Green Flash

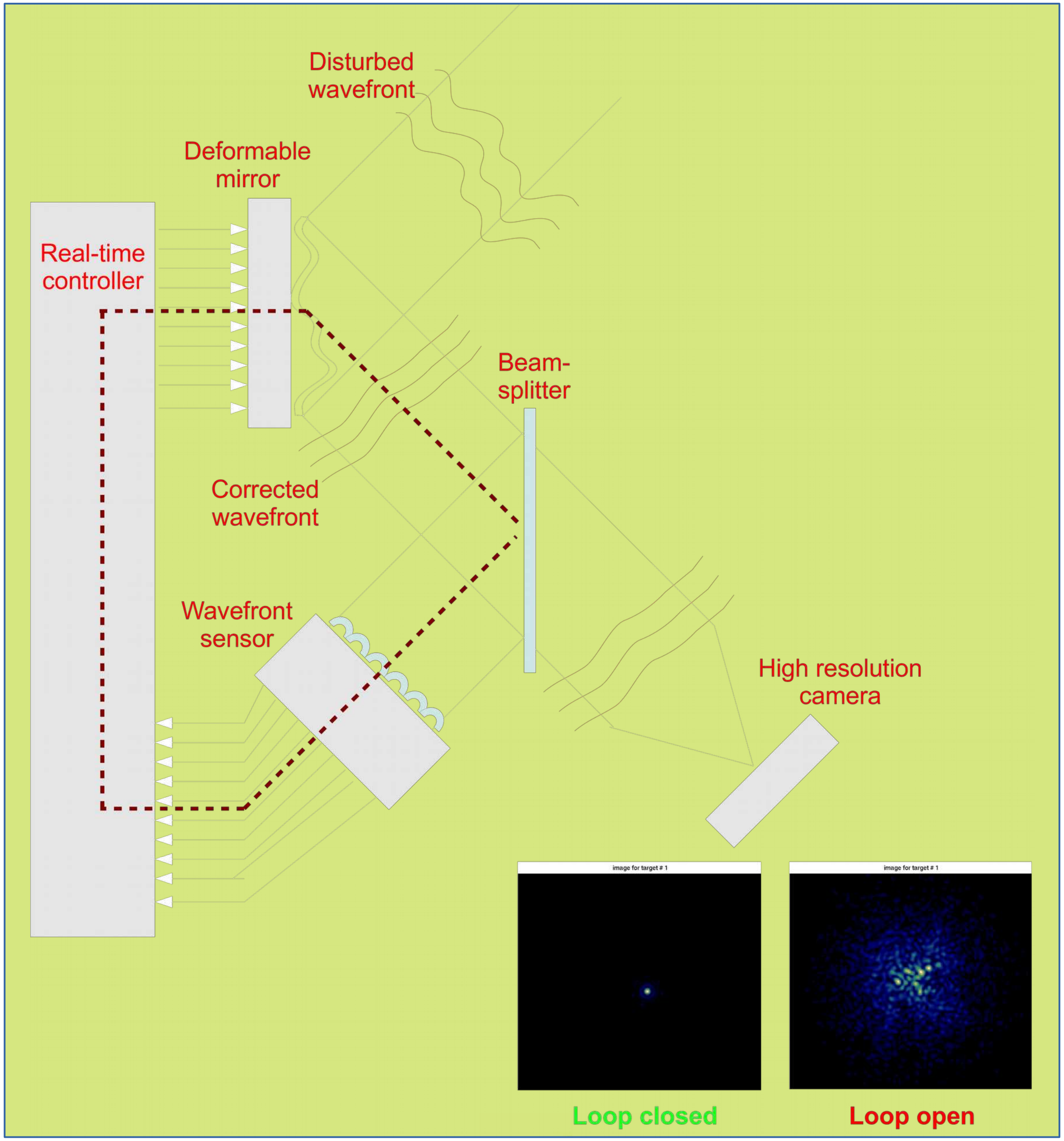
High performance computing for real-time science

Project overview and management (WP 1 & 2)

Project #671662 funded by European Commission under program H2020-EU.1.2.2 coordinated in H2020-FETHPC-2014

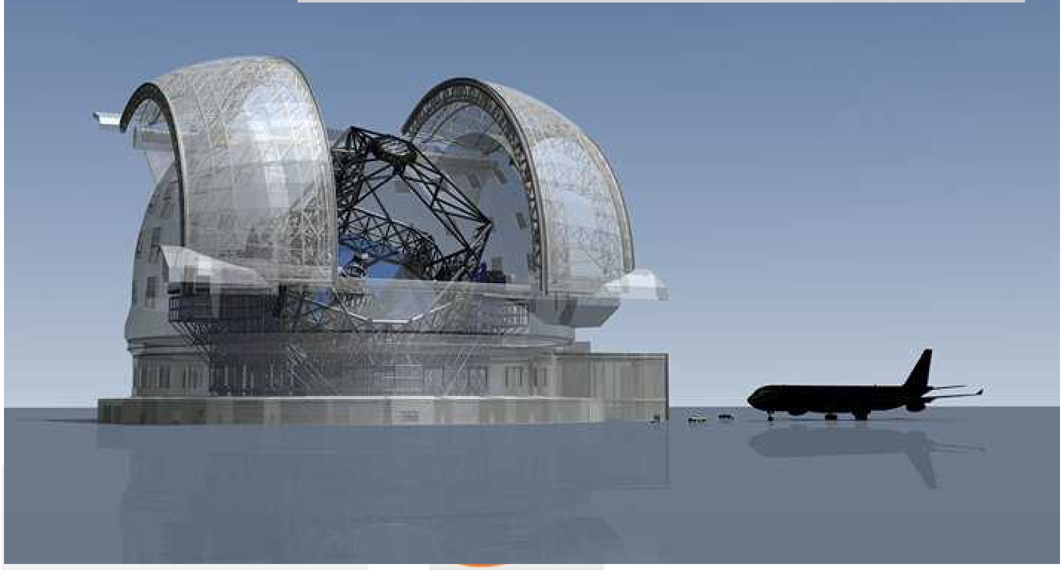
Adaptive optics

- Compensate in real-time the wavefront perturbations
- Using a wavefront sensor to measure them
- Using a deformable mirror to reshape the wavefront
- Commands to the mirror must be computed in real-time (1ms rate)



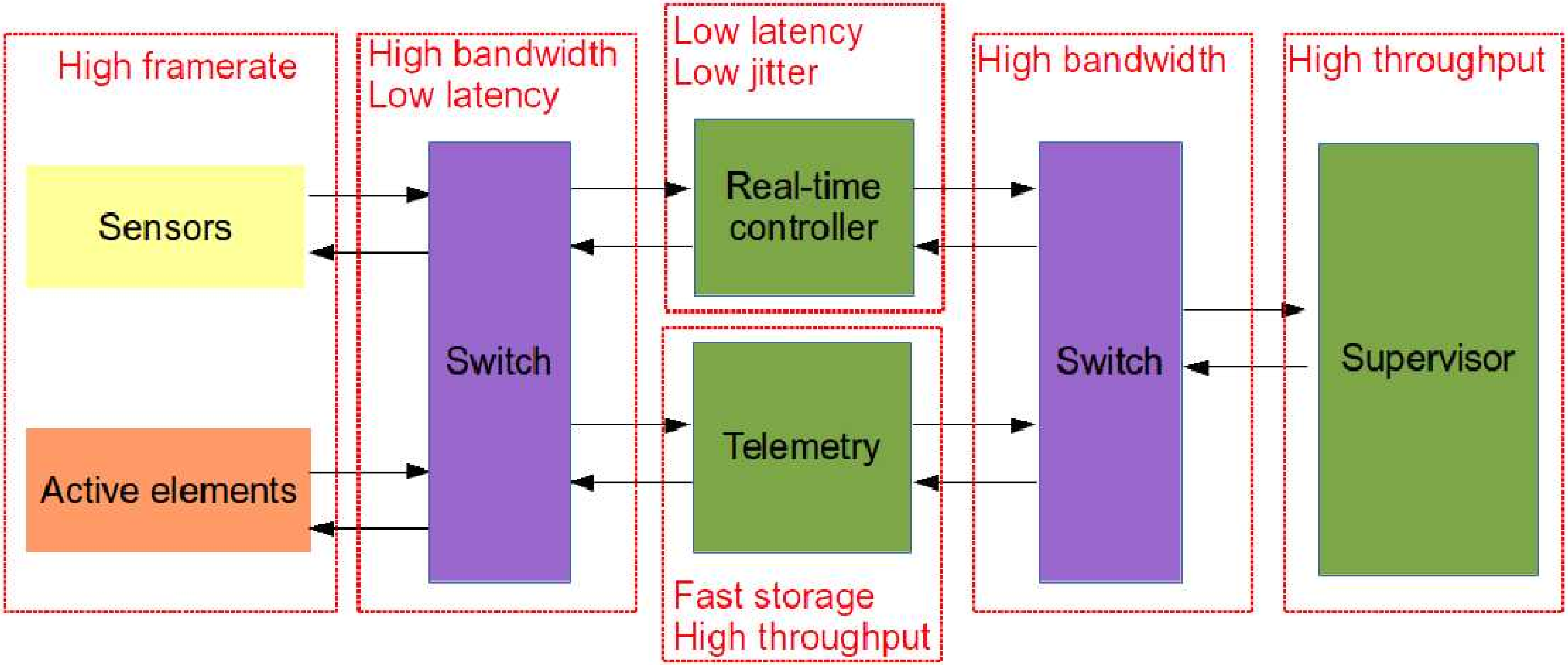
European Extremely Large Telescope

- 39m diameter telescope : x5 in diameter => x25 in system complexity
  - 100m dome, 2800 tones structure rotating @ 360°, seismic safe (Chile)
  - 1.2 G€ project, first light foreseen in 2024
  - Construction led by ESO (European Southern Observatory), international organization funded by 15 European countries
  - Telescope components + science instruments built by European research labs + industrial partners

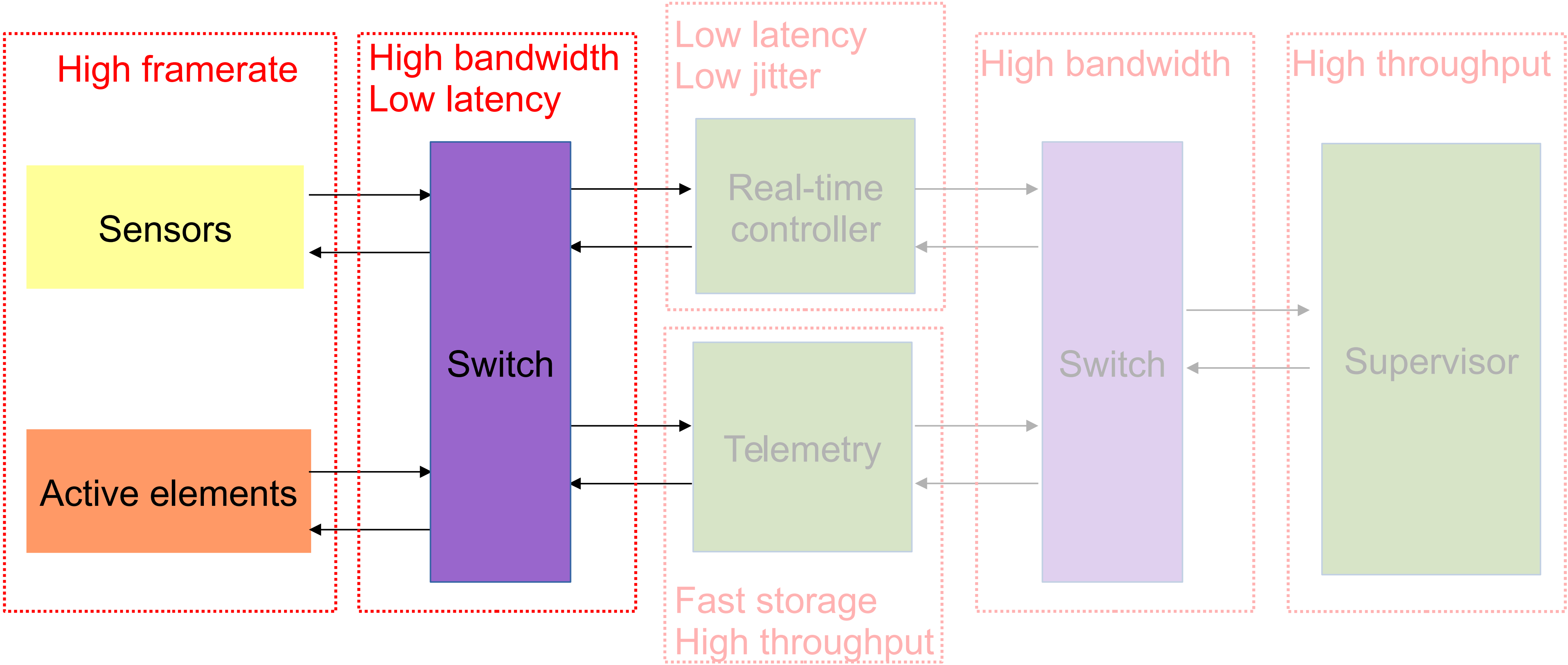


l'Observatoire de Paris LESIA
Laboratoire d'Études Spatiales et d'Instrumentation en Astrophysique
Durham University

AO RTC concept

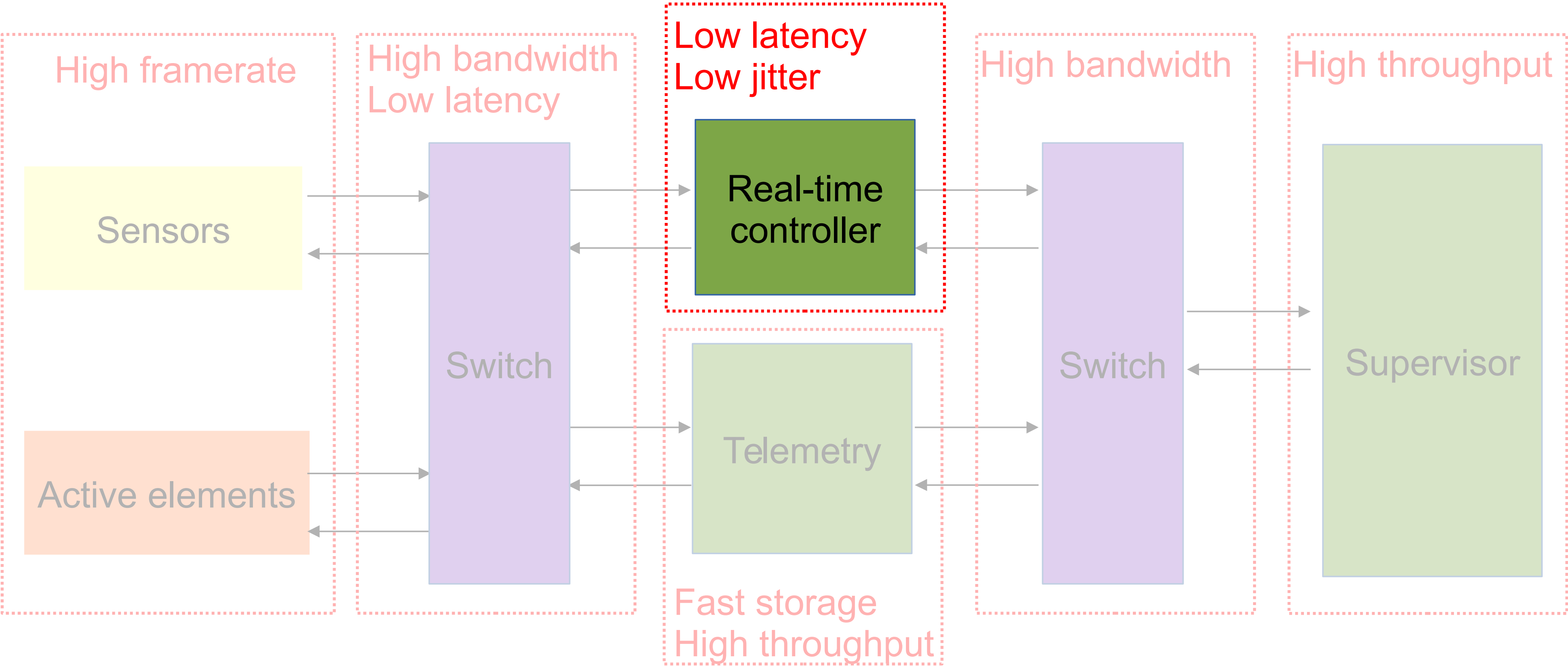


AO RTC concept : RT simulator



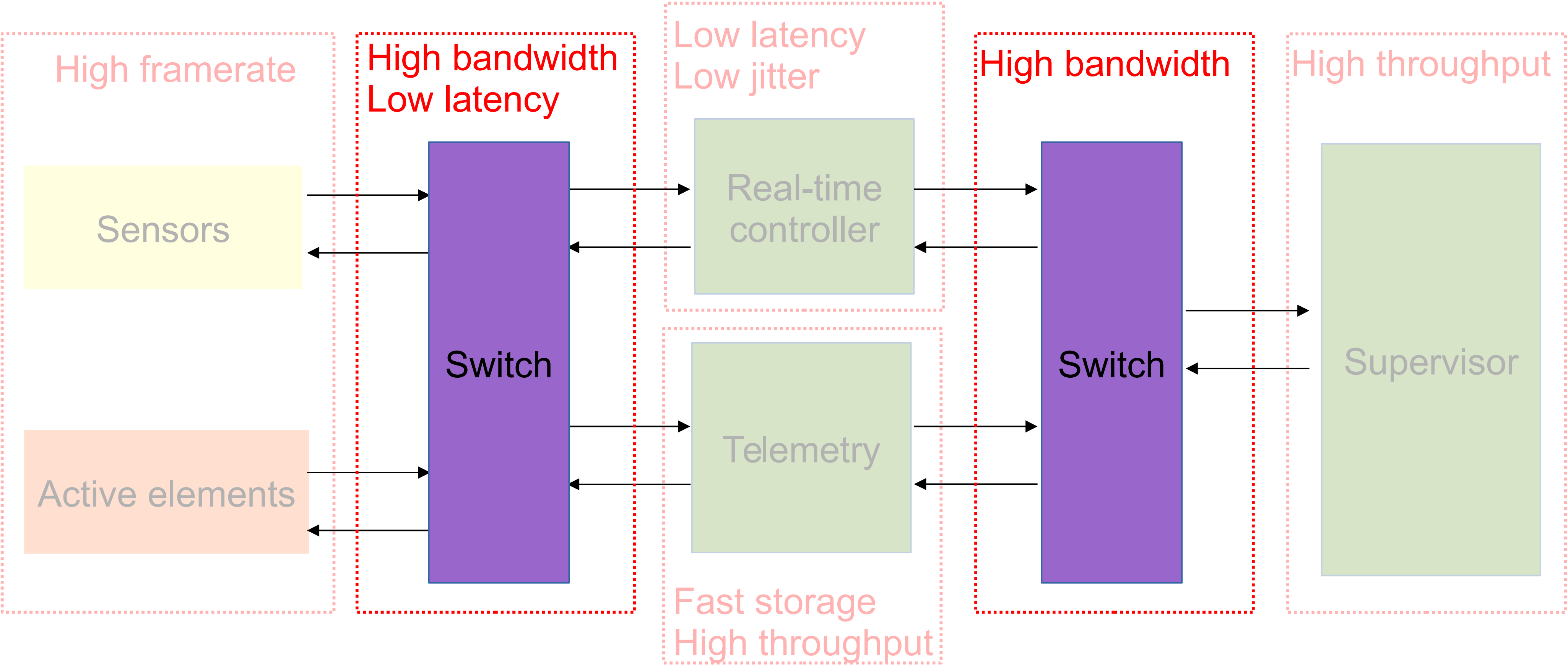
01/22/2016

AO RTC concept : data pipeline

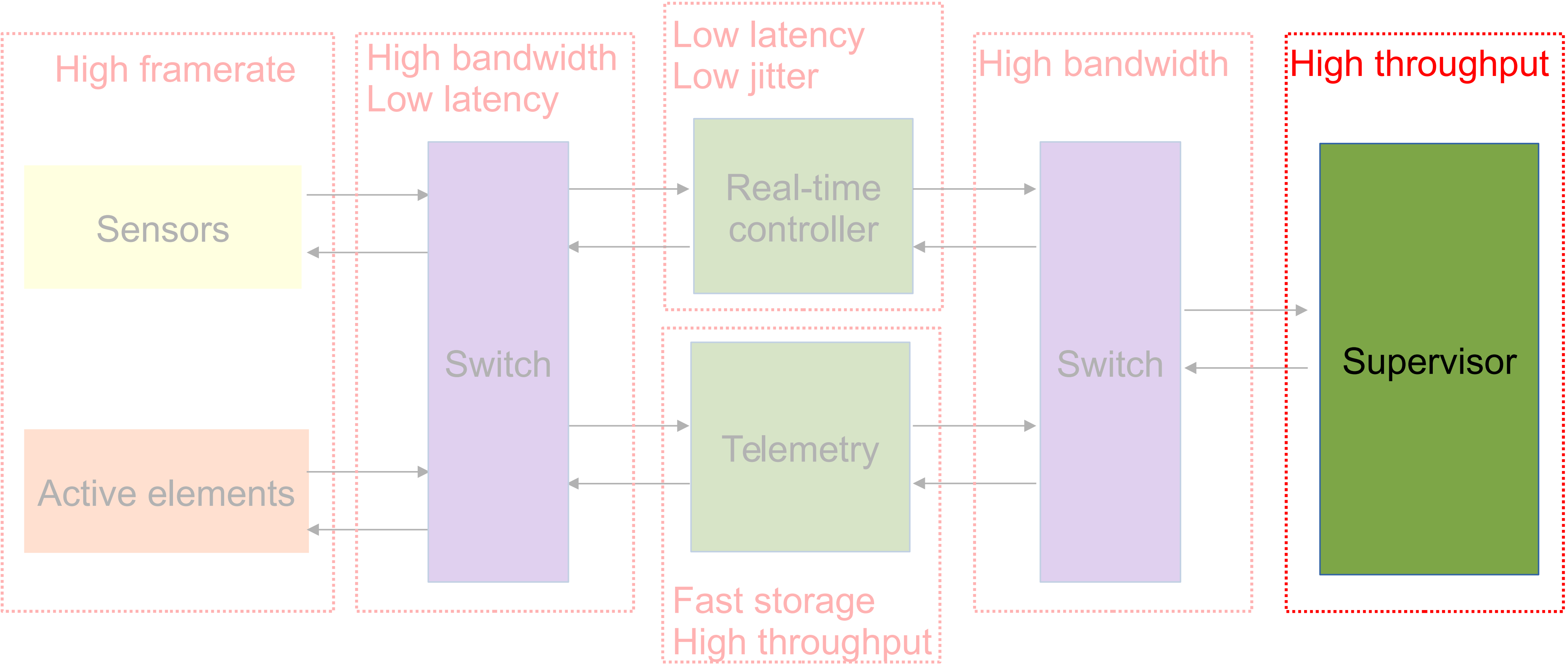


01/22/2016

AO RTC concept : smart interconnect

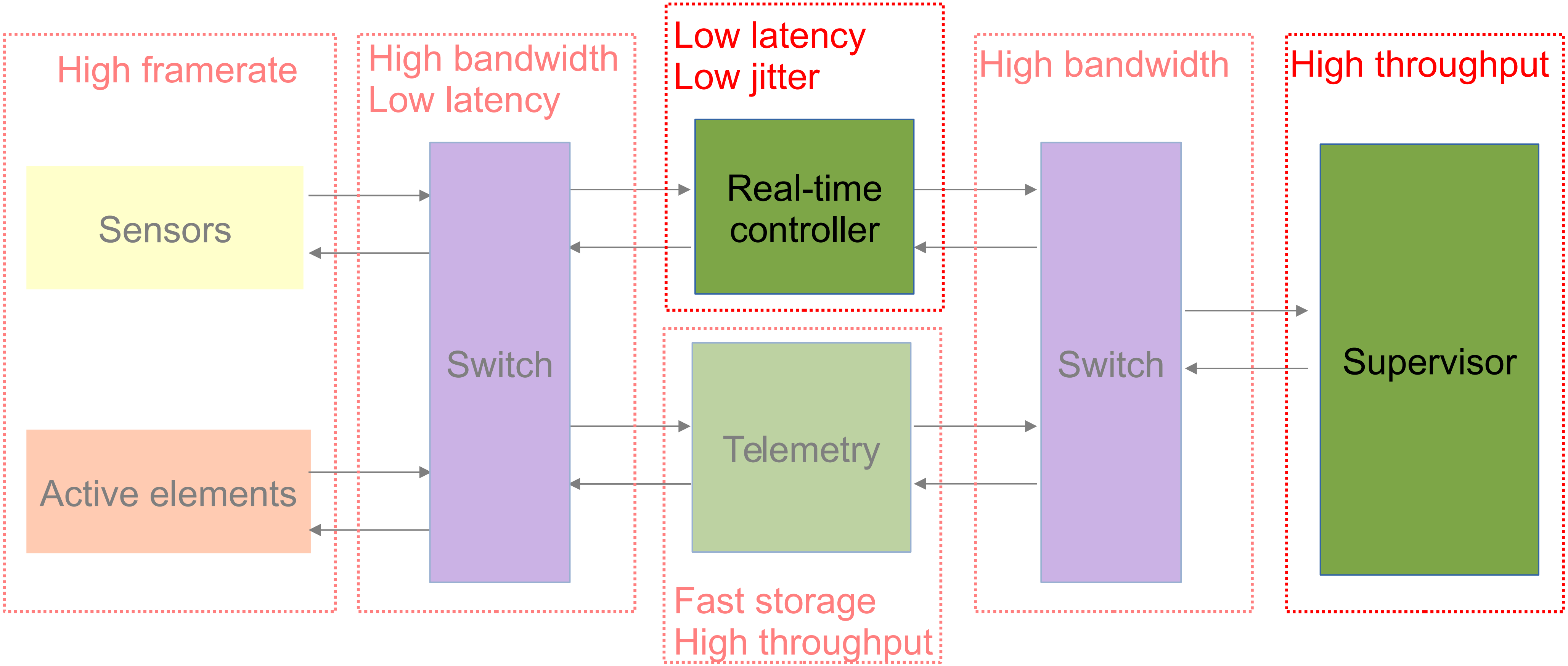


AO RTC concept : supervisor



01/22/2016

AO RTC concept : SW & MW



01/22/2016

Introduction to Green Flash

- Program objectives: 3 research axes
  - 2 technological developments and 1 validation study
- Real-time HPC using accelerators and smart interconnects
  - Assess the determinism of accelerators performance
  - Develop a smart interconnect strategy to cope for strong data transfer bandwidth constraints
- Energy efficient platform based on FPGA for HPC
  - Prototype a main board, based on FPGA SoC and PCIe Gen3
  - Cluster such boards and assess performance in terms of energy efficiency and determinism
- AO RTC prototyping and performance assessment
  - Assemble a full functionality prototype for a scalable AO RTC targeting the MAORY system
  - Compare off-the-shelf solutions based on accelerators and new FPGA-based concept

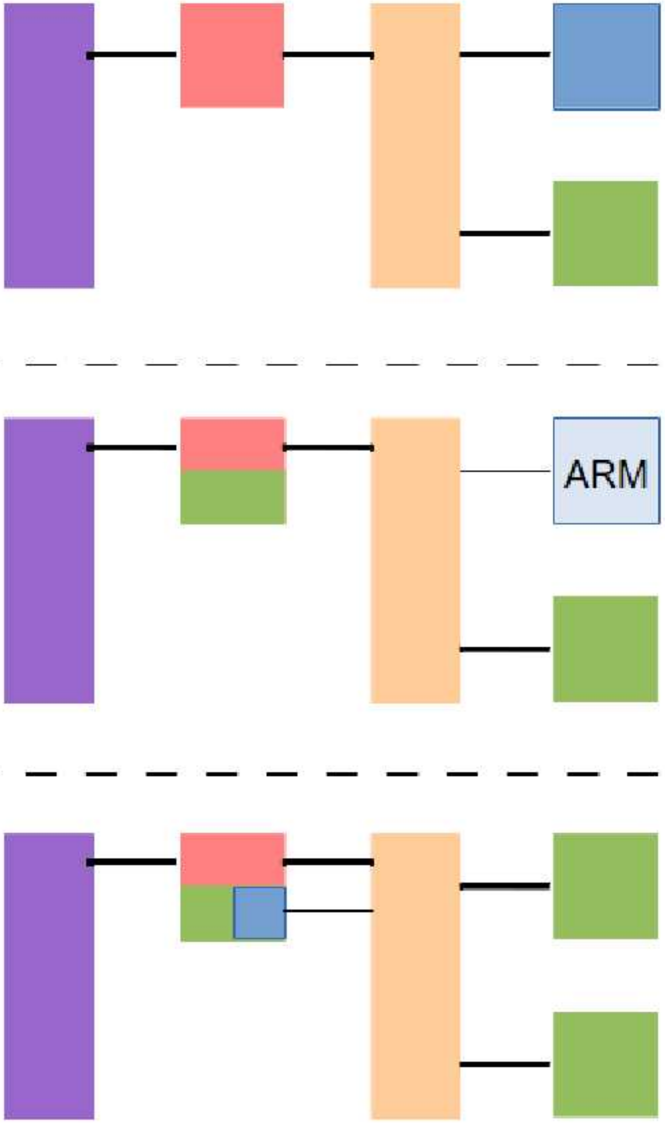
What this is about … really

- Find the best trade-off for ELT sized AO systems RTC
  - Comprehensive assessment of existing technologies
  - Development of new custom solutions for comparison
  - Propose new development processes to reduce cost and increase maintainability
- Build a full featured RTC prototype at the largest scale possible
  - Technology down-selection from a number of criteria : performance, cost, compliance to standards, obsolescence, maintainability
  - State of the art system to be assessed in the llab, with a simulator

Objectives of Green Flash

- Real-time HPC using accelerators and smart interconnects
  - (1.1) Prototype cluster : 1.5 TMAC/s, 250 Gb/s of streaming data, max. jitter of 100µs of 1 sec of operations based on COTS accelerators
  - (1.2) Develop COTS NIC solution based on FPGA (TCP/UDP through 10G Ethernet)
  - (1.3) Complement FPGA development tool (QuickPlay) ecosystem with data handling and computing blocks for smart interconnect strategy
  - (1.4) Assess performance of linear algebra (MVM, Cholesky facto.) on prototype cluster
- Energy efficient platform based on FPGA for HPC
  - (2.1) Prototype a main board, based on FPGA SoC (Arria 10), including PCIe Gen3 and 10G Ethernet
  - (2.2) Provide support for this board in QuickPlay, including smart interconnect features
  - (2.3) Cluster such boards and assess performance in terms of energy efficiency and determinism on linear algebra + streaming data
- AO RTC prototyping and performance assessment
  - (3.1) Assemble a full functionality prototype for a scalable AO RTC targeting the MCAO system on E-ELT
  - (3.2) Implement a real-time simulator for performance assessment
  - (3.3) Fully characterize the AO RTC prototype performance under realistic conditions with simulator

Assessing new HPC concepts



Current trend in heterogeneous HPC

- 🟪 Network fabric
- 🟥 Network controller
- 🟧 Local interconnect (PCIe)
- 🟦 Host CPU
- 🟩 Accelerator

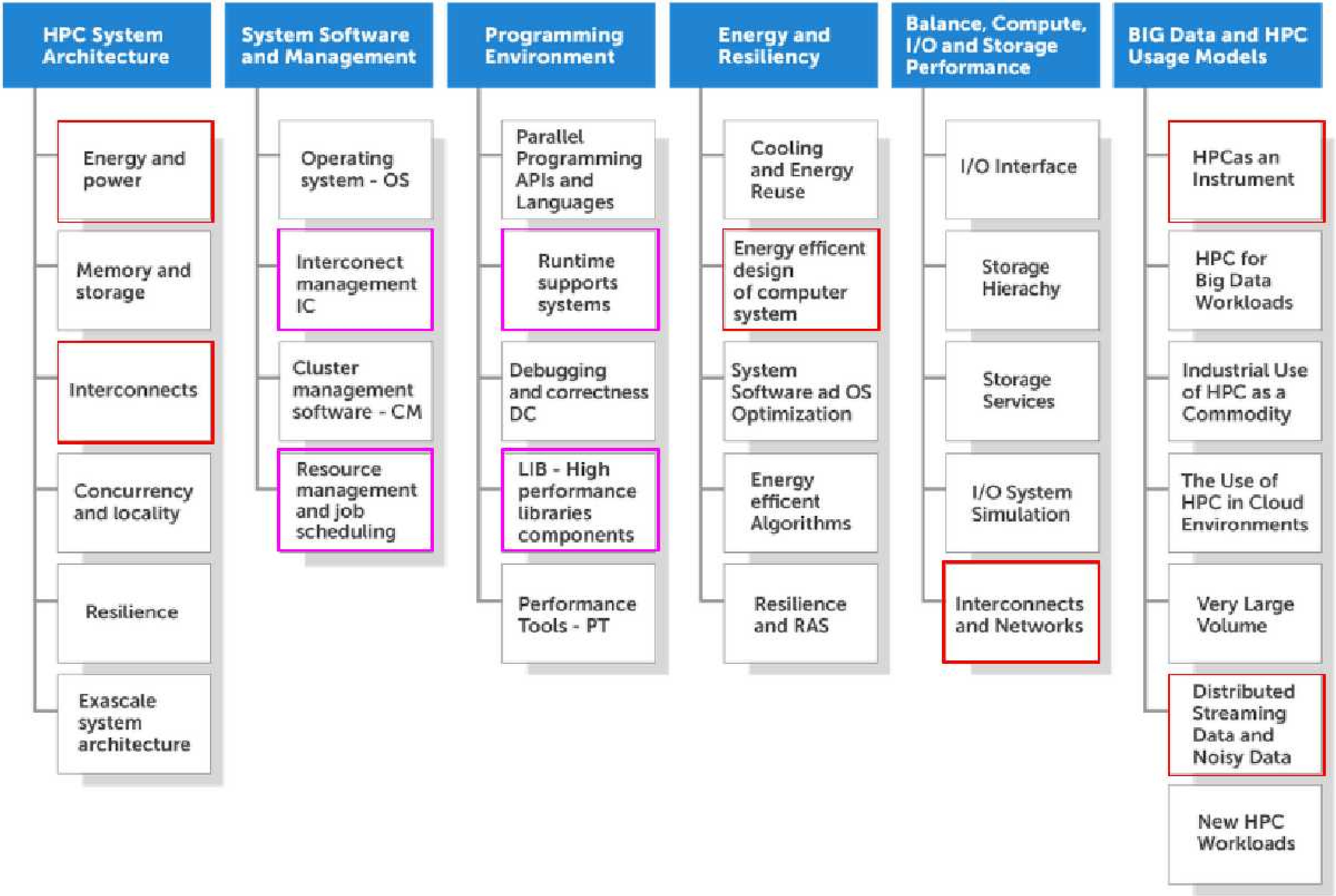
Accelerators, ARM64 & smart interconnect

Dynamically reconfigurable NIC
Expandable programming environment on FPGA
Features accessible in middleware

FPGA-based microserver

Main board with FPGA + ARM HPS
Integrated ecosystem for interconnect handling and accelerators support
Scalable environment

Addressing HPC roadmap to exascale



HPC System Architecture
- Energy and power
- Memory and storage
- Interconnects
- Concurrency and locality
- Resilience
- Exascale system architecture

System Software and Management
- Operating system - OS
- Interconnect management IC
- Cluster management software - CM
- Resource management and job scheduling

Programming Environment
- Parallel Programming APIs and Languages
- Runtime supports systems
- Debugging and correctness DC
- LIB - High performance libraries components
- Performance Tools - PT

Energy and Resiliency
- Cooling and Energy Reuse
- Energy efficent design of computer system
- System Software ad OS Optimization
- Energy efficent Algorithms
- Resilience and RAS

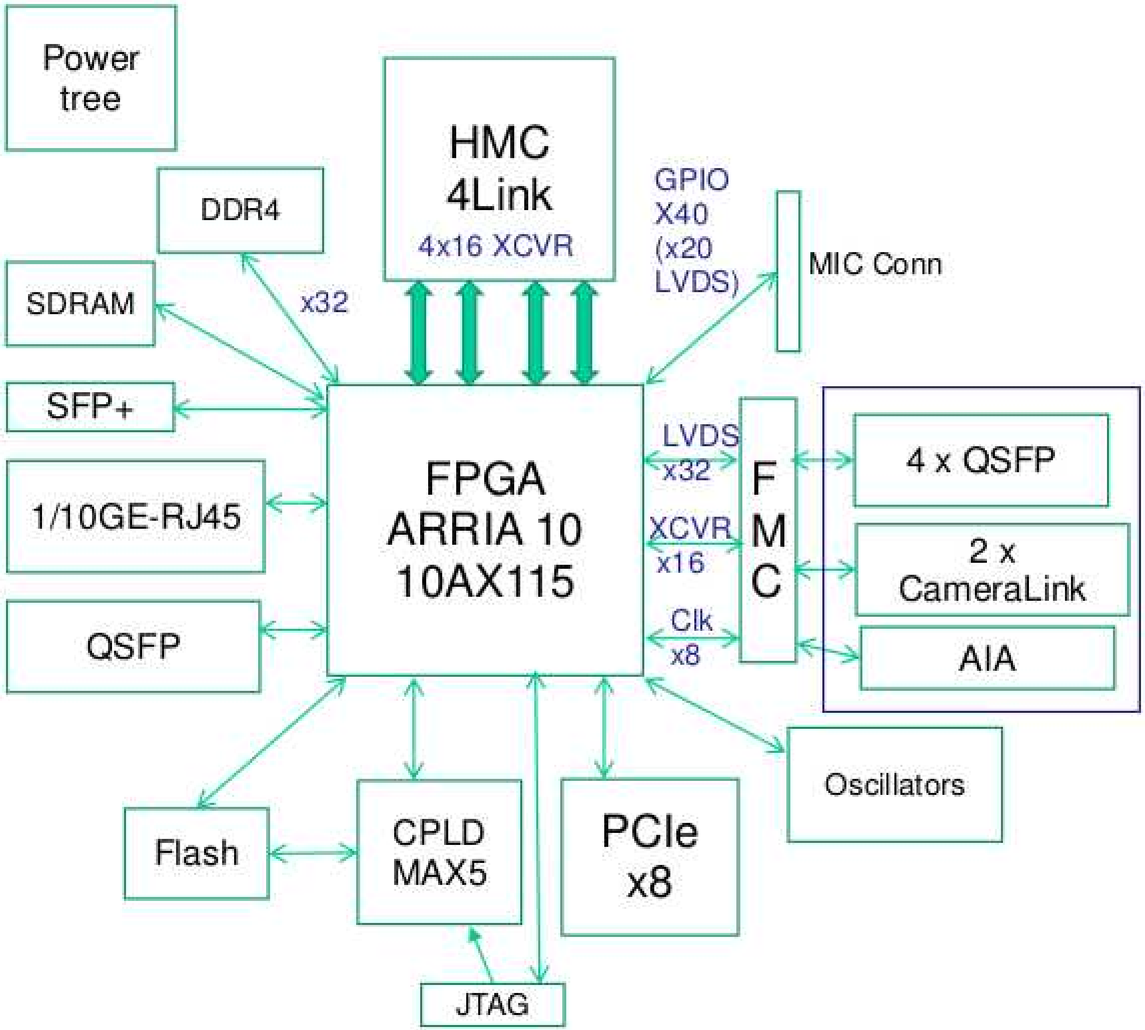
Balance, Compute, I/O and Storage Performance
- I/O Interface
- Storage Hierachy
- Storage Services
- I/O System Simulation
- Interconnects and Networks

BIG Data and HPC Usage Models
- HPCas an Instrument
- HPC for Big Data Workloads
- Industrial Use of HPC as a Commodity
- The Use of HPC in Cloud Environments
- Very Large Volume
- Distributed Streaming Data and Noisy Data
- New HPC Workloads

Green Flash project

- Partners
  - 2 academic partners
    - LESIA, Observatoire de Paris, P.I. Damien G.
    - CfAI, University of Durham
  - 2 industrial partners
    - Microgate : Italian SME designing FPGA solutions for various applications (including astronomical AO)
    - PLDA: French SME developing FPGA solutions (mostly IP cores, world leader in PCIe IP)

FPGA solution : µXcomp



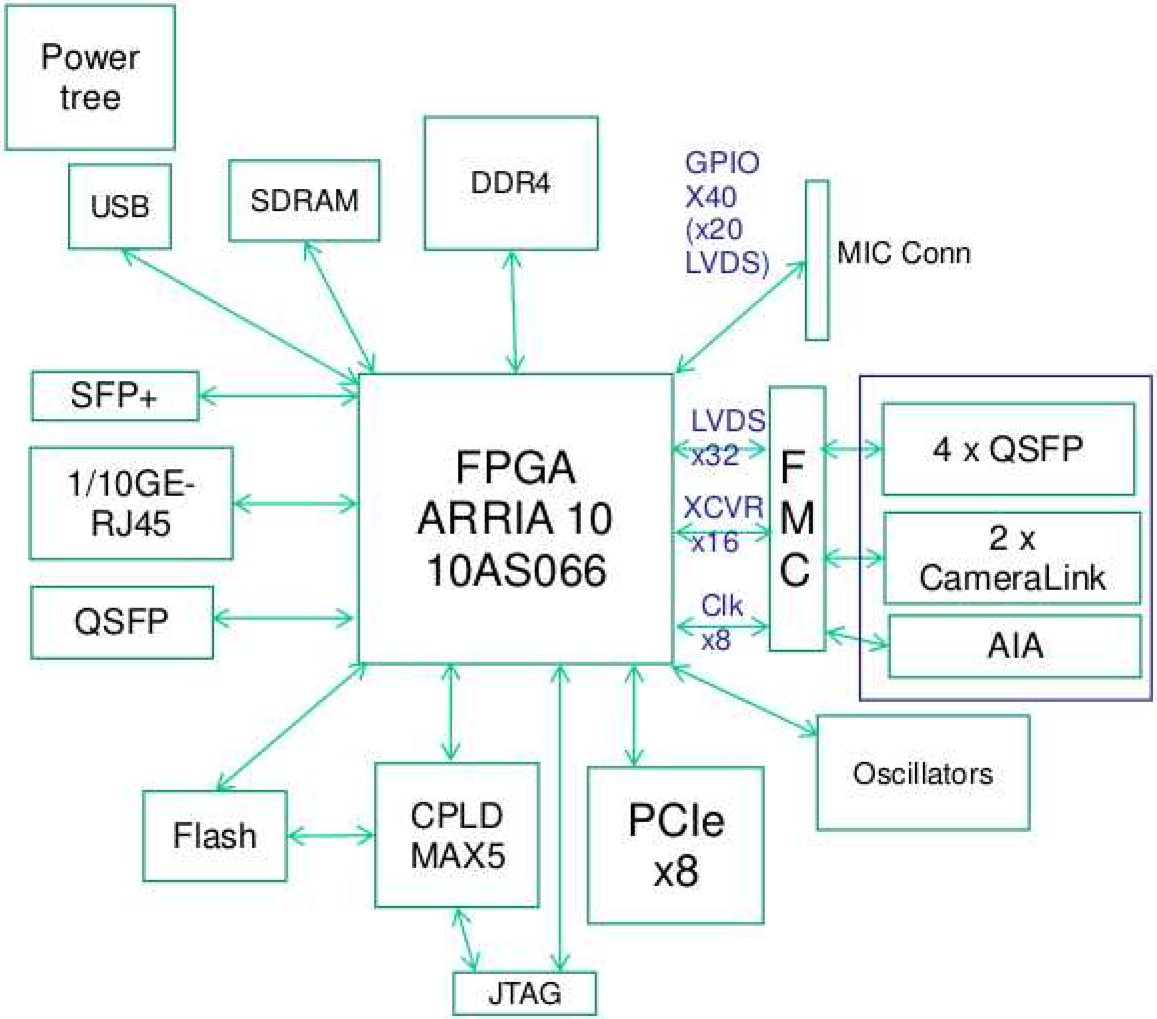
Based on ARRIA 10AX115:

- 1518 DSP blocks
- 6.6MB int. RAM
- 96 XCVR

Board features:

- Optimized for **heavy deterministic computation** in floating-point
- **Large Bandwidth between HMC and FPGA** - 4 links 16 lanes/link up to 15Gbps/lane = 120GB/s bidirectional
- Extremely **low jitter**
- More **power efficient** compared to GPUs
- Offers a lot of different interfaces on board or via the FMC connector and extension cards

FPGA solution : µXlink



ARRIA 10AS066 SoC:

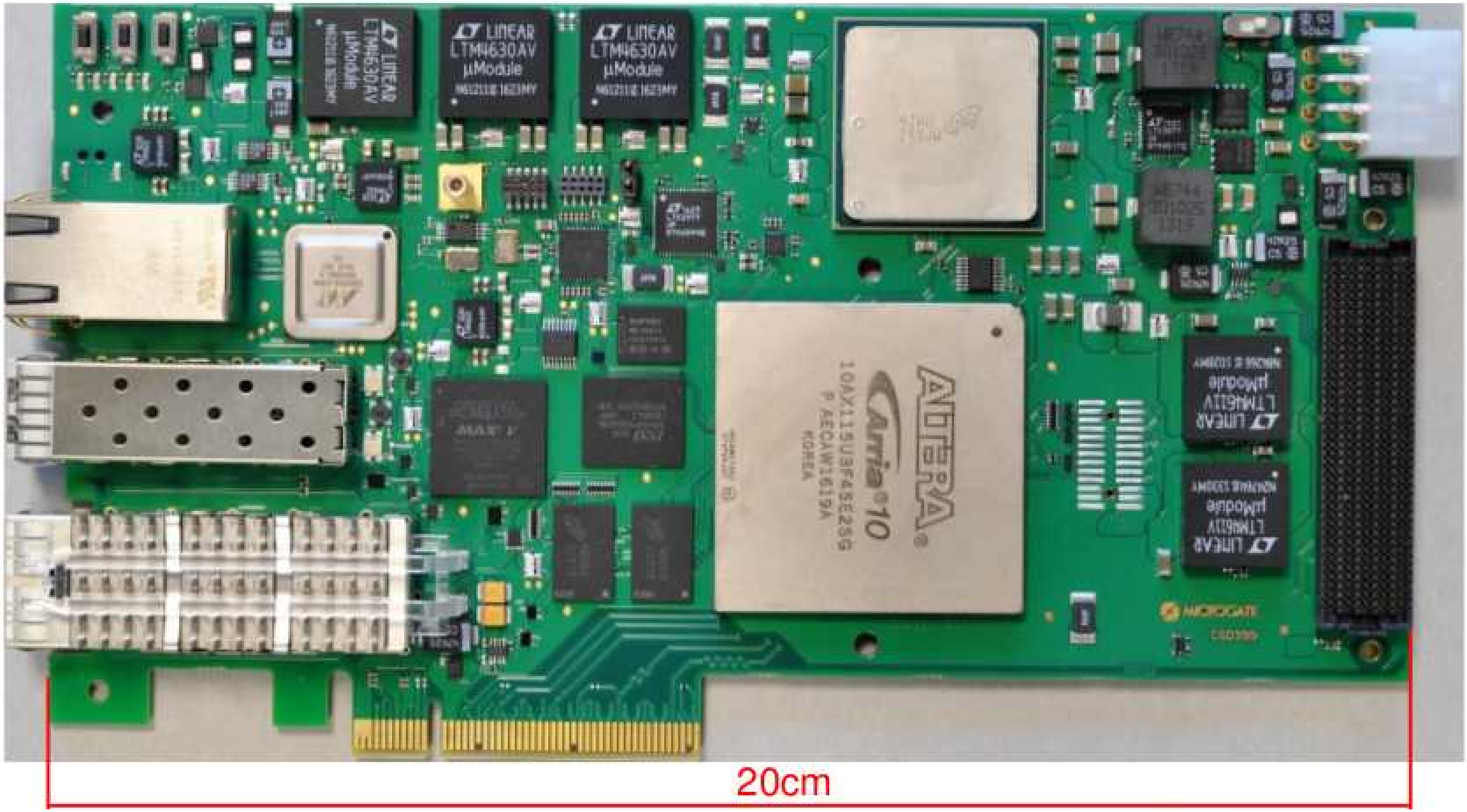
- 1.5GHz ARM dual-core Cortex-A9 on-chip processor
- 1855 DSP blocks
- 5.2MB int. RAM
- max. 48 XCVR

Board features:

- ARM **embedded processor** for **stand-alone** real-time box
- Powerful **PCIe root port** because of ARM and OS
- Management of accelerator cards on the PCIe interface
- Running control software using a full OS (e.g. Linux)
- Easy implementation of different communication protocols
- Offers a lot of different interfaces on board or via the FMC connector and extension cards

FPGA solutions: status

The first prototype of the two FPGA boards, the µXComp board is manufactured and is currently under test. After the validation of the interfaces and the communication between FPGA and HMC some more boards of this type will be produced and made available to the team.



Green Flash

Accelerators for real-time HPC

- Assess various HW accelerator options on a real-time application
  - GPU : lead by OdP with contribution from UoD
  - Xeon Phi : lead by UoD
  - FPGA : lead by UoD with contribution from OdP
- Assess performance of same hardware on complex data pipeline
  - Supervisor module for AO : lead by OdP
  - Criterion optimization and large matrix inversion

l'Observatoire de Paris LESIA Laboratoire d'Études Spatiales et d'Instrumentation en Astrophysique

Durham University

MICROGATE

PLDA

WP 4

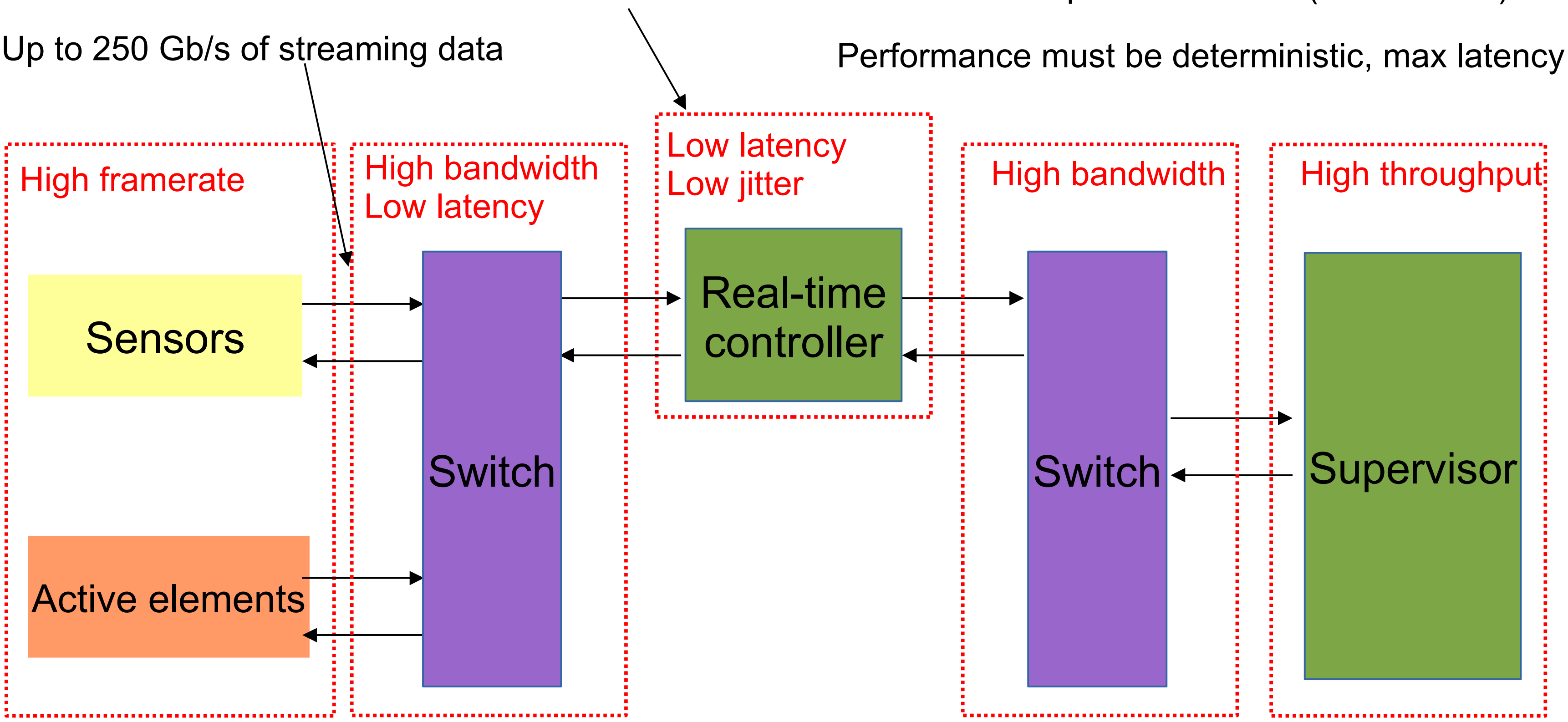
Real-time pipeline.
Includes sensors pixels streams processing and MVM for control of active elements

Pixel Streams processing, 2 options:
* tens of GFLOPS in simple arithmetics or
* hundreds of GFLOPS in batched Fourier Transform

MVM : up to 5 TFLOP/s (2.5 TMAC/s)

Performance must be deterministic, max latency : 2ms

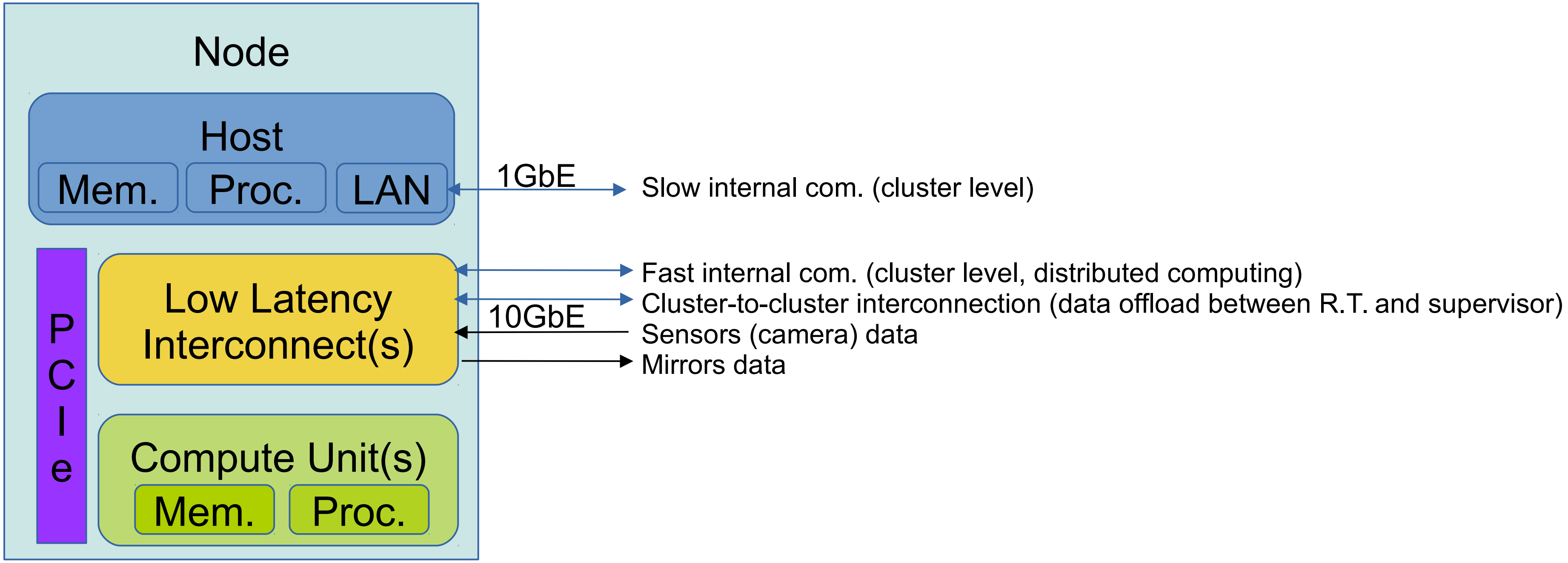
Up to 250 Gb/s of streaming data



Generic platform based on accelerators

One generic node architecture, two applications :
- Real-time memory bound linear algebra (AO linear control, a.k.a. real-time pipeline)
- High throughput compute bound linear algebra (AO supervisory tasks, a.k.a. supervisor)

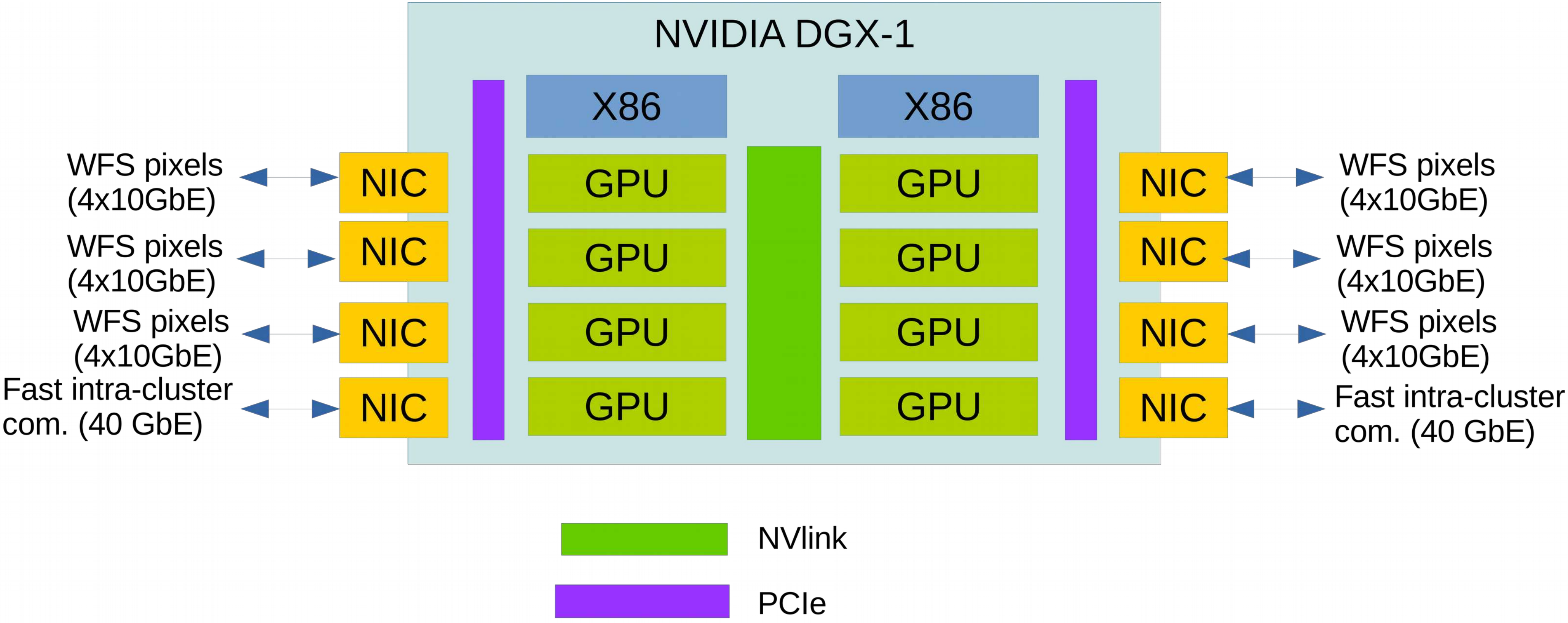
For each application, nodes are interconnected into a cluster.
For the full featured prototype, the two clusters are interconnected



Node
Host
Mem.
Proc.
LAN
1GbE
Slow internal com. (cluster level)
P C I e
Low Latency Interconnect(s)
10GbE
Fast internal com. (cluster level, distributed computing)
Cluster-to-cluster interconnection (data offload between R.T. and supervisor)
Sensors (camera) data
Mirrors data
Compute Unit(s)
Mem.
Proc.

RT data pipeline with GPUs

- Prototype using latest generation GPU cluster



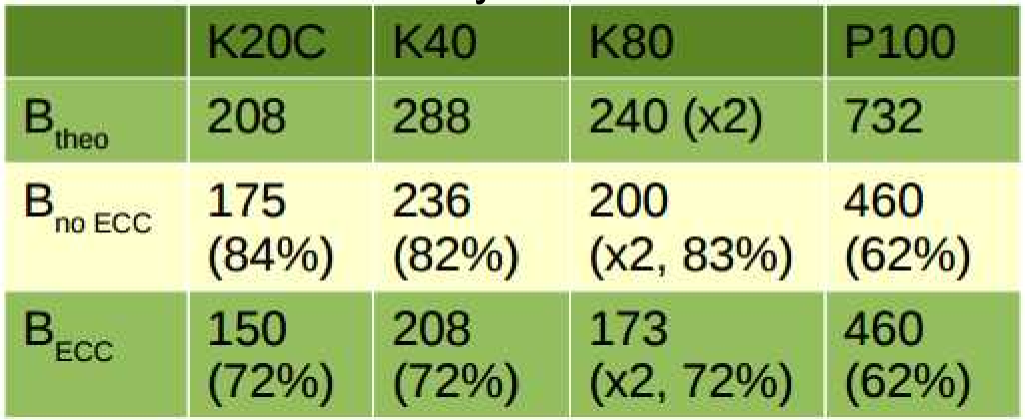
- Concept studied at LESIA

System dimensioning

MCAO @ E-ELT scale

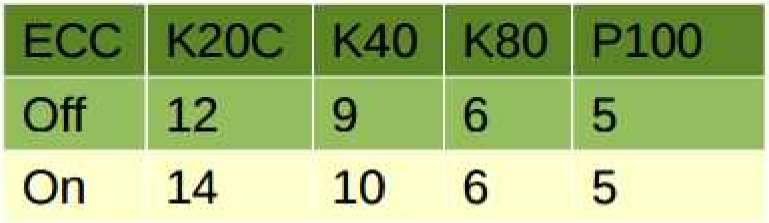
- POLC control scheme + LGS WFS : 2.5 TMAC/s with 250 Gb/s of streaming data
- Upper limit from instruments specification capture during PDR (actual first light instruments may require less)

Memory bandwidth



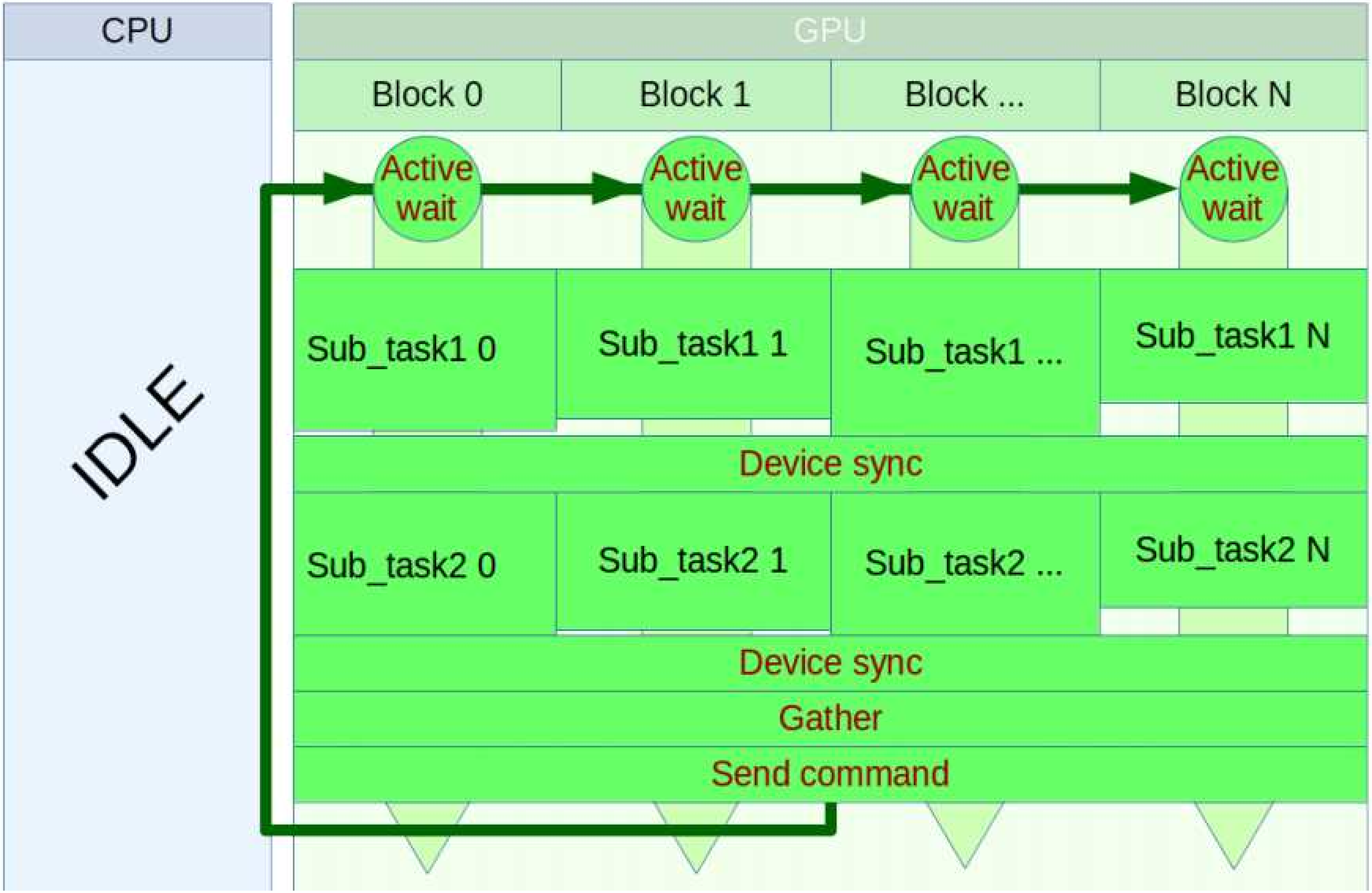
	K20C	K40	K80	P100
B_{theo}	208	288	240 (x2)	732
$B_{no\ ECC}$	175 (84%)	236 (82%)	200 (x2, 83%)	460 (62%)
B_{ECC}	150 (72%)	208 (72%)	173 (x2, 72%)	460 (62%)

Number of GPUs required

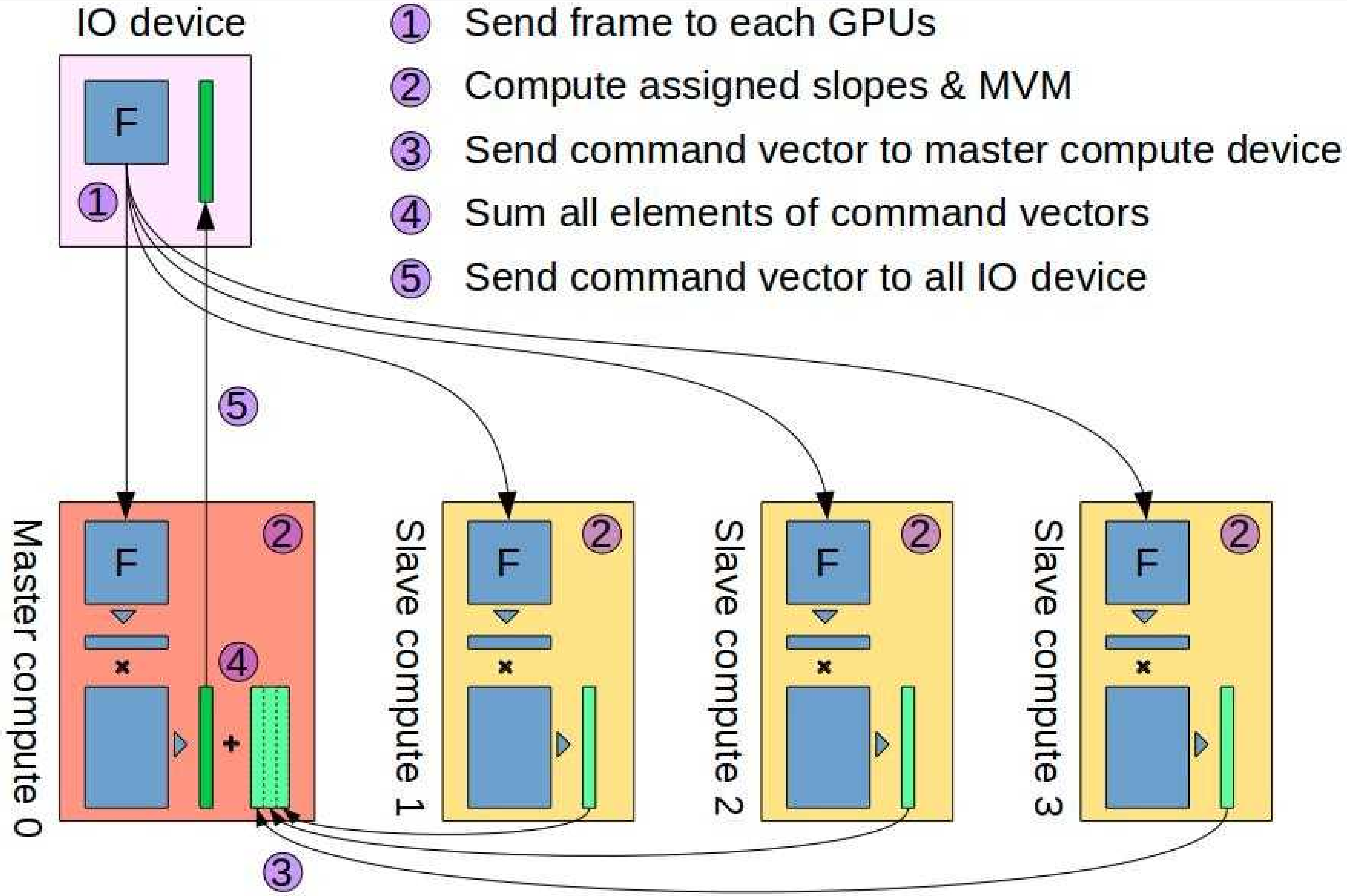


ECC	K20C	K40	K80	P100
Off	12	9	6	5
On	14	10	6	5

Persistent kernel implementation

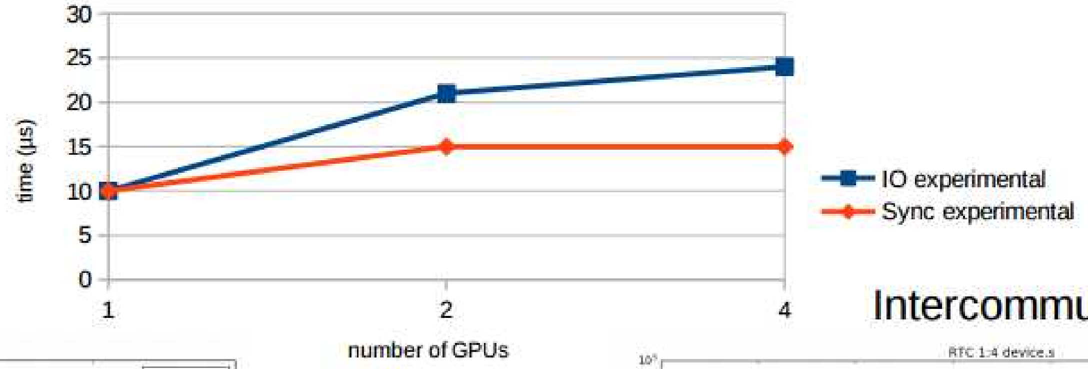


Multi-GPU prototype

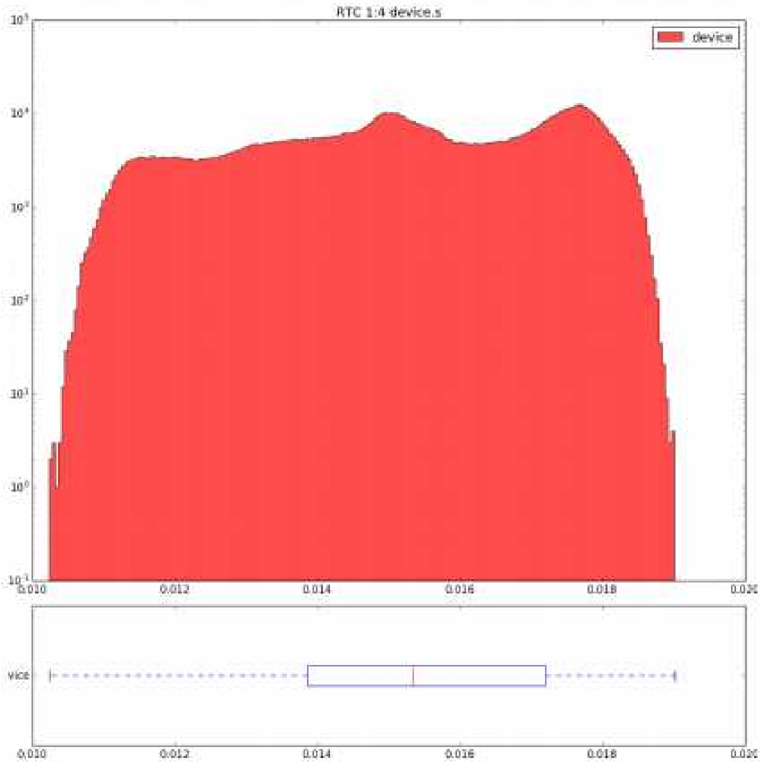


1. Send frame to each GPUs
2. Compute assigned slopes & MVM
3. Send command vector to master compute device
4. Sum all elements of command vectors
5. Send command vector to all IO device

Persistent kernel implementation

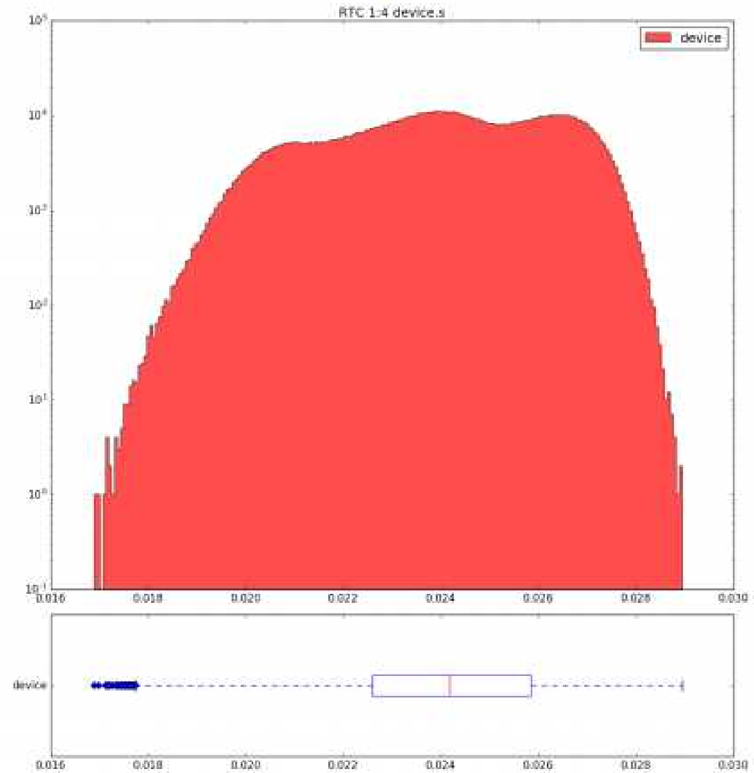


Synchronize jitter



Average : 15µs   Jitter : 8.8µs

Intercommunication jitter



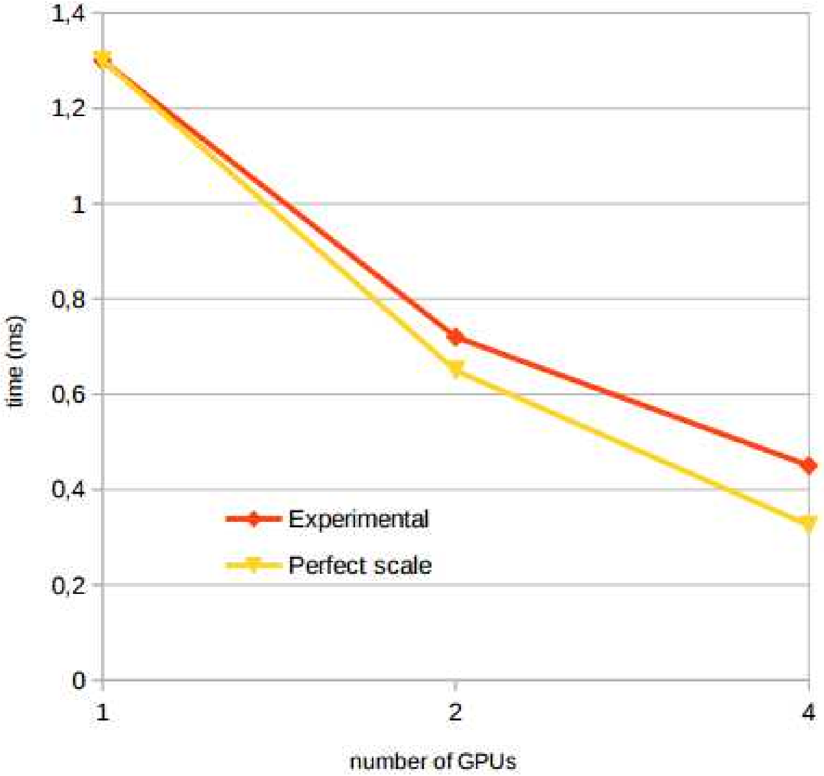
Average : 24µs   Jitter : 12µs

l'Observatoire de Paris LESIA
Laboratoire d'Études Spatiales et d'Instrumentation en Astrophysique
Durham University
MICROGATE
PLDA

Persistent kernel implementation

Strong scalability

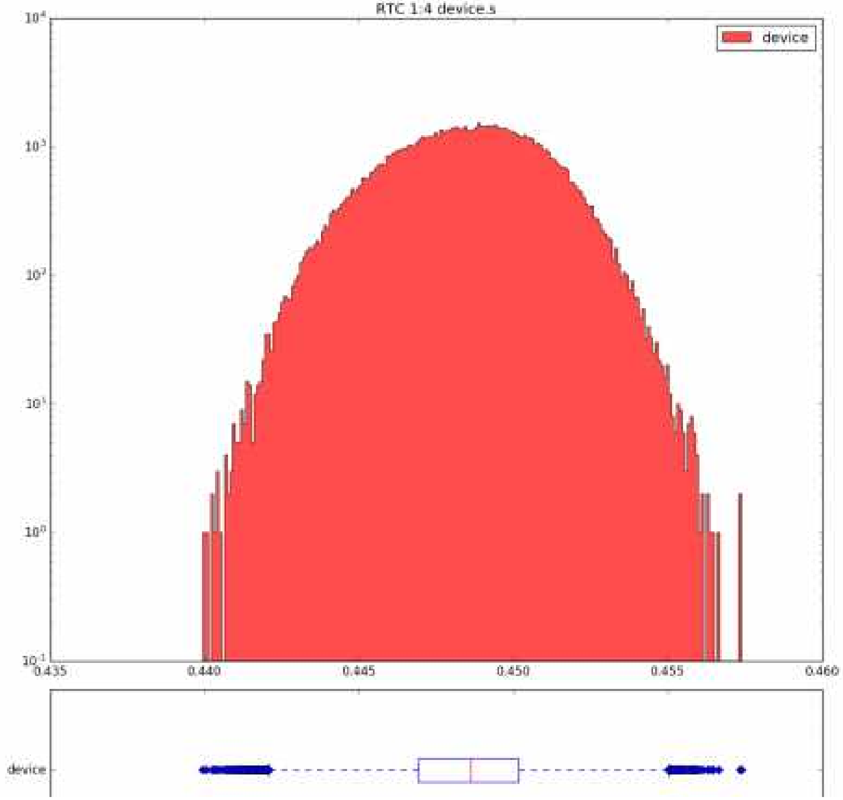
Constant case with **10,048 slopes x 15,000 commands**



Histogram

Case with **10,048 slopes x 15,000 commands** on 4 devices

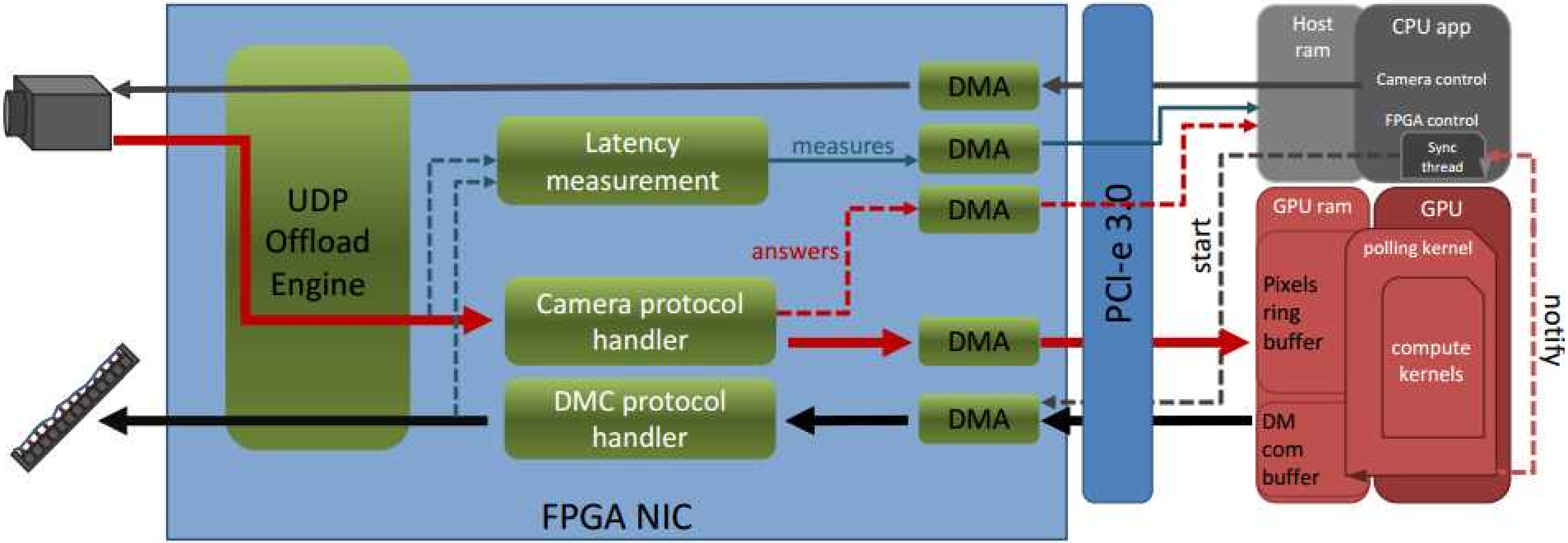
Average : 0.45ms   Jitter : 17µs



Data acquisition

FPGA writes/reads directly to/from GPU memory

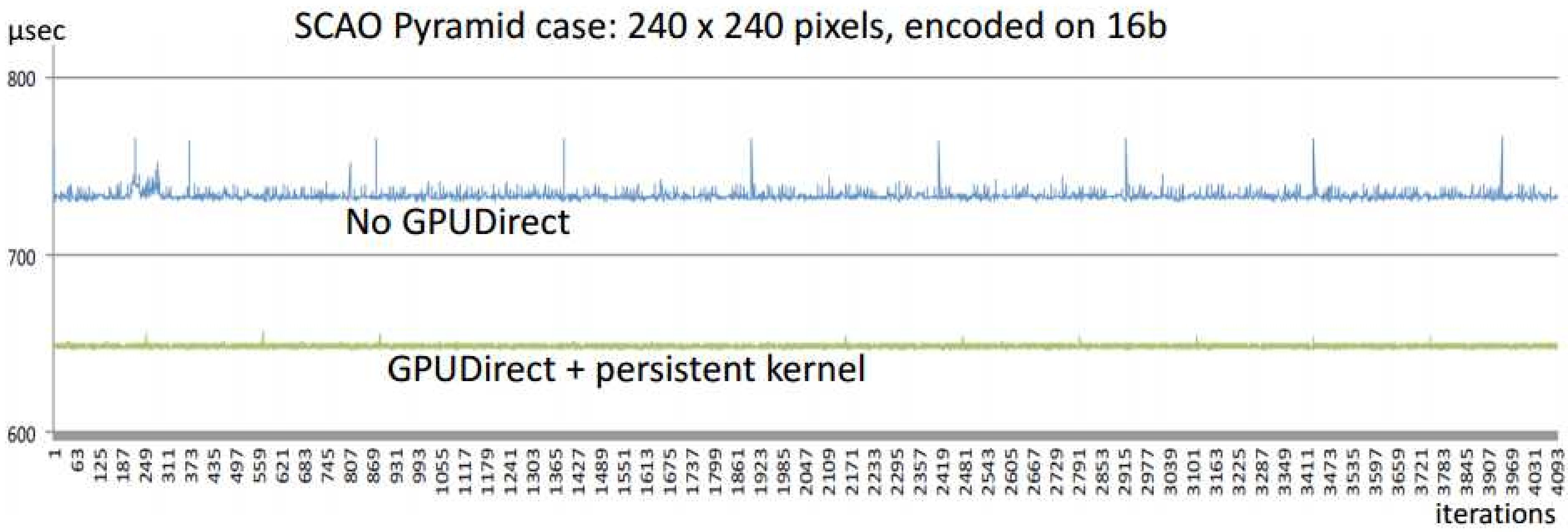
Using only writes would be better though



Data acquisition + persistent kernels

FPGA PLDA XPressG5
GPU Tesla C2070
OS Debian wheezy

Camera EVT HS-2000M
10GbE network



SCAO Pyramid case: 240 x 240 pixels, encoded on 16b

l'Observatoire de Paris LESIA
Laboratoire d'Études Spatiales et d'Instrumentation en Astrophysique

Durham University

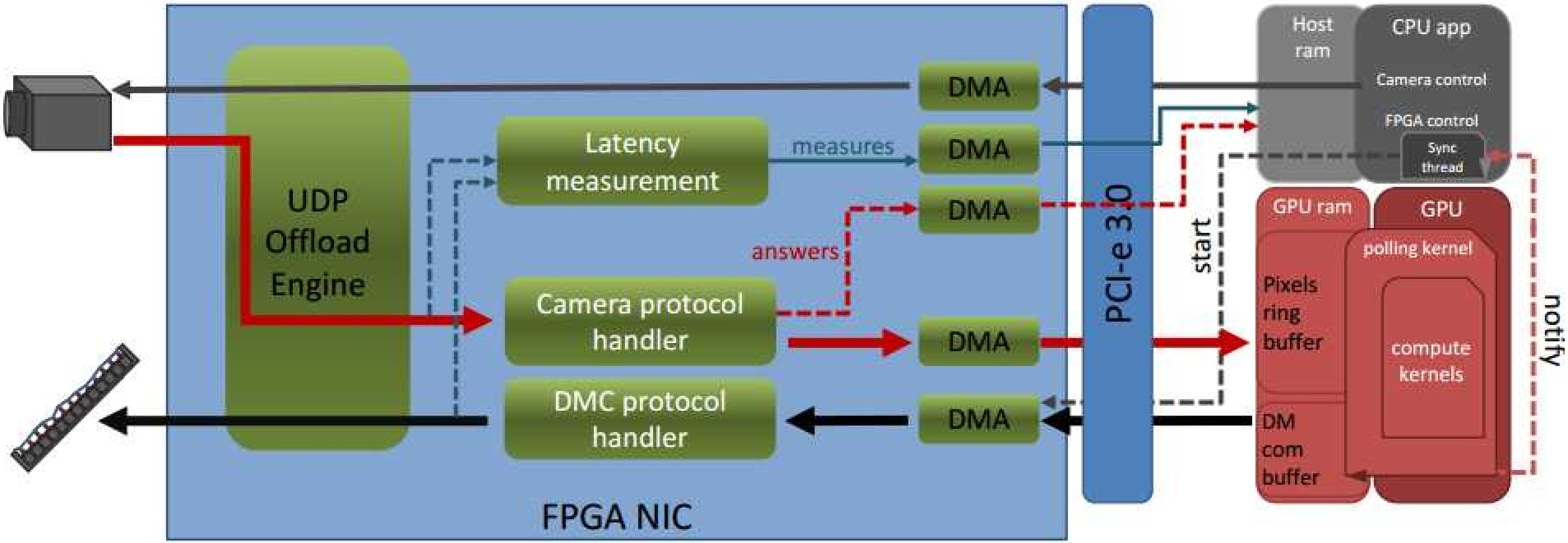
MICROGATE

PLDA

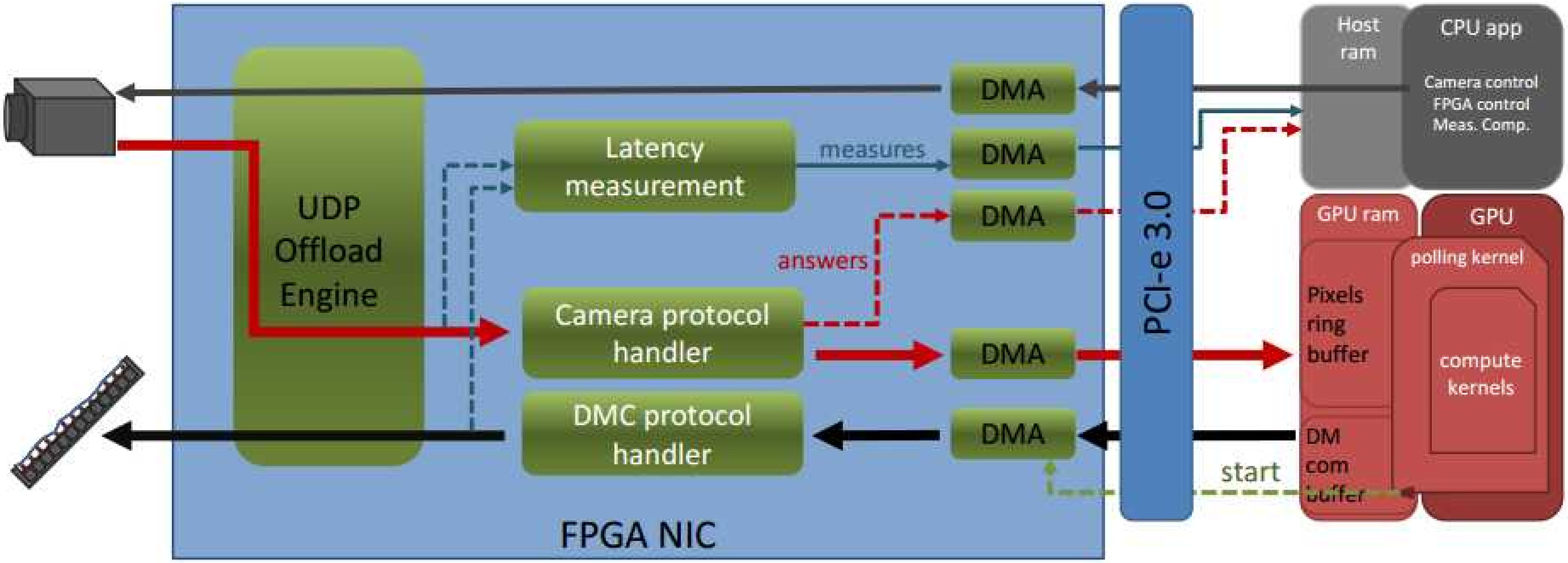
Data acquisition

FPGA writes/reads directly to/from GPU memory

Using only writes would be better though



FPGA/GPU optimized sync.



Little to no improvements, but CPU free for other kind of computations

WP 4

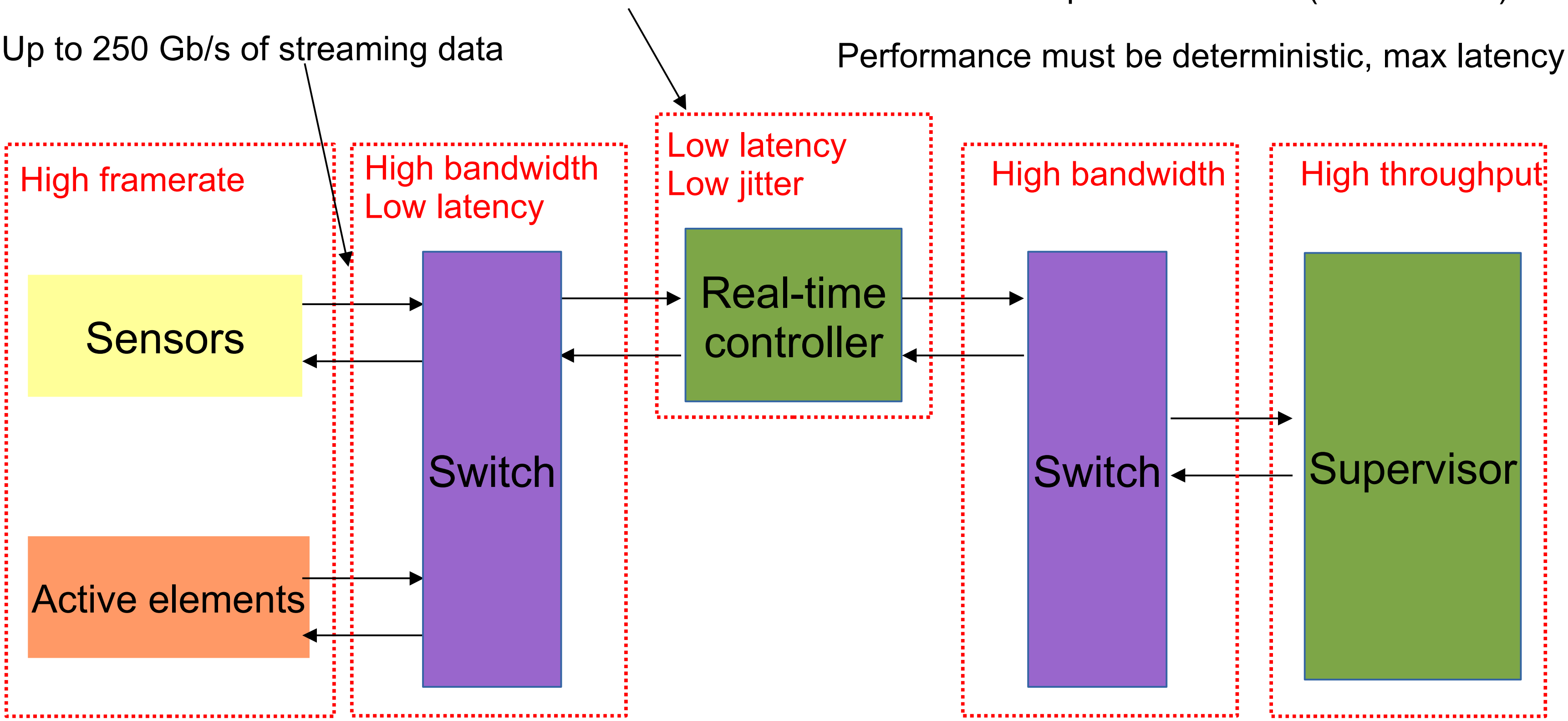
Real-time pipeline.
Includes sensors pixels streams processing and MVM for control of active elements

Pixel Streams processing, 2 options:
* tens of GFLOPS in simple arithmetics or
* hundreds of GFLOPS in batched Fourier Transform

MVM : up to 5 TFLOP/s (2.5 TMAC/s)

Up to 250 Gb/s of streaming data

Performance must be deterministic, max latency : 2ms



WP 4

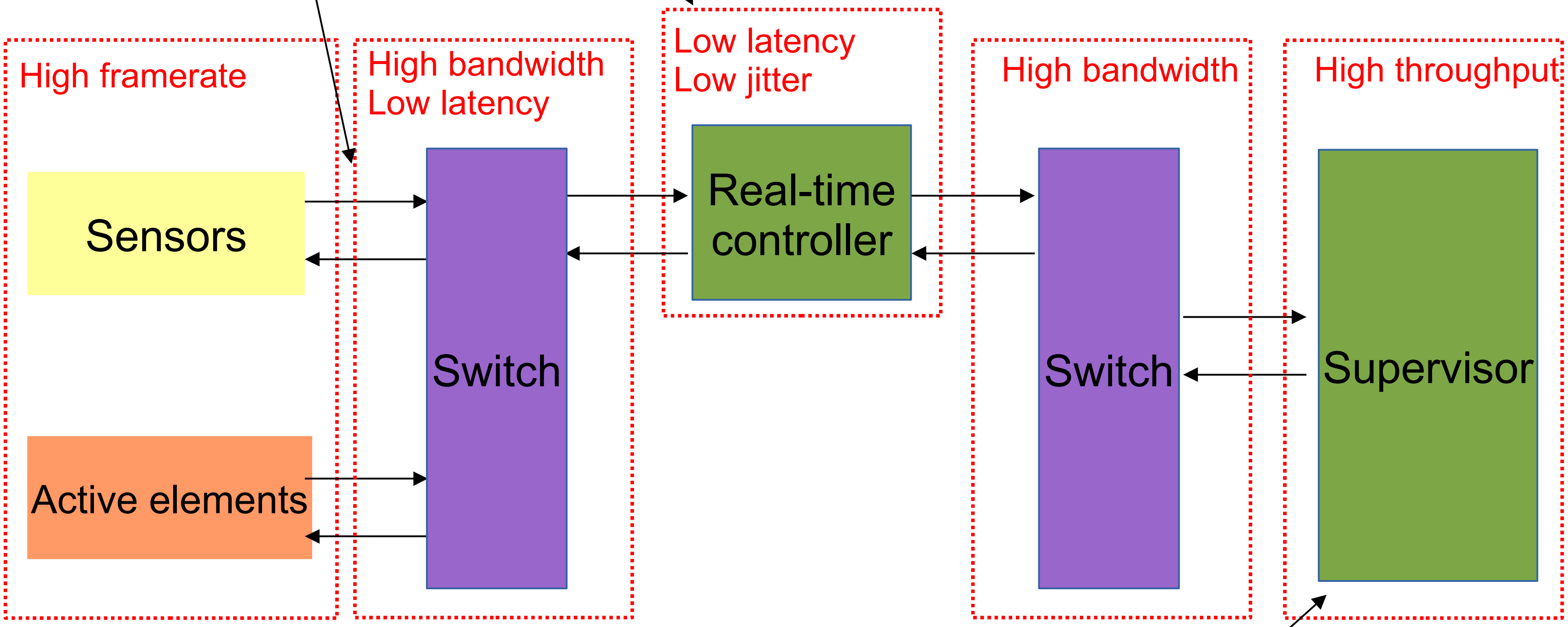
Real-time pipeline.
Includes sensors pixels streams processing and MVM for control of active elements

Pixel Streams processing, 2 options:
* tens of GFLOPS in simple arithmetics or
* hundreds of GFLOPS in batched Fourier Transform

MVM : up to 5 TFLOP/s (2.5 TMAC/s)

Up to 250 Gb/s of streaming data

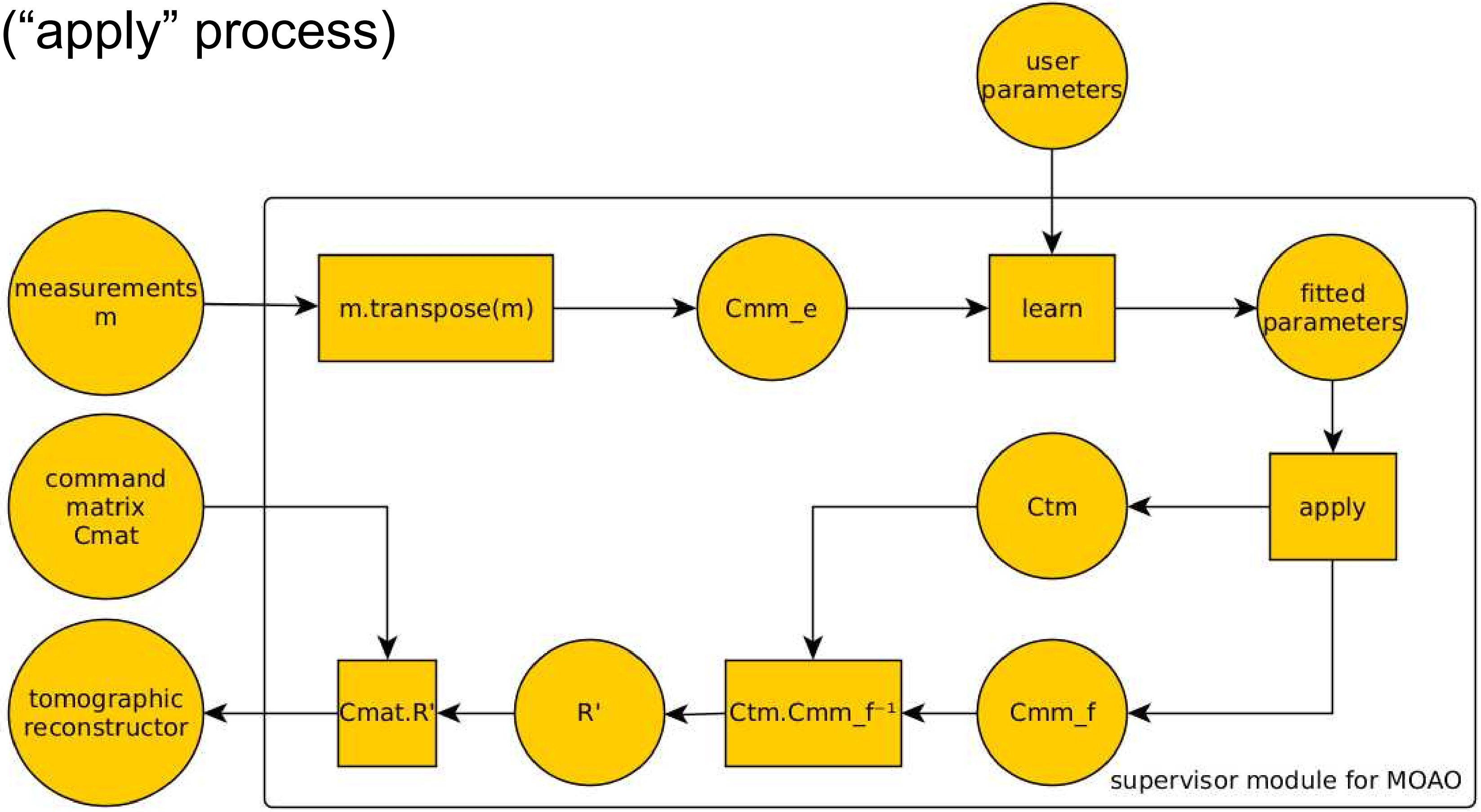
Performance must be deterministic, max latency : 2ms



Supervisory module. Use the output data stream from RT pipeline to re-optimize the control matrix
2 stages : function optimization (gradient descent) and
Choleski inversion : up to 100 TFLOP/s

Loop supervision module

Mix of cost function optimization for parameters identification ("Learn" process) and linear algebra for reconstructor matrix computation ("apply" process)



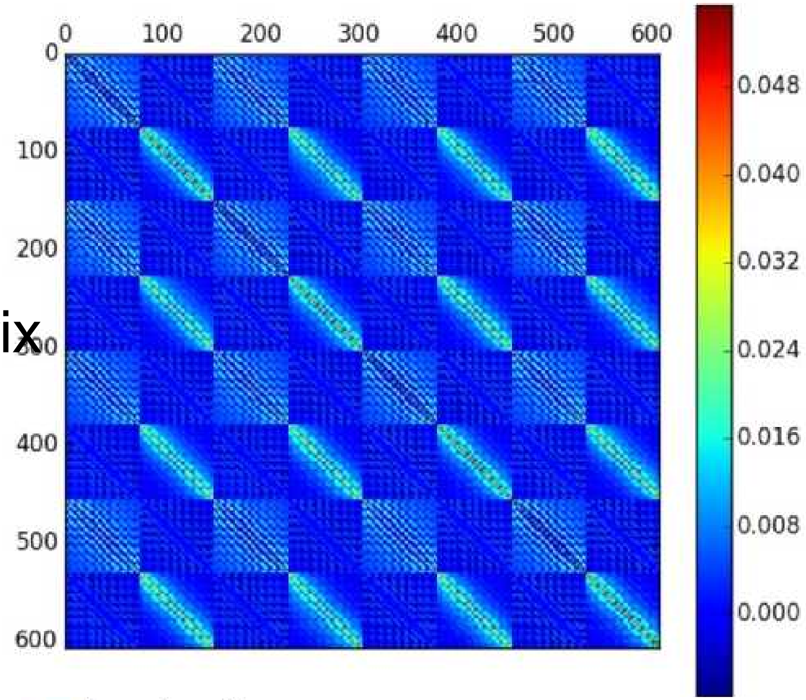
Loop supervision module

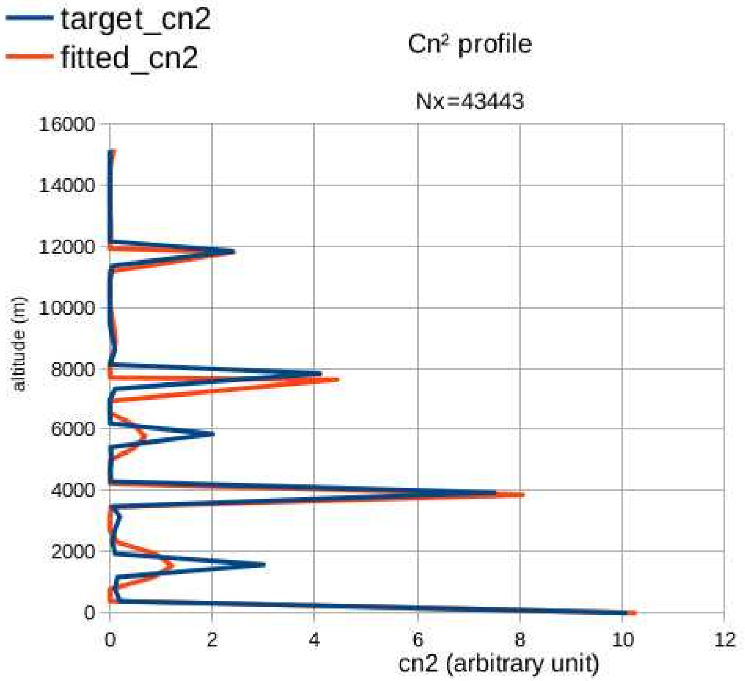
Parameters identification ("Learn" process)

- Fitting measurements covariance matrix on a model including system and turbulence parameters
- Using a score function

$$F(x)=\sum_{k=1}^{N^2}\left[Cmm_k-f_k(x)\right]^2$$

- Levenberg-Marquardt algorithm for function optimization
- Exemple of turbulence profile reconstruction
- Dual stage process (5 layers + 40 layer





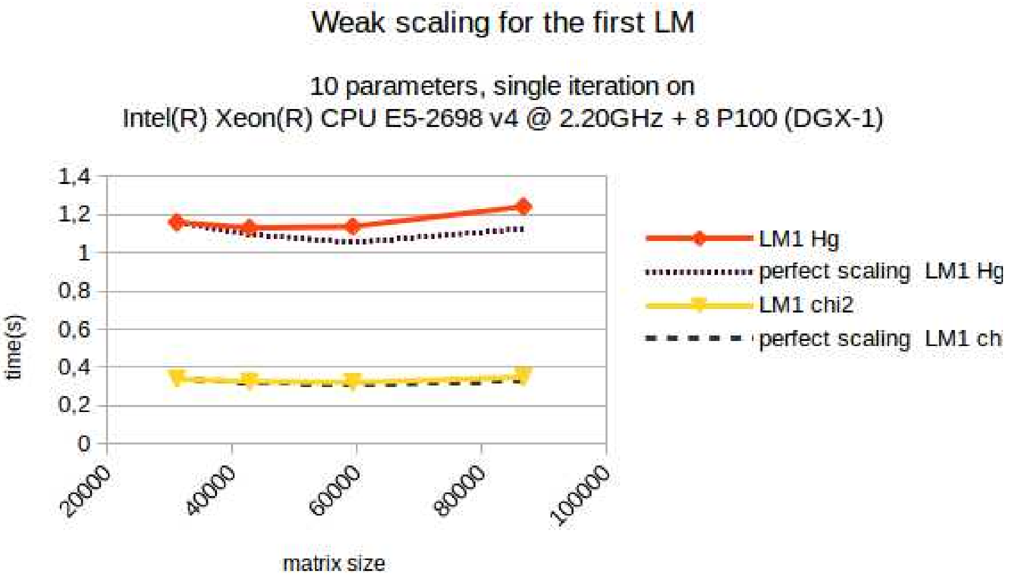
Loop supervision module

Performance for parameters identification (“Learn” process)

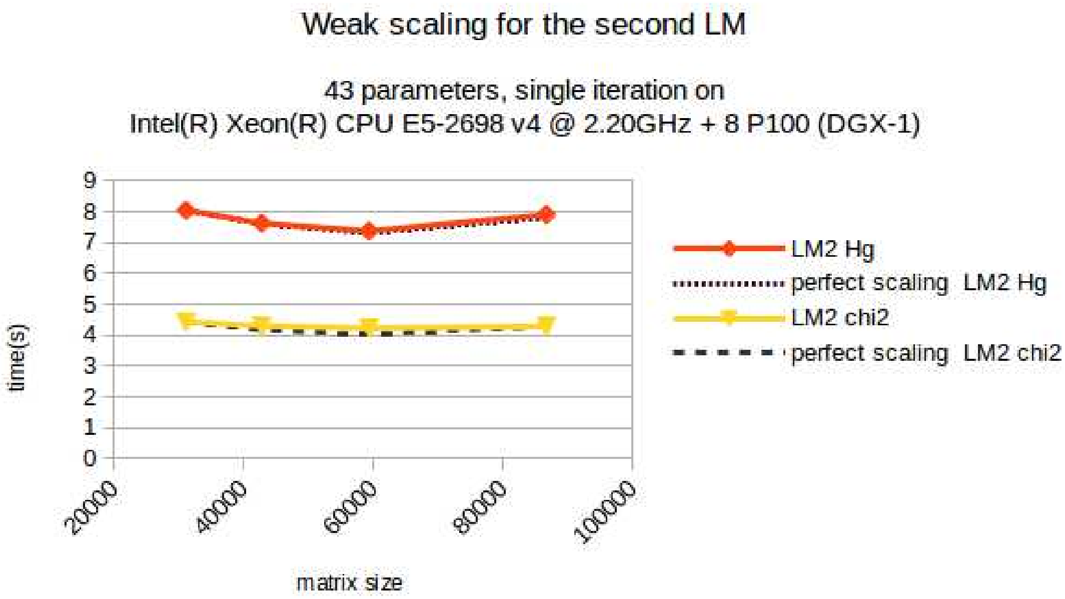
Multi-GPU process, including matrix generation and LM fit

Time to solution for a matrix size of 86k :240s (4 minutes)

- first pass (5 layers) : 25s
- Second pass (40 layers) : 213s


Weak scaling for the first LM

10 parameters, single iteration on
Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz + 8 P100 (DGX-1)


Weak scaling for the second LM

43 parameters, single iteration on
Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz + 8 P100 (DGX-1)

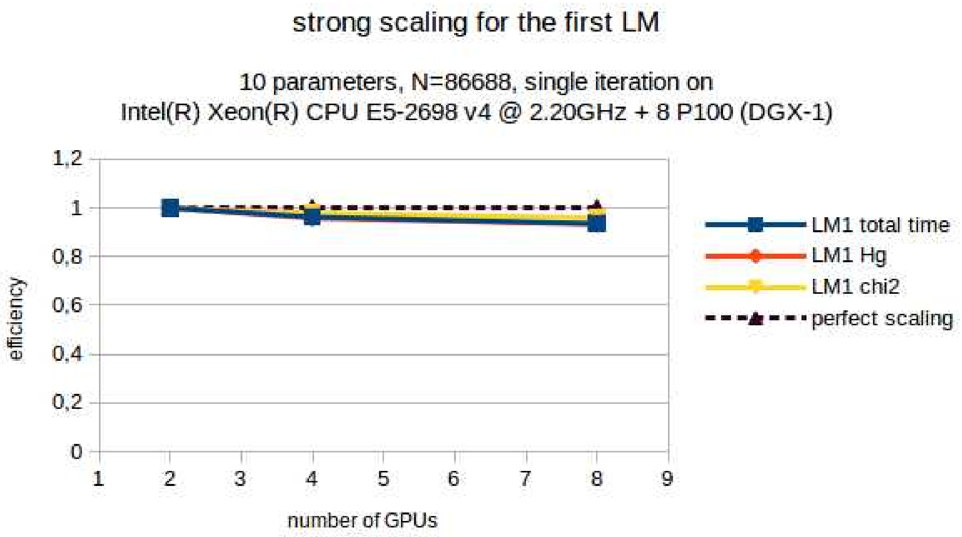
Loop supervision module

Performance for parameters identification (“Learn” process)

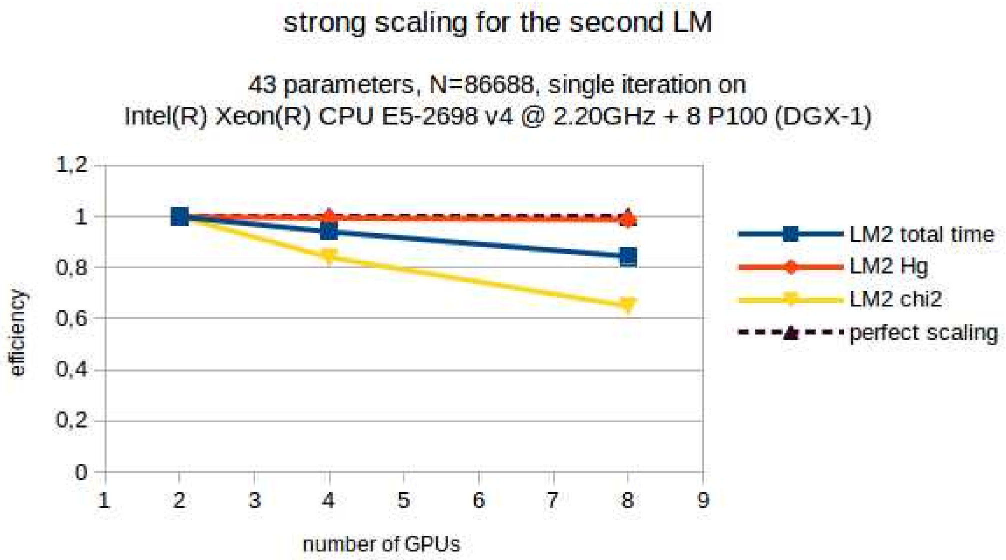
Multi-GPU process, including matrix generation and LM fit

Time to solution for a matrix size of 86k :

- first pass (5 layers) : 25sec
- Second pass (40 layers) : 213sec


strong scaling for the first LM

10 parameters, N=86688, single iteration on
Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz + 8 P100 (DGX-1)


strong scaling for the second LM

43 parameters, N=86688, single iteration on
Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz + 8 P100 (DGX-1)

Loop supervision module

Reconstructor matrix computation (“apply” process)

- Compute the tomographic reconstructor matrix using covarince matrix between “truth” sensor and other WFS and invert of measurements covariance matrix

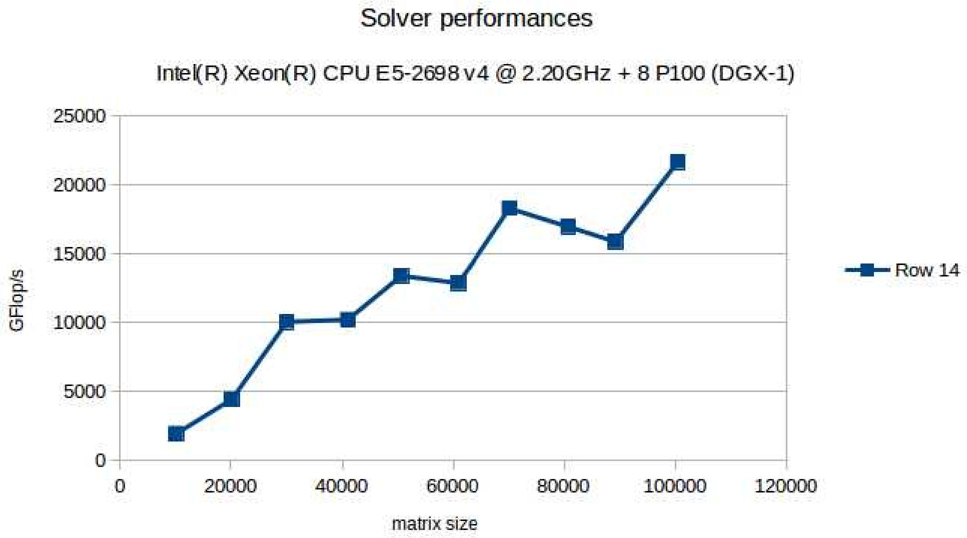
$$R' = Ctm \cdot Cmm_f^{-1}$$

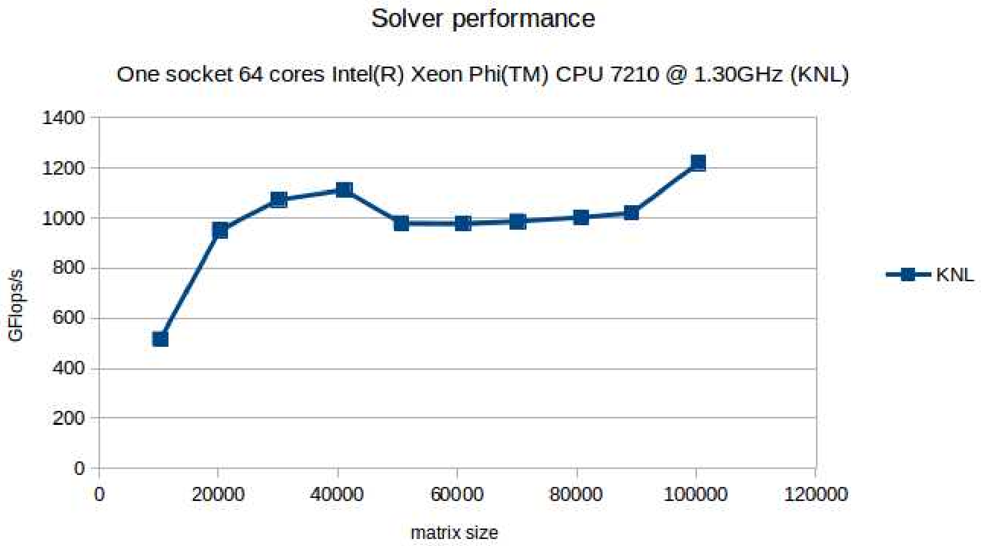
- Can use various methods. “Brute” force : direct solver
- Standard Lapack routine : “posv” : mostly compute-bound, high level of scalability
- Highly portable code : explore various architectures by using standard vendor provided maths libraries

Loop supervision module

Performance for reconstructor matrix computation (“apply” process)

Comparing last generation of GPU (NVIDIA P100) and last generation of Intel Xeon Phi (KNL)



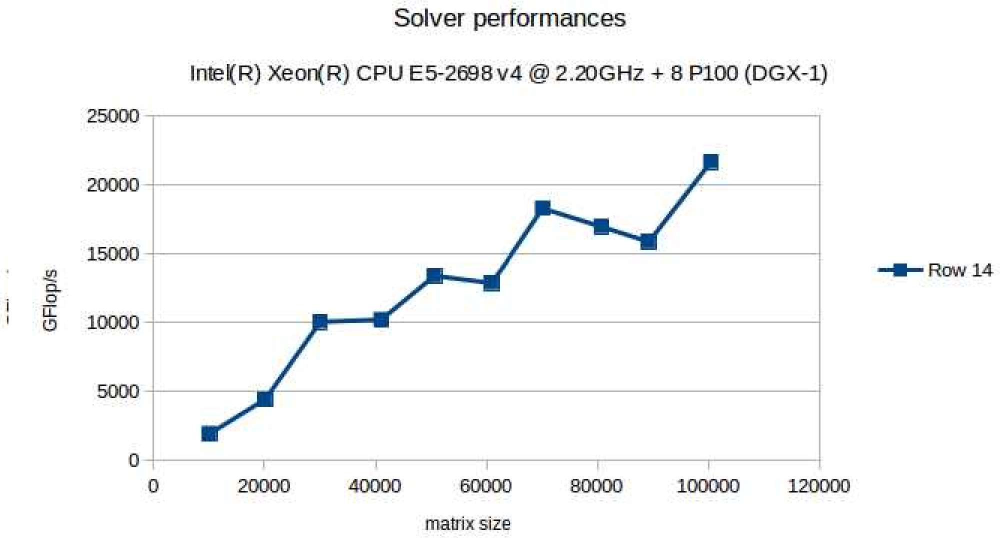


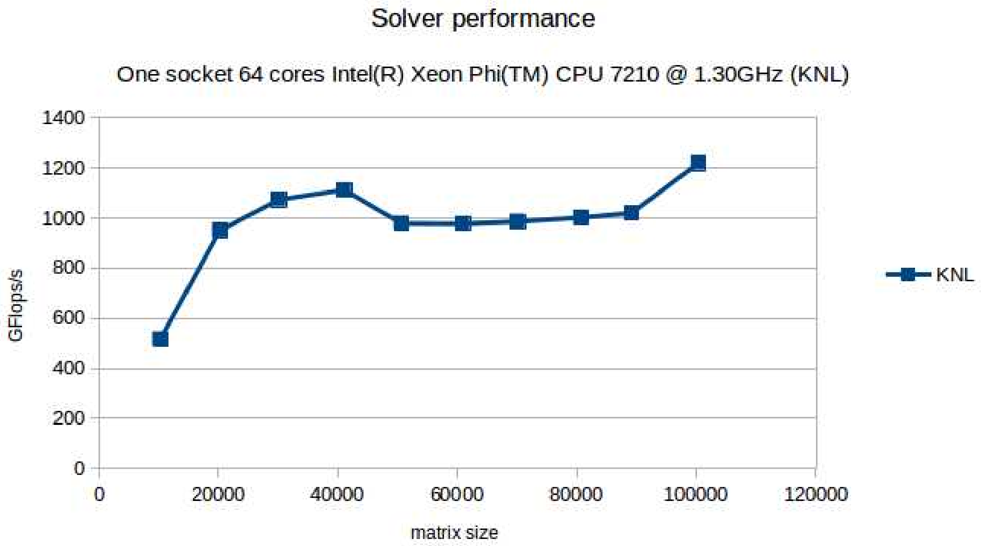
8 GPUs together reach more than 21 TFLOP/s while a single KNL can only reach about 1.2 TFLOP/s in peak performance

Loop supervision module

Performance for reconstructor matrix computation (“apply” process)

Comparing last generation of GPU (NVIDIA P100) and last generation of Intel Xeon Phi (KNL)



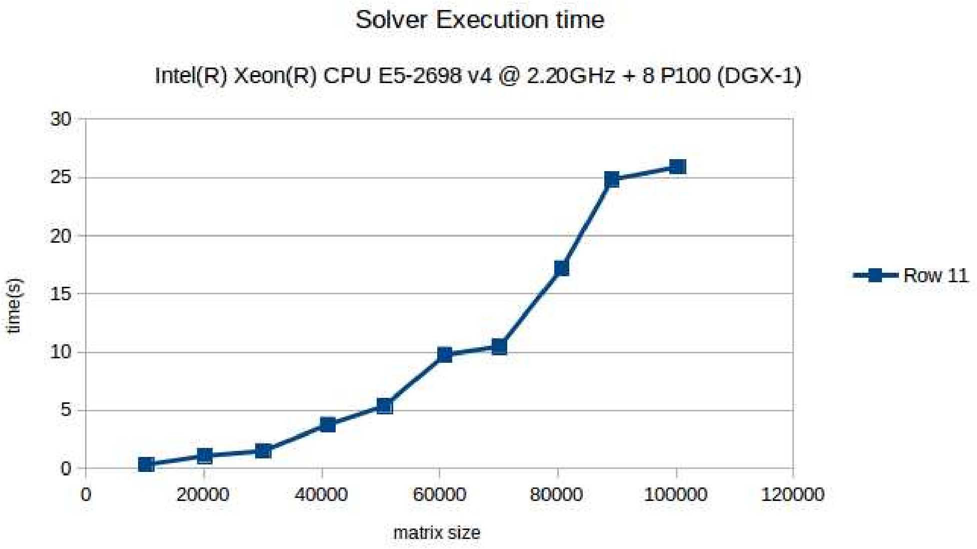


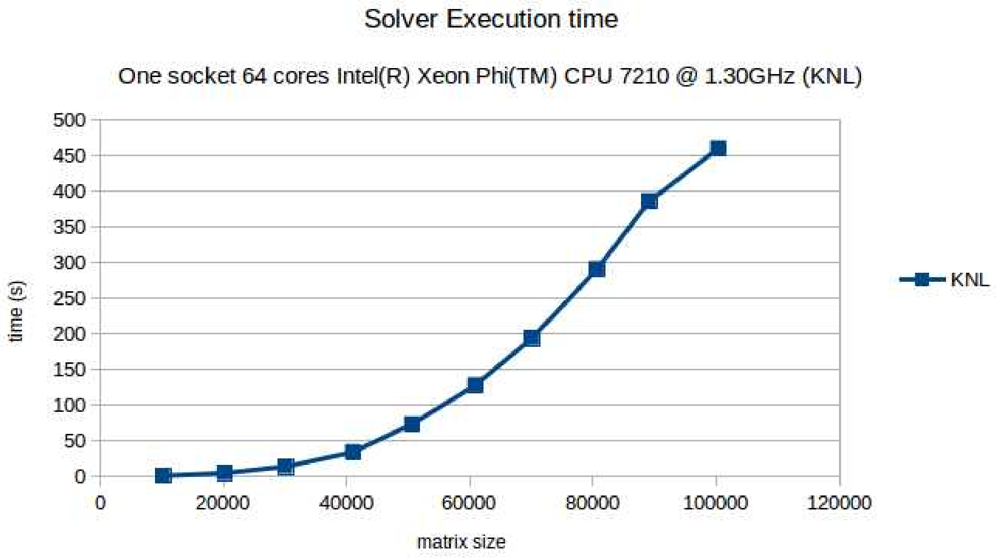
GPUs can deliver better peak perf. (saturation not reached, expect >2.5 or more) and the NVlink interconnect seems to perform very well

Loop supervision module

Performance for reconstructor matrix computation ("apply" process)

- Comparing last generation of GPU (NVIDIA P100) and last generation of Intel Xeon Phi (KNL)





- Record time-to-solution on DGX-1 : MAORY / HARMONI full scale (100k x 100k matrix) : 25sec to compute tomographic reconstructor

RT pipeline with Xeon Phi

A many-core CPU (64-72 cores, depending on model)

A standard CPU - self hosted and self booting

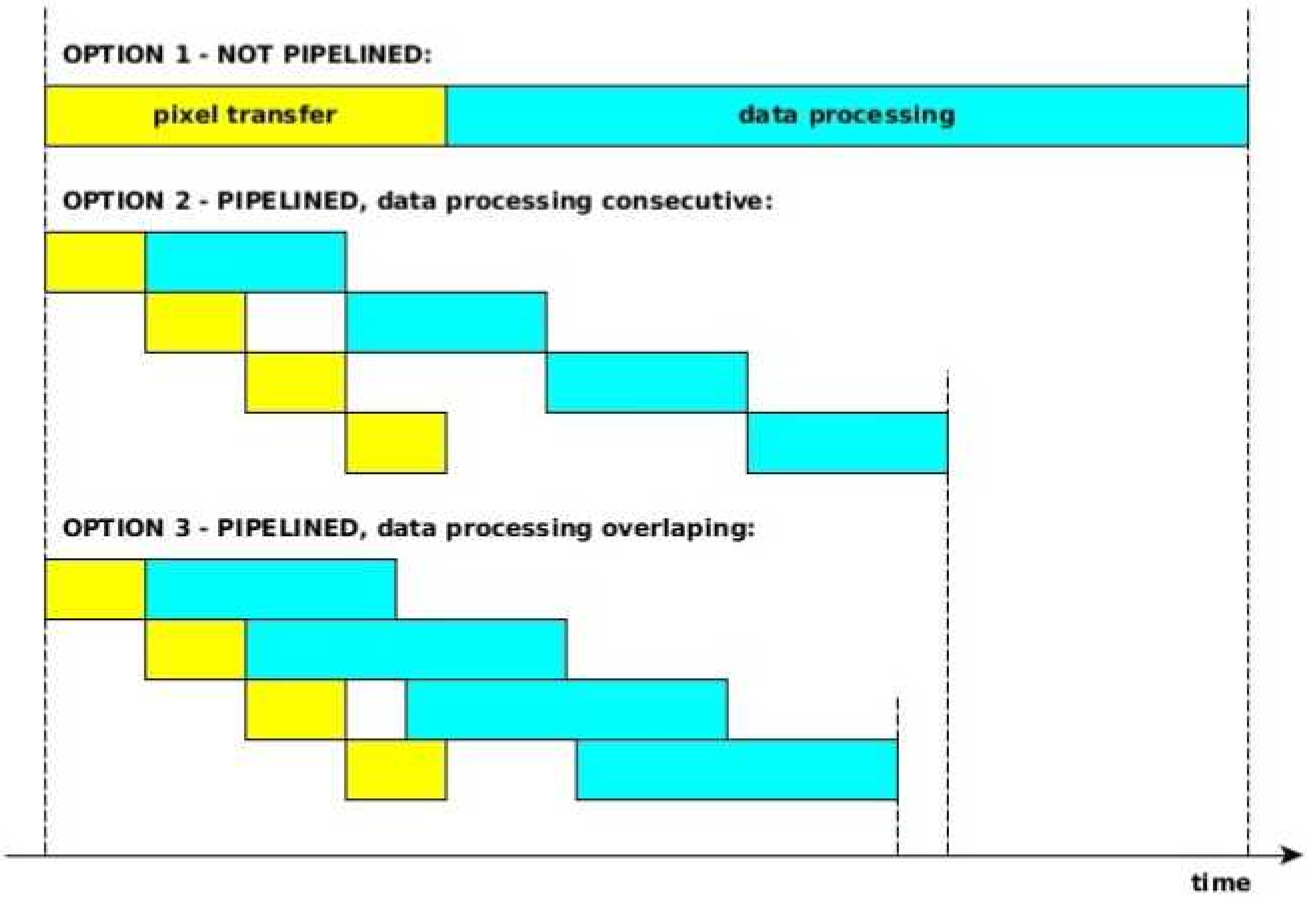
- Runs standard Linux, standard compiler tools
  - We currently use Centos
- Not an accelerator (in accelerators work package for historical reasons)
  - Though an accelerator version will be made available

A key design decision (important for AO RTC): 16GB High bandwidth memory

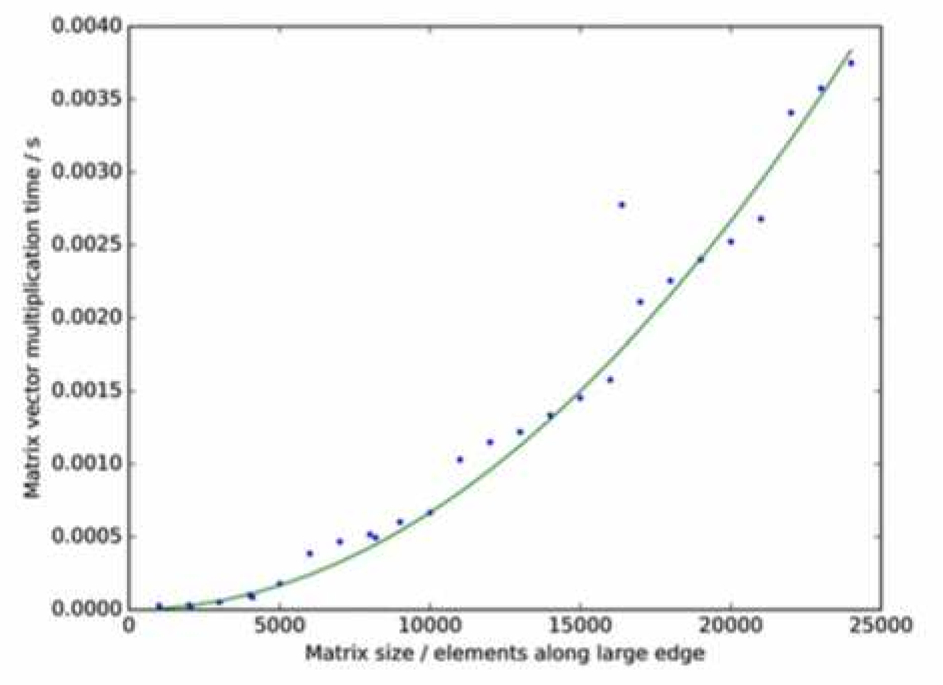
- 320GB/s theoretical, xxx GB/s measured at Durham

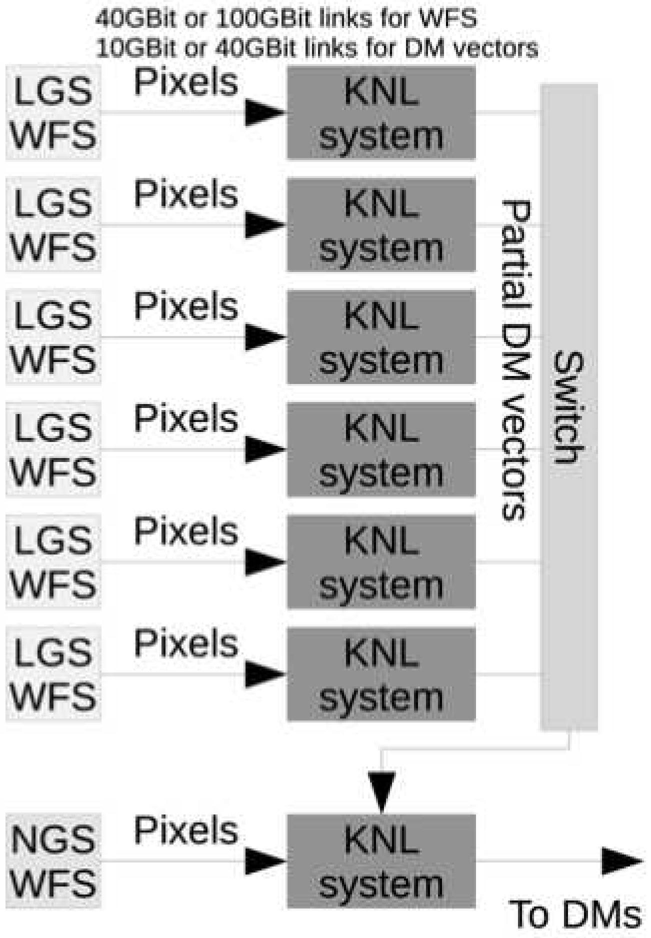
Wide vectorisation unit: 16 floats operated on simultaneously in each core

RT pipeline with Xeon Phi


OPTION 1 - NOT PIPELINED:
pixel transfer
data processing
OPTION 2 - PIPELINED, data processing consecutive:
OPTION 3 - PIPELINED, data processing overlaping:
time

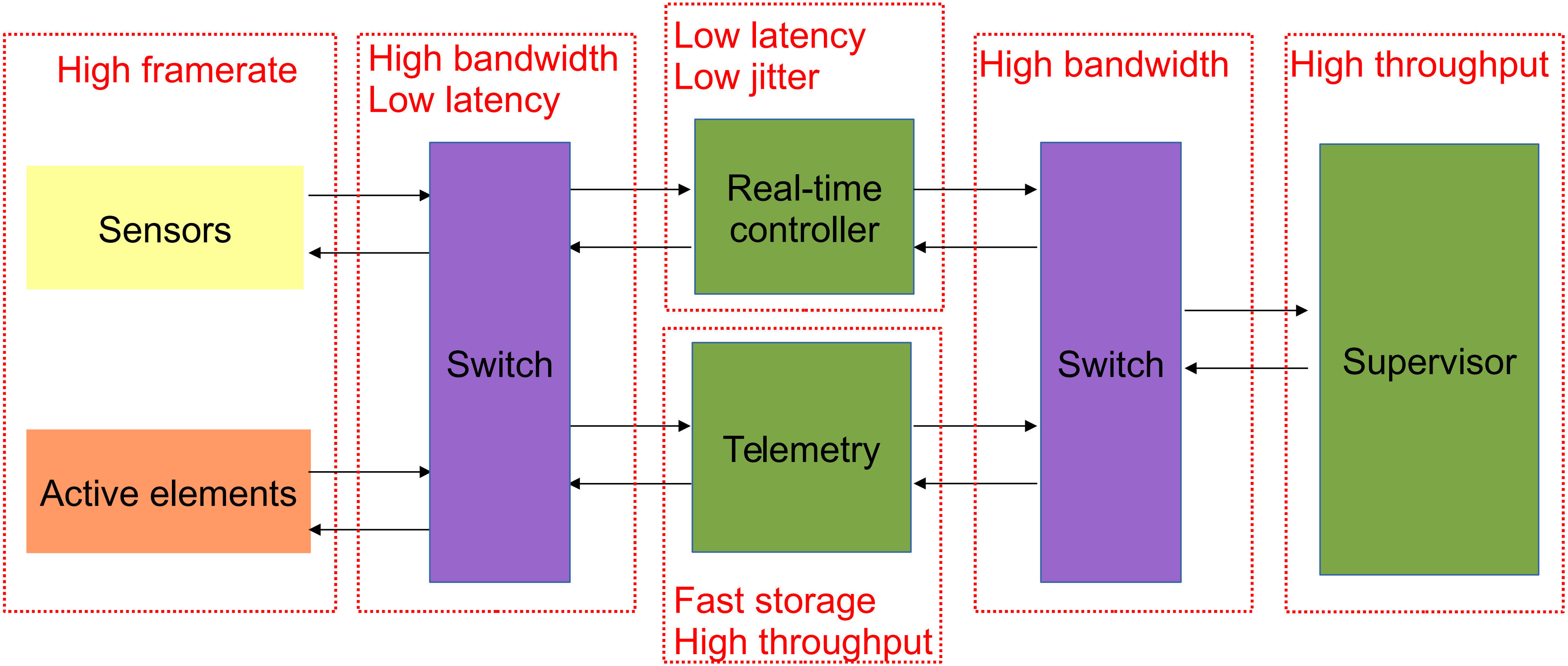
RT pipeline with Xeon Phi





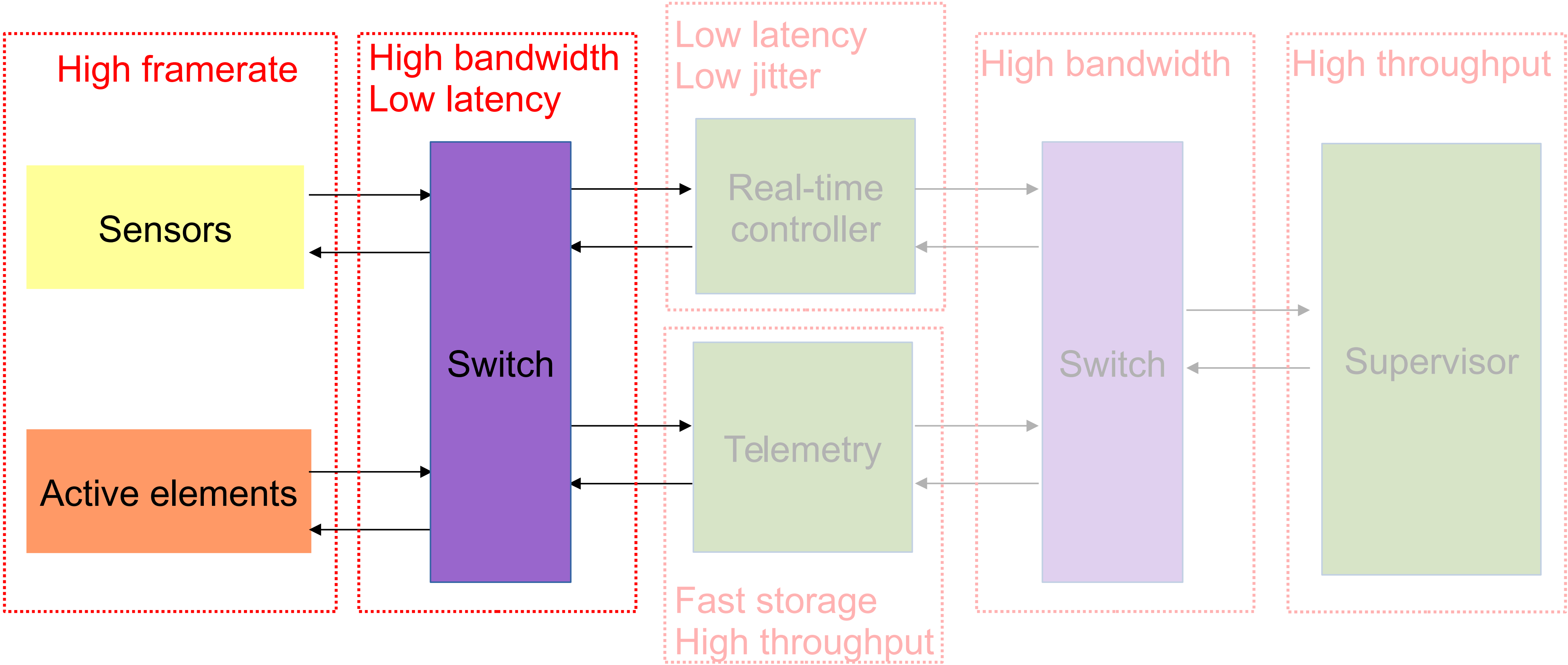
40GBit or 100GBit links for WFS
10GBit or 40GBit links for DM vectors

AO RTC concept

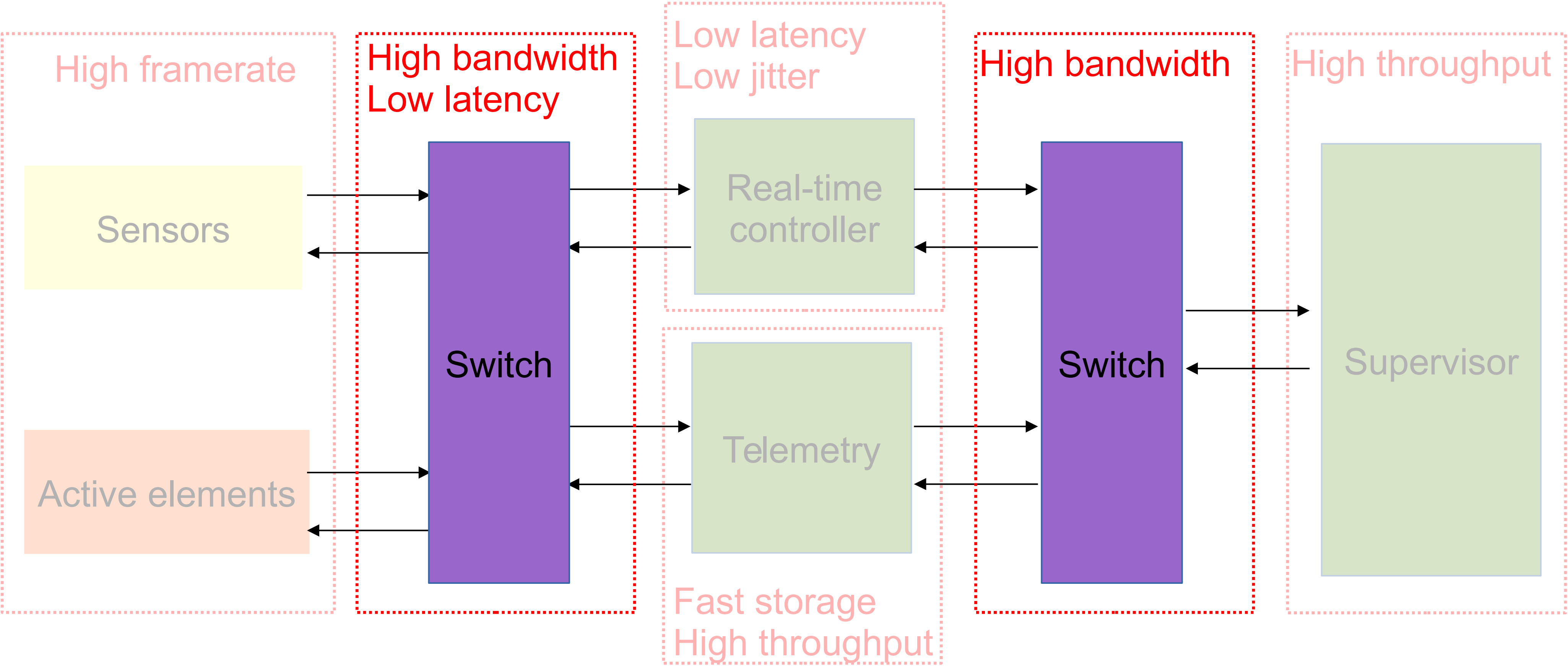


01/22/2016

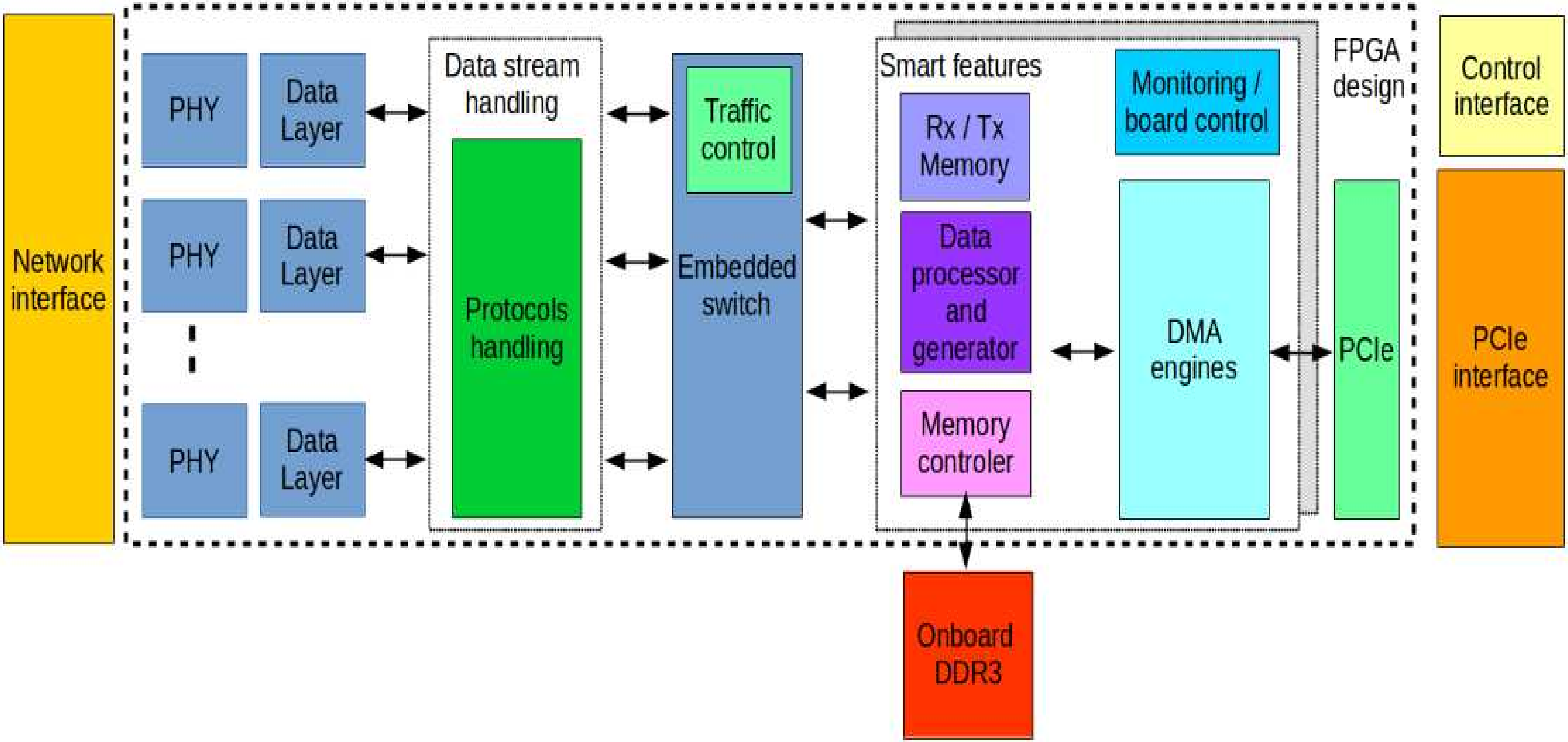
AO RTC concept : data streams



AO RTC concept : local / global interco.



Smart interconnect architecture

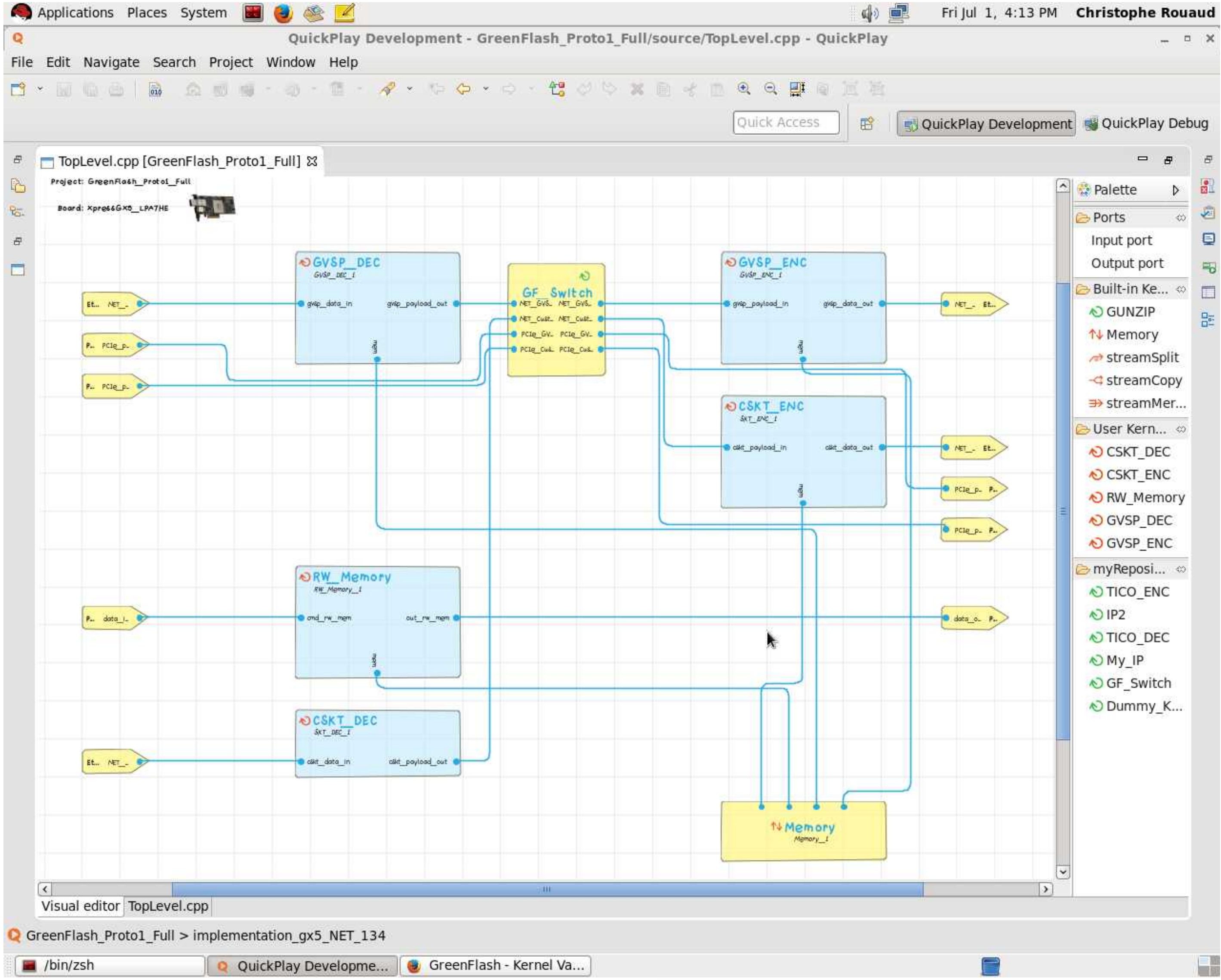


Smart interconnect concept

- Eased devel. process using the QuickPlay tool from PLDA

QuickPlay™

QuickStore™

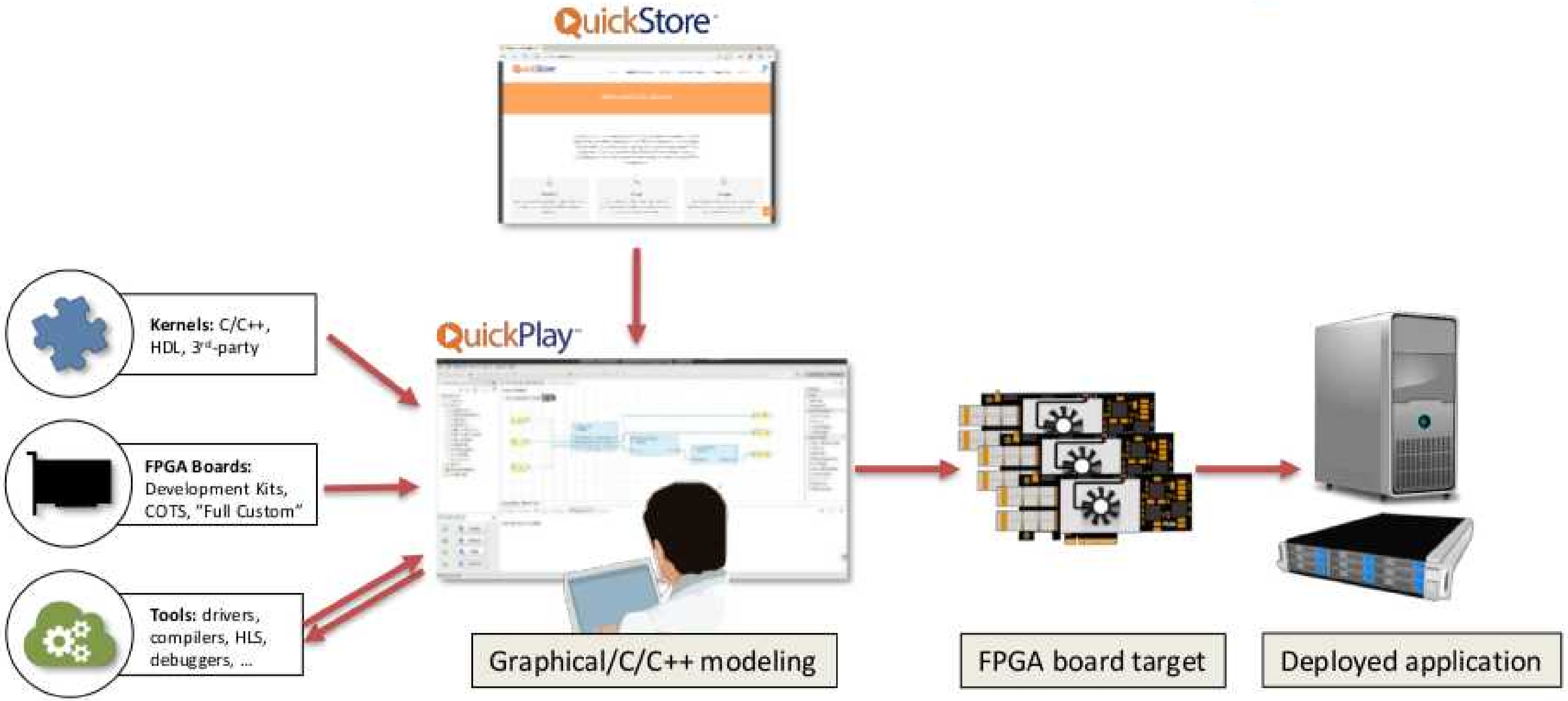


l'Observatoire de Paris LESIA
Laboratoire d'Études Spatiales et d'Instrumentation en Astrophysique
Durham University
MICROGATE
PLDA

QuickPlay

QuickPlay™

Introducing QuickPlay



QuickStore

Kernels: C/C++, HDL, 3^{rd}-party

FPGA Boards: Development Kits, COTS, "Full Custom"

Tools: drivers, compilers, HLS, debuggers, ...

QuickPlay

Graphical/C/C++ modeling

FPGA board target

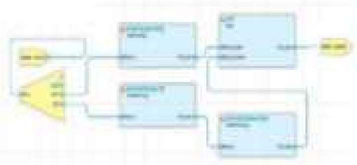
Deployed application

QuickPlay

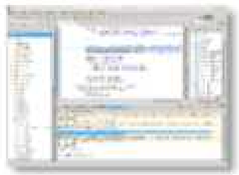
QuickPlay™

FPGA Design with QuickPlay IDE

1. **MODEL**
   - C/C++ functional modeling

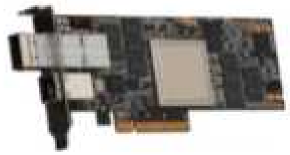


2. **VERIFY & VALIDATE**
   - Desktop execution of system functional model



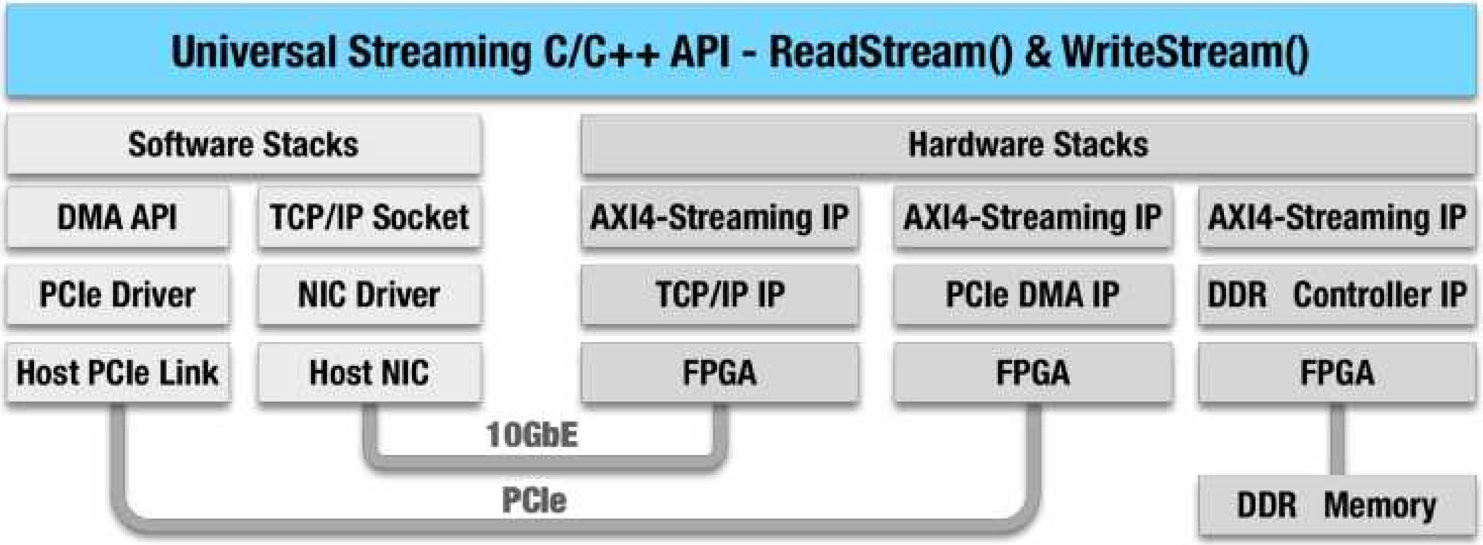
3. **BUILD**
   - Hardware implementation: HLS, Logic Synthesis, P&R

4. **EXECUTE**
   - FPGA based system hardware execution



QuickPlay

QuickPlay™ Hardware Accelerator Abstraction Layer



Universal Streaming C/C++ API - ReadStream() & WriteStream()

Software Stacks

DMA API

TCP/IP Socket

PCIe Driver

NIC Driver

Host PCIe Link

Host NIC

Hardware Stacks

AXI4-Streaming IP

AXI4-Streaming IP

AXI4-Streaming IP

TCP/IP IP

PCIe DMA IP

DDR Controller IP

FPGA

FPGA

FPGA

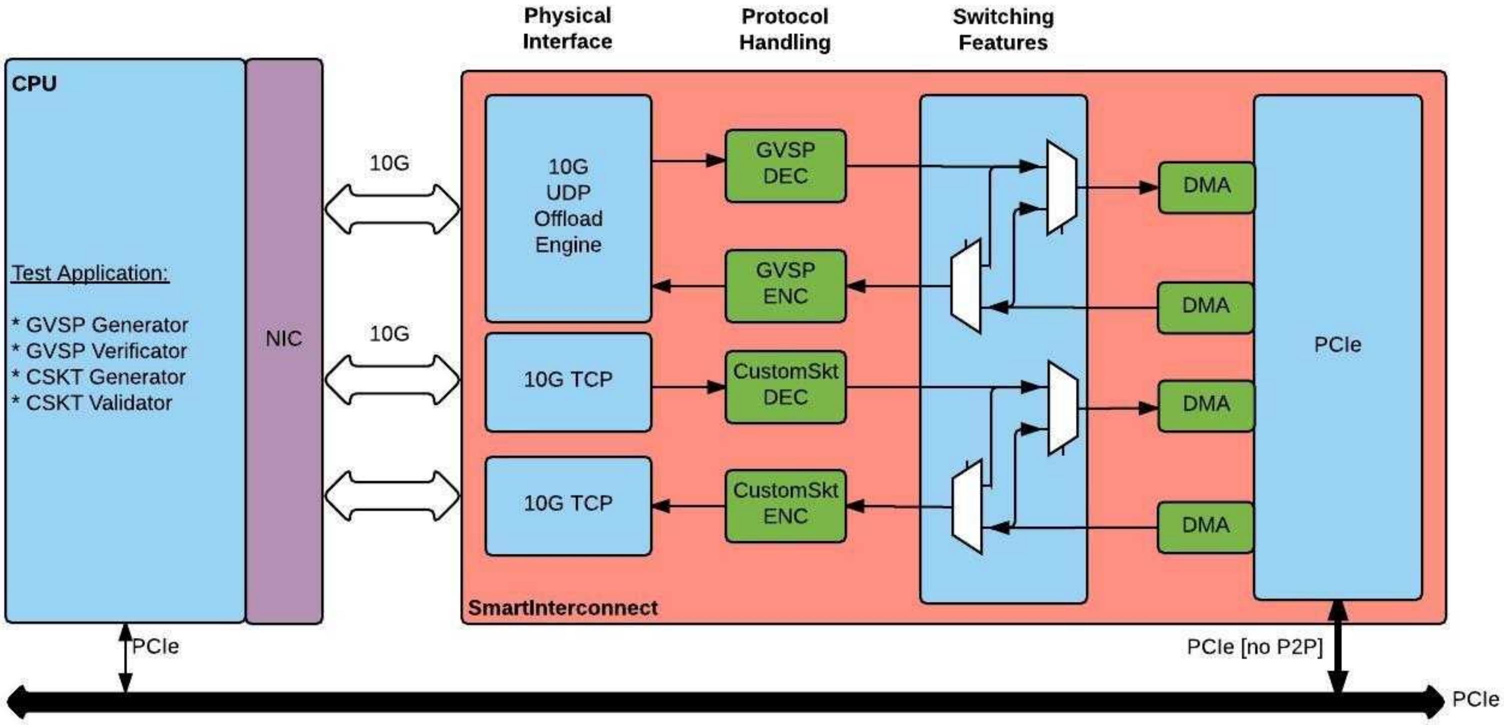
10GbE

PCIe

DDR Memory

Smart interconnect concept

- Link with high level API / application

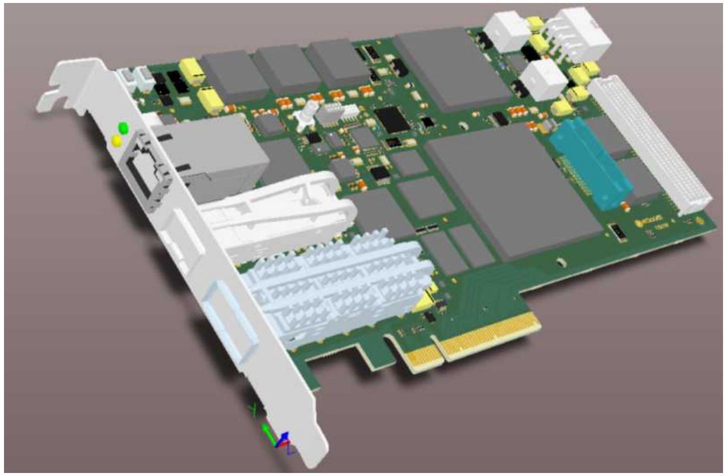


Smart interconnect prototyping

- Single generic design / multiple target boards
  - ExpressK-US board (hosting a Kintex UltraScale from Xilinx)
  - ExpressGX V board (hosting a Stratix V from Altera)
  - µXlink board from microgate (hosting a Arria 10 board from Altera)

REFLEX CES
Custom Embedded Systems

REFLEX CES
Custom Embedded Systems



MICROGATE

Faster Technology

LESIA
Laboratoire d'Études Spatiales et d'Instrumentation en Astrophysique

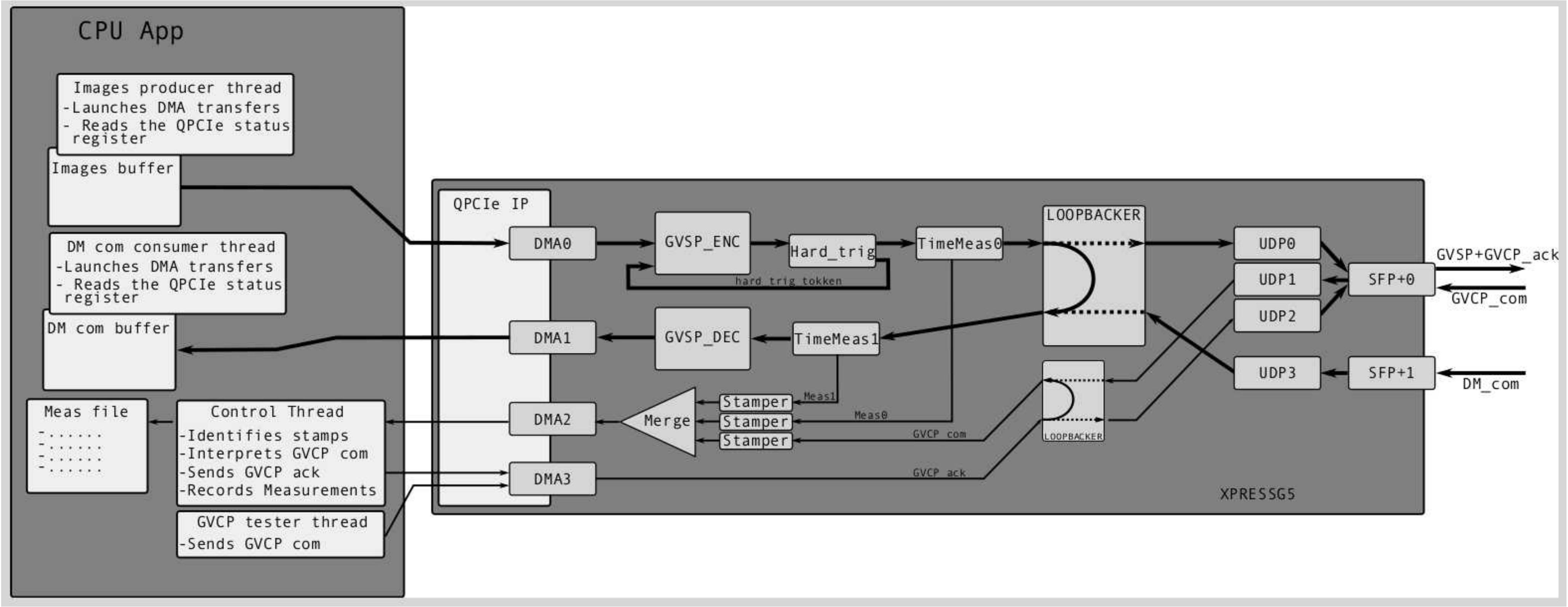
Durham University

MICROGATE

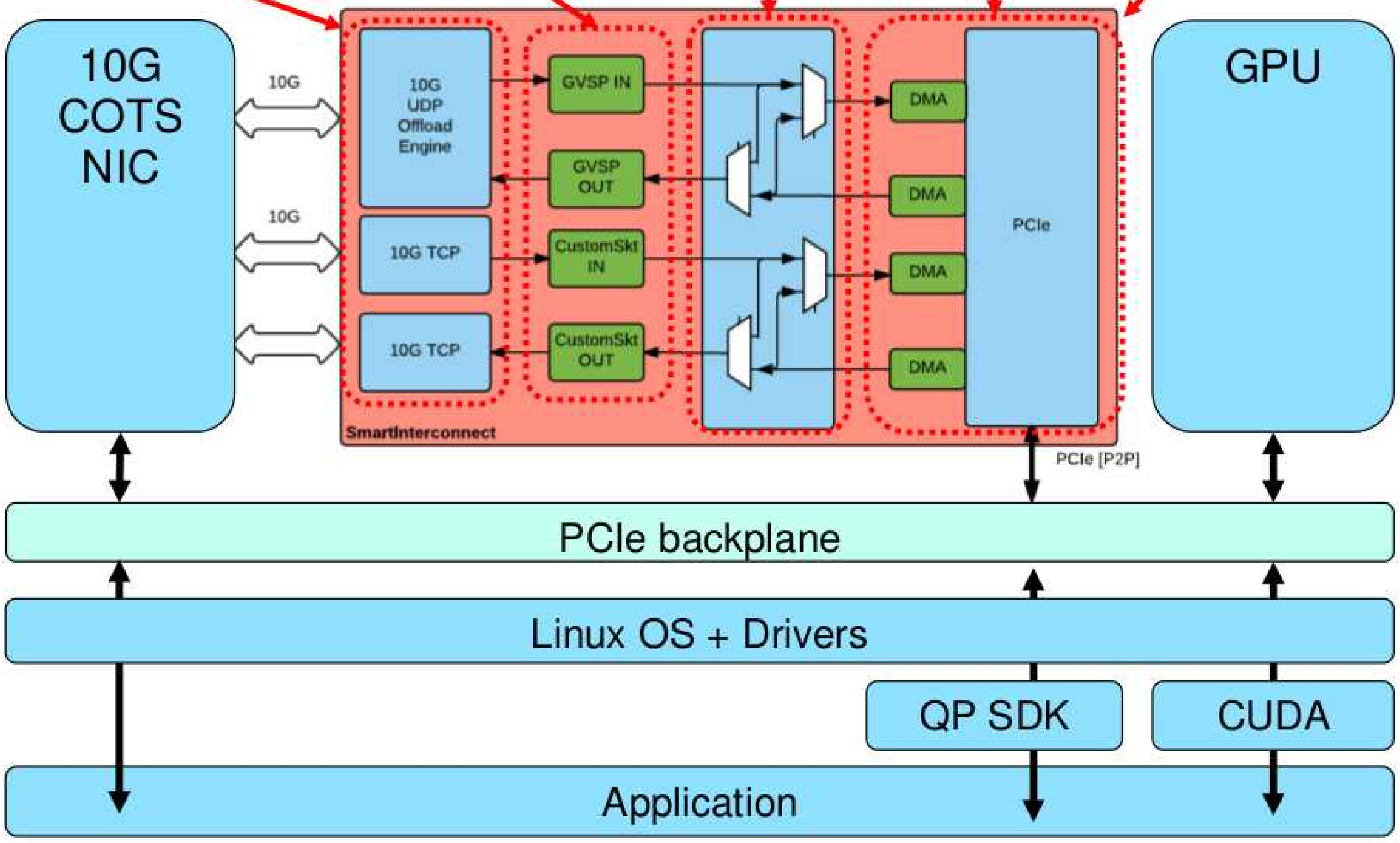
PLDA

Smart interconnect concept

- Fake camera / fake DMC concept developed at LESIA



Smart interconnect prototyping



3x10G Ethernet ports with TCP and UDP offloading IP Cores

GeV partial support (GVSP) and Custom Socket support (CSKT)

Elementary Switch (HDL)

Configurable PCIe with P2P capability

XpressGX5 XpressKUS boards supported

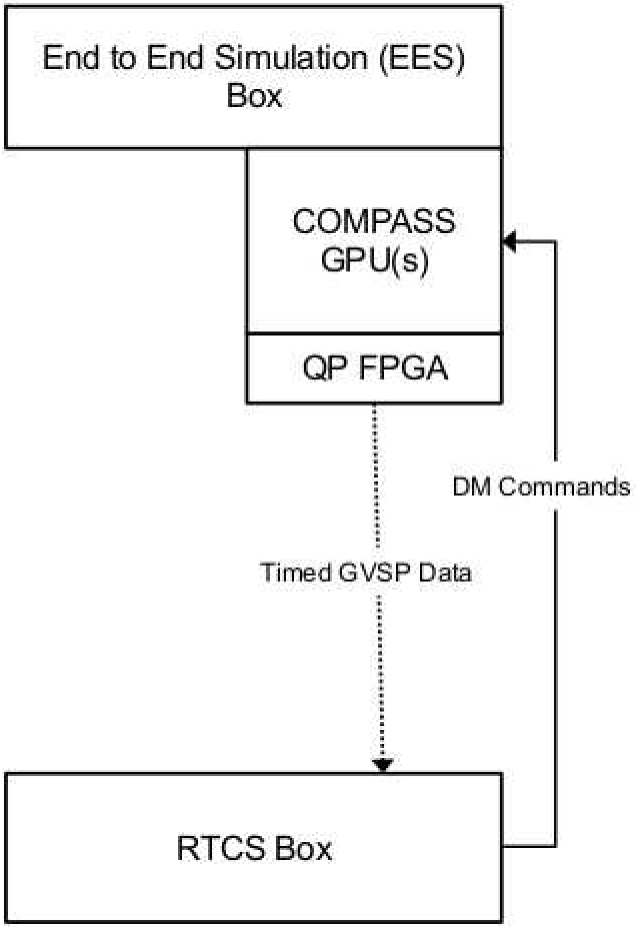
01/22/2016

Real-time simulator concept

- **Two primary modes of operation:**
  - Simulation Rate Mode
  - On-sky Rate mode

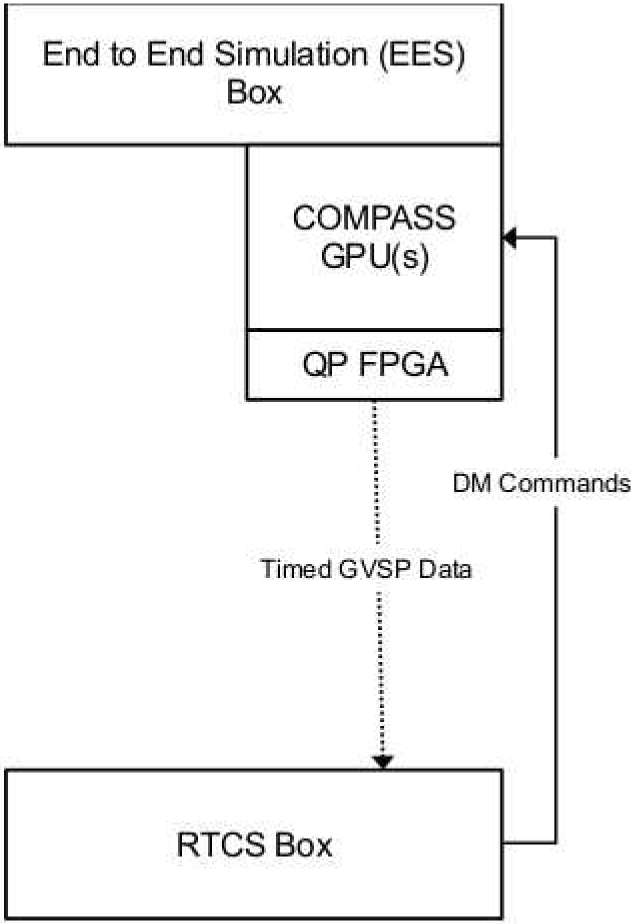
Real-time simulator concept

1. Data Generated by COMPASS
2. Data sent directly from GPU to FPGA
3. Data encapsulated as GVSP
4. Data sent to RTCS
5. DM commands received by COMPASS, providing feedback to simulation



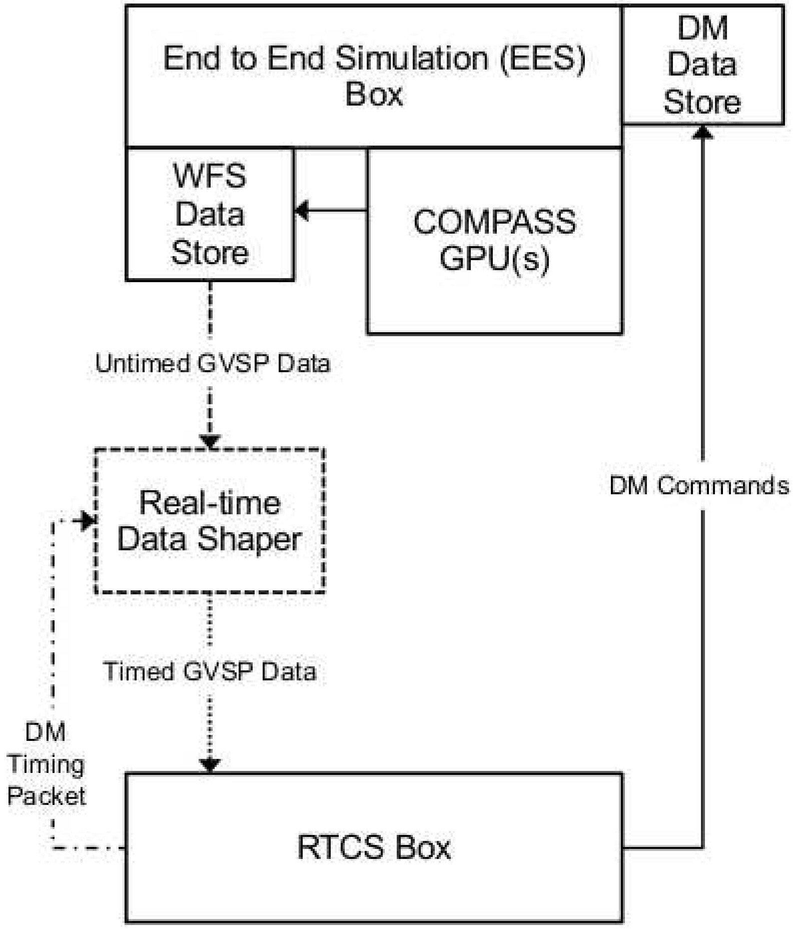
Real-time simulator concept

- COMPASS operates as fast as possible (~100s Hz at ELT scale)
  - Future goal is on-sky rate
- Data extracted from GPU by FPGA to avoid simulation slow down
- Can operate in closed loop and assess AO performance
- Can quickly change AO parameters



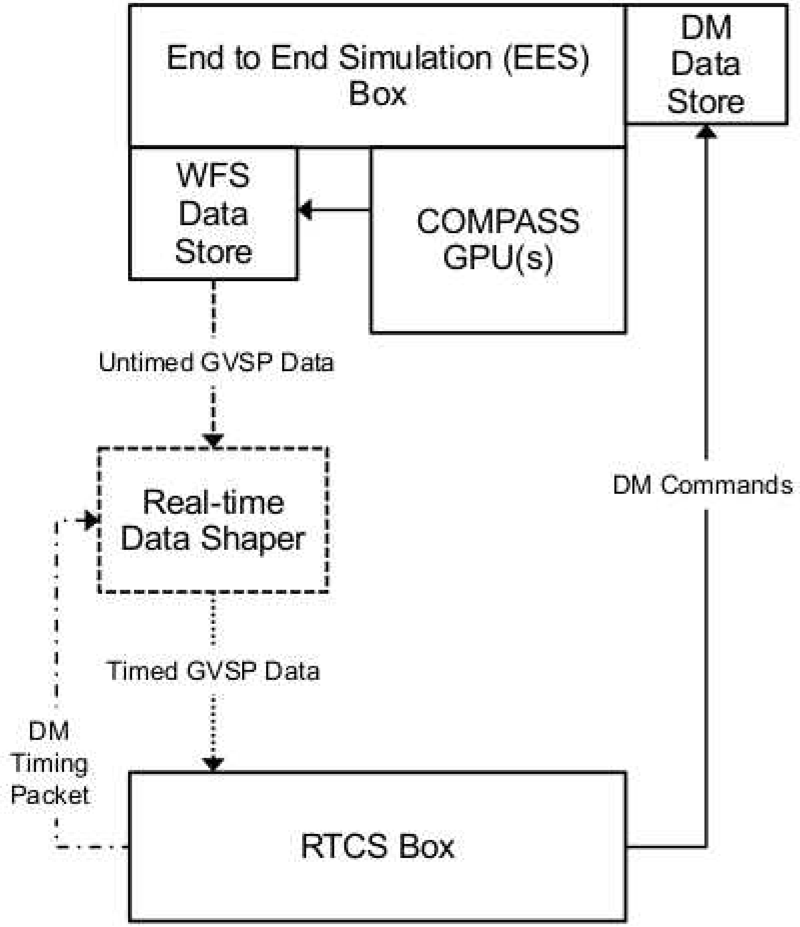
Real-time simulator concept

1. Large WFS data set generated by COMPASS, encapsulated as GVSP and stored
2. Data sent to 10G Ethernet at on-sky frame rate
3. Data packets intercepted by RTDS, and buffered
4. Data sent out with deterministic timing to RTCS box
5. DM Commands received and saved in data store for later analysis

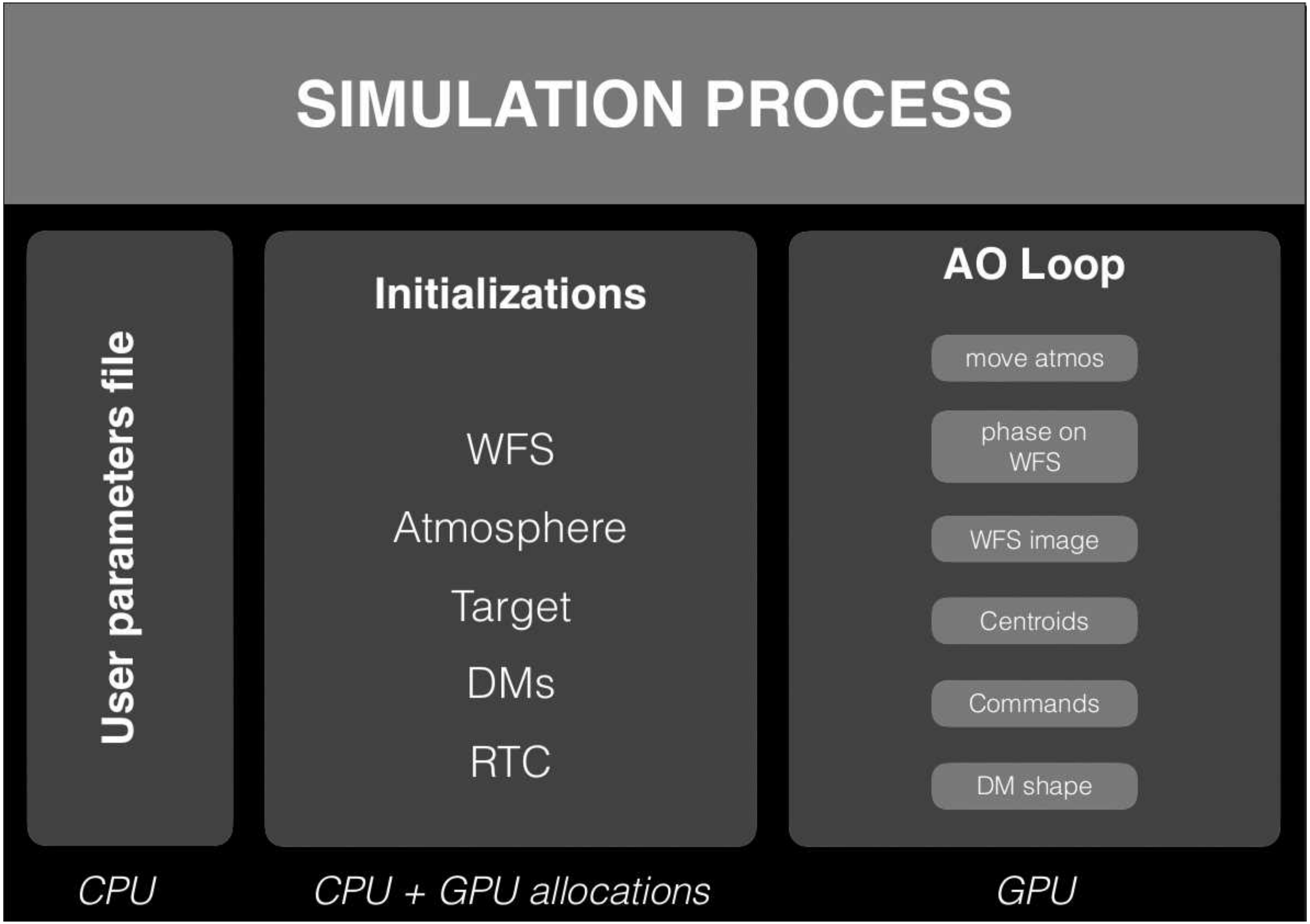


Real-time simulator concept

- Can operate at on-sky rate and measure latency accuracy
- Minimal jitter from RTDS
- Realistic data provided
  - but no AO feedback
- Data stored to ensure reconstruction successful
- Flexible design allows data from any source to be used



The COMPASS platform



SIMULATION PROCESS

User parameters file

Initializations

WFS

Atmosphere

Target

DMs

RTC

AO Loop

move atmos

phase on WFS

WFS image

Centroids

Commands

DM shape

CPU

CPU + GPU allocations

GPU

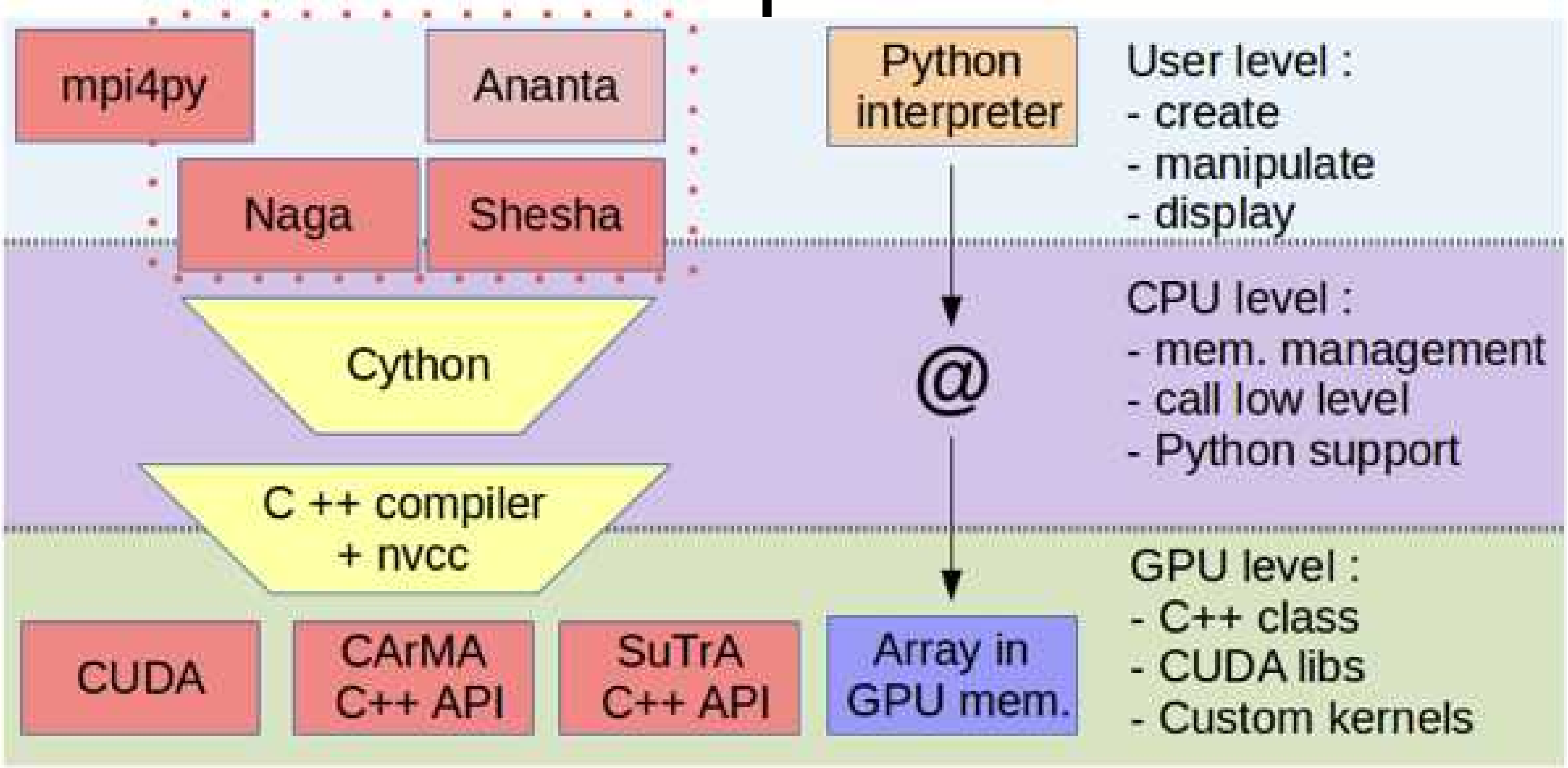
l'Observatoire de Paris LESIA
Laboratoire d'Études Spatiales et d'Instrumentation en Astrophysique

Durham University

MICROGATE

PLDA

The COMPASS platform

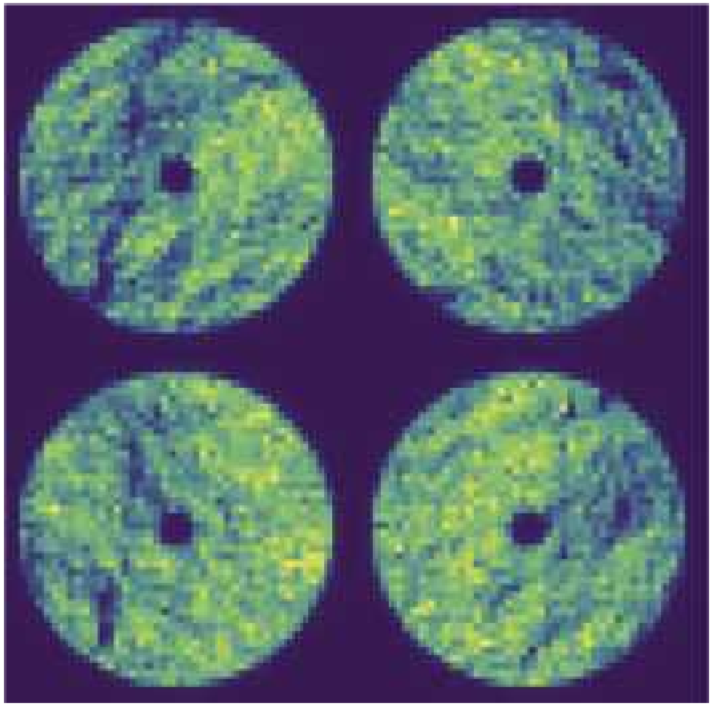


User interface coded in Python for long tem maintenance

Main computations relies on GPU:

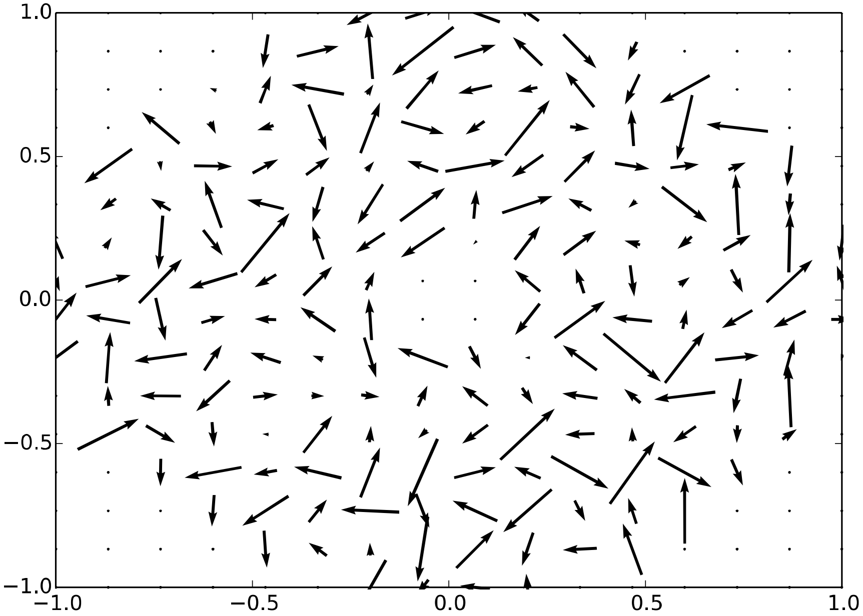
- **CArMA**: C++ Api for Massively parallel Applications
- **SuTrA**: Simulation Tool for Adaptive optics
- Use optimized libraries such as CUBLAS, CUFFT, MAGMA…

Features



Wavefront Sensor models:

- Shack-Hartmann
- Pyramid
- Laser Guide Star



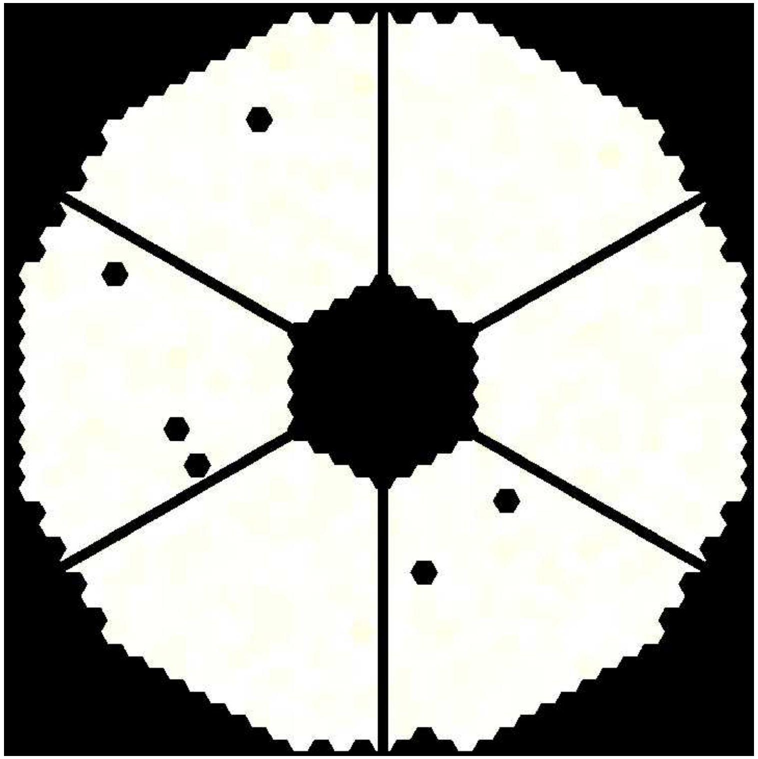
Centroiding methods:

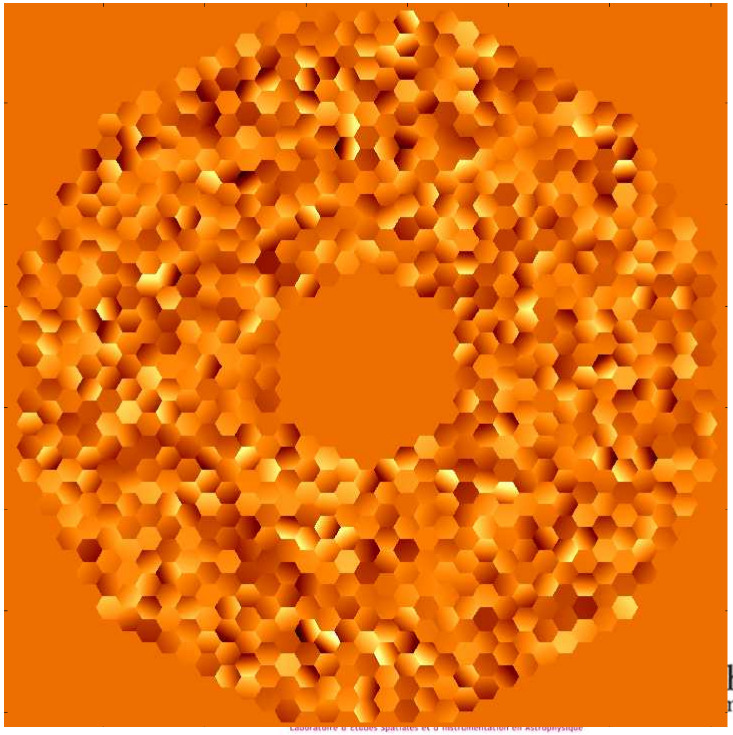
- Center of gravity (cog)
- Thresholded cog
- Weighted cog
- Brightest pixels
- Correlation

Features

Controllers:

- Least square
- Modal optimization
- Minimum variance
- CuReD
- Projection



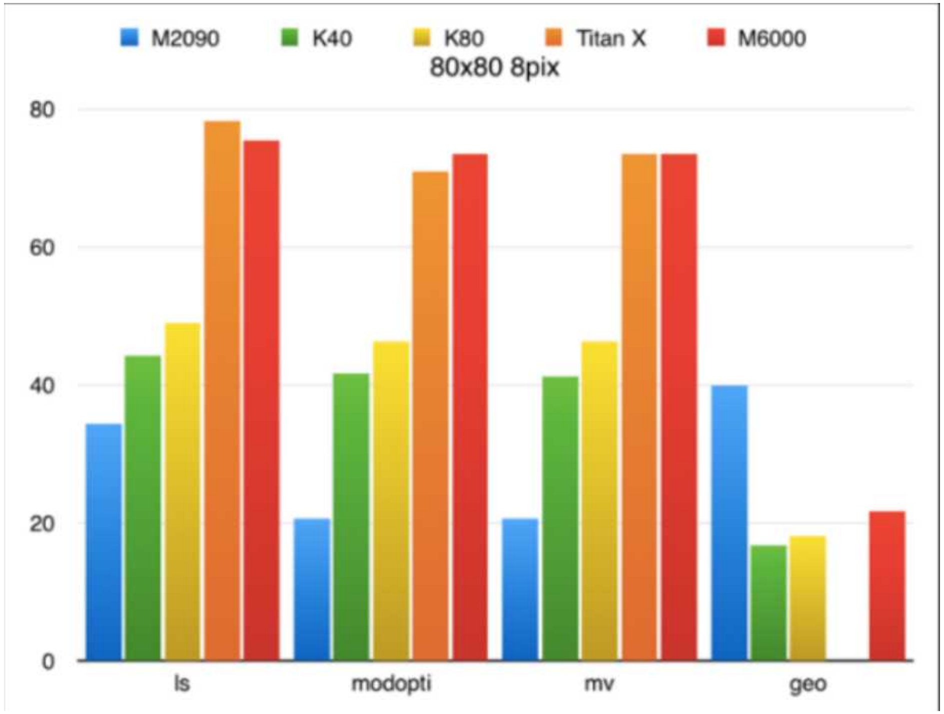


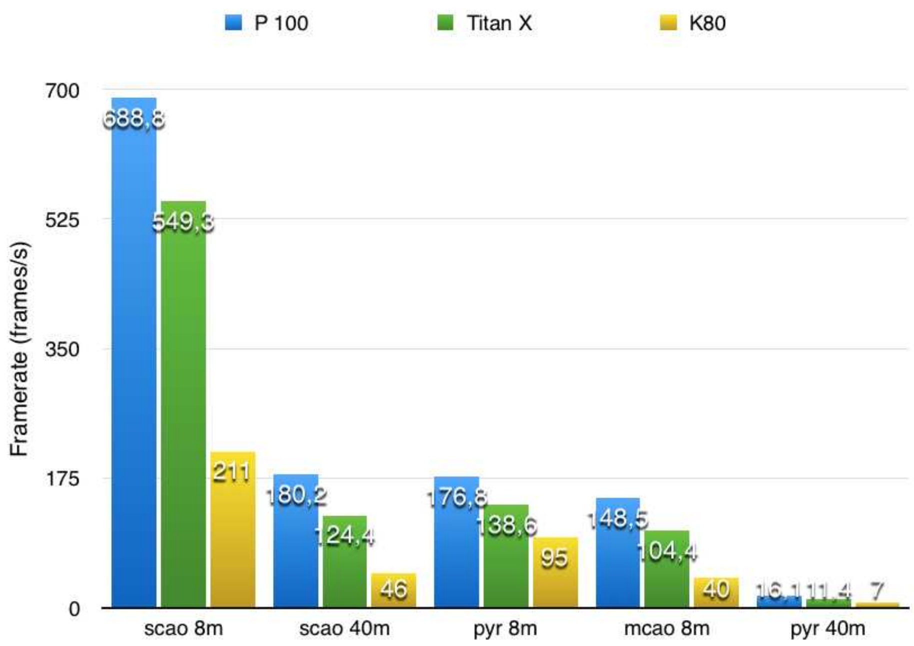
E-ELT:

- Hexagonal pupil
- Spiders
- Phase aberration
- M4 influence functions

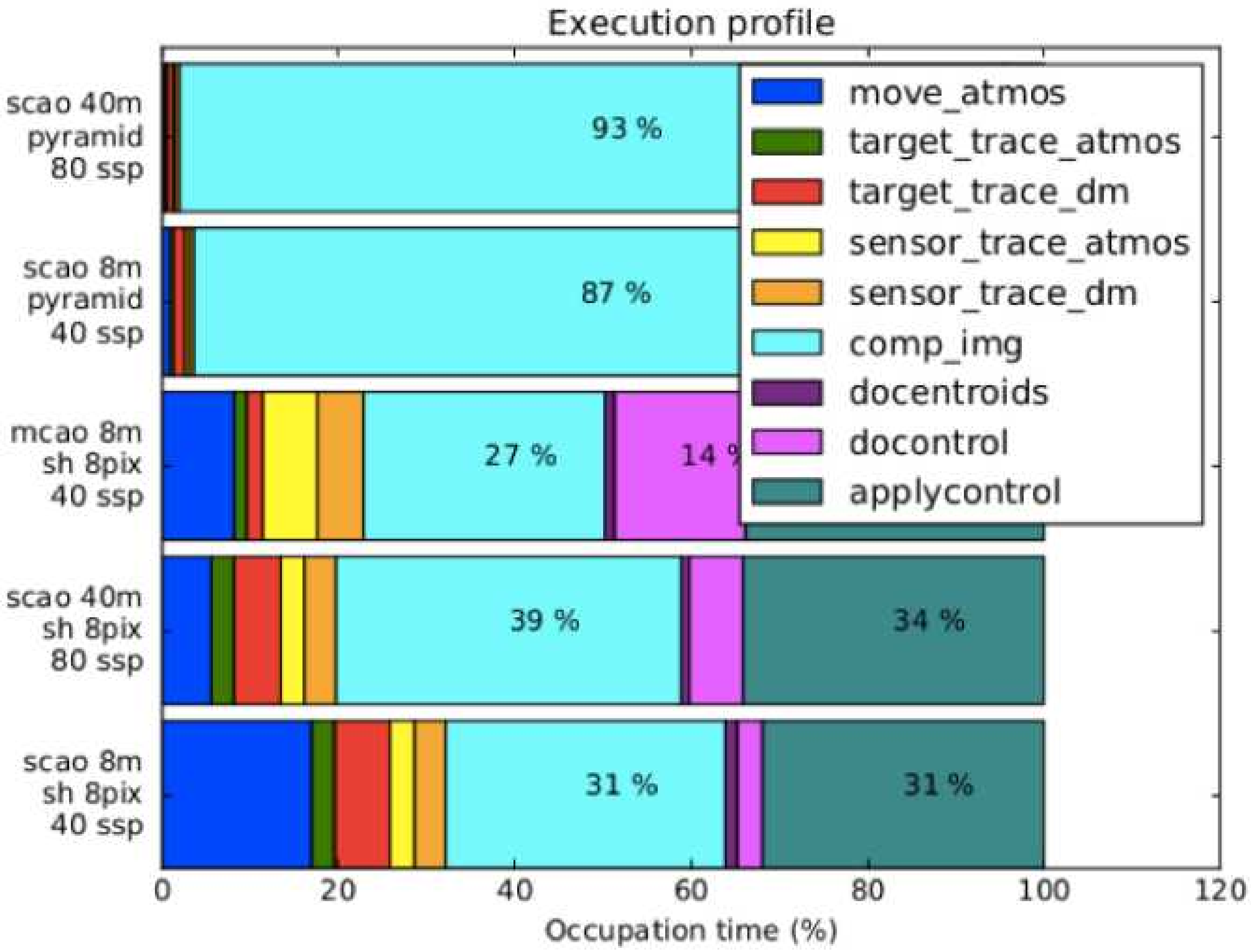
Real-time simulator

- Using COMPASS for E2E should provide a scalable solution over the long term
  - Execution times from F. Ferreira





COMPASS performance

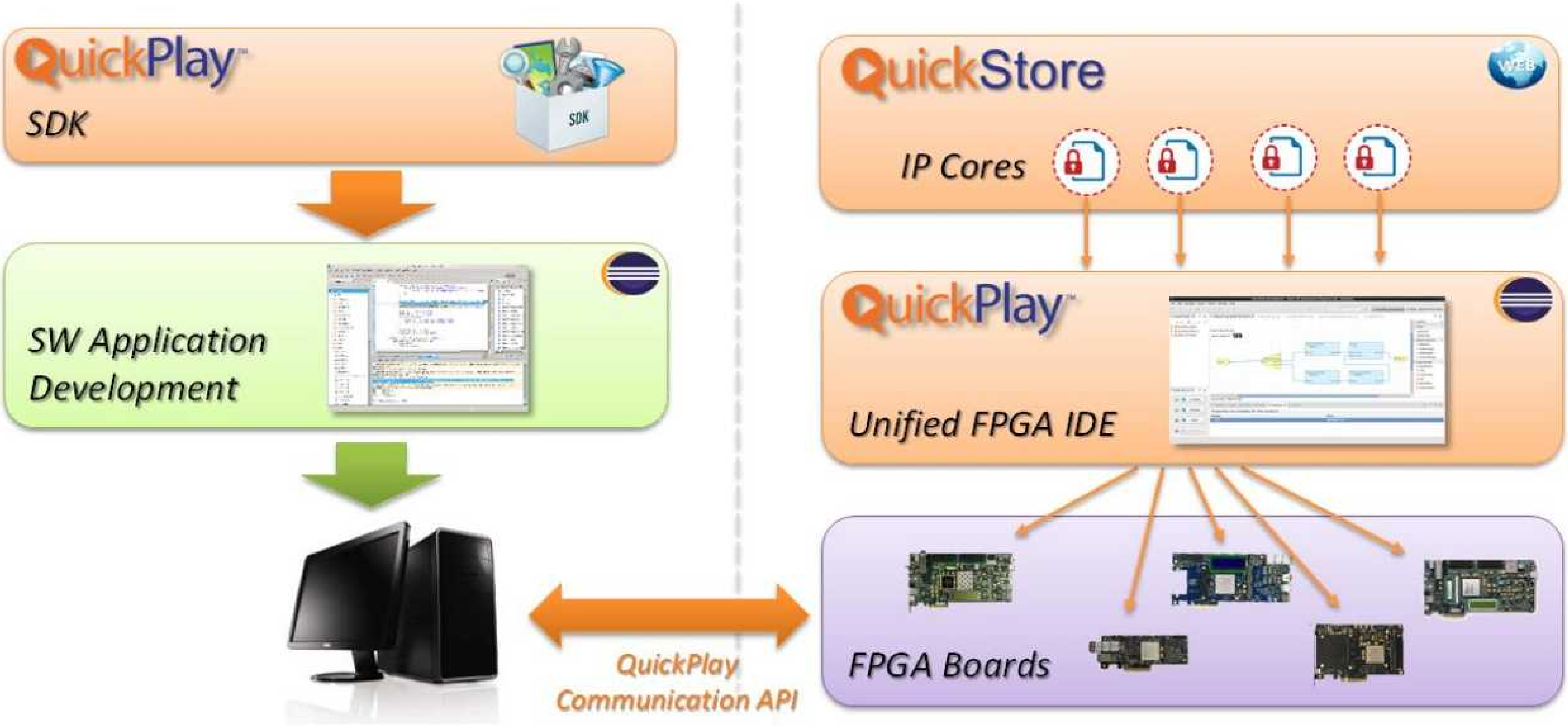


01/22/2016

Green Flash ecosystem

- FPGA development environment
  - Using Quick Play
- Middleware
  - Solutions studied at UoD
- Software and libraries
  - Solutions studied at OdP
- Try to rely on standards as much as possible

FPGA ecosystem



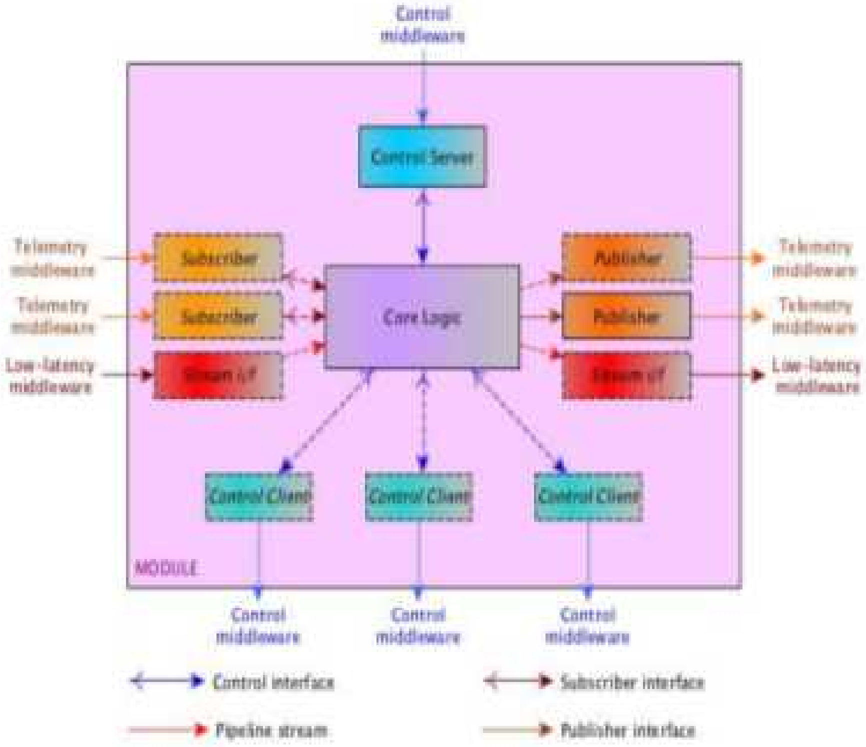
l'Observatoire de Paris — LESIA
Laboratoire d'Études Spatiales et d'Instrumentation en Astrophysique
Durham University
MICROGATE
PLDA

Middleware

- 3 Middleware domains:
  - Control
  - Telemetry
  - Low-latency pipeline
- Closely related to design abstraction

Middleware

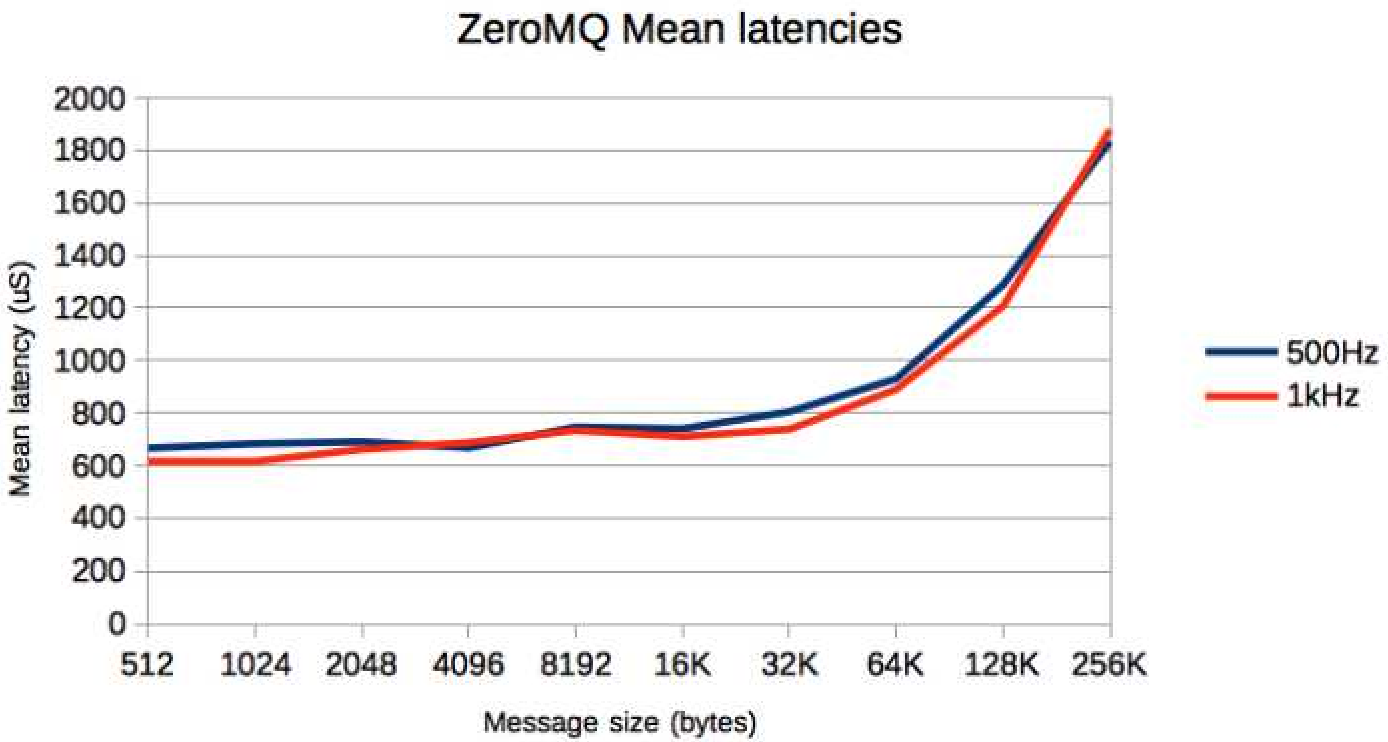
- Interfaces abstracted from core logic
- Module internal interfaces independent of middleware
- Middleware dependencies in separate components



Middleware

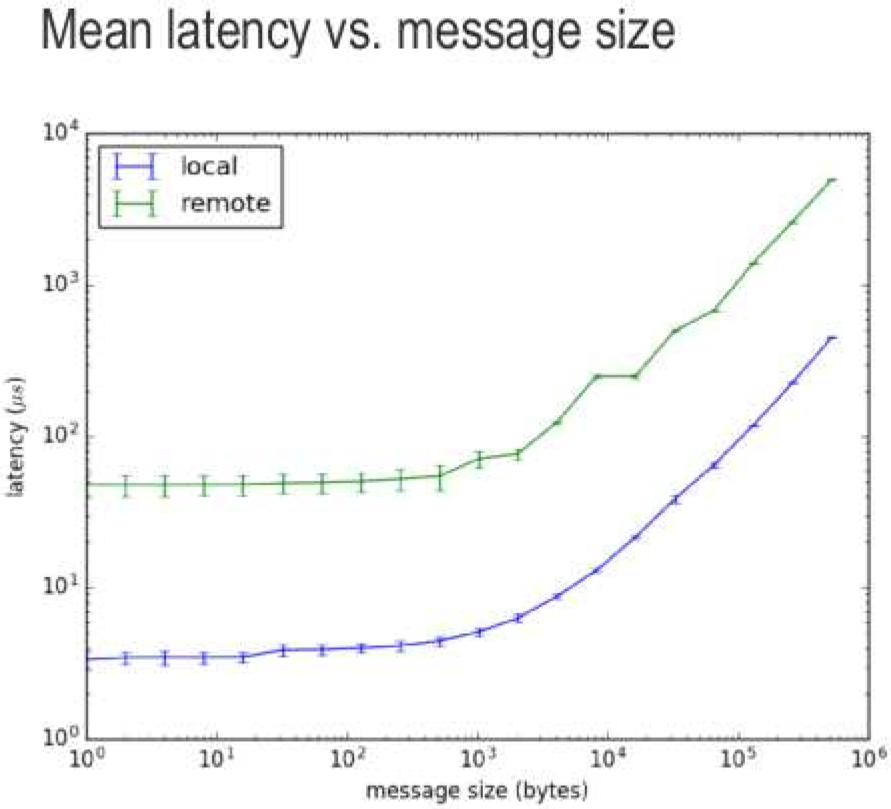
- Evaluations in progress
- Control:
  - DDS, ICE
- Telemetry
  - DDS, ZeroMQ/Google protocol buffers
- Real-time pipeline
  - ZeroMQ, MPI

Middleware : ZeroMQ



01/22/2016

Middleware : MPI



Middleware : conclusions

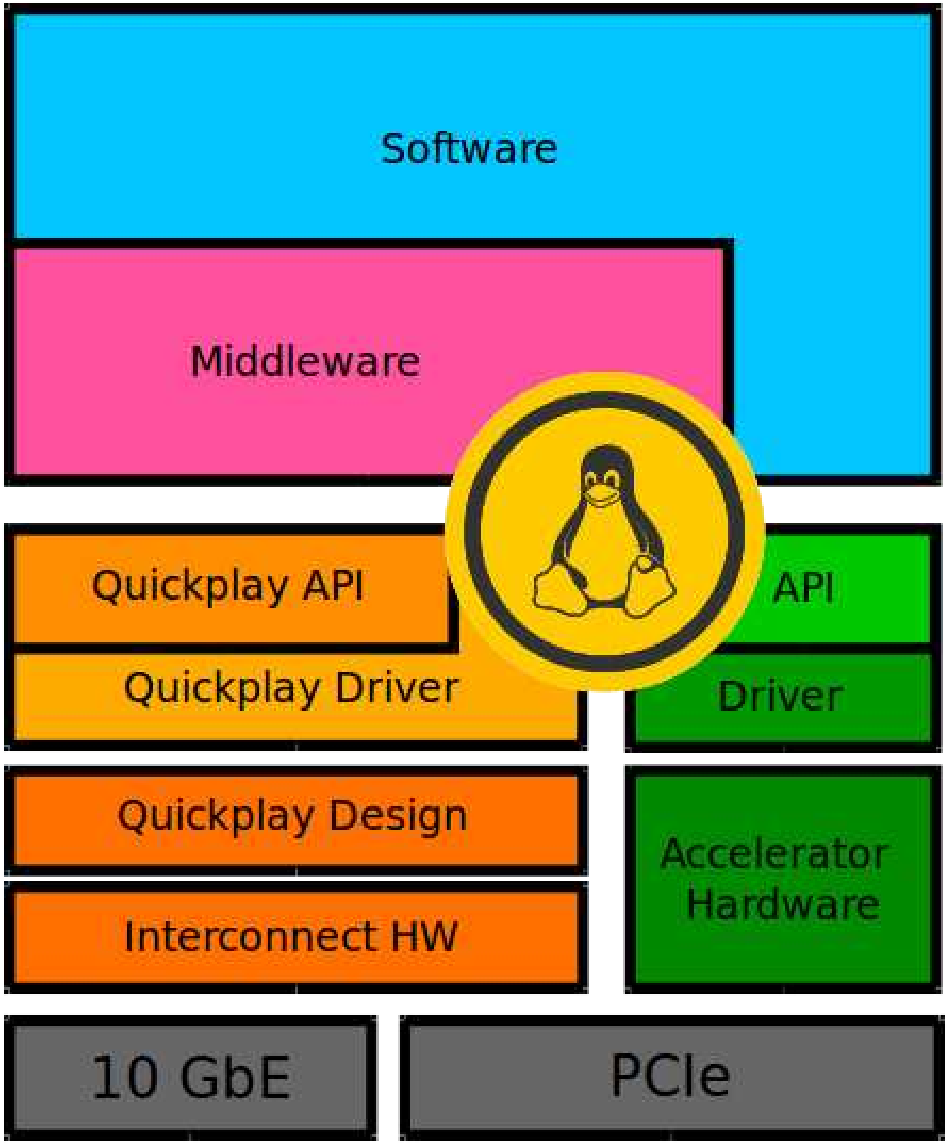
- ZeroMQ: unsuitable for real-time pipeline
  - Excessive latency: > 3 x budget, probably owing to internal buffering and message aggregation
- MPI: latency and jitter adequate
  - ~ 5% of latency budget, for small messages
  - Hence, limited number of network hops allowed
  - Hence, some constraints on implementations using MPI

Ecosystem

- Task 7.3 : algorithms and libraries
  - Definition of various algorithms and libs used in different sub-systems
    - » RT data pipeline : compare custom code and standard libs
    - » Supervisor : only based on standard libs
- Use of accelerators / distributed memory systems
  - portability is key to mitigate obsolescence risk
  - use vendor-based (possibly proprietary) core libs for performance

SW / MW stack

- Under development in Paris & Durham
- Run a standard HPC ecosystem on the prototype boards and clusters
- Performance assessed w/r AO application (mainly linear algebra)

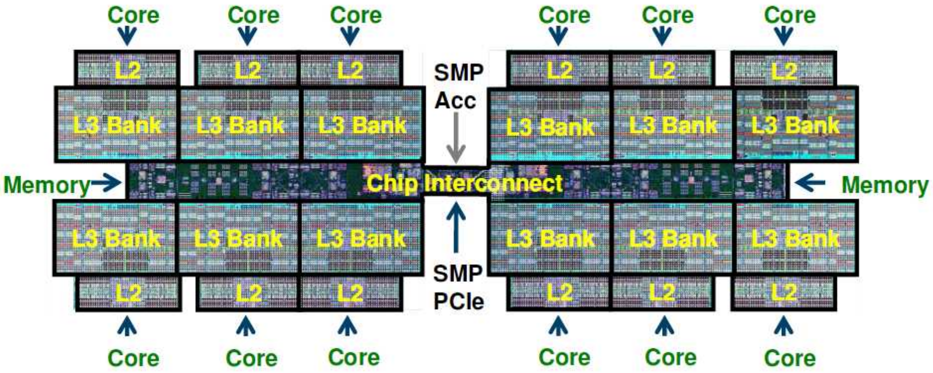


HW landscape

Intel KNL

Nvidia GPUs

IBM Power 8



Unified framework

Supervisor module : compute bound, most demanding stack

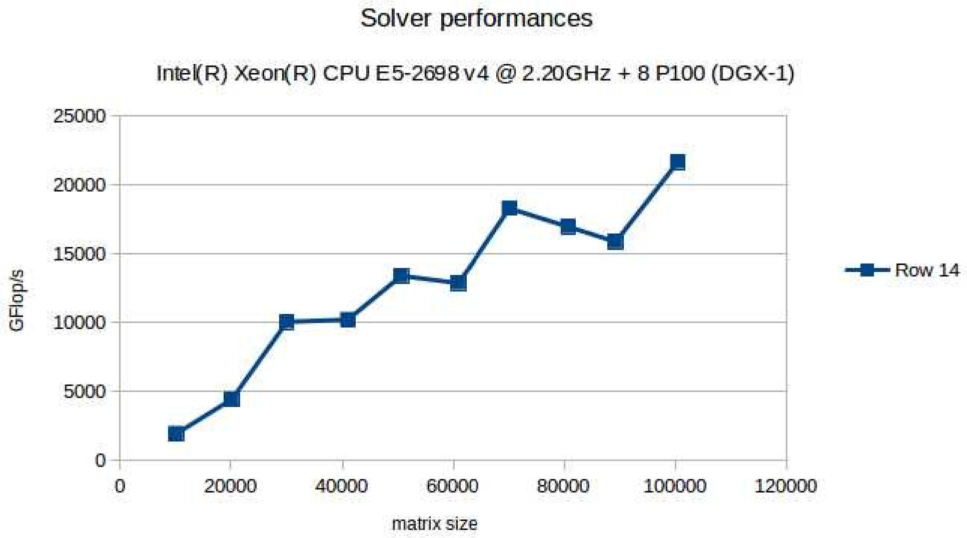
Based on the Chameleon library

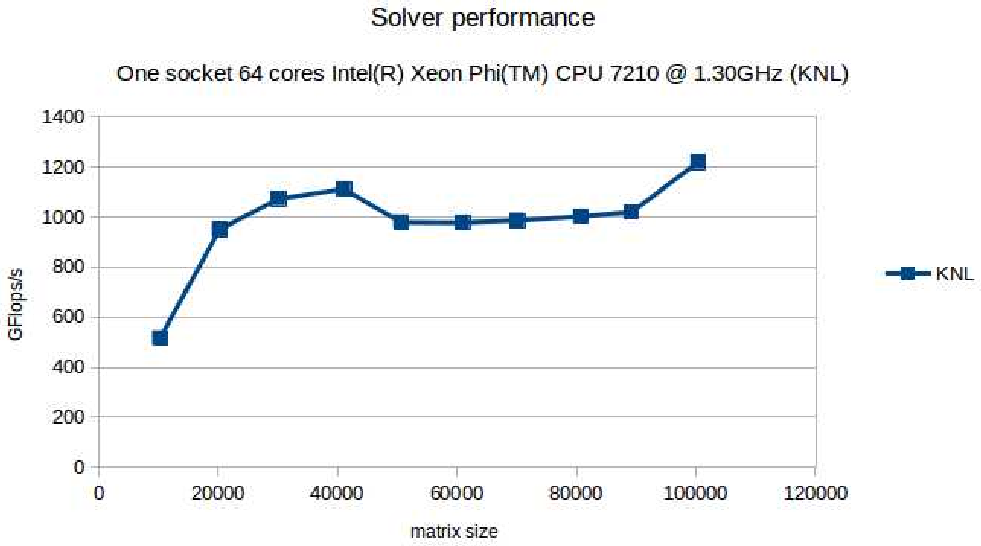
- Tile algorithms
- Sequential task-based programming model
- Dynamic runtime systems: StarPU, Quark, OpenMP, PaRSEC
- Oblivious to the underlying hardware

Unified framework

Performance for reconstructor matrix computation ("apply" process)

Direct comparing between last generation of GPU (NVIDIA P100) and last generation of Intel Xeon Phi (KNL). **Same SW stack**





l'Observatoire de Paris LESIA Laboratoire d'Études Spatiales et d'Instrumentation en Astrophysique

Durham University

MICROGATE

PLDA

Summary

Project on track

- PDR occurred in Jan. 2016 with feedback from community
- Prototyping activities have started with preliminary results (see per WP presentation)
- Prototyping mid-term review scheduled on Feb. 1rst 2017

Collaborations initiated

- Good feedback from the community on different aspects (HPC + instrumentation)
- Evaluate the convergence and minimize additional effort

More work needed on dissemination

- Populating public website
- Contributions to international conference and publications

Already enhancing the readiness level of commercial solutions

- Contribution to QuickPlay development
- Design of an innovative FPGA board (see Roberto's presentation)