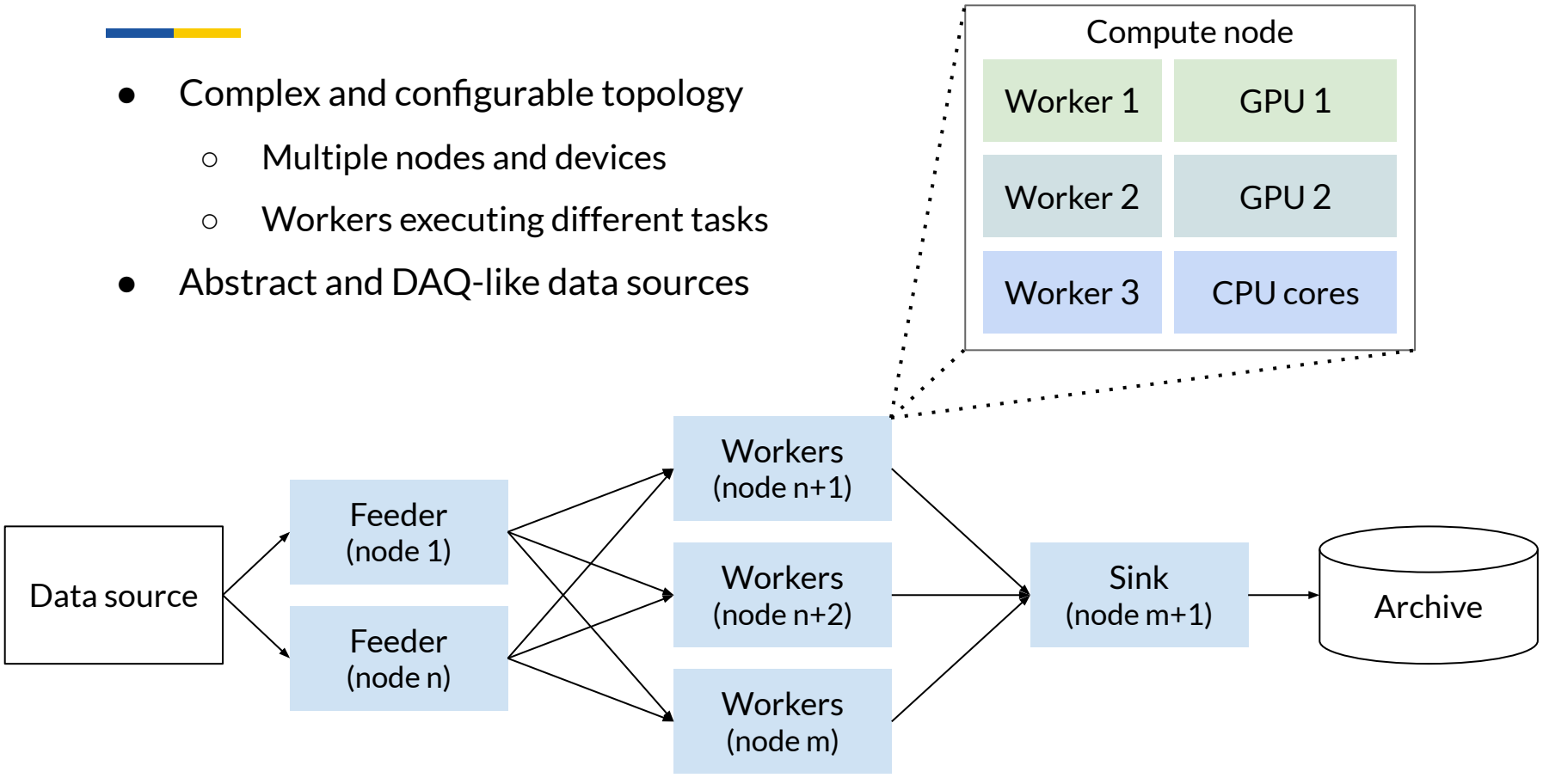
Simple setup

- Two nodes connected with Ethernet network
- *Feeder* FairMQ device on Node 1
  - Reads data from simulations file and sends serialized data
- *Worker + Sink* devices on Node 2
  - Receive data, run processing and dump output

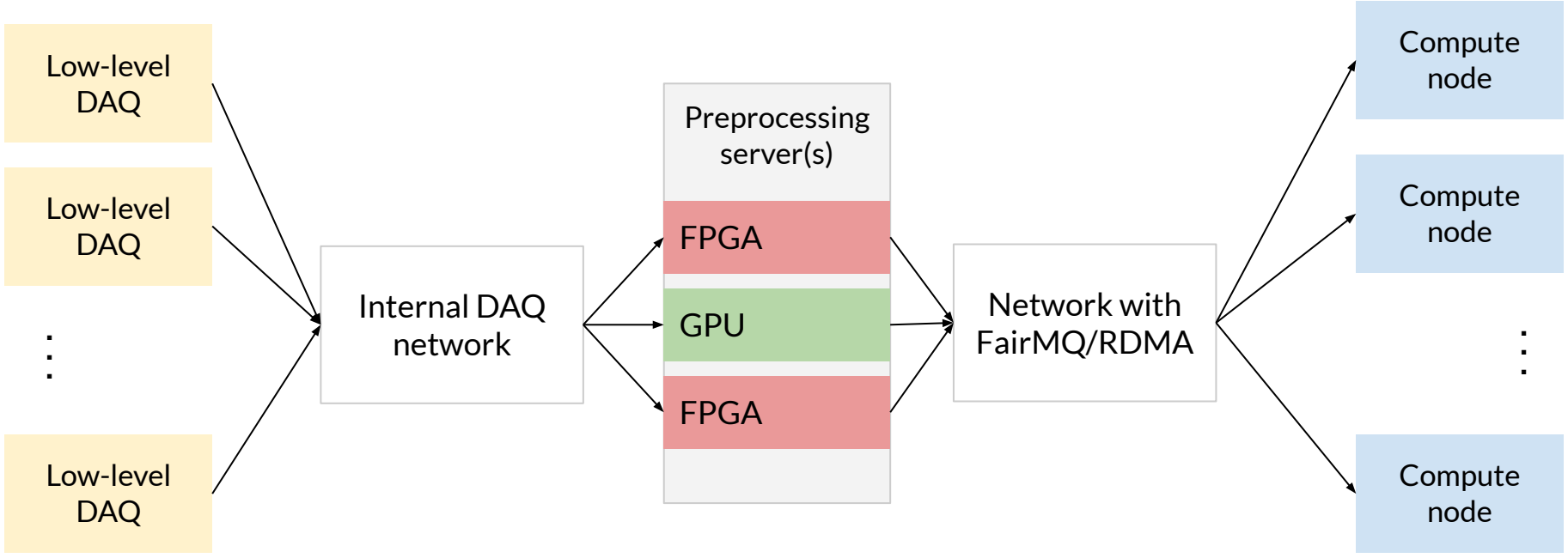


Advanced setups

- Complex and configurable topology
  - Multiple nodes and devices
  - Workers executing different tasks
- Abstract and DAQ-like data sources



Ultimate Goal



Development summary

Completed

- Framework foundations, basic abstractions
- Integration with SYCL and FairMQ
- Basic PandaRoot-R2 integration
- Two tracking algorithms implemented

Ongoing

- Implementing more elements of reconstruction pipeline
- Constructing complex FairMQ topologies with deployment

Future

- Infrastructure (parameters repository, monitoring)
- DAQ output simulation
- Testing in experimental test setup
- …