Green Flash
High performance computing for real-time science

Project overview & status
Green Flash @ AO4ELT5

1) Biasi et al. “FPGA based microserver for high performance real-time computing in AO” [P3055]
3) Perret et al. “A generic and scalable heterogeneous architecture for real-time computing and performance measurements in AO” [P3037]
4) Doucet et al. “Efficient supervision strategy for tomographic AO systems on ELTs” [P3054]
6) Jenkins et al. “ELT scale real-time control on Intel Xeon Phi and manycore CPUs” [P3036]
7) Ferreira et al. “ROKET: erROr breaKdown Estimation Tool for adaptive optics systems” [P1057]
8) Vidal et al. “MICADO SCAO numerical simulations” [P1056]
9) Petit et al. “RTC strategies for Harmoni SCAO and LTAO modes” [P3035]
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 one or several 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 systems to be assessed in the lab, with a simulator
Assessing new HPC concepts

Current trend in heterogeneous HPC

Accelerators, ARM64 & smart interconnect

FPGA-based microserver

- Network fabric
- Network controller
- Local interconnect (PCIe)
- Host CPU
- Accelerator

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

- Main board with FPGA + ARM HPS
- Integrated ecosystem for interconnect handling and accelerators support
- Scalable environment
AO RTC concept
AO RTC concept: RT simulator

- **High framerate**
- **Sensors**
- **Active elements**
- **High bandwidth Low latency**
- **Switch**
- **Low latency Low jitter**
- **Real-time controller**
- **Telemetry**
- **Fast storage High throughput**
- **High throughput**
- **Supervisor**
Real-time simulator concept

- 2 modes: simulation rate / real-time rate
Real-time simulator concept

- Data store concept
  - PCIe carrier with 4 SSDs (up to 12 GB/s)
Real-time simulator SW

• COMPASS simulator
  – GPU based, scalable, versatile, very fast!
AO RTC concept: data pipeline

- High framerate
- Sensors
- Active elements
- High bandwidth
- Low latency
- Low jitter
- Switch
- Real-time controller
- Telemetry
- Fast storage
- High throughput
- Supervisor
- High bandwidth
- High throughput
FPGA solutions: µXcomp

Based on ARRIA 10 AX115:
- 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 solutions: µserver concept

- Based on Arria 10 but no HMC
- To be used as a main unit in a cluster
RT data pipeline with GPUs

• Prototype using latest generation GPU server
RT data pipeline with GPUs

- Prototype using latest generation GPU server
Persistent kernels
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 kernels

Synchronize jitter

- Average: 15μs
- Jitter: 8.8μs

Intercommunication jitter

- Average: 24μs
- Jitter: 12μs
Persistent kernels

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
RT data pipeline with Xeon Phi
Xeon Phi solution

SCAO MVM on single
Xeon Phi, Knights Landing

- 1 million iterations
- (~10 minutes)
- 750 ± 17μs
Xeon Phi prototype
Xeon Phi testing facility
RT data pipeline with COTS FPGA

COTS FPGA cluster
FMC to 10GbE

- 2 SFP+ 10GbE interface
- 2 SATA like internal connection interface
- Based on the FMC HPC connection
COTS FPGA cluster

peak memory bandwidth of 76.8 (38.4) GB/秒, i.e. 19.2 (9.6) GFLOPS
AO RTC concept: smart interconnect

- High framerate
- Sensors
- Active elements
- High bandwidth
- Low latency
- Switch
- Low latency
- Low jitter
- Real-time controller
- Telemetry
- Fast storage
- High throughput
- Switch
- Supervisor
- High bandwidth
- High throughput
Smart interconnect concept
Smart interconnect concept

- Eased develop. process using the QuickPlay tool from PLDA
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
  - FPGA
  - FPGA
  - FPGA
  - DDR Memory

10GbE

PCle
Smart interconnect prototype

• Link with high level API / application
Smart interconnect prototype

- 3x10G Ethernet ports with TCP and UDP offloading IP Cores
- GeV partial support (GVSP) and Custom Socket support (CSKT)
- Configurable PCIe with P2P capability
- Elementary Switch (HDL)
- XpressGX5 XpressKUS boards supported

Diagram showing connectivity and components:
- 10G COTS NIC
- 10G UDP Offload Engine
- 10G TCP
- GVSP IN
- GVSP OUT
- CustomSk1 IN
- CustomSk1 OUT
- PCIe backplane
- Linux OS + Drivers
- QP SDK
- CUDA
- Application
Smart interconnect prototyping

- Single generic design / multiple target boards

<table>
<thead>
<tr>
<th>FPGA family</th>
<th>Board name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stratix V</td>
<td>Reflex XpressGX5</td>
</tr>
<tr>
<td>Kintex-7</td>
<td>Reflex XpressK7 160/325 (v2.0)</td>
</tr>
<tr>
<td></td>
<td>Xilinx KC705 (v2.1)</td>
</tr>
<tr>
<td>Kintex UltraScale</td>
<td>Reflex XpressKUS (v2.0)</td>
</tr>
<tr>
<td></td>
<td>Xilinx KCU105 (v2.1)</td>
</tr>
<tr>
<td>Arria10</td>
<td>Microgate (\mu)XComp (2017.5)</td>
</tr>
<tr>
<td></td>
<td>Reflex XpressGX4A10 (2017.5)</td>
</tr>
<tr>
<td></td>
<td>Bittware A10PL4 (2017.5)</td>
</tr>
</tbody>
</table>
AO RTC concept: supervisor

- High framerate
- Sensors
- Active elements

- High bandwidth
- Switch
- Low latency
- Low jitter
- Real-time controller
- Telemetry
- Fast storage
- High throughput

- High bandwidth
- Switch

- High throughput

- Supervisor
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

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

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

Reconstructor matrix computation ("apply" process)

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

\[ R' = C_{tm} \cdot C_{mm}^{-1} \]

- 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
AO RTC concept: SW & MW

- High framerate
- Sensors
- Active elements
- High bandwidth
- Low latency
- Switch
- Telemetry
- Fast storage
- High throughput
- Real-time controller
- Low latency
- Low jitter
- High throughput
- Supervisor
Middleware

3 Middleware domains:
- Control
- Telemetry
- Low-latency pipeline
Middleware: ZeroMQ

Unsuitable for RT data pipeline

- Excessive latency x3 budget
- Probably due to internal buffering
Middleware: MPI

Latency & jitter acceptable

- ~5% of budget for small message sizes
- But limited NW hops allowed, some constraints on implementation
### Middleware: down-selection

<table>
<thead>
<tr>
<th>ID</th>
<th>Criterion</th>
<th>Description</th>
<th>Weighting</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS-MW-1</td>
<td>Reliability</td>
<td>The middleware should be able to guarantee delivery of uncorrupted data, or at the least, detect and signal non-delivery or data corruption.</td>
<td>3</td>
</tr>
<tr>
<td>DS-MW-2</td>
<td>Latency</td>
<td>2mS (goal: 1mS) between first pixel received and last actuator demand delivered. This is the total latency budget for the pipeline, the majority of which must be available to be expended on processing; a nominal 10% of the budget has been allowed in this assessment for communications.</td>
<td>3</td>
</tr>
<tr>
<td>DS-MW-3</td>
<td>Jitter</td>
<td>100μS peak-to-peak; as in the case of latency, this is the budget for the pipeline. Contributions to jitter from different sources (processing, communication,...) sum quadratically; a nominal 30% of total jitter has been allowed.</td>
<td>3</td>
</tr>
<tr>
<td>DS-MW-4</td>
<td>Throughput</td>
<td>Within the pipeline: the most demanding case in terms of aggregate throughput is METIS LTAO mode, with a frame rate of 1kHz and 6 LGS/3 NGS WFS. The input bandwidth for pixel data is ~ 200Gb/s (25 Gb/s). However, this is not carried by a single connection, and pixel input data is not carried by the middleware. A more realistic requirement on bandwidth per link within the pipeline is the transport of pixel data for a single WFS, from a calibration module to a centroider module; for a single LGS WFS, the required bandwidth is 2.19 Gb/s (274 MB/s). If calibration and centroiding are performed within the same hardware module and data is not required to be transported on the network at pixel rates, the requirement is reduced to transporting frames of centroids from a single WFS, and for a LGS WFS at 1kHz this is evaluates to 350 Mb/s (44 MB/s).</td>
<td>3</td>
</tr>
</tbody>
</table>
# Middleware: down-selection

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Weighting</th>
<th>Technology</th>
<th>Remarks</th>
<th>Score</th>
<th>Weighted score</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS-MW-1 Reliability</td>
<td>3</td>
<td>ZeroMQ</td>
<td>No guaranteed delivery</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPI</td>
<td>Reliable QoS available</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>DS-MW-2 Latency</td>
<td>3</td>
<td>ZeroMQ</td>
<td>Unable to meet requirement</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPI</td>
<td>Required performance achieved in testing</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>DS-MW-3 Jitter</td>
<td>3</td>
<td>ZeroMQ</td>
<td>Unable to meet requirement</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPI</td>
<td>Required performance achieved in testing</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>DS-MW-4 Throughput</td>
<td>3</td>
<td>ZeroMQ</td>
<td>Required performance achieved in testing</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPI</td>
<td>Required performance achieved in testing</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>DS-G-1 Cost</td>
<td>1</td>
<td>ZeroMQ</td>
<td>Available free/open source</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPI</td>
<td>Available free/open source</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>DS-G-2 Ease-of-use</td>
<td>1</td>
<td>ZeroMQ</td>
<td>Commensurate with facilities provided</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPI</td>
<td>Commensurate with facilities provided</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>DS-G-3 Long-term support</td>
<td>2</td>
<td>ZeroMQ</td>
<td>Single supplier, commercial support available</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPI</td>
<td>Several implementations available, and very widely used.</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>DS-G-4 Standards compliance</td>
<td></td>
<td>ZeroMQ</td>
<td>No standard</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPI</td>
<td>De-facto HPC standard</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>DS-G-5 Familiarity</td>
<td>2</td>
<td>ZeroMQ</td>
<td>Expertise in consortium</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPI</td>
<td>Expertise in responsible partner</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>DS-G-8 Source of supply</td>
<td>2</td>
<td>ZeroMQ</td>
<td>Single supplier</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPI</td>
<td>Multiple implementations</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>Overall Score</td>
<td></td>
<td>ZeroMQ</td>
<td></td>
<td></td>
<td>22</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPI</td>
<td></td>
<td></td>
<td>57</td>
</tr>
</tbody>
</table>
Summary

Project on track
- PDR occurred in Jan. 2016 and MTR in Feb. 2017 with feedback from community
- Prototyping activities are entering final phase with downselection and final prototype(s) architecture to be defined by end 2017 during FDR

Collaborations initiated
- Good feedback from the community on different aspects (HPC + instrumentation)
- Evaluate the convergence and minimize additional effort
- More than happy to collaborate more!

Excellent feedback for European Commission
- Mid-term progress review in Brussels last week

Already enhancing the readiness level of commercial solutions
- Contribution to QuickPlay development environment
- Design of innovative FPGA boards (see Roberto's poster)
1) Biasi et al. “FPGA based microserver for high performance real-time computing in AO” [P3055]
3) Perret et al. “A generic and scalable heterogeneous architecture for real-time computing and performance measurements in AO” [P3037]
4) Doucet et al. “Efficient supervision strategy for tomographic AO systems on ELTs” [P3054]
6) Jenkins et al. “ELT scale real-time control on Intel Xeon Phi and manycore CPUs” [P3036]
7) Ferreira et al. “ROKET: erROr breaKdown Estimation Tool for adaptive optics systems” [P1057]
8) Vidal et al. “MICADO SCAO numerical simulations” [P1056]
9) Petit et al. “RTC strategies for Harmoni SCAO and LTAO modes” [P3035]