Remote direct memory access (RDMA) for low-latency high-throughput data acquisition and redistribution

IFDEPS 2024, Port Jefferson 17-20 March

Pablo Fajardo
Detector & Electronics Group
Instrumentation Services and Development Division
ESRF

ESRF | The European Synchrotron

DISTRIBUTED DATA ACQUISITION FOR ON-LINE PROCESSING

Motivation

- A single data receiver approach is no longer sustainable with increasing data throughputs.
- Need of tools to implement an efficient scheme for multi-receiver data distribution and processing.

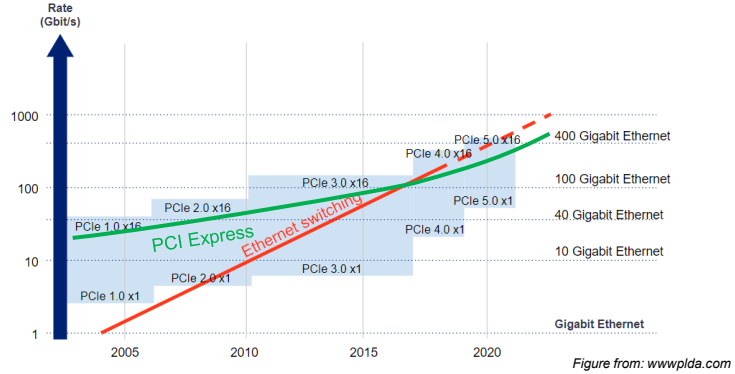
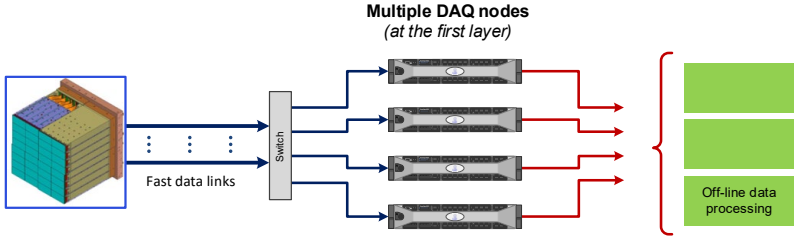


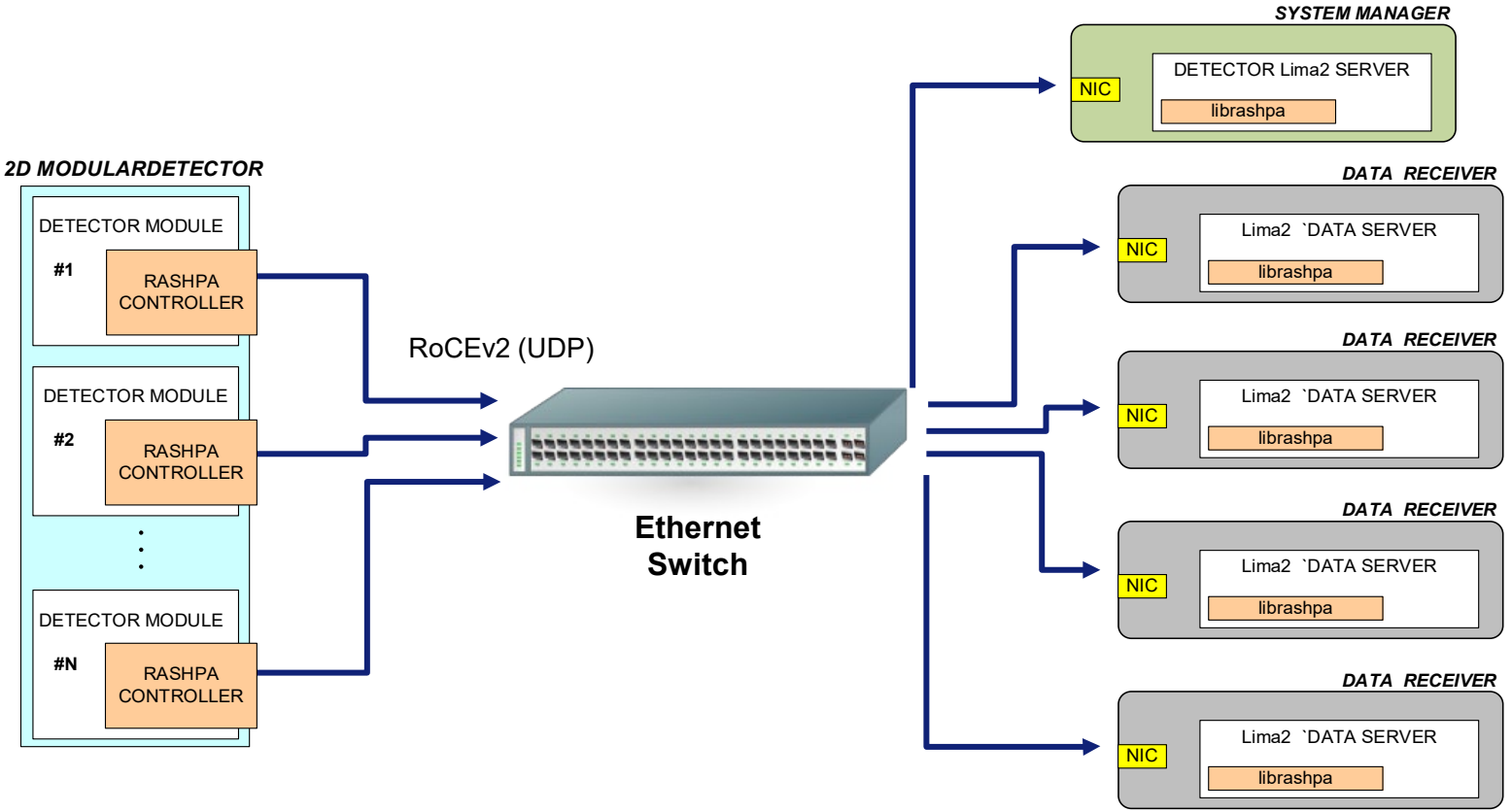
Figure from: wwwplda.com



The ESRF approach to build fully distributed DAQ systems relies on two components:

- **RASHPA** : Transfer and redistribution from the detector data into the processing nodes
- **Lima2** : Software framework for building detector servers with parallel processing (MPI based)

RASHPA/Lima2 DISTRIBUTED DAQ SYSTEMS

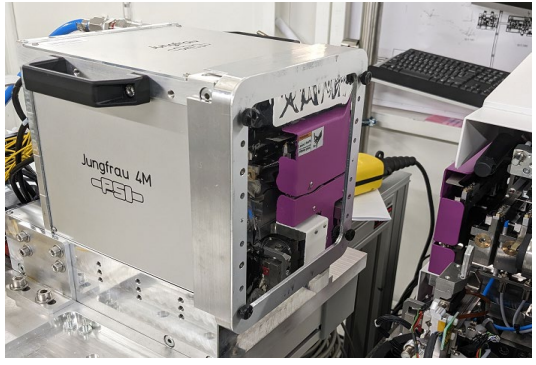


LIMA2: DISTRIBUTED CONTROL SOFTWARE FOR ADVANCED DETECTORS

- **High-level software** framework for 2D detector control and data acquisition.
- Based on **MPI** for parallel/**distributed** processing.
- Software processing pipelines can be **tailored to the specific applications**.
- Support for **in-house developments** (RASHPA based) and **other detectors**.

Jungfrau 4M for SSX experiments at ID29:

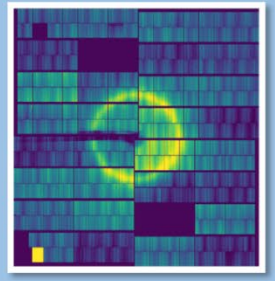
- 16 × 10 GbE links (50 % usage @ 1 kHz): 8 Gbytes/s of raw data.
- Lima2 **in production** since January 2023: ~500 Hz operation in continuous mode.
- **Processing-intensive** tasks performed on-line on **GPU**.
- Data is drastically reduced by pyFAI **peak finder & background suppression.**
  - Tune parameters to **extract useful data** from noise & background.
  - Seek scientific community consensus for **optimized data reduction** approaches.
- **Scalable topology**: more backend computers will allow higher detector frame rates.



Jungfrau 4M (PSI)

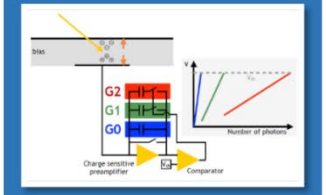
LIMA2 JUNGFRAU 4M & SMX PROCESSING

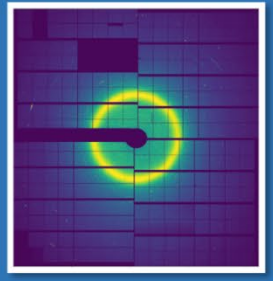
Image Reconstruction



UDP packets data from detector are geometrically assembled (CPU)

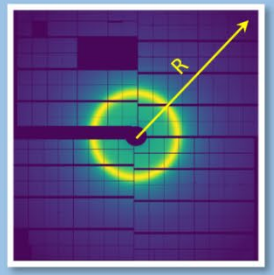
Background and Gain Corrections

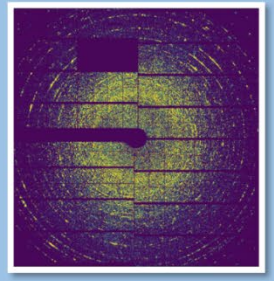




Per pixel & per frame gain selection: 3 pedestal + 3 gain maps

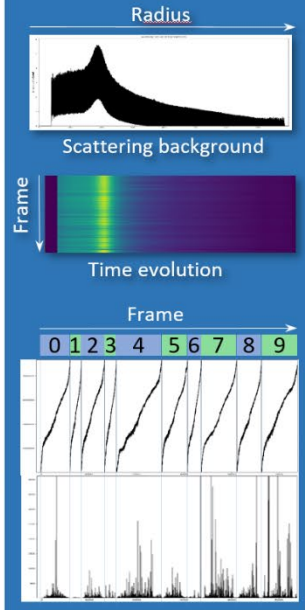
pyFAI-based Data Reduction





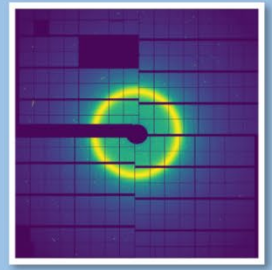
Peak finding and background extraction using Sigma Clipping

Sparse Data Saving



Peak pixel position & intensity per frame (CSR)

Offline re-densification

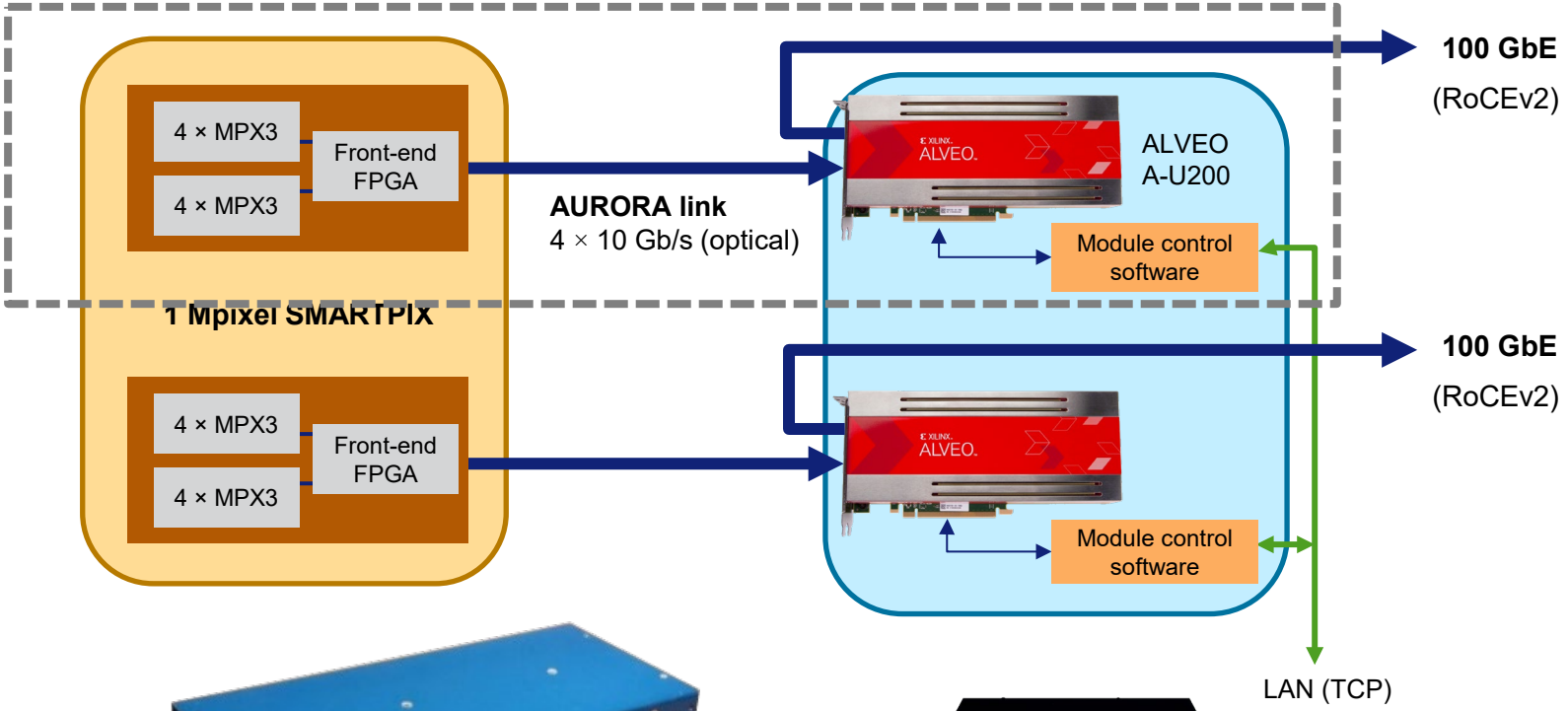


Performed to feed SMX data processing software

Jungfrau 4M specific

SXM specific

A RASHPA-BASED DETECTOR: SMARTPIX



SMARTPIX 1M

3000 fps (6 GB/s)

NETSTOR enclosure

RASHPA AS THE DATA REDISTRIBUTION LAYER

What is RASHPA?

- A **data dispatching** layer that transfers data from the detector to the **first level of computing** nodes
- A component to be integrated in a DAQ system for **high data throughput** detectors
- Uses remote direct memory access (**RDMA**)
- Designed/optimized for **modular/segmented** detectors and **2D detector data** (images)

Some features:

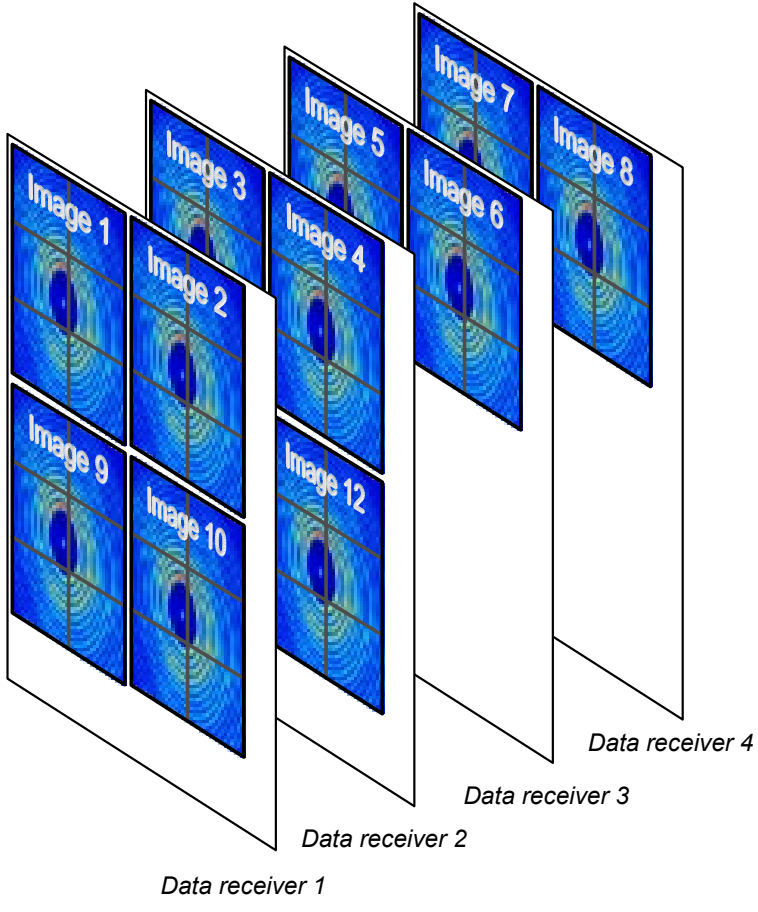
- **Zero-copy**: data is pushed by the detector modules into destination buffers, no software intervention
- Uses a **switchable network** supporting RDMA transfers, the number of processing nodes is arbitrary
- Data is dispatched and redistributed in a **configurable** way, to match the application needs
- **Fully COTS backend** components: no dependence with specific hardware, the achievable performance can follow the evolution of the market

THE DATA TRANSFER PROCESS (DTP)

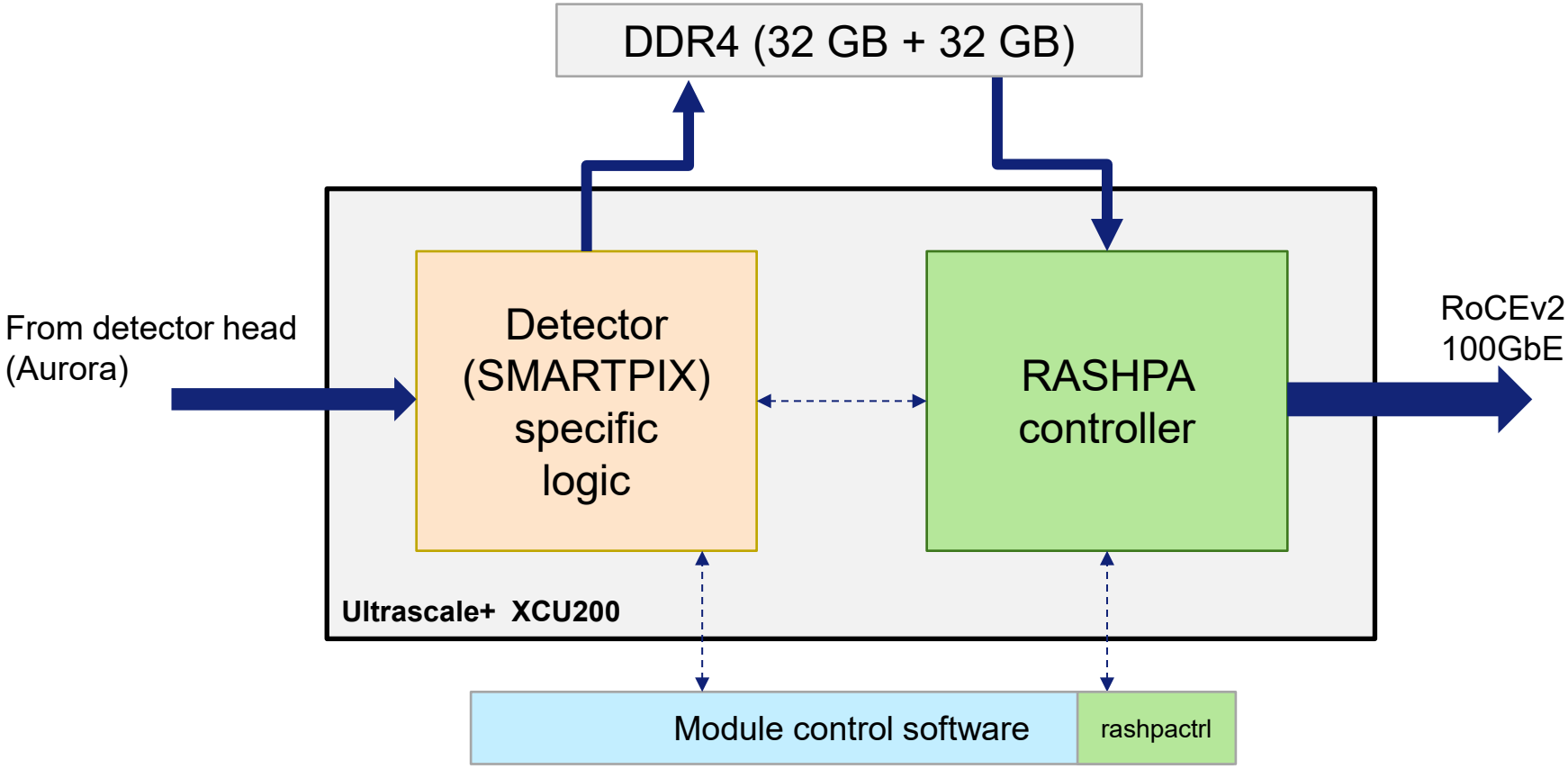
- Image frames (data blocks) are transferred from the detector modules to the destination buffers
  - The destination buffers can be **any (virtual) memory area** in the data receiver nodes… (RAM, GPU, …)
- A DTP must be configured by defining a set of rules:
  - **What to transfer**: which data blocks (images) , which fraction of the data block (ROI)
  - **Where**: which destination buffers in the data receivers and how to dispatch the data between them
- Several data transfer processes can run in parallel on the same data
  - **Multiple DTPs** allow to distribute image regions into groups data receivers, to separate processing
  - And transferring **metadata**
  - The same data can be retransmitted in more than one DTP (e.g. advance online data monitoring)
- The geometric reconstruction of image data from multiple modules is implicit in the DTP operation
- In an active DTP, new data is transferred as soon as they are available
  - Asynchronous **events** are issued to inform the data receivers and system manager of **progress and errors**
  - A ‘data credit’ mechanism is specified **to slow down** the data transfer (but not implemented so far)

DTP EXAMPLE

- Example of a single Data Transfer Processes :
  - 6-module 2D detector
    - One RASHPA controller per module
  - Data gets redistributed into 4 data receivers
- Operation of multiple simultaneous DTPs is possible



SMARTPIX FPGA



RASHPA TELEGRAMS

- Configuration/status data is coded and transferred between RASHPA components as opaque “telegrams”
- RASHPA does not implement its own communication channels: the control system is responsible of transferring RASHPA telegrams between the RASHPA controllers and the *librashpa* instances.

Example of status information telegram

```
<rashpa version="1.0">
    <status_info>
        <module name="first_module">
            <dchan index="0x0" … />
            <dchan index="0x1" … />
            ...
        </module>
    </status_info>
</rashpa>
```

Example of module information telegram

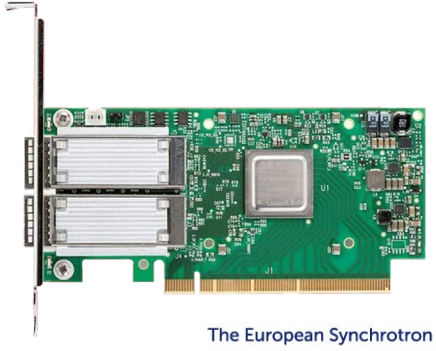
```
<rashpa version="1.0">
    <module_info>
        <module name="my_module">
            <dlink_info>
                <endpoint type="PCIe"> … </endpoint>
                ...
             </dlink_info>

            <dataset name="my_image">
                <frame>
                    <format> … </format>
                    <position … />
                    <supported> … </supported>
                </frame>
            </dataset>
            <dataset name="metadata">
                <fixedsize … />
            </dataset>
            ...

            <capabilities>
                <dchan index="0">
                    <dataset name="mainImage"/>
                    <dataset name="metadata"/>
                    ...
                    <limits … />
                    <supported>
                        <transfer_mode … />
                        <frame_ops> … </frame_ops>
                    </supported>
                </dchan>

                <dchan from="2" to="10"> … </dchan>
                ...
            </capabilities>
        </module>
    </module_info>
</rashpa>
```

DATA TRANSMISSION

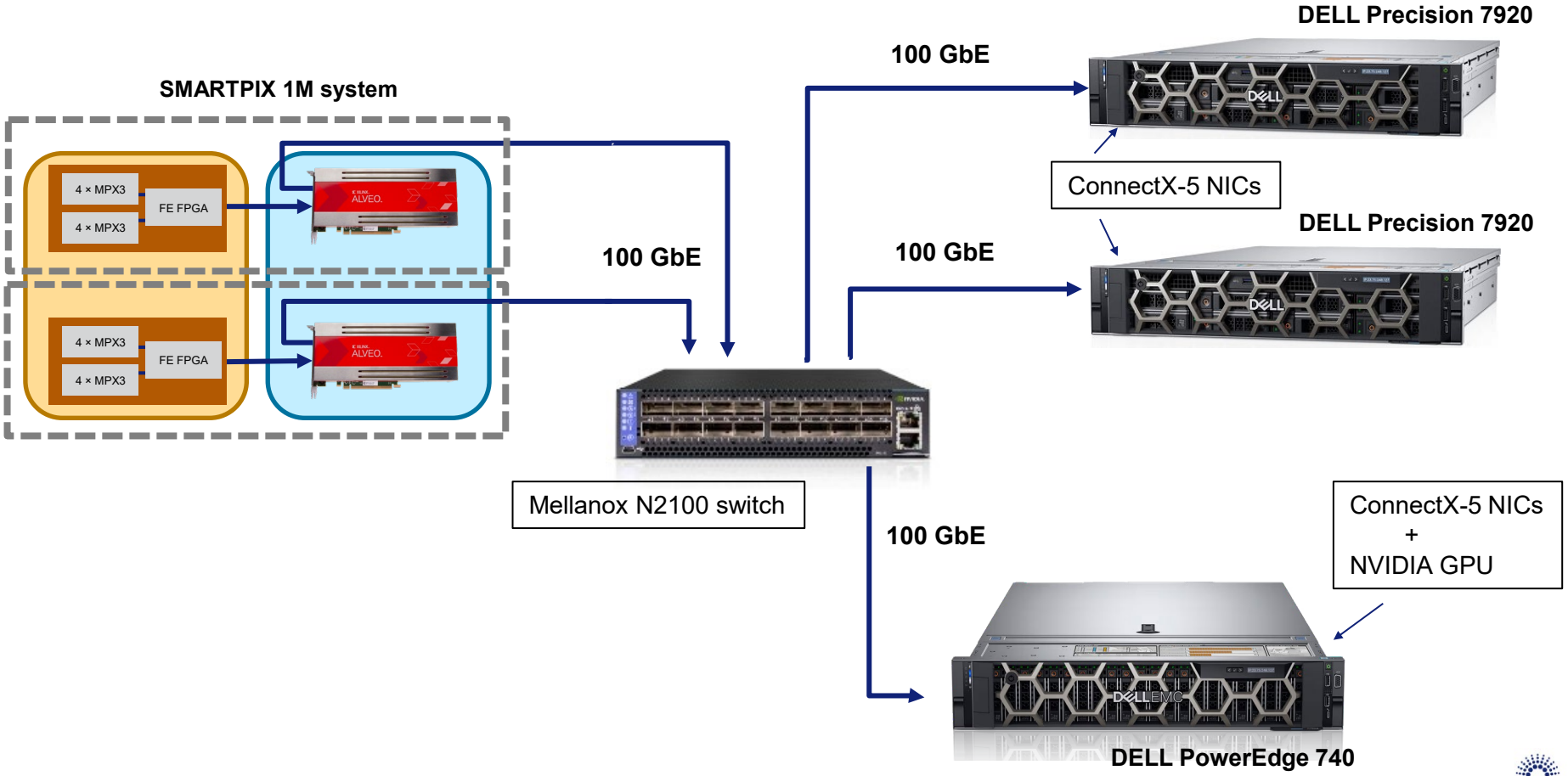
- Data transmission protocol (RASHPA functional requirements)
  - Support of RDMA routable/switchable data transfers
  - Event dispatch mechanism
  - High data bandwidth

- Our implementation is based on **RoCEv2**
  - Other candidates: InfiniBand, iWARP (and even PCIe over cable)

- About RoCEv2:
  - RoCE = ***RDMA over Converged Ethernet***
  - Protocol **developed by Mellanox** (NVIDIA) for low-latency high-bandwidth communication in **data centers**
  - Supported in **Linux** (and all major OS)
  - Encapsulates RDMA packets in **UDP** datagrams (IP routable protocol)
  - Major manufacturers of network interfaces (NIC) **support RoCEv2** :
    - Intel, Broadcom, Mellanox, Marvell, …



A RASHPA-BASED SYSTEM (EXAMPLE)



OTHER CHARACTERISTICS

Among the potentially useful aspects of RASHPA

- **Frees CPU resources** for other purposes (on-line data processing, compression, etc.)
- **Generic** design for standardization and reusability
  - Is **scalable** by construction
  - The current implementation (RASHPA controller and librashpa) is **not tied to a specific detector**
  - “RASHPA” **learns** about the detector and the computing node topology **at configuration time**
- **Low-latency** data transfer (sub-millisecond)

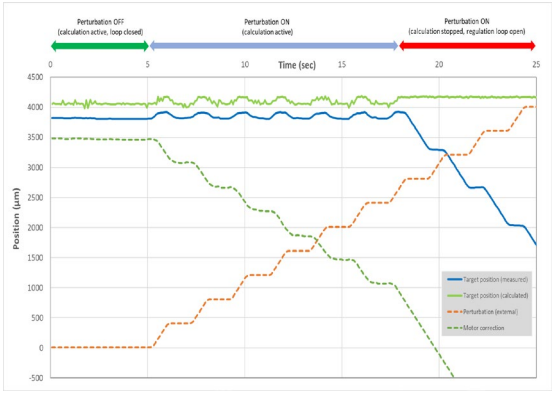
LOW LATENCY DATA ACQUISITION AND PROCESSING

- Data transfer and processing in millisecond scale opens the door to fast **experimental feedback**

- Use the 2D data information to **change dynamically** the experiment conditions:
  - To **reduce sample radiation damage** by adapting the beam intensity and the sample scanning sequence
    - By dynamically **changing** the **beam intensity** and the **scanning/mapping sequence**
    - A proof-of-concept at an ESRF beamline (microfocus diffraction) is under discussion
  - To **correct beam/sample position drifts** during data acquisition in nanofocusing experiments
  - To detect **rare events**

- A simple lab demonstration of fast feedback capabilities. Closing the position control loop of a mechanical device by means of a SMARTPIX/RASHPA GPU-based metrology chain.
  - Multiple data receivers with different roles (one dedicated to fast feedback)
  - Determination of the position of a target object at 1kHz sampling rate

SMARTPIX 1M (ESRF)



Active feedback system based in a SMARTPIX/RASHPA/Lima2/GPU/IcePAP chain to control the position of an object

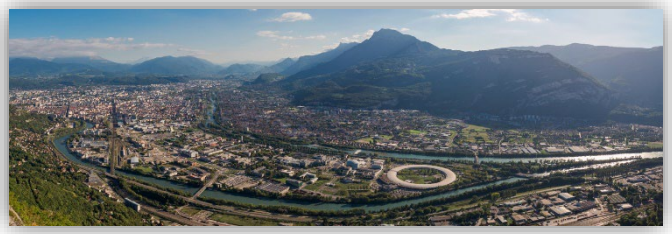
ACKNOWLEDGEMENTS

RASHPA development

- Fabien LeMentec
- Nicolas Janvier
- Wassim Mansour
- Aurèlien Bideuad
- Raphaël Ponsard

Lima2 development

- Alejandro Homs
- Samuel Debionne
- Laurent Claustre



THANK YOU FOR YOUR ATTENTION

RASHPA CONTROL/CONFIGURATION/ACQUISITION STEPS