



A Real-time Control Computer for the E-ELT

Document: GF-D2.1

Prototypes mid term report

Version 1.0

14th March 2017

Change Record

Version	Date	Author(s)	Remarks
0.1	1 Mar 2017	D. Gratadour	Initial skeleton
0.2	2 Mar 2017	D. Gratadour	Contribution from Mic
0.3	10 Mar 2017	C. Roaud	Contribution from PLDA
0.41	11 Mar 2017	D. Perret	Contribution from OdP
0.42	11 Mar 2017	M. Lainé	Contribution from OdP
0.43	11 Mar 2017	J. Bernard	Contribution from OdP
0.5	11 Mar 2017	D. Gartadour	Added panel review reports
0.6	13 Mar 2017	J. Osborn	Added contribution from UoD
0.7	13 Mar 2017	N. Doucet	Contribution from OdP
0.8	14 Mar 2017	N. Dipper	Added contribution on down select process
0.9	14 Mar 2017	D. Gratadour	Complemented down select process
1.0	17 Mar 2017	J. Osborn	Updated contribution from UoD

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Damien Gratadour Page: 3 of 49
Prototypes mid term report		

Applicable Documents (AD)

These are the Green Flash PDR documents

No.	Title	Reference	Issue	Date
AD01	Introduction	GF-PDR-01		
AD02	Management plan and WP definition	GF-PDR-02		
AD03	Requirements Specification	GF-PDR-03		
AD04	System Architecture	GF-PDR-04		
AD05	Distributed GPUs for real-time HPC	GF-PDR-05		
AD06	FPGA Solution for hard real-time	GF-PDR-06		
AD07	Interconnect Strategy	GF-PDR-07		
AD08	Interface Control Document	GF-PDR-08		
AD09	Supervision Strategy	GF-PDR-09		


Reference Documents (RD)

These are documents external to the Green Flash project

No.	Title	Reference	Issue	Date
RD01	FPGA microserver design report	GF-D3.1		
RD02	GPU cluster for RT-box prototype	GF-D4.1		

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 4 of 49
Prototypes mid term report		


No.	Title	Reference	Issue	Date
RD03	GPU cluster design for supervisor	GF-D4.3		
RD04	MIC cluster for RT-box design & test	GF-D4.3a		
RD05	Smart interconnect demonstrator	GF-D5.1		
RD06	Smart interconnect demonstrator #2	GF-D5.2		
RD07	Prototype board support package definition	GF-D5.7		
RD08	Scalability of QuickPlay designs	GF-D5.10		
RD09	List of features to integrate in QuickPlay	GF-D7.1		
RD10	Simulator interface definition	GF-D6.1		
RD11	QuickPlay integration plan	GF-D7.2		
RD12	Green Flash nine months periodic report			
RD13	Simulator concept and interface	GF-DU-SCI-001		

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 5 of 49
Prototypes mid term report		

Acronyms and abbreviations

Table 1 Acronyms and Abbreviations


AO	Adaptive Optics
AXI4	Advanced Microcontroller Bus Architecture (AMBA) AXI4
CPU	Central Processing Unit
CUDA	NVIDIA GPU based software development language
DDS	Data Distribution Service
DMA	Direct Memory Access
DDR	Double Data Rate SDRAM (memory)
E-ELT	European ELT
ESO	European Southern Observatory
FPGA	Field Programmable Gate Array
GPU	Graphics Processing Unit
HDK	Hardware Development Kit
HDL	Hardware Description Language
HLS	High Level Synthesis
HPC	High Performance Computing
IP	FPGA IP (Intellectual Property) core
I2C	Inter-Integrated Circuit
M4	E-ELT adaptive mirror
MPI	Message Passing Interface
MTR	Mid Term Review
NIC	Network Interface Controller
PCIe	Peripheral Component Interconnect express
PtP	Precision Time Protocol
RD	Reference Document
RTC	Real-Time Control
RTL	Register Transfer Level
RTPS	Real-Time Publish Subscribe
SDK	Software Development Kit
SOC	System On Chip
SPARTA	ESO VLT AO Real-time Control System
UDP	User Datagram Protocol
VLT	Very Large Telescope
WFS	Wave-Front Sensor


Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 6 of 49
Prototypes mid term report		

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 7 of 49
Prototypes mid term report		

Table of Contents

Scope.....	9
FPGA for hard-RT prototyping.....	9
Outline.....	9
Work done during the first half of the prototyping period.....	9
Initial requirements.....	9
Solutions design.....	10
It is a complex board with 18 layers and a high number of components. 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.....	13
Board testing.....	13
Prototyping using hardware accelerators.....	15
Outline.....	15
Mainstream accelerators technologies for a real-time application.....	16
Cluster of GPUs for real-time.....	16
Intel Xeon Phi.....	25
COTS FPGA technologies.....	34
Supervision strategy.....	37
Learn process.....	39
Apply process.....	42
Xeon Phi for supervisor applications.....	44
Smart interconnect prototyping.....	48
Outline.....	48
Work done during the first half of the prototyping period.....	48
Smart Interconnect prototype design - Results.....	50
Simulator prototyping.....	51
Requirements.....	52
Proposed Solution.....	52
Prototypes ecosystem.....	53
Middleware.....	53
FPGA development environment.....	56
Selection criteria for the final design review.....	57
General description of the down selection process.....	57
Detailed example with the smart interconnect concept.....	60
Reports from mid-term review panel members.....	61
Reports from Marcos Suarez, SPARTA architect, ESO.....	62
Report from Laura Schreiber, Real-time computer engineer for MAORY project, INAF.....	69
Report from Yann Clénet, Co-PI for the MICADO project.....	71
Report from Markus Feldt, Co-Investigator of the METIS project.....	73

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 8 of 49
Prototypes mid term report		

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 9 of 49
Prototypes mid term report		

Scope

This document aims to outline the output of the Green Flash prototyping mid-term review that was held in Paris on Feb. 1st 2017. Representatives for all the project partners participated to this review as well as a panel from experts from the astronomical AO community. A number of presentations were made by project partners that are accessible on the project website here : <http://greenflash-h2020.eu/-prototyping-mid-term-review-.html>

Additionally, the panel members provided short reports on the material presented during the this MTR. These reports can be found at the end of this document. The core of this document contains a description of the activities led at each partner during the first half of the prototyping period as well as the main results obtained to date. Moreover, the final prototype down selection process was addressed during the review and is presented in the last section as it stands today.

FPGA for hard-RT prototyping

Outline

This aspect of the prototyping activities, under the responsibility of Microgate, provides a concept study based on FPGA boards for the stackable, energy efficient stand-alone microserver for data-intensive applications. It involves the prototyping of one main board with an SoC FPGA containing a hard-wired ARM processor and several interface and the production of several FPGA based computational boards to be clustered in a microserver. The performance in terms of communication bandwidth and computation throughput will be assessed for the AO application on a single board and on the small scale cluster.


Work done during the first half of the prototyping period

During the preliminary design phase (performed before the granting of the GreenFlash project by the EC) we defined the internal requirements and the architecture of the microserver system.

Initial requirements

The internal requirements can be summarized as follows:

- The misroserver shall allow stand-alone operation using SoC FPGA-CPU , while preserving compatibility with standard servers
- It shall be possible to interface it to different accelerator cards (FPGA based, GPU based, CPU based)
- The system shall be modular to adapt to the Real Time Reconstructor requirements of different AO instruments; in this frame, it shall also provide different interfaces to wavefront

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 10 of 49
Prototypes mid term report		

- cameras and deformable mirrors
- It shall be expandable to cope with high computational throughput demands, in the range of few floating point TMACs
- It shall guarantee low latency (maximum two ms to complete the pipeline processing) and low jitter, <100µs
- It shall be energy efficient in comparison to other hardware solutions with similar performance
- The hardware design shall be compatible with the PLDA QuickPlay development tool


Solutions design


The solution implemented comprises the design and prototyping of two different boards based on FPGAs. One board, *called $\mu XComp$* , will act as a computational board that can perform the real-time computation in a deterministic way with low latency and low jitter. To guarantee these performances, the board will be based on the Arria 10 FPGA by Altera, embedding >1500 DSP cores, each capable of performing a full MAC operation in one cycle. Moreover, the board is equipped with the Hyper Memory Cube technology that allows memory transfer rates up to 10x faster than SDRAM DDR4 technology. The second board, *called $\mu XLink$* , will have an FPGA with an hard-wired ARM processor in the same chip (SoC) and is used as an interface and control board to connect to several computational boards and to WFSs and DMs. These board will be also based on the Arria 10 device, but in its SoC version.

Some highlights from these boards :

- Both based on Altera ARRIA 10 FPGAs – Newest Altera FPGA Chip on the market
- High number of hard-wired DSPs and Transceivers
- Each DSP can perform a full single precision (32-bit) floating-point MAC
- The large number of transceivers allows to realize a large number of different interfaces e.g. 10G Ethernet, Infiniband ...
- Backplane communication interface based on PCIe x8 up to Gen3
- Include a novel external memory chip HMC (Hybrid Memory Cube) – fast DRAM memories stacked vertically using true-silicon-via combined with up to 64 high-speed transceiver serial links (up to 120GB/s each direction)
- Low power consumption

During the first half of the prototyping period, the HW design of the $\mu XComp$ board was completed; the routing of the board was finalized as well as the component procurement to fabricate the first prototypes. The first prototype of the $\mu XComp$ was also produced by Microgate and is one of two types of FPGA boards to realize the hard real-time data pipeline in a microserver. The first prototype of the two FPGA boards, the $\mu XComp$ board is manufactured and is currently under test.

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 11 of 49
Prototypes mid term report		

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 12 of 49
Prototypes mid term report		

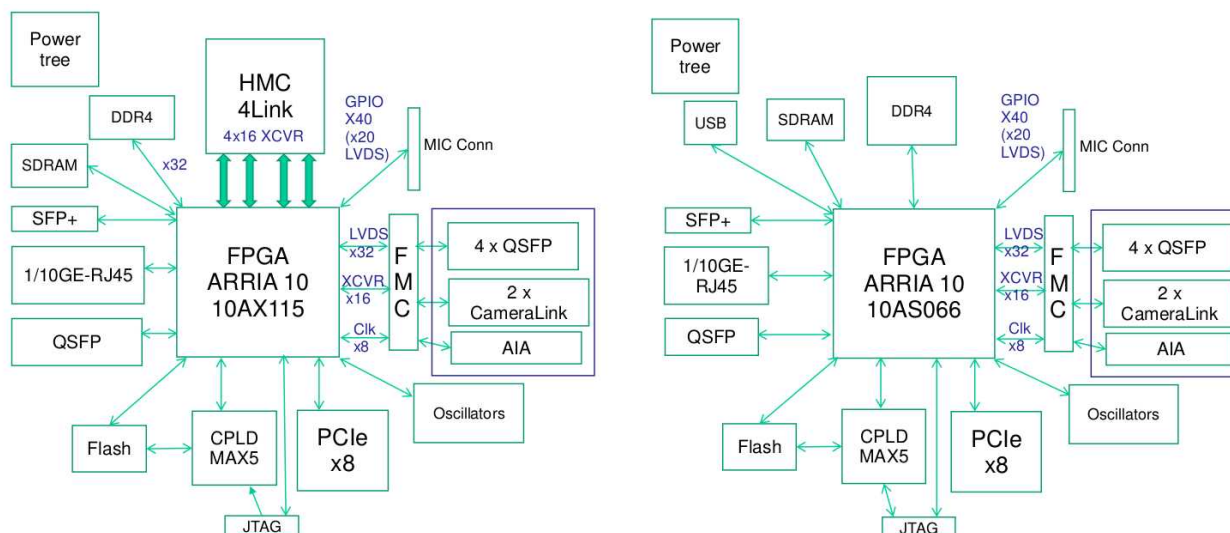



Illustration 1: block diagram for the two boards developed at Microgate (left μ XComp and right μ XLink)



Illustration 2: Picture of the first μ XComp board manufactured at Microgate

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 13 of 49
Prototypes mid term report		

It is a complex board with 18 layers and a high number of components. 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.

Board testing

Board features:


- PCIe up to x8 Gen 3 (64Gb/s each direction)
- Front-panel interfaces: 1/10Gb Ethernet via fiber (SFP+) and via copper (RJ45), 40Gb Ethernet or Infiniband (QSFP)
- FMC standard extension boards attachable on the back (up to length 130mm to fit in a PCIe full length slot (320mm))
- Sustained performance 30 GMAC/s (single precision floating-point)
- Memory transfer rate 120GByte/s each direction
- Compatible with QuickPlay tool requirement

Board facts:

- Board size (111x200 mm) compliant with PCIe standard single slots full height and $\geq 3/4$ length
- # Layers: 18 (9 signal, 9 power-ground)
- # Components: 1442
- # Tracks: 71388 (300 LVDS pairs)
- # Vias: 12305

Tests completed

- Voltages measured of all Power rails
- Current driving capability for each rail measured (up to 100W total power consumption tested)
- Power-up and Power-down sequence programmed and measured
- External PLL chip programmed and clk outputs measured
- Max5 CPLD logic development started for housekeeping
 - SPI interface implemented for ADC to read out voltages and currents of the different power rails.
 - I2C interface implemented to read out the temperature sensors and set the thresholds
 - Flash interface implemented to access the flash memory via JTAG.
 - JTAG switches tested adding the HMC JTAG
- PCIe interface x8 Gen 2 implemented in new ARRIA 10 and tested -> works
- MVM calculation 222X5316 floating-points implemented for DP control -> works good with execution time of $< 30\mu s$ (only internal memory)

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 14 of 49
Prototypes mid term report		

Tests still to be performed:

- HMC interface testing up to 4 links with 16 Transceiver lanes per link
- DDR4 memory testing
- 10G Ethernet
- Real-Time Interface to AO mirrors

After these tests are done and show the main features of the board working we will launch the production of other 2 μ XComp boards – one for PLDA and one for Observatoire de Paris

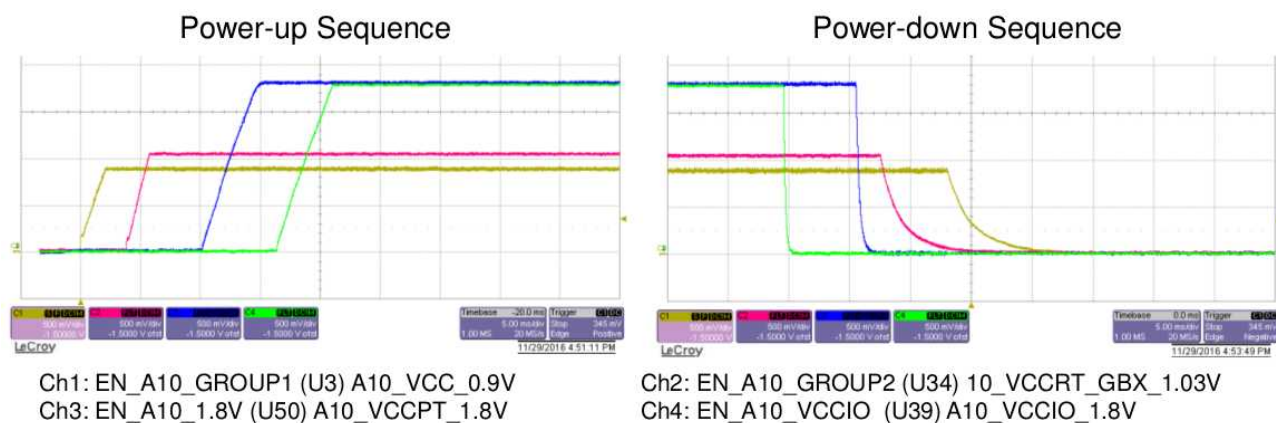



Illustration 3: Measured power-up and power-down sequences on the μ XComp board

On the firmware side, a preliminary version of the PCIe board drivers for the Linux OS was completed at Microgate and some sample AO design at low level were implemented, i.e. not using yet the Quickplay tool by PLDA. We also implemented preliminary input and output interfaces for the real-time pipeline.

For the design of the second μ XLink board already a preliminary design is started and the final design will start in March. A lot of knowledge gained with the μ XComp can be reused for the design of the μ XLink that will reduce the development time of the second board significantly. The first prototype of μ XLink is expected to be available by summer 2017. The assembly of a Microserver containing one μ XLink and one μ XComp is predicted by the end of the 2017.

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 15 of 49
Prototypes mid term report		

Prototyping using hardware accelerators


Outline

The goal of WP 4, under the responsibility of Observatoire de Paris with a significant contribution from University of Durham, is to investigate a classic accelerated architecture for adaptive optics real-time control where the system is based on a standard CPU server accelerated by various alternative technologies: GPU, MIC (Intel Xeon Phi) and FPGA. The advantage of such a system is that it is based entirely on commercial off-the-shelf components and the accelerator hardware can be abstracted from the software and upgraded with new technology as it emerges. The advantages and disadvantages of the three technologies are being assessed in terms of throughput, latency and jitter along with ease of programming.

Mainstream accelerators technologies for a real-time application

Cluster of GPUs for real-time

Most of the control laws used require a matrix inversion and matrix-vector multiply, at a frequency around 1 kHz. We have studied a GPU based solution because of their great energy efficiency, but the main focus of our work is on their deterministic behavior. The goal is as much to maximize performance on a single iteration as to minimize the jitter on this peak performance, including data transport. In order to meet the specifications (jitter, throughput), we chose a very low level approach using a FPGA based network and used persistent kernels to handle all the computation steps that include pixel calibration, slopes and command vector computation. This approach simplifies the latency management by reducing the communication but leads us to re-implement an entire AO control loop and some GPUs standard features : communication mechanisms (guard, peer-to-peer), algorithms (generalized matrix-vector multiplication, reduce/all reduce) and new synchronization mechanisms on a multi node - multi GPU system. Thanks to the use of this strategy coupling custom (direct) FPGA-GPU data transfer and persistent kernels on the GPU, we were able to demonstrate very low jitter on a realistic pipeline, dimensioned to the SCAO case on the E-ELT, including data transfer from a simulated 240x240 camera, providing pyramid WFS data, and corresponding RTC computation (a vector of about 10k wavefront measurements per frame to produce a 5k command vector).

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 16 of 49
Prototypes mid term report		

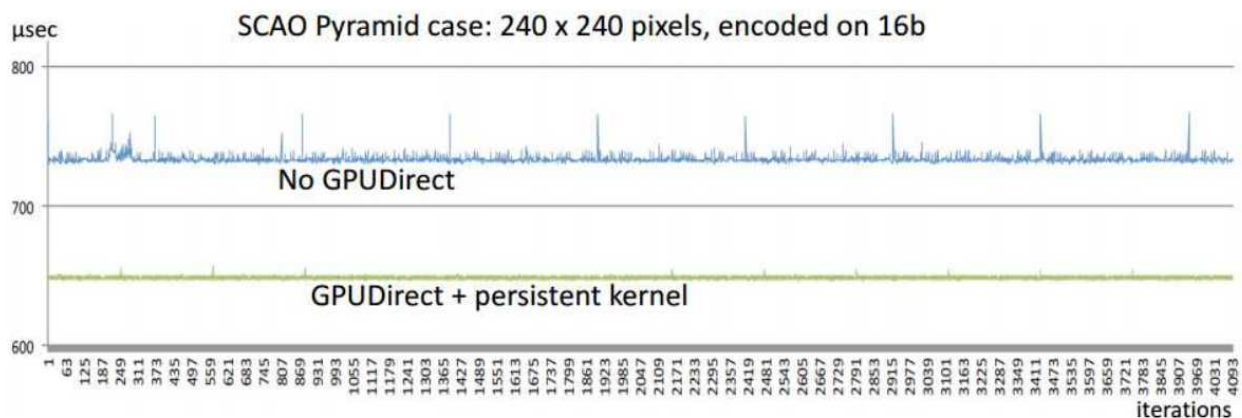


Illustration 4: Performance of a single GPU RTC prototype for a SCAO system on the E-ELT

The obtained results on a single GPU are consistent with the constraints of a real system, with a time-to-solution of 650μs

In order to scale up to the specifications of the Green Flash project, targeting a MCAO system on the E-ELT, we have designed and started the implementation of a generic computing node based on FPGAs, GPUs and CPUs. Each component of the node is used for what it is best at and can directly access other components' address spaces, allowing efficient peer-to-peer transfers, dynamic data streams reconfiguration and thus future advanced middleware strategies. To meet the performance requirements, this generic platform is based on the NVIDIA DGX-1 server as depicted illustration 5.

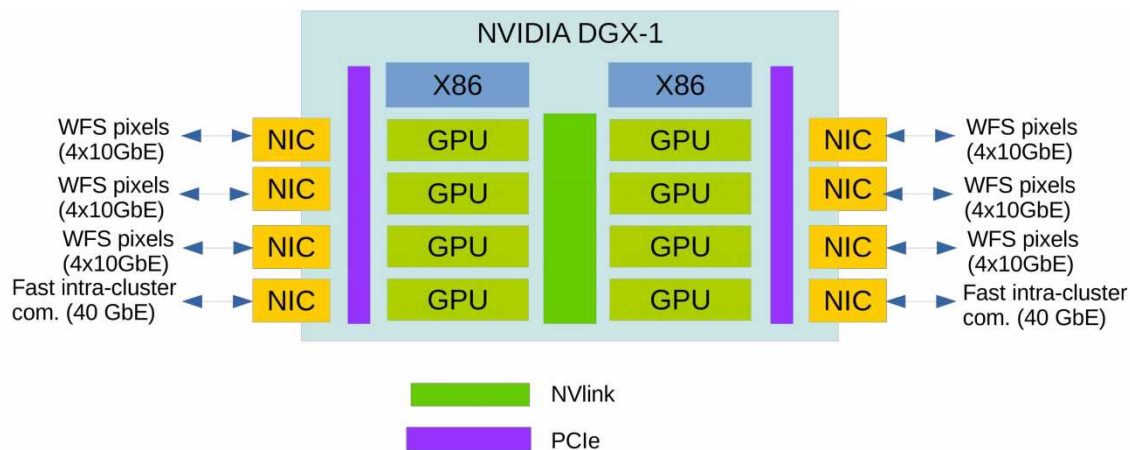


Illustration 5: GPU-based RTC prototype

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 17 of 49
Prototypes mid term report		

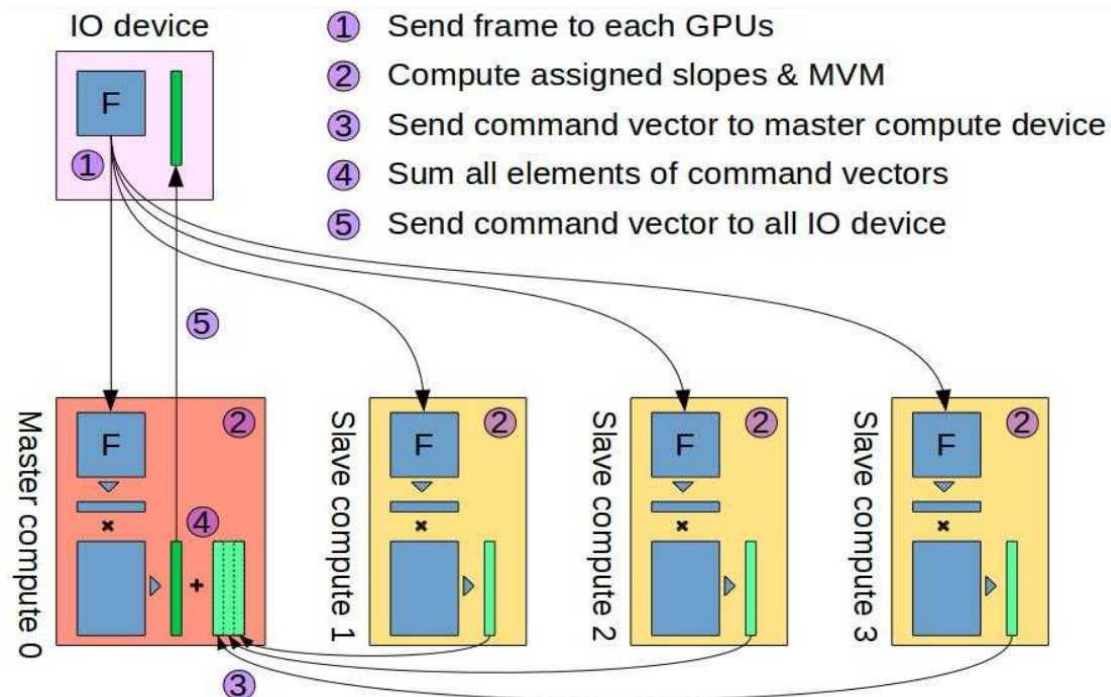


Illustration 6: Multi-GPU RTC test bench implementation

The strategy for the multi-GPU implementation is depicted on the left. One device is used for I/O and sends the data to a master device, synchronizing the execution and to several slave devices.

We performed a first series of test on a fraction of a MCAO system with:

- a 16 bit x 512² pixel frame
- 2 x 5024 slopes
- 15k commands
- 1-4 GPU for computing

The obtained results are assessed both in terms of execution time and communication latency and depicted in the two figures below

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 18 of 49
Prototypes mid term report		

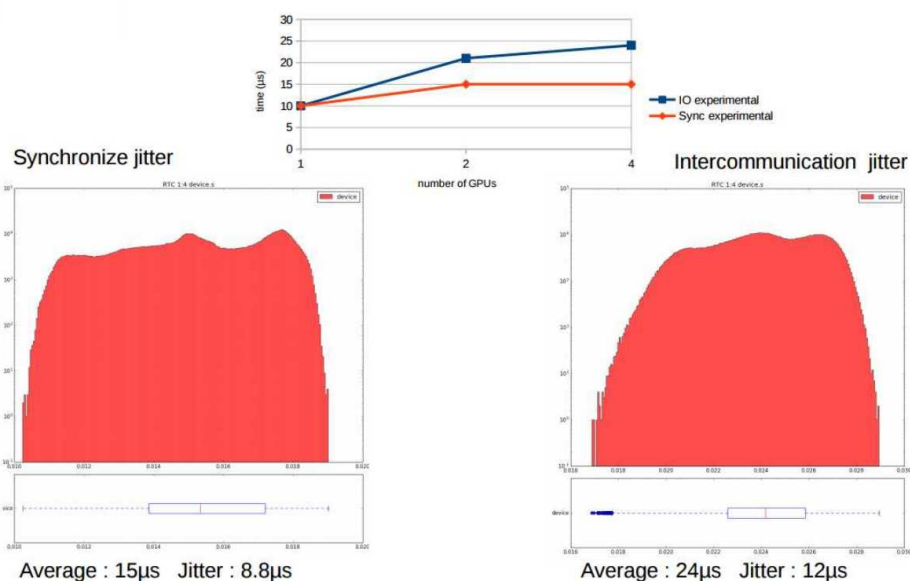


Illustration 7: Communication and synchronisation times and jitter on a multi-GPU implementation

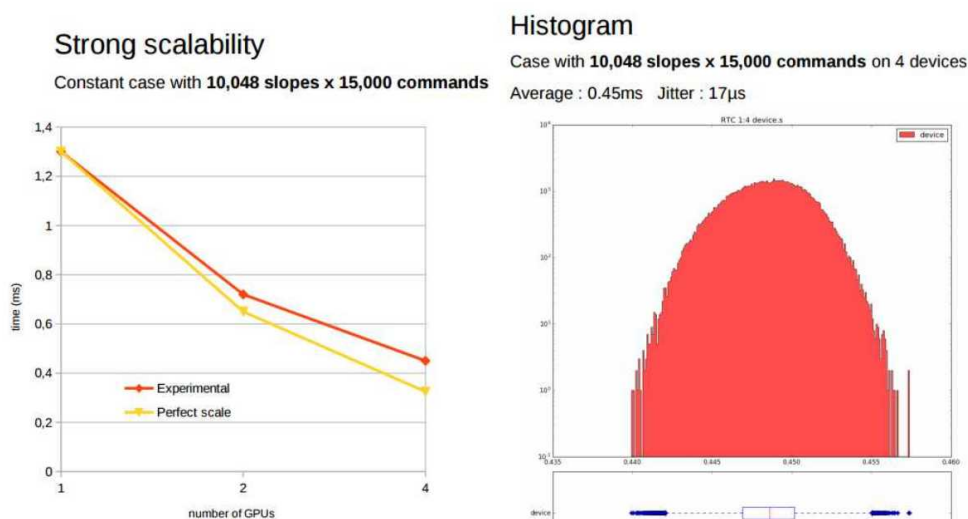



Illustration 8: Execution time on a multi-GPU implementation of a fraction of a MCAO case (10k measurements x 15k commands)

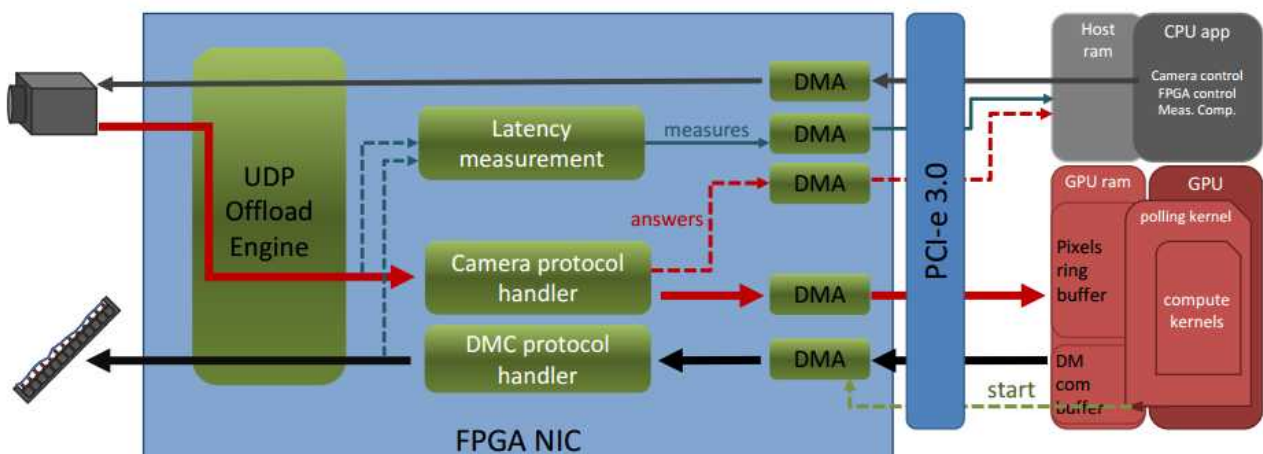
Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 19 of 49
Prototypes mid term report		

During the last stage of prototyping we will increase the computing complexity of the prototype up to the E-ELT MCAO case (80k measurements by 15k commands).

As stated above, data transfer to and from the GPUs is as much a concern as data transfer between GPUs. Additional developments are also led on the data acquisition scheme to enable direct communication between a third-party device (WFS camera) and the GPUs, through a dedicated FPGA interface. Our FPGA developments are based on a hardware agnostic environment brought by PLDA; its High Level Synthesis tool and IP-based programming model allow non-experts to quickly access the advantages of FPGAs by using C kernels. This is implemented in a dedicated prototype described below.


System description:

- Camera: Emergent Vision HS-2000M (10GbE GigEvision).
- FPGA Board: PLDA Xpress5 (Altera Stratix V, 4x10GbE SFP+).
- PCIe IP: PLDA QuickPCIe. It supports 8-lane PCIe 3.0, but our mainboard chipset is limited to PCIe 2.0.
- GPU: Nvidia Tesla C2070 (GPUDirect capable)
- UDP Offload Engine: PLDA QuickUDP
- Custom GigE Vision IP made at OdP.
- OS: Debian Wheezy, Linux 3.2



Detailed procedure:

- Creation of a buffer in the GPU memory: `cuMemAlloc()`
- Request of a token associated with the buffer: `cuPointerGetAttribute()`

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 20 of 49
Prototypes mid term report		

- Passing the token to a kernel module with a `ioctl()` request
- In the kernel module, getting the physical address of the buffer: `nvidia_p2p_get_pages()`
- Programming the NIC DMA engine registers with the physical address, size and everything needed.
- The main application launches the DMA engine and the camera using the GVCP protocol, and then waits for the buffer to get filled (by polling a flag in a DMA engine register).
- When the image header is detected in the FPGA, time t_0 is recorded.
- When the first image has arrived in the GPU memory, the application launches the computation on the GPU, and waits for the end of it.
- Then it launches a second FPGA NIC DMA engine to transfer the results (mirror commands) from the GPU to the FPGA (so here, it's a reading process over PCIe).
- When the results header is detected, time t_1 is recorded, and the value $(t_1 - t_0)$ is transferred to the main memory.

This way, we get the latencies shown on illustration 4. The size of the images was 64x64, pixels being coded on 16 bits. The latency improvement brought by peer-to-peer over PCIe is clearly visible, while the jitter doesn't change much when computation is performed.

Additional prototyping is now being conducted to enable the next generation of camera simulator (codenamed fakeCam design), which will be able to produce in real-time WFS images by simulating stars, atmosphere turbulences and deformable mirror behavior, enabling to precisely measure the performance of RTCs for AO and the impact of the jitter on Strehl ratio. The design of this modular prototype is depicted in illustration 9.

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 21 of 49
Prototypes mid term report		

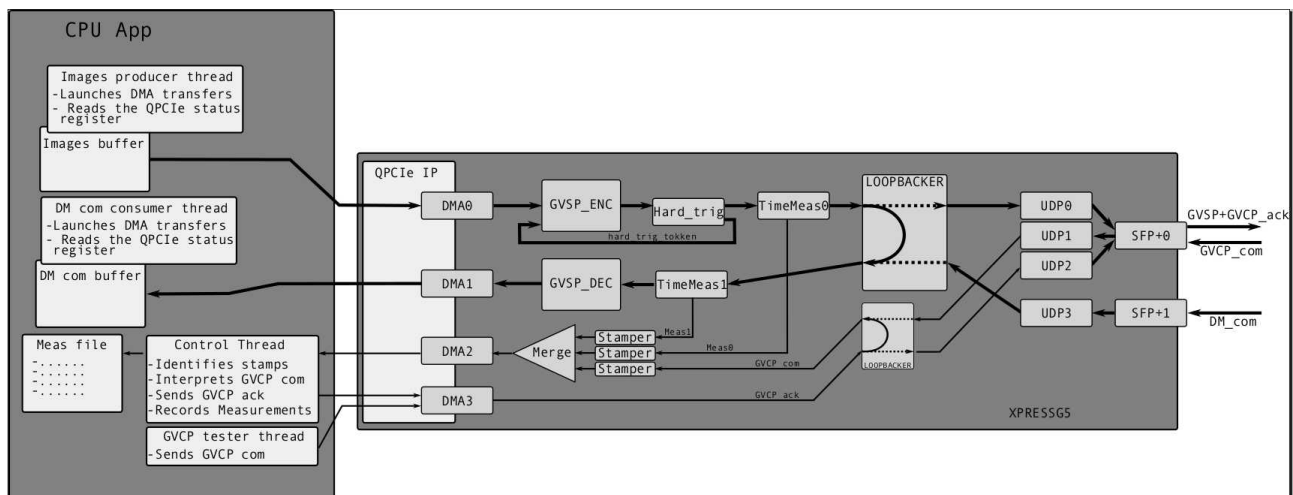


Illustration 9: Next generation camera simulator (a.k.a. fakeCam) design


FakeCam prototype design description :

- **GVSP_ENC:** C Kernel. Encodes images in GVSP format. Wait for image availability and **Hard_trig** signal
- **Hard_trig:** HDL kernel. Send flags are a regular (tweakable) rate.
- **Time_meas:** HDL Kernel. Time measurement between GVSP packets sent throughn AXIstream.
- **Loopbacker:** HDL Kernel. Configurable Loopback.
- **GVSP_DEC:** C Kernel. De-encapsulates data from GVSP frames
- **Stamper:** Mix HDL/C. Stamps packets depending on their source so as to be able to merge various data flux

Initial performance measurements with a scaled up dimensioning (2048*512*16bit images) lead to encouraging results with a data rate of about ~700 images/s. Jitter can be adjusted using the **hard_trig** function that can simulate efficiently the trigger usually sent to real cameras. When the **hard_trig** box is just a standard link (without any blocking mechanism) the jitter reaches about 30µs, which is well within specifications. The Work on this prototype will continue until the end of the prototyping period, in order to provide an efficient data acquisition / transfer module that could be implemented in the data simulator, in the RTC box and in the supervisor.

Intel Xeon Phi

The architecture described is based on the use of Multi-Integrated-Core (MIC) technology by Intel, also known as Xeon Phi.

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 22 of 49
Prototypes mid term report		

The Intel Xeon Phi is a many-core CPU (64-72 cores, depending on model). It is a standard CPU, it is self hosted, self booting and runs standard Linux and uses standard compiler tools. The Phi has 16 GB of high bandwidth memory (320GB/s) and wide vectorization unit (16 floats operated on simultaneously in each core).

Our prototype design is based on our experience from previous work with Xeon Phis (published in GF-D4.3a “MIC cluster for RT-box design and test”, AD01) and on our on-going preliminary tests focused on the Green Flash objectives. We provide results for both previous generation (Knights Corner, KNC), and the new generation (Knights Landing, KNL) systems, for completeness, though we have not yet had sufficient time to fully evaluate the KNL system. An important difference between the two systems is that the KNC is an accelerator, the device is connected to the computer via a PCI-e slot, whereas the KNL is not an accelerator, but rather, the core CPU of a server is a KNL.

The Xeon Phi has the advantage of offering a full open-source capability, with no requirements for closed source drivers or other code. Such a system can be written entirely in conventional software languages, and therefore has long lifetime expectations, and is easily transferable to other CPU-based technologies as they become available, with little or no change. Such lifetime considerations are extremely important for instrument systems with projected lifetimes spanning several decades.

The study has followed a staged process, stepping up as resources become available.

1. Knights Corner accelerator and native tests
2. Knights Landing CPU tests
3. Prototype design and definition
4. Xeon Phi for supervisor applications

Below we review the work so far but direct the reader to AD01 for more details.

Computational requirements

Here we summarize the size of the computational load. In GreenFlash, we are considering two types of AO systems: MCAO and SCAO.

The SCAO system has two options for the WFS: a Shack-Hartmann WFS and a Pyramid WFS. The Pyramid WFS produces about ten times less data than the SH WFS and is also computationally less intensive than the SH WFS. Therefore only the SH WFS is considered when designing the system, as it is more challenging.

The RTC shall be capable of controlling an 80x80 (74x74) AO system at the following frame rates:

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 23 of 49
Prototypes mid term report		

• MCAO: 500 Hz, 2 ms camera integration time, • SCAO: 1000 Hz, 1 ms camera integration time. This leaves 2 ms for the RTC processing in case of MCAO, and 1 ms in case of SCAO.

Knights Corner

The Knights Corner is used as an accelerator device. The device is connected to the computer via a PCI-e slot. There are two basic modes of using the Knights Corner: the accelerator mode and the native mode.

- In the accelerator mode, the main program is running on the CPU and it “offloads” the computationally intensive tasks to the Knights Corner, because it can perform the calculation faster than the CPU. The input data is copied from the CPU to the Knights Corner, then the Knights Corner performs the calculation and in the end the output is copied back to the CPU. The calculation must be performed fast enough to make up for the extra time spent on copying the data between the CPU and the Knights Corner.
- In the native mode, the Knights Corner runs its own operating system and can in several perspectives be seen as an independent node with a key limitation that it has no interfaces to the outer world and cannot be directly connected, for example, to a camera and a DM. In this mode, the Knights Corner is similar to Knights Landing, which indeed is a separate node but can be interfaced to a camera and a DM. As a preparation for the Knights Landing, we studied the Knights Corner in the native mode.

For our tests, we used a single Knights Corner model 5110P.

Both the accelerator and native modes have been implemented. Results and analysis can be found in AD01. The results obtained using Xeon Phi Knights Corner, in the native mode agree with those using Xeon Phi in the accelerator mode. The MVM for an 80x80 SCAO system can be performed in 1.5 ms. The current implementation in the full AO RTC still needs to be improved. From our experience with other hardware, we expect that the full AO processing cycle will be at the most 50% longer than the MVM alone.

Knights Landing

An important difference between the Knights Corner and the Knights Landing is that the Knights Landing system is not an accelerator, but rather, the core CPU of a server is a Knights Landing. Therefore we do not need to consider aspects such as data offload to an accelerator, and the system design is greatly simplified.

RTC pipeline operations

The Xeon Phi KNL has been tested with a simple code performing basic RTC operations including image calibration, slope calculation and reconstruction. For an SCAO ELT case (74x74 subapertures) a frame rate of 1.2 kHz has been achieved (800 μ s computation time). This simple

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 24 of 49
Prototypes mid term report		

case is to provide a maximum performance benchmark but is not pipelined (illustration 10) and is therefore not suitable for real RTCS operation.

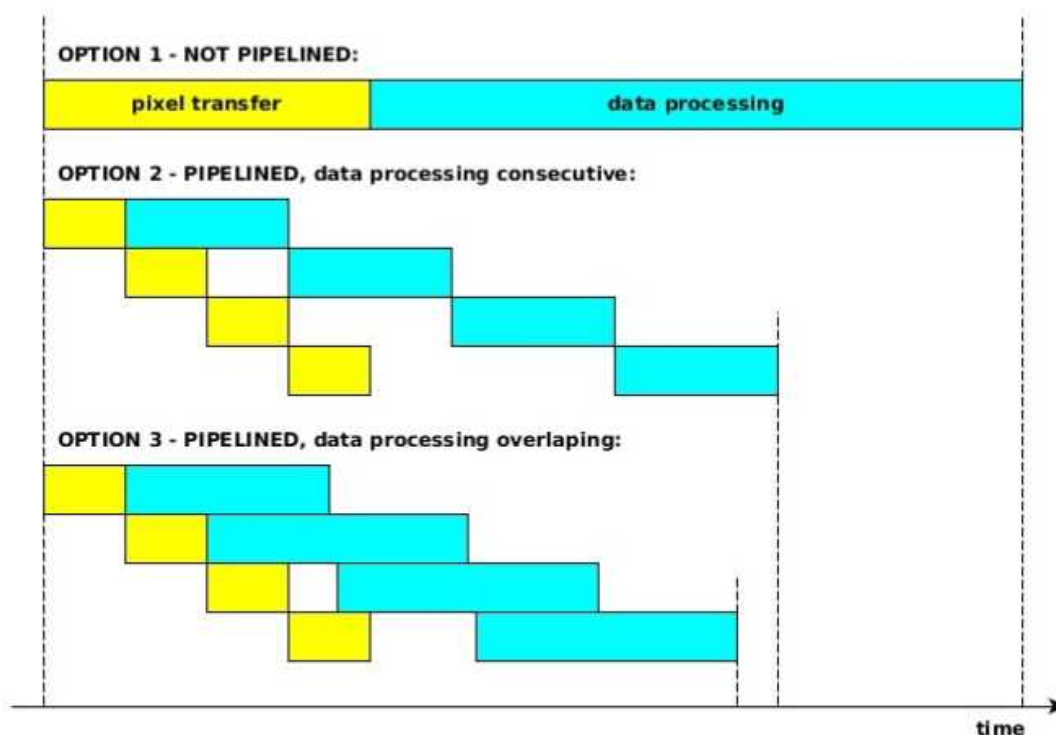



Illustration 10: Three options for data processing: non-pipelined, pipelined with consecutive data processing and pipelined with overlapping data processing

When the last set of pixels has been received and processed, the partial DM commands of all the sets are added together to obtain the final DM commands which are then passed on to the DM. The pipelined data processing will be implemented for SCAO and MOAO but for the Shack- Hartmann WFS only (not for the Pyramid WFS), with the following reasoning:

- For the Pyramid WFS one could in principle also implement it, the details being slightly different, but due to a much smaller amount of WFS data for the Pyramid WFS the expected gain in time would be negligible.

For MCAO, it is also necessary to compute pseudo-open-loop slope measurements, which requires an additional MVM (with a smaller matrix). However, this operation can be performed prior to first pixels arriving, and therefore does not increase the pipeline computation time significantly.

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 25 of 49
Prototypes mid term report		

A more complicated test case has also been performed by implementing the Durham AO Real-time Controller (DARC). DARC is the RTC used for the on-sky Canary AO instrument. The standard version of DARC compiles without modification on the Phi KNL, but offers low performance. We have therefore been making optimisation to improve this performance, and DARC is now able to operate at over 1kHz for the EELT SCAO case (74x74 sub-apertures). Optimisations include:

- More extensive use of vectorisation and memory alignment
- Optimisation of thread synchronisation primitives
- Improved algorithm for final DM vector computation
- Balancing of thread computation, CPU shielding, boot-time kernel parameters
- Compiler optimisation options
- Environment variables.

Future work will investigate other optimisations, namely in the area of image calibration and reconstruction. However we anticipate mostly incremental gains in performance.

Table 1 shows the summary of performance achieved with the Xeon Phi and illustration 11 shows the MVM computation time for various matrix sizes. We see that this scales quadratically with matrix size as expected, though (probably due to the internal implementation of the MKL library used), there is some variation from a quadratic scaling depending on exact matrix size.

	Knights Corner	Knights Landing predicted	Knights Landing preliminary results
MVM computation time	1.2-1.3 ms	0.9 ms	0.76 ms
Memory Bandwidth	165 GB/s	250 GB/s	258GB/s TRIAD, 235 GB/s MVM

Table 1: Summary of the matrix-vector multiplication results obtained with the Knights Corner and with the Knights Landing. The Knights Landing results in the last column are preliminary only. Matrix size is 9440 x 4720 elements.

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 26 of 49
Prototypes mid term report		

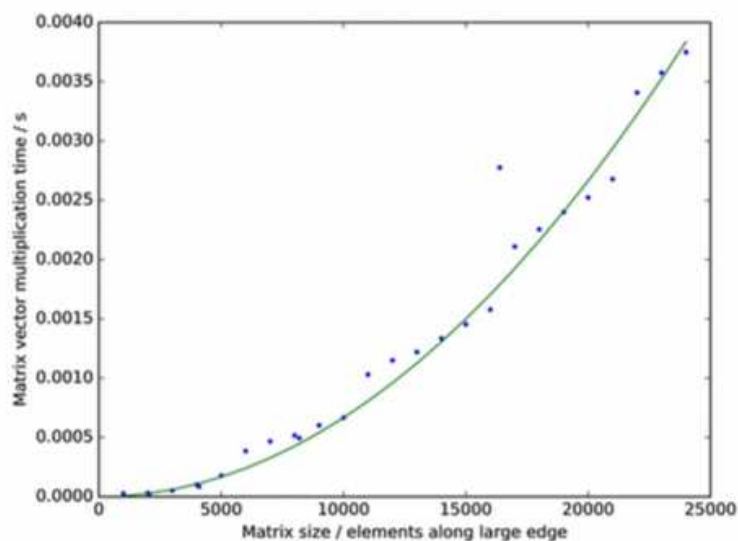



Illustration 11: MVM time on KNL system. The matrix is twice as long as it is wide and the longer dimension is given in the x-axis. A quadratic fit is shown for comparison.

Xeon Phi prototype design

In addition to optimizing existing hardware for AO RTC we have also ordered the hardware for a full prototype system. This system consists of Xeon Phi server including 4 x 7210 and 4 x 7250. We also have a 10Gb/s camera which will be interfaced to the system over 10Gb/s ethernet. Illustration 12 shows with more detail a solution for a E-ELT MCAO system using 7 Xeon Phi systems (including the NGS). Since the NGS are lower order, they are handled together by one Phi. The 6 LGS are each connected directly to a KNL system which calibrates, computes wavefront slopes and performs partial reconstruction. These partial DM vectors, of approximately 10k elements (40kB) are then passed to one Phi which combines and passes the result to the DMs. This Phi is also responsible for computation of pseudo-open-loop slopes, which it does during dead-time while waiting for NGS pixels to arrive.

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 27 of 49
Prototypes mid term report		

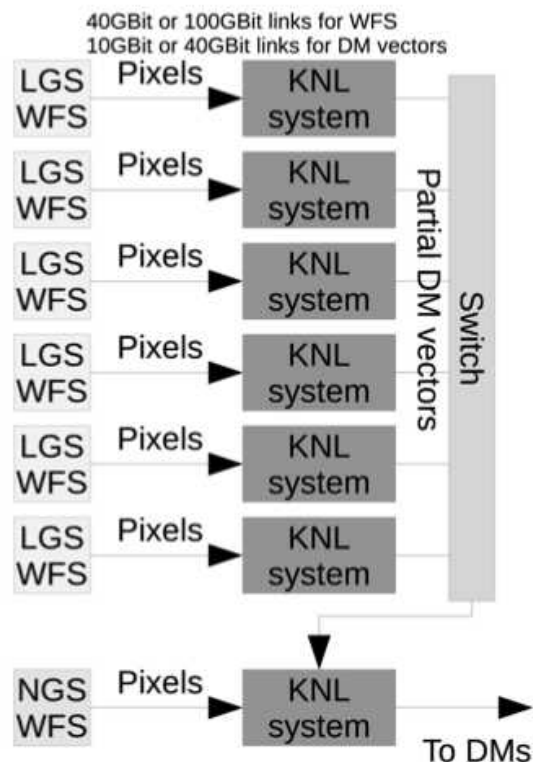



Illustration 12: Conceptual design for a full MCAO ELT system using Xeon Phi systems

COTS FPGA technologies

A COTS FPGA cluster is studied for the real-time data pipeline, especially the demanding reconstruction computing (MVM). The cluster is designed to have a scalable multiple FPGA architecture. The performance can be improved by adding more FPGA hardware and each FPGA node runs identical firmware. The FPGA will provide low latency and very low jitter, ideal for systems that require such deterministic characteristics. A standard protocol such as UDP based on 10GbE can be used for communication to the rest of the system.

This FPGA cluster is designed as an accelerator for the AO control loop data processing, especially for the reconstruction. The main design is in line with the Greenflash project requirement. Some features are highlighted as, -

- Scalable: both the communication bandwidth and the processing power can be adjusted for different computing requirement or different budget;

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 28 of 49
Prototypes mid term report		

- Upgradeable: When new hardware is available, the FPGA application can be upgrade to the new platform with relatively small effort;
- Programmability: An FPGA application can be constructed without much concern about the running hardware or low level signal timing, which should dramatically shorten the development time and/or cost.

Other cluster criteria, such as reliability and manageability, would still remain as key specification.

Proposed Solutions


This design of the FPGA cluster consists of an access node and many internal work nodes. The single access node provides an external interface for accessing the cluster computing resource. The communication of such interface is standard UDP, preferably through a 10GbE Ethernet port. This node translates the external request to internal command and communicates with the internal working nodes.

The internal work nodes are connected by their internal interface, which can be some standard only available with FPGAs for better efficiency and easier programming, for example, Xilinx Aurora.

The access node and work node are logical nodes, which means an access node and a work node can possibly coexist on the same piece of hardware, especially for many AO applications, such as WPU and MVM, are memory bandwidth constrained, while the translation between the external request and the internal command does not require large buffer.

The daisy chain cluster is simple to implement and manage. Although the processing power has good scalability, its communication bandwidth is almost fixed. In case of the interface based on 10GbE was saturated due to demanding data exchange, either the communication interface can be upgraded, e.g. 40GbE or multiple such FPGA chain should be employed.

Normally an AO real-time control system does not need to continuously run over days or months, which makes the reliability less restrictive. In the event of hardware failure, the hardware board can be swapped in a short time, since it is possible for all the work nodes to have the same copy for firmware and the access node firmware can be loaded in field.

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 29 of 49
Prototypes mid term report		

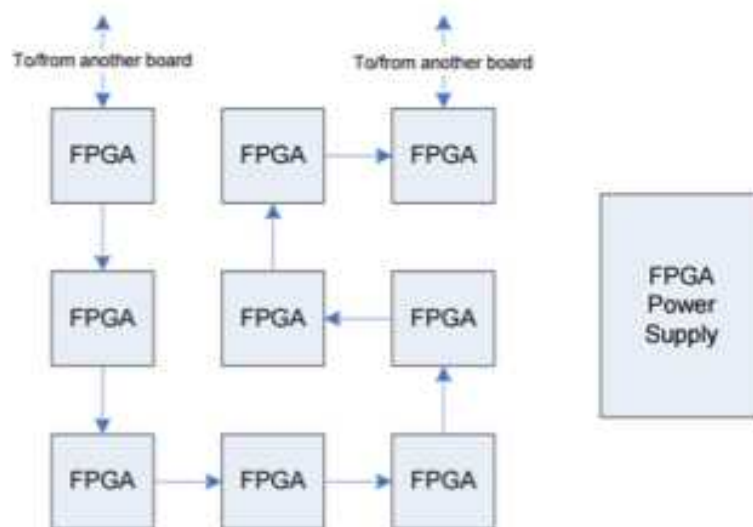



Illustration 13: A simple FPGA implementation is proposed as a daisy chain of FPGAs. The command from the access node is passed on through the chain, and each FPGA can insert its result into the chain and being passed for next stage processing, finally the result comes to the access node, which translates the result and sends out through its external 10GbE UDP interface

The matrix-vector-multiplication(MVM) can be easily distributed over multiple computing nodes by dividing the matrix among the FPGAs and broadcasting the incoming vector for the partial MVM result, and eventually accumulating the partial results together to complete the MVM calculation.

Selection of a COTS FPGA hardware is a 3D job among different FPGA product line, different board manufacture and different FPGA maker, i.e. Xilinx, Altera, Lattice, etc. The high end FPGA boards may not be suitable for a cluster due to their high cost, while economic FPGA solutions don't always offer the latest technologies, such as 10GbE and DDR3/4 memory interface.

Professional line of FPGAs should be considered, for example, Xilinx Kintex Ultra Scale series. Such FPGA boards are available from Xilinx as their official development hardware, as well as many other FPGA hardware vendors, like PLDA.

Illustration 14 defines the FPGA cards under consideration. Both can be prepared using the

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 30 of 49
Prototypes mid term report		

QuickPlay tool meaning that the MVM kernel can be developed in C. Meanwhile common component can still be developed at RTL level with VHDL to provide better efficiency and reduced latency.

PLDA Quickplay helps in a few ways in the FPGA development especially for the AO real-time system. Firstly, it brings up a system architecture based on the Kahn Process Network (KPN), which connects multiple processing kernels via FIFO based stream interfaces. This model matches the data flow of an AO RTCS. Secondly, Quickplay provides a high level synthesis allowing the application model to be described in C style programming language. So the function can be simulated and debugged with the readily available C tools. Furthermore, the Quickplay is an integrated environment. The host system interface and the standard communication interface are all provided and tested. The developer can almost concentrate entirely on the user application.

FPGA Boards

Board	DDR	Speed	Peak BW (GB)	GFLOPs (80%)	Cost (EUR)	Source
KCU105	DDR4	1200	19.2	3.84	3252.43	Digikey
XpressKUS	DDR3	933	14.928	2.9856	4990	PLDA

FPGA Cost


Instrument	Subaps	DM channels	Matrix size	Freq	BW (GB)	BW (GFLOPS)	KCU 105 boards	KCU 105 cost (kEuro)	XpressKUS Bords	Xpress cost (kEuro)
Harmoni	21904	4326	189513408	800	606.44291	151.6107264	40	130.0972	51	254.49
Micado	32856	10000	657120000	500	1314.24	328.56	86	279.70898	111	553.89
MOSIAIC	32856	36479.6	2397147475	250	2397.1475	599.2868688	157	510.63151	201	1002.99
HIRES	32856	4326	284270112	500	568.54022	142.135056	38	123.59234	48	239.52
EPICS	40000	49053	3924240000	3000	47090.88	11772.72	3066	9971.95038	3944	19680.56

Illustration 14: FPGA cluster examples. Two types of FPGA board are evaluated as defined in the upper table. The lower table shows an estimate of the number and cost of the boards to solve varying problem sizes corresponding to same of the E-ELT instrumentation.

Note that the FPGA cards shown are generic FPGA hardware, which haven't utilized the full potential of the FPGA performance regarding the key specification of the MVM, i.e. the memory bandwidth. For a few smaller design cases the FPGA can still show reasonable cost. However, for most ELT level instruments the FPGA needs optimized hardware to provide affordable solution.

Supervision strategy

The supervisor module feed the real-time box at a regular rate with a tomographic reconstructor

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 31 of 49
Prototypes mid term report		

matrix, computed from a statistical analysis of the measurements. This process involves dense arithmetics of the covariance matrix generated from the WFS measurements, the size of which being of the order of $110k \times 110k$. These matrix arithmetics are composed of matrix factorizations and basic linear algebra operations. The overall process can be represented as an embarrassingly parallel problem with a numerical complexity scaling with N^3 , from which an optimized implementation should maximize the usage of computing cores rather than the memory bus. However, the underlying standard algorithms require frequent synchronizing of global communications, which represents a bottleneck that may impede performance. Optimizing the compute performance means larger update rate of the reconstructor matrix hence better image quality at the output of the telescope and incidentally larger science return. A full pipeline, following the so-called learn & apply approach, including the experimental covariance matrix generation using noisy data from the instrument, includes the identification of critical turbulence parameters through a fitting process of the latter on a theoretical model and the computation of the corresponding tomographic reconstruction. The supervisor module is thus a mix of cost function optimization for parameters identification (“Learn” process) and linear algebra for reconstructor matrix computation (“apply” process). The whole supervision algorithm is depicted in illustration 11.

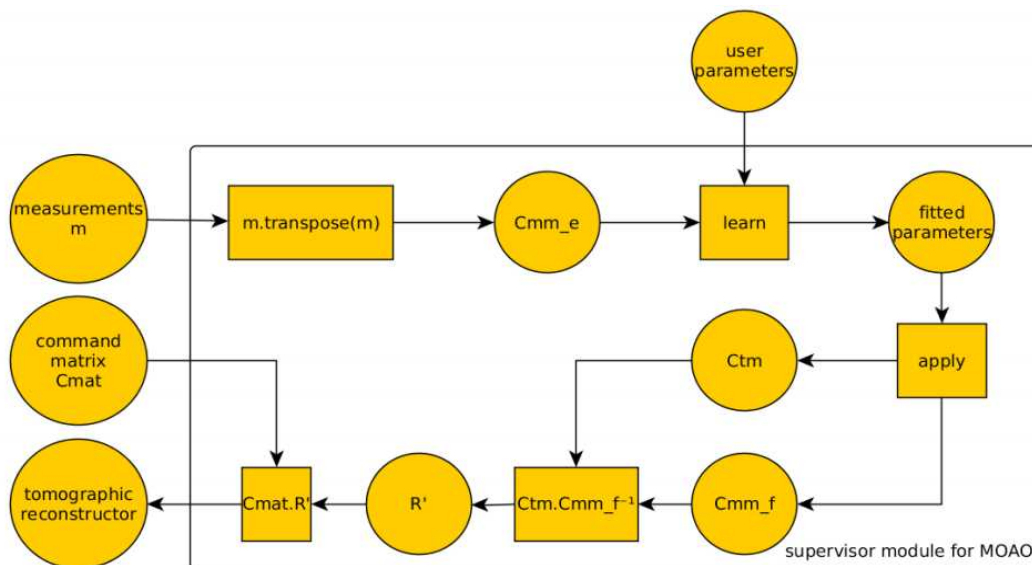


Illustration 15: AO supervision algorithm

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 32 of 49
Prototypes mid term report		

Learn process

The parameters identification stage (“Learn” process) is intended to fit the measurements covariance matrix on a model including system and turbulence parameters. To do so, we use a score function optimized using the Levenberg-Marquardt algorithm.

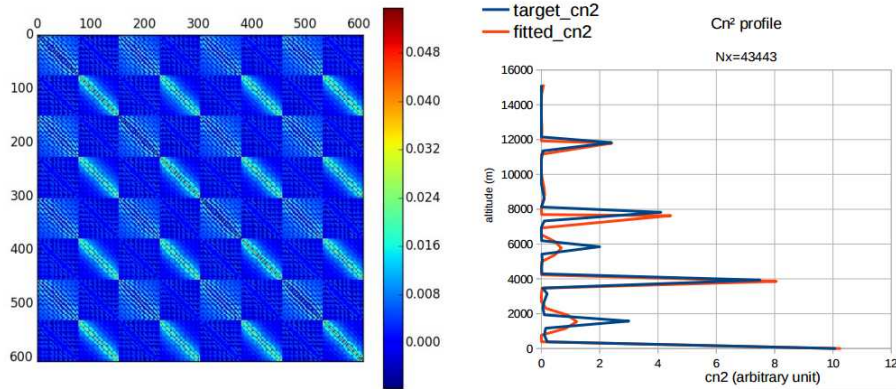



Illustration 16: Example of a measurement covariance matrix (left) and fitted turbulence profile (right)

A multi-GPU implementation of the latter has been produced during the first phase of the project. It consists of a dual stage process. In the first pass, only a limited number of turbulence layers are considered (5 layers) and in the second pass more layers are added to meet the system specifications (up to 40 layers). Initial performance analysis was done on a realistic E-ELT MCAO case on the multi-GPU DG-X1 platform, including matrix generation and LM for a matrix size of 86k and is outlined below.

The time-to solution reached is 240s (4 minutes) including 25s for the first pass and 213s for the second pass. The weak and strong scaling of this process are depicted in illustration 13 for various matrix sizes and various number of GPUs. It shows an excellent behavior, very close to the perfect case for weak scaling and an impressive >90% efficiently for strong scaling on multiple GPUs. This is very encouraging for the supervision strategies on the E-ELT since the time-to-solution is already very close to the initial instrument specifications (with a reconstructor update every minute).

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 33 of 49
Prototypes mid term report		

Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz + 8 P100
First LM : 25.5s Second LM : 213,8s

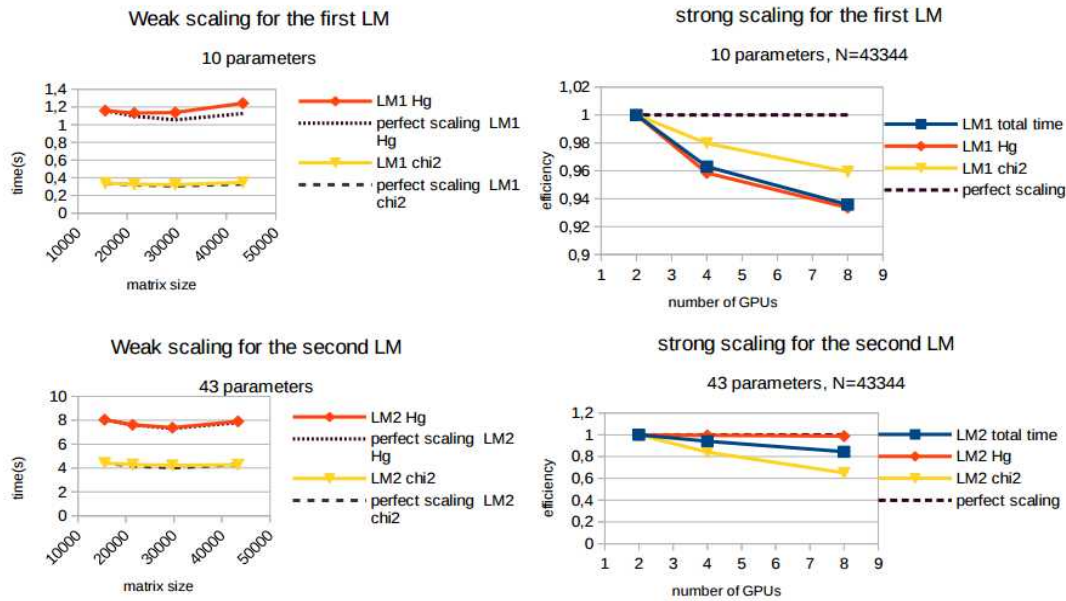



Illustration 17: Weak and strong scaling of the multi-GPU learn process

Apply process

Concerning the reconstructor matrix computation (“apply” process), it aims at computing the tomographic reconstructor matrix using covariance matrix between “truth” sensor and other WFS and invert of this covariance matrix. Several methods have been assessed: LU or Cholesky factorizations or “brute” force using a direct solver. The latter seems to be the most efficient one since it is mostly compute-bound and exposes a high level of scalability. In the context of the Green Flash project, and in collaboration with the Extreme Computing Research Center at KAUST, we have developed an efficient implementation of the control matrix computation for AO on multicore system with multiple GPUs using high-performance numerical library for solving large dense linear algebra problems. The high performance implementation relies on the use of a dynamic run time system to schedule computational tasks simultaneously on various compute devices and a data flow programming model based on the use of direct acyclic graphs for an efficient scheduling in which the tasks are executed out-of-order and scheduled according to a critical path for the execution. The code was built to be highly portable, so as to explore various architectures by using standard vendor provided maths libraries. The obtained results for various matrix sizes are depicted in illustration 14 and 15.

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 34 of 49
Prototypes mid term report		

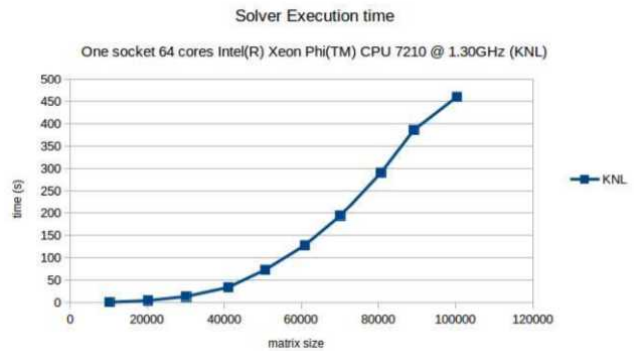
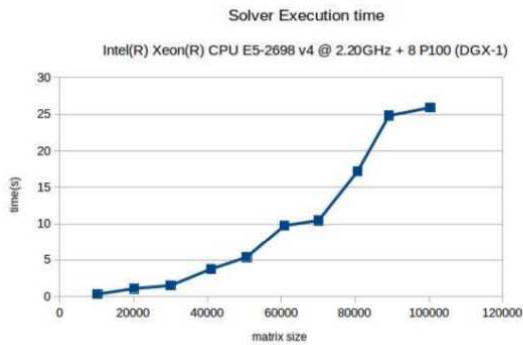


Illustration 18: Time to solution to compute the tomographic reconstructor matrix for various matrix sizes on two different architectures : NVIDIA P100 GPUs and Intel KNL Xeon Phi

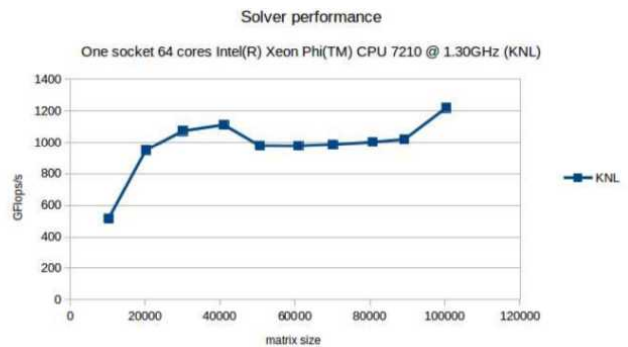
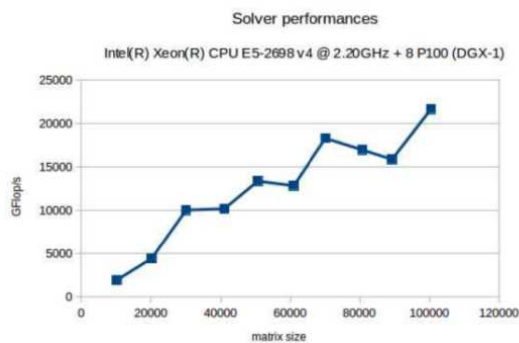



Illustration 19: Peak performance achieved for the direct solver for various matrix sizes on two different architectures : NVIDIA P100 GPUs and Intel KNL Xeon Phi

GPUs can deliver better peak performance. The saturation is not reached and we expect >2.5 or better with larger matrices. Moreover, the NVlink interconnect seems to perform very well in this multi-GPU platform. Finally, record time-to-solution is obtained on the DGX-1, as a MAORY / HARMONI full scale process (100k x 100k matrix) is addressed in 25sec to compute tomographic reconstructor, which is well within the system specifications.

There is, however, room for improvements as the current performance scalability can be further enhanced by reducing data motion and increasing data locality within GPU memory. Furthermore, the various covariance matrices are hierarchically low rank, which could be approximated and therefore, exploited to reduce the arithmetic complexity and the memory footprint. Last but not least, we would like to port the whole framework to ARM platforms and to assess the performance obtained with various hardware accelerators (e.g., Intel Xeon Phi, AMD APUs) both in terms of


Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 35 of 49
Prototypes mid term report		

compute performance (time-to-solution) and energy efficiency. This involves the implementation of an ARM-based cluster and the validation of the whole required software stack, on this rather emerging platform. For instance, the ATLAS library is currently the only one supported as the de facto BLAS library. On the other hand, several accelerators solutions could be studied depending on the availability of corresponding drivers, e.g., for the ARM, IBM Power8+, AMD APUs and the Intel Xeon Phi platforms. Performance will be assessed in terms of throughput and scalability using the clAmdBlas and MKL libraries for AMD APUs and Intel Xeon Phi, respectively. Performance comparisons will be done against results obtained so far on multiple GPUs using NVIDIA CUBLAS library. From a data structure perspective, the idea will be to exploit the hierarchically low rank structure of the off-diagonal blocks of the various covariances matrices being operated on. These blocks could be approximated, which will translate into reduction of the arithmetic complexity as well as the memory footprint, both being paramount to sustain the real-time requirements and the energy efficiency of the AO instruments. This new data structure (H-Matrix) necessitates the development of new numerical algorithms to perform matrix computations, such as matrix factorizations and basic linear algebra matrix operations. This is the goal of a PhD thesis starting in co-tutelle between OdP and the KAUST University. These algorithmic developments will take place within the Hierarchical Computations on Manycore Architectures (HiCMA) project, developed as an open-source library among the ECRC group at KAUST.

All this topics will be addressed during the second phase of prototyping of the Green Flash project to build a full pipeline for the supervision strategy. The final implementation should be consistent with the required rate of operation and its performance should be assessed on various architectures for future down-selection as the instruments enter the final design phase. To support this activity on the supervisor module, a PhD thesis has started in November 2016, through a *co-tutelle* scheme between OdP and KAUST.

Xeon Phi for supervisor applications

Whilst this work is not strictly a part of the accelerator workpackage, we are also investigating the application of Xeon Phi hardware to the supervisor. Calculation of an AO control matrix is necessary to enable wavefront reconstruction when an MVM is used. Since this is the default case that we are considering here, it is therefore applicable to consider the use of the Xeon Phi for control matrix calculation. Typically, to compute a control matrix, there are a number of operations that are performed, principally including matrix-matrix multiplications, and a matrix inversion. The control matrix should be updated on time-scales within which the atmospheric profile changes, which can typically be as short as tens of seconds to minutes. Instrumental effects may also require control matrix update, for example relative rotation of wavefront sensors and DM actuators. Therefore, it must be possible to compute a control matrix on a ten second time-scale. For the SCAO case, matrix multiplication of an approximately 5kx10k element matrix with its transpose is first required. The inversion of a 5kx5k element square matrix follows, followed by matrix

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 36 of 49
Prototypes mid term report		

multiplication of a 5kx10k element matrix. For the MCAO case, the same order of operation can be used (depending on the type of control matrix desired), though the matrices in question are of approximate size 10kx60k and 10kx10k. Illustration 7 shows matrix inversion time on the Xeon Phi KNL system using LU decomposition. The BLAS functions SGETRF followed by SGETRI were used, taken from the Intel Math Kernel Library. It would be trivial to replace these functions with standard BLAS calls from other libraries (e.g. OpenBlas, ACML, etc). The line shown on the plot is a cubic function passing (arbitrarily) through the point at 10,000 elements. Therefore, it can be seen that matrix inversion time follows the cube of matrix size as expected. We note there are a few sharp deviations from this line, which are probably due to the internal implementation of the MKL functions. Future releases of MKL may reduce these spikes, however, this means that the precise size of the matrix is an important consideration for inversion time. We suspect that this is due to the internal architecture of KNL, and would be fixed in future releases, since this is a new technology. For many algorithms, powers of two sizes give better performance (e.g. FFTs). We also note that inversion via Cholesky decomposition is known to be about twice as efficient as LU decomposition, and therefore inversion times could halve. However, Cholesky decomposition does not necessarily compute a solution depending on the matrix form, and so we have not considered this here: once the form of the matrix is known, Cholesky decomposition can then be investigated. Matrix inversion is compute- dominated, and it is evident that only for larger matrix sizes does placing the matrix in HBW memory lead to reduction in computation time. From this information, we see that it would take less than 2 seconds to invert a 10kx10k matrix and about 0.4s for a 5x5x matrix (SCAO case).

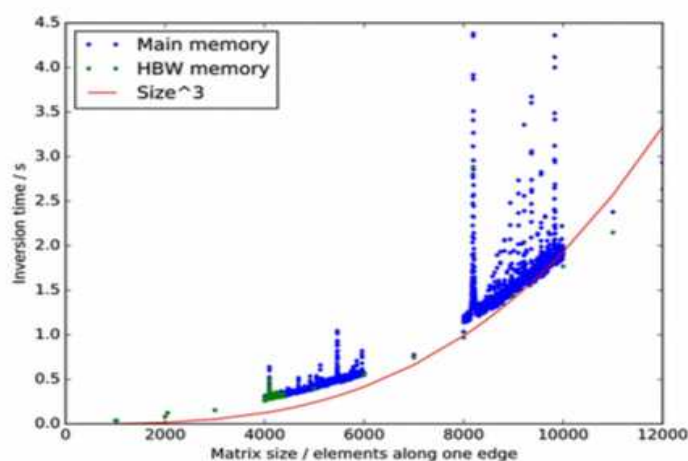



Illustration 20: Preliminary matrix inversion time using LU decomposition on a KNL Xeon Phi as a function of matrix size. Cases with the matrix in main and high bandwidth memory are shown, along with a best fit cubic function.

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 37 of 49
Prototypes mid term report		

Smart interconnect prototyping

Outline

The goal of WP5, under the responsibility of PLDA with significant contribution from Observatoire de Paris, consists in developing a prototype of Smart Interconnect. Such system, built around FPGA based boards, aims at providing a high performance low latency interconnect solution, based on standards and exploring the use of new smart features in the context of real-time control for AO.

Work Package was originally built around 3 tasks:

1. Development of a High bandwidth FPGA NIC card
2. Integration of standard middleware on top of Smart Interconnect
3. Improvement of development environment and integration of IP blocks

Work done during the first half of the prototyping period

At development environment level, following achievements were reached on QuickPlay:

- Improved maturity through:
 - Improved development flow, easing and speeding up IPs and boards integration and allowing integration of 3rd party IP providers.
 - High Standard development methodology (continuous delivery, automated validation and analysis capability)
 - Improved Performances and features through:
 - Improved proprietary HLS through directives usage, allowing higher computing performances
 - Vivado HLS integration
 - Improved SDK performances (PCIe and TCP layers)
 - Multi-board support (ready to be delivered)
 - Improved emulation and configuration (static or dynamic) of C and HDL Kernels (
 - Improved Tool accessibility, allowing SW engineer to easily target application on FPGA
 - Improved genericity through
 - Increased target families (Altera Stratix V / Arria 10, Xilinx Kintex-7 / Kintex-US, Virtex-US)
 - BSP delivery for increased set of boards (Reflex XpressGX5 / XpressKUS / XpressGXA10 boards)
- Note: Basic support of Microgate μ XComp board (PCIe + Ethernet 10G) is done under QuickPlay but could not be validated on HW, waiting for board availability.
- Increased IP portfolio and features:
 - 10G Ethernet RAW mode

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 38 of 49
Prototypes mid term report		

- UDP offload engine with UDP multicast
- DDR4
- Peer-to-Peer for PCIe
- C-Kernels for AO : GVSP (GeV) and CSKT (matrix exchanges) codecs

Development of High Bandwidth FPGA NIC card was driven by QuickPlay tool enhancement, and came to reality through Smart Interconnect development, which is a perfect application example since it requires all features proposed for such smart FPGA NIC.

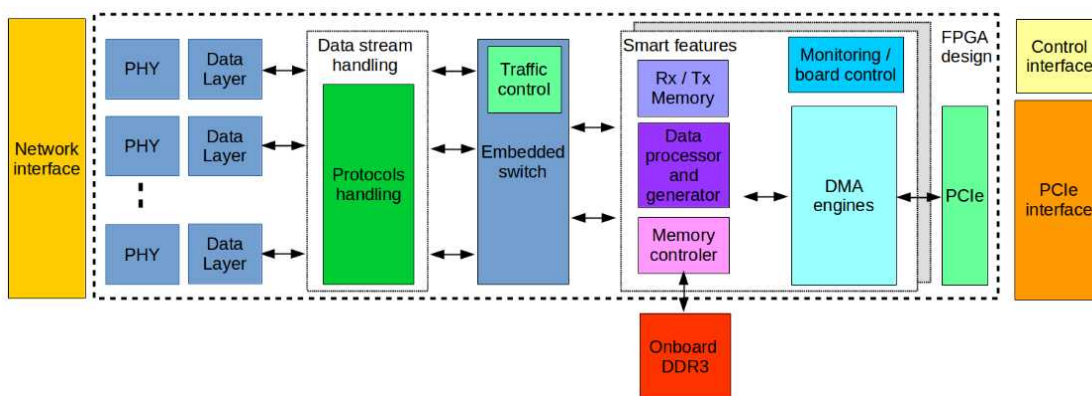


Illustration 21: Smart interconnect generic design

Indeed, Smart Interconnect development required:


- BSP availability for multiple boards
- Integration of specific IPs and features with TCP/UDP offload engines at 10 Gbps, PCIe gen3, DDR4 memory (and HMC to come)
- Data processing capability, intrinsic to FPGA and made simpler through QuickPlay and C-Kernels usage

Smart Interconnect prototypes allowed to demonstrate:

- QuickPlay Genericity, allowing design to be targeted on any of the supported board with performance difference
- Maximal TCP/UDP performances, with
 - 9 Gbps of effective data transported
 - A latency and jitter fully compliant with Greenflash project requirements

Smart Interconnect prototype design - Results

Focusing on GreenFlash prototype objective, a prototype of Smart Interconnect was built having in

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 39 of 49
Prototypes mid term report		

mind to be used for AO RTC demonstrator. This prototype is architected this way:

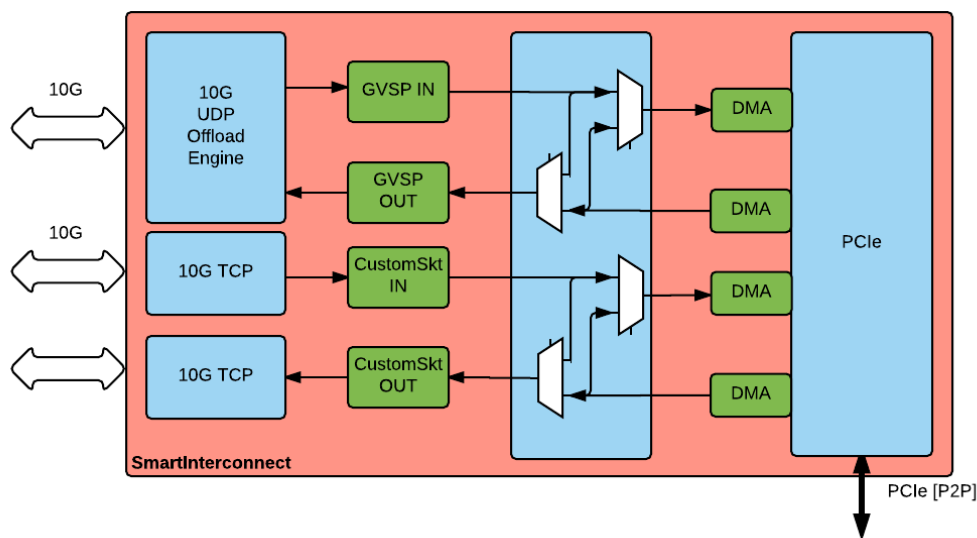


Illustration 22: Exemple design for unitary tests


Latest developments allowed to demonstrate following features and performances of such Smart Interconnect:

- Gen3 x8 PCIe End-Point with DMA and Peer-to-Peer capability (tested with Tesla K40 GPU)
- 10G Ethernet interfaces with full TCP/UDP bandwidth (9 Gbps of effective data carried)
- Real-time GeV (GVSP) and matrix (CSKT) encoding/decoding through C-Kernels
- Simple switching features through HDL Kernel integration

Simulator prototyping

The simulator has two main use cases:

- Real-Time Controller performance testing
 - Verify that RTC prototype(s) can meet key performance requirements in terms of frame rate, latency and jitter
 - Ensure results are valid when running at on-sky rates
- Algorithm and Interface development
 - Ensure that the algorithms implemented in the RTC give the expected results
 - Develop and test interfaces between components
 - Develop and test human interfaces to RTC

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 40 of 49
Prototypes mid term report		

The simulator has two main use cases:

- Real-Time Controller performance testing
 - Verify that RTC prototype(s) can meet key performance requirements in terms of frame rate, latency and jitter
 - Ensure results are valid when running at on-sky rates
- Algorithm and Interface development
 - Ensure that the algorithms implemented in the RTC give the expected results
 - Develop and test interfaces between components
 - Develop and test human interfaces to RTC

Requirements

There are three sets of requirements for the simulator:

- Adaptive Optics Requirements:
 - Simulate E-ELT scale AO system
 - Multiple Wavefront Sensors
 - Laser and Natural Guide Stars
 - Shack-Hartmann and Pyramid WFS Types
 - Received feedback to simulated DM(s) to provide AO performance estimate
- Computational Requirements:
 - Provide raw WFS pixel data
 - Frame rate up to 800Hz
 - Less than 1 μ s jitter
 - Provide 10 minutes of continuous data at full speed
 - Measure latency accurately by timing DM command time of arrival
- Interface Requirements:
 - Transport data over 10G Ethernet
 - Data should as closely as possible appear to originate from a real camera
 - Simulator should receive DM commands as feedback from GreenFlash telemetry

Proposed Solution

No single solution can meet all of the requirements. End-to-end simulations cannot reach on-sky rates with no jitter and deterministic hardware solutions cannot provide realistic data. Therefore, a combined approach is proposed. The solution consists of an end to end simulation to perform experiments with feedback to simulated DMs, hardware based (FPGA) solution to send data deterministically at high frame rates and an end to end simulation used to record large data sets to be sent by FPGA.

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 41 of 49
Prototypes mid term report		

Two primary modes of operation have been identified. Illustration 23 and 24 demonstrate the simulator architecture for the two (simulator rate and on-sky rate) modes. In simulation rate mode, data generated by the simulation is sent directly to the RTCS and DM commands are returned, providing feedback to the simulation. For on-sky rates the simulated WFS data is stored and then sent to the RTCS at real-time rates. An FPGA based data shaper intercepts this data and ensures that the RTCS receives the signal with deterministic timing. DM commands returned by the RTCS are saved in the data store for later analysis.

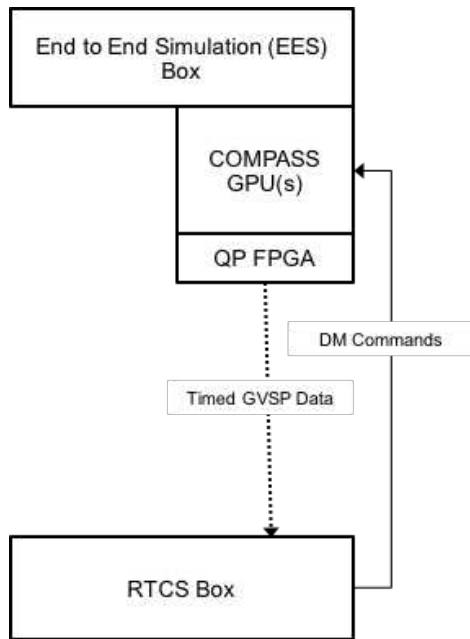


Illustration 24 Simulator architecture for the simulation rate mode where data generated by the simulation is sent directly to the RTCS and DM commands are returned and re-combined in the simulation.

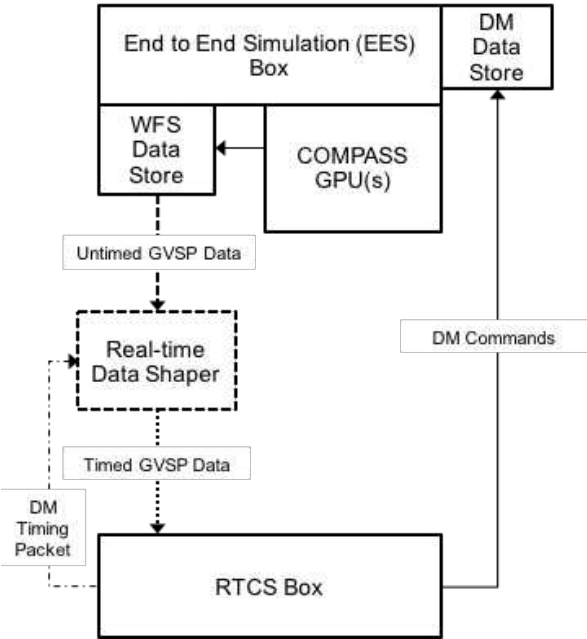



Illustration 23 Simulator architecture for the on-sky rate mode. In this case simulated WFS data is sent to a data store from where it is sent to the RTCS at on-sky frame rates. The data is intercepted by a real-time Data Shaper which is used to format the signal to have deterministic timing capabilities. DM commands from the RTCS are saved in the data store for later analysis.

Prototypes ecosystem

Middleware

There are three identified middleware domains.

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 42 of 49
Prototypes mid term report		

- Control
- Telemetry
- Low-latency pipeline

Abstraction is essential to realise high-level design on different hardware/middleware combinations.

The three middleware domains each have their own requirements.

- Control:
 - Request/reply pattern
 - Service discovery/location transparency
- Telemetry:
 - Publish/subscribe pattern
 - Hard throughput, weak latency/determinism requirements
- Low-latency pipeline:
 - Hard latency, determinism, throughput requirements
 - Fan-out/fan-in pattern – distribute workloads
 -

The following evaluations are in progress:

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

For the real-time pipeline the goal is to limit the total RTCS latency to one frame, at a frame rate of 1kHz this equates to a total latency of 1000 μ s. The goal for jitter on the latency is 100 μ s in any 1 second period. The latency in the middleware must therefore be significantly less than these values.

In illustration 25 we evaluate ZeroMQ for the real-time pipeline. The figure shows that the mean latency, even for small message sizes is unacceptable. The jitter on the latency is also unacceptably high.

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 43 of 49
Prototypes mid term report		

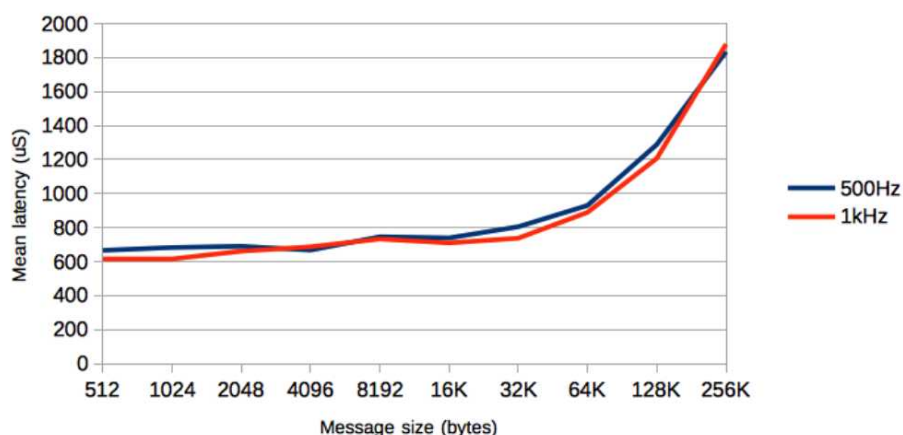


Illustration 25: Mean latency for ZeroMQ middleware running at 500Hz and 1kHz. For message sizes smaller than 32kb the mean latency is approximately 0.7 ms. This is a significant fraction of the total RTC latency and is therefore not acceptable.

Illustration 26 shows a similar plot for MPI. In this case we see that for small message sizes the mean latency is less than 50µs with acceptable jitter.

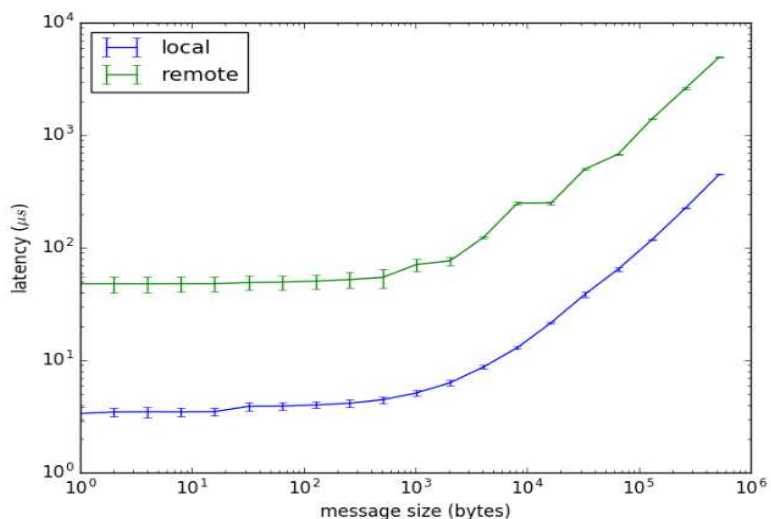



Illustration 26: Mean latency for MPI middleware for local and remote communication. The error bars indicate the standard deviation of the measurements.


Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 44 of 49
Prototypes mid term report		

FPGA development environment

FPGA development environment (aka QuickPlay) was upgraded keeping in mind Green Flash specific objectives. Table hereunder lists different releases since beginning of the project and important associated features.

Release	Major features introductions
2.0 March 2016	Xilinx board support PCIe Gen3 support 10G Ethernet Multi-link support Enhanced HDL kernels creation and configuration
2.1 June 2016	Xilinx Dev-Kit support (KC705 / KCU105) QuickStore addition Multiple implementations support Import of Vivado HLS kernels 64-bit integer type addition UDP Multicast support DDR4 Support
2.2 October 2016	Ethernet RAW mode support SDK support for Windows and Debian based OS 32-bit target support
2017.02 February 2017	Configurable PCIe (speed / lane numbers / number of DMA streams) Enhanced PCIe transfers through new DMA modes P2P support in SDK
Coming Next Q1 2017	Arria10 Devices support (Reflex XpressGXA10 and Bittware A10PL4 boards) SDK Multi-board support

In order to allow Green Flash consortium members to take full advantage of QuickPlay, PLDA organized training sessions (basic and advanced) to both University of Durham and Observatoire de Paris (respectively 3 and 2 persons trained). Also, PLDA support teams ensures a fortnightly meeting with both organisations.

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 45 of 49
Prototypes mid term report		

Selection criteria for the final design review

General description of the down selection process

Green Flash is a 3 year research program on techniques for real-time HPC. The standards and techniques developed within the project can be applied to any instrument requiring real-time HPC. Even though the development of techniques / options will continue throughout the 3 years, during year 3, a final down-selection will be made aimed at a prototype RTC to meet the requirements of the E-ELT MCAO system MAORY

We are investigating many technology options and building prototypes

- These prototypes will be completed by Oct 2017
- There needs to be a down-select process amongst these to provide a final prototype RTC system capable of E-ELT scale MCAO (MAORY)
- The final down-select will be made in late 2017.
- Full integration and testing to Sept 2018 (WP8)

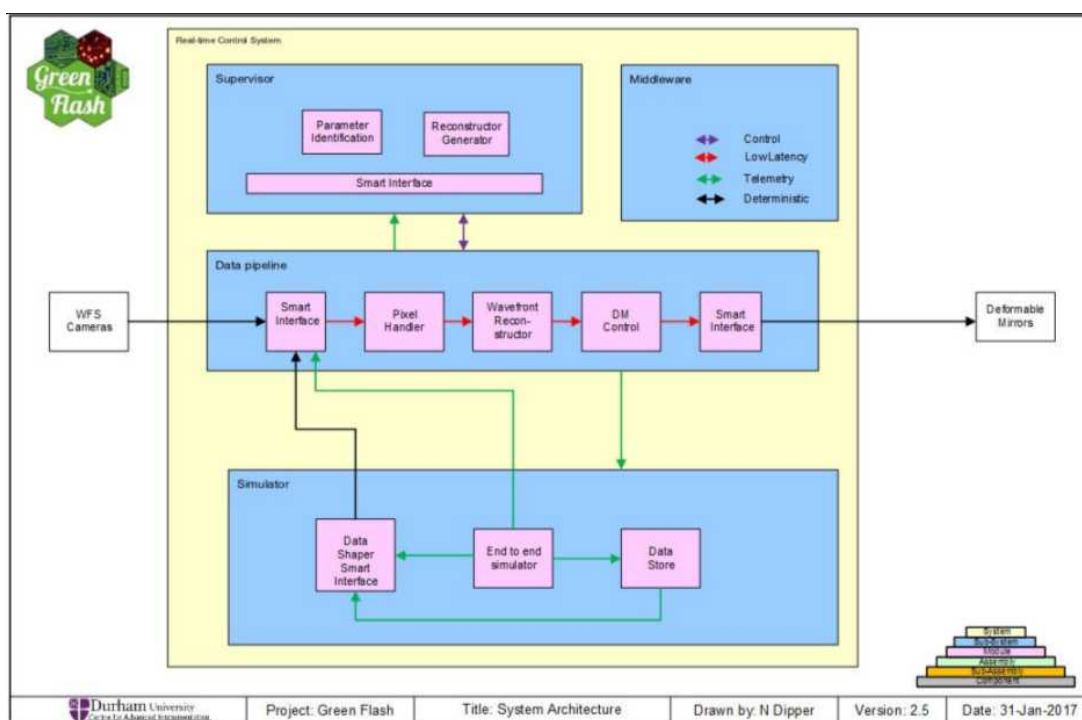


Illustration 27: Updated version of the top-level system architecture

The top-level architecture was defined at PDR

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 46 of 49
Prototypes mid term report		

- This has been slightly modified and simplified to assist in the down-selection
- The down-selection process is under development
- It is being fully defined in a ‘System Down-select Criteria’ document
- Based on selection at multiple system engineering levels
- Top level requirements flow down to each level
- Lower level selection (component, assembly, sub-assembly) will be made during the prototyping phase of each sub-system
- Down-selection for the final prototype will be at the sub-system level

Requirement	Weighting	Option 1	Option 2	Option 3
R3.10 (Frame rate)	<1 to 5>	<1 to 5>	<1 to 5>	<1 to 5>
R3.20 (Latency)
R3.30 (Modularity)
R5.10 (Telemetry)
R7.20 (Interfaces)
R7.5 (Configuration)
R8.32 (Calibration)
R8.34 (Temporal filtering)
Total Score:	-			

Illustration 28: Example of a down-selection matrix for a given module (RT pipeline)

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 47 of 49
Prototypes mid term report		

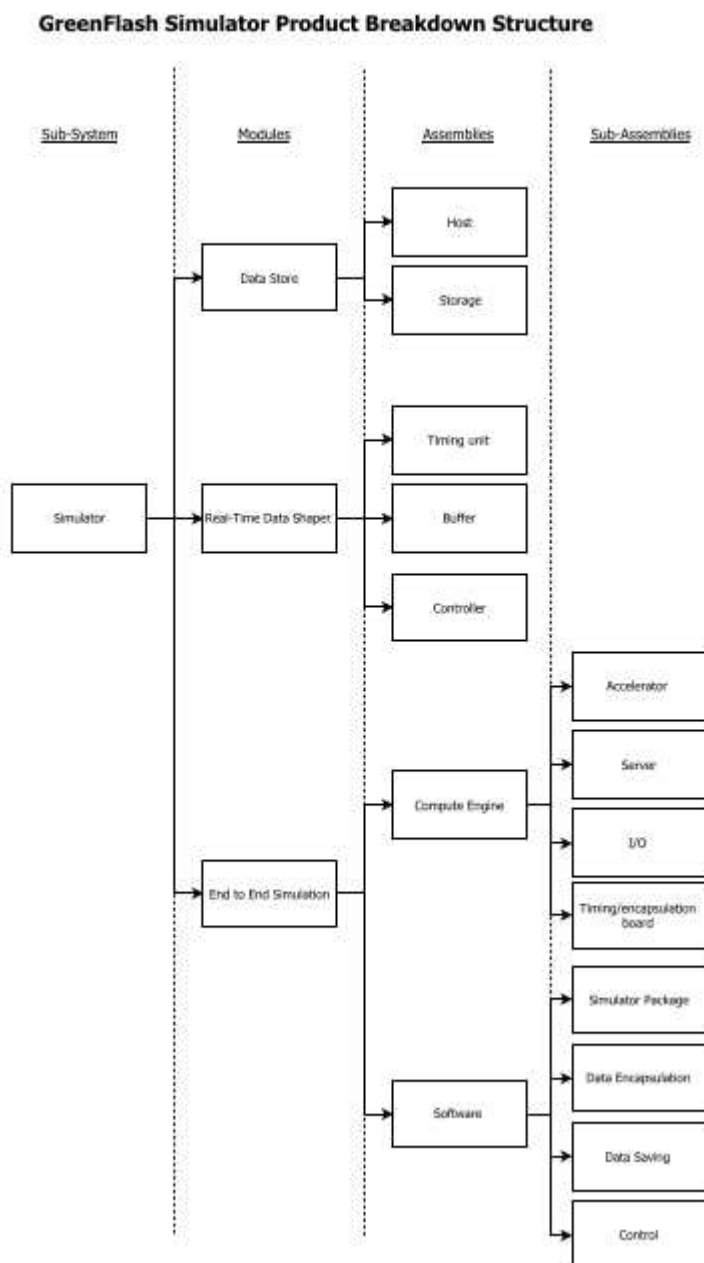



Illustration 29: Example of a product breakdown structure for the RT simulator

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 48 of 49
Prototypes mid term report		


Detailed example with the smart interconnect concept

The smart Interconnect is not a sub-system by itself, but rather an assembly of interfacing subsystems. Depending on sub-system it participates to and architectural breakdown of this sub-system, the Smart Interconnect can inherit a variable set of requirements. For instance if is used for the Data Pipelining subsystem, the Smart Interconnect could support variable number of interfaces presented in above diagram and possibly handle (part of) the “Pixel Handler” function, or even more (depending on its computational capabilities). The down selection process for the smart interconnect concept should thus allow to select Smart Interconnect assemblies tuned to their specific usages in Green Flash. We plan to develop tables of available options, to populate over time, in which sub-systems will be able to pick the most suitable assembly to fulfill their specific requirements.

Board	Board Configuration					Performance		Cost	Power (board)
	FPGA	Daughter card	PCIe sub-system	Ethernet sub-system	Memory sub-system	Bandwidth	Computation capability (Y/N)		
XPressKUS	Xilinx Kintex UltraScale 040	Faster Technologies S-14	PCIe x8 gen 3	1 QSFP	2 x 4 GB DDR3		Y		
XpressGX5	Altera Stratix V EA7	N/A	PCIe x8 gen 3	1 QSFP	2 x 4 GB DDR3		Y		
XpressGXA10	Altera Arria10	N/A	PCIe x8 gen 3/4	2 QSFP	? x ? GB DDR4		Y		
Microgate uXComp							Y		
Microgate µXLink (SOC)							Y		
Bitware A10PL4		N/A					Y		
Bitware XUPP3R	Xilinx Virtex UltraScale Plus	N/A					Y		
Mellanox 40G adapter	N/A	N/A					N		

Development Solution	Elements / components						Cost	Obsolescence	Availability of upgrades
	Interface	Features	Modularity	Reprogrammability	Ease of access (API, drivers, language)	Design scalability			
QuickPlay	10G Ethernet subsystem (Ethernet, TCP, UDP, PCIe)		BSP on-demand IP & Kernels drop-down	Highest	Custom C++ API	Multi-Board support Single-FPGA Provider agnostic Open to new technologies		QuickPlay “covers” hardware obsolescence issues (vendor & technology agnostic)	
Vendor BSP + propr. IP									
Vendor BSP + custom IP									
Custom BSP + IP									
Standard COTS NIC				N/A					

Illustration 30: Example of down-selection matrices at the sub-assembly level for the interconnect

Observatoire de Paris Durham University Microgate PLDA		Title: Prototypes mid term report Version: 1.0 Status: Final Authors: Green Flash team Page: 49 of 49
Prototypes mid term report		

Reports from mid-term review panel members

Report on Green Flash Mid-term Review documentation

Marcos Suarez Valles, ESO, January 31st 2017.

RTC dimensioning for the E-ELT case

When comparing the RTC dimensioning used for the Green Flash prototyping with the applicable instrument requirements and the foreseen WFS/actuator capabilities “as of today”, significant over-dimensioning can be seen in certain cases. This can have an impact in your prototype design and eventually drive it towards too big realizations. In addition, it makes it sometimes difficult to grasp from the documentation the immediate applicability/viability of your prototype for an E-ELT RTC as per today’s baseline and what the resulting size would be. Since size (and mainly the complexity that goes with it) are serious concerns, it would be nice to understand them for an RTC that may not meet the goal specifications but would be compliant with the baseline ones. This may be of particular significance if you tackle in a later phase an MCAO demonstrator relevant for the E-ELT.

Potential documents exhibiting this over-dimensioning are 4.1 and 4.4, amongst others. Taking for instance document 4.1, some specific MCAO dimensions (i.e. the design driver) to be noted may be:

- CMOS Shack-Hartmann WFS:

As of today, a size of $\sim 800 \times 800$, with a maximum nominal readout frequency of 500Hz (goal 700Hz) is purposed by the ongoing development. Document 4.1, section 2.3 seems to consider a size of $1600 \times 1600 @ 500\text{Hz}$, resulting in a throughput of 6x20.48Gbps and therefore requiring several 10Gbps links per WFS. Reducing the size here to 800×800 yields a maximum of 6x7.1Gbps even if the goal frequency of 700Hz is used. This may have a serious impact in A) the image transfer time to the GPU and B) some of your node topologies in section 4.2, where you are currently reserving 4x10GbE links per WFS.

- Control matrix size:

Document 4.1, section 2.3, seems to consider $\sim 15\text{k}$ DoF to be controlled, resulting in a CM of size 39.1Gb and an update bandwidth requirement of 653Mbps. This does not seem to be in agreement with the current instrument requirements. A decrease in the number of DoF to be controlled may significantly reduce the CM size and the update bandwidth. This would have a significant impact in the complexity of the computation, which you are currently estimating in 2.4TFLOP/s. Such reduction may involve that the number of required K80 GPUs decreases significantly. In connection to the above point (i.e. less 10GbE WFS connections required per node), this may make the full RTC eventually fit in a single node in the future.

Figures 1 and 2 in document 4.1 seem to assume that A) WFS frame k is readout during the integration cycle $k+1$ and B) the readout time will be smaller than the integration time. This may not be the case for the CMOS WFS, where the readout time may elapse the full integration time¹. In both the *sequential* and *fragmented* processing, the GPU is assumed to complete POL computation, then catch up with the blocked readout process/jump on the full frame received and possibly produce a command within the current integration cycle. For the CMOS WFS in *fragmented* processing, the GPU will not be able to produce a result until a finite time after the full frame has been received - i.e. already inside the next loop cycle. Depending on the length of this pure delay, the GPU processing for frame k and frame $k+1$ may be overlapping - i.e. resource over-dimensioned to start processing a new frame while the previous one is being completed. For the CMOS WFS in *sequential* processing, when frame k starts to be transferred to the GPU, frame $k+1$ is already being readout: the full GPU computing time is added directly to a full loop cycle time, in terms of computing delay -i.e. from the first

¹ In fact, each the integration interval for detector region is different (i.e. it is the time elapsed between two detector region readouts), so it may be more convenient to talk about readout time and loop cycle.

pixel being available. Even though you report in section 4.3.1.3 that the *sequential* approach has double the performance as the *fragmented* one, you may want to carefully consider the WFS readout scenario described herein, in order to determine which strategy is the best in terms of absolute latency.

Question: in document 4.1, section 3.1.1, page 10, some *Operation Intensity factors* are provided. Do you confirm that, out of the 4 memory operations required for each MVM result - i.e. 3 reads and 1 write, you are assuming 3 of them to be cached - i.e. the vector read and the result read and write?

Section 2.1.1 in document 4.1 describes the POLC algorithm on which the prototype is baselined. The performance requirement is derived taking into account two matrix vector multiplications of the same size. **Question:** have you considered replacing the MVM associated to matrix *D* by some more efficient technique that may take advantage of sparsity properties -e.g. interpolation via stencils, sparse matrix multiplication etc.?

Section 4.3.2 in document 4.1 addresses the expected precision for 16-bit floating-point processing around the value 0. In connection to this, section 5 in document 4.4 states that 16-bit floating-point has shown to provide *ample* precision for OA. **Question:** could you be more specific as for what is being meant by *ample*? Has this been proven for an integral controller over sufficient loop cycles and/or demonstrated in an actual system? If so, where dedicated techniques (e.g. Kahan summation) required for implementing the integral effect? Is there any publication available concerning this result? Note that, although the reported precision around the value 0 (i.e. 10^{-8}) seems compatible with a dynamic range well in excess of 16-bit, integral blocks do not necessarily operate around the value 0.

Strategy for RTC control matrix update

Document 4.3 presents an algorithm for CM matrix update based on a fitting stage of the covariance matrix followed by an inversion via Cholesky decomposition. Up to now, the covariance matrix inversion has been considered in the literature the dominant computing step for periodic control matrix re-computation - as it is still the case, for instance, in document 4.4, section 4.3. The proposal in document 4.3 shifts the focus to the fitting step, which clearly drives the whole algorithm complexity. This is so to the extent that the expectation is to invest 680 s in the fitting step and only 30 s in the matrix inversion, using a cluster of 32 GPUs. This even surpasses by far the complexity expected for the RTC hard real-time pipeline.

The complexity of the fitting step probably stems from the fact that up to 43 parameters are fitted and up to 40 layers are used, over several iterations. Clearly, the control matrix needs to include noise and turbulence statistics, but also geometric and optical information. The former is introduced via the covariance matrix, whereas the latter may be accounted for by parameter fitting, (synthetic) interaction matrix, interpolation stencil operators, etc. For instance, one could imagine that few geometric parameters (e.g. mis-registration, rotation, magnification, etc.) are fitted via synthetic interaction matrix computation at the time scale they are expected to vary at the telescope, whereas noise and turbulence statistics are introduced with a different periodicity via regular MMSE that makes use of the synthetic interaction matrix plus the measured covariance. **Question:** has the baseline in document 4.3 been adopted following instrument requirements and/or the expectation that a much higher performance will be achieved and/or as a precondition for loop stability? Will the Green Flash prototype also address a more classical approach for control matrix re-computation in order to trade it off with the baseline fitting algorithm proposed in document 4.3?

Some basic dimensioning aspects in document 4.3 may not be completely clear:

- Page 6, mentions a covariance matrix size of 90kx90k (and later extends it to 100kx100k) in connection with the E-ELT First Light instruments. This does not seem to be in agreement with the dimensioning in document 4.1, section 2.3, which assumed 76.8k slope measurements.
- Page 6, also indicates that a 64GB matrix must be uploaded periodically to the so called RTC box. This seems to be the size resulting from a covariance matrix of size 90kx90k expressed in double precision. However, I would expect that the matrix to be uploaded to the RT box is the control matrix, expressed in single precision and with a much smaller size.

- Page 7 seems to focus system dimensioning on the inversion via Cholesky decomposition (which later on is shown not to be the dominant step) and considers a cluster of 100 GPUs for the inversion of a 100kx100k covariance matrix, assuming that the required performance is around 150-200TFLOP/s. I would expect 1200 TFLOP (i.e. $N^3 + 2 \cdot N \cdot m$, with $N=10^5$ and $m=10^4$, as per your document) to be the number of operations required by a single inversion of the covariance matrix. With the suggested update rate of ~1 min, this results in 20TFLOP/s of computing power, which should not require 100 GPU devices.
- Section 2, page 12, states that the inversion via Cholesky decomposition itself may be implemented in 30s by a cluster of 32 GPU devices for ($N=10^5$, $m=10^4$). This goes beyond the initial specification of ~1 min updates. I understand that the computation is performed with 32 devices because they are already available -i.e. imposed by the fitting step. **Question:** *if we would consider only the inversion via Cholesky decomposition, may 16 devices be used for an increased computing time of 60s or is the on-board GPU memory the limiting factor here?*

Document 4.4, section 4.3, indicates that a single Xeon Phi device is expected to perform the covariance matrix inversion step for ($N=6 \times 10^4$, $m=10^4$) in 10s. This statement should be put in contrast with the need for 32 GPU devices to complete the same computation in 30s for ($N=10^5$, $m=10^4$) stated in document 4.3. In terms of covariance matrix size, there is a factor ~4.7 between the two test cases but, considering raw FLOPS/s, the factor is only ~1.5. **Question:** *what is the cause of this difference in the required number of computing devices between the two documents? Are there concerns about the parallelization of the algorithm and/or memory sharing across GPUs that would significantly increase the number of GPU devices and/or make the Xeon Phi (bigger on-board memory) approach much more advantageous?*

FPGA development with QuickPlay

Documents 5.2 and 7.2 describe several iterations in the process of generating a Smart Interconnect prototype at PLDA that (at least partly) should reproduce prototype for PCIe P2P communication already successfully realized at OBSPM. If my understanding is correct, the OBSPM prototype was based on IP cores, at least some of them from PLDA (e.g. QuickPCIe), and possibly using VHDL. The intent here seems to be synthesizing the same OBSPM design from QuickPlay using C/C++ functional kernel descriptions compiled by the tool into VHDL. This objective may not have been achieved yet.

A number of limitations have been identified in the process, mainly related to:

- Lack of QuickPlay support for certain features of the UDP and IP protocol suite (e.g. UDP broadcast/multicast, DHCP); lack of support for certain memory types (e.g. DDR4, HMC).
- Lack of timing performance in the synthesized firmware -e.g. UDP/TCP throughput limit of 1Gbps, PCIe throughput limit of 5.5Gbps, etc.
- Problem in accessing GPU registers from the FPGA.
- Lack of QuickPlay support for certain emulation abstractions at kernel level (e.g. HW registers, parallel code execution).

Some of the above limitation are allocated to the QuickPlay API, whereas some others stem from kernel design. The latter seem to have required through kernel optimisation - e.g. data path sizes, re-writing based on HSDL recipes in order to meet timing constraints and reduce the use of logic blocks, etc. The former group of limitations require modifying QuickPlay itself.

Some of the reported constraints resemble early experiments performed at ESO based on C-to-VHDL compilers. At the time, the conclusion was that performance-critical aspects needed to be allocated to IP cores, whereas the compiler was mainly helpful in *gluing* these IP cores together and implementing less critical interfaces requiring highly complex logic. A key feature for the compiler was then the ability to integrate IP cores from many sources. **Question:** *in respect of the above, how does the FTE invested in your original prototype at OBSPM compare with the one invested so far in the porting to QuickPlay? What is your estimate of the FTE still required to completion? How do the two developments compare in terms of the skills required, the length of the iterations, etc.?*

The kind of interventions/optimizations described in document 5.2 for a prototype of still limited extent are likely to be expected as part of mostly any new development (unless extremely tight HW standards are imposed and IP core diversity strongly constrained). Two of the fundamental criticisms to FPGA-based implementations have been the long development cycles and the need for specific profiles. *Question: what are the typical synthesis times for the development described in document 5.2? What would be the expectation on the time required for a BSP to be available in order to use a certain new FPGA board under QuickPlay? How much of the kernel tuning requires extensive knowledge of the underlying board/FPGA architecture?*

Document 7.2. states that full compliance to networking standards will not be provided in the course of the Green Flash project, in particular in connection to openDDS and openMPI. The feasibility of implementing and maintaining complex protocols like DDS running in the FPGA domain has always been an open question. *Question: is this decision motivated by cost/FTE trade-offs, expected difficulty/complexity in supporting those protocols in FPGA/QuickPlay, other reasons?*

RTC telemetry data

The propagation of RTC telemetry seems not to be directly addressed by any of the Green Flash deliverables. Even if designing an RTC telemetry network is not a target of the project, the presence of telemetry data circulation is potentially a source of non-deterministic behaviour that may need to be taken into account while prototyping the RTC hard real-time pipeline:

- For GPU-based realizations, like the one described in document 4.1, with pixel data directly fed to the GPU over the PCIe bus by an FPGA board, the circulation of additional pixel telemetry data (even if sub-sampled) may be quite disruptive (the pixel frame transfer time is already regarded in section 2.4 as the real dimensioning factor), since it will be difficult to prioritize the usage of the PCIe bus. This may become more demanding if telemetry pixel data from different tap points (i.e. processing stages in the pipeline are required simultaneously) and may require some kind of telemetry data throttling over the PCIe bus and/or buffering inside the GPUs.

In addition to the pixel telemetry data, slopes, commands, possibly data at intermediate pipeline stages and internal status will likely be required for telemetry at loop rate. This might affect the way some algorithms are written (e.g. to make some intermediate data explicit) and the number/entity of the reductions across GPUs. In addition, the aggregated PCIe traffic may not be negligible -i.e. 2.1Gbps considering only slopes, intensities and commands at 500Hz, with the dimensioning assumed in document 4.1.

Please note that disturbance injection (e.g. at slope and command level) is an essential part of the telemetry data i.e. it must be possible to calibrate the AO system. Following the algorithm in section 4.1.1, each GPU requires a part of the slope disturbance vector, whereas some privileged device may need to add the command disturbance before being sent to the actuators. Note that slope disturbance at 500Hz may result in additional ~1.2Gbps, with the dimensioning assumed in document 4.1.

- For CPU-based realizations, like the one described in document 4.4 based on bootable MIC devices, some of the above concerns may also apply. In addition, telemetry data propagation is known to introduce jitter in the hard real-time computation, coupled via the CPU cache and the OS kernel network layer. Mitigations may include some form of OS real-time scheduling and parameter tuning, isolation of CPU cores and NUMA handling, allocation of NICs to cores, amongst others. The support of this features for MIC devices using mainstream OS kernels remains to be explored -at least partially.
- For FPGA-based realizations, like the one described in document 3.1, the handling of telemetry data may require supporting some form of reliability for the corresponding network interface. This may result in significant processing time invested in packet acknowledgement, buffering, retransmission, etc. depending on the protocol chosen. It remains to be proven that the required overhead can be taken over by a single adjoint ARM processor that, at the same time, performs command and configuration functions.

The telemetry data may also result in a requirement for additional network interfaces in the FPGA board -e.g. it does not look immediate that the same 10GbE network used for deterministic sensor and actuator traffic can take in addition significant telemetry traffic subject to retransmissions.

Comments to the Green Flash Mid Term Review

The statements contained herein express a personal, technical view based on previous, long-term experience with the development, commissioning and operation of the VLT Adaptive Optics (AO) real-time Computer (RTC) platform. They shall not be understood as the official position of ESO or the E-ELT Project Office in respect of the Green Flash project undertakings. The technical comments and suggestions in this document are not necessarily aligned with the future E-ELT RTC standards, which are still under development at this stage of the project.

By its Mid-Term Review (MTR), the Green Flash project has shown clear progress in a number of areas where synergies with the AORTC domain apply. Short comments in this respect are provided below, focusing only on few aspects deemed critical for the future applicability/success of the solutions proposed by the project. Feedback is provided from both technical (i.e. pointing to potential design issues, overlooks, challenges, etc.) and operational perspectives (i.e. focusing on maintainability, upgradeability and obsolescence). Where applicable, comments are put in the context of the feedback already provided by the Preliminary Design Review (PDR) and follow-up by Green Flash is analyzed.

Comments to Overall Project Strategy

The portfolio of Green Flash solutions for AO RTC has been extended to include standard CPU technology and development tools. Specifically, the Xeon Phi family of bootable devices recently introduced by Intel has been selected. The MTR documentation now addresses initial benchmarking of these devices with respect to GPU realizations for both the hard real-time and supervisory domains. This fulfills the PDR recommendation to explore the applicability of today's CPU architectures to the AO RTC problem. The project is encouraged to further progress in this line and:

- Monitor the evolution of multi-core and/or Many Integrated Core (MIC) hardware and explore possible alignment of AO RTC with standard High Performance Computing (HPC) technology.
- Evaluate the amount of OS- and HW-dependent tuning required for a compliant AO RTC implementation and the usability of standard parallel programming tools and libraries.

Green Flash has advanced significantly towards a deterministic, GPU-based, AO RTC hard real-time implementation using direct I/O. The recent prototypes seem to point out that no access to undisclosed, proprietary GPU Application Programming Interface (API) is required. This mitigates the PDR concern in respect of the need for Non-Disclosure Agreements (NDA). Still, in order to achieve real-time determinism, the solution proposed relies completely on the availability of a custom *smart interconnect* FPGA board to be developed by the project partners. This poses a clear risk in terms of single-source procurement and obsolescence. It is recommended that the project attempt to mitigate this by:

- Clearly identifying those smart interconnect component blocks that will remain the intellectual property of the Green Flash commercial partners, extensively describing their interfaces and adopting open standards for them.
- Demonstrating that smart interconnect functionality may be built based on Component Off-The-Shelf (COTS) hardware and standard FPGA firmware development tools, even if overall performance degrades.

As per the MTR documentation, Green Flash performance specifications seem to sometimes exhibit significant over-dimensioning when compared to the current First Light E-ELT instrument baseline configurations. In particular, this is the case for the Multi-Conjugate AO (MCAO) sensor/actuator sizes and (as a result) wavefront reconstruction complexity. This may eventually drive prototyping activities towards too big realizations in terms of computing power and network throughput, thus jeopardizing simpler, compliant AO RTC designs. It is recommended that the project addresses this by:

- Closely following up the evolution of the E-ELT First Light instruments requirements in terms of sensor/actuator sizes and control laws.
- Clearly separating baseline and goal specifications for design and prototyping purposes when the increment in complexity is seen to have a strong impact in system dimensioning and limiting it will provide additional room for technical compromises.

Technical Comments and Specific Obsolescence Concerns

The propagation of AO RTC telemetry data seems not to be currently addressed by Green Flash in the various hard real-time solutions proposed. Even through the design of a telemetry network is arguably not a deliverable of the project, the presence of telemetry data circulation is a potential source of non-deterministic behaviour that needs to be considered for the validation of the various prototypes. The impact can be severe for most of the technologies in use by the project:

- For GPU-based realizations, with sensor images directly fed to the GPUs by an FPGA board over a shared bus, the circulation of additional pixel telemetry data through the same channel may be disruptive, since the pixel frame transfer time has been identified by the project as a key dimensioning factor. This may become even more constraining if the extraction of pixel frames simultaneously at different points in the computing pipeline is specified. Moreover, the need for prioritizing real-time sensor/actuator traffic on the shared bus may require the implementation of telemetry data throttling and/or buffering inside the GPUs.

In addition to pixel frames, AO telemetry will require slopes, commands, data at intermediate pipeline stages and control logic state information to be propagated at loop rate. On top of this, disturbance injection (e.g. at slope and command level) is essential for AO system calibration and will add to the overall telemetry throughput, which may amount to several Gbps. Telemetry might also affect the way some algorithms are written (e.g. to make some intermediate data explicit) and the number/entity of the reductions across GPUs.

- For CPU-based realizations, some of the above concerns may also apply. In addition, telemetry data propagation is known to introduce jitter in the hard real-time computation, coupled via the CPU caches and the OS kernel network layer. Mitigations may include some form of OS real-time scheduling and parameter tuning, isolation of CPU cores, NUMA handling, allocation of network interfaces to cores, etc. The support of these features for MIC devices remains to be explored - at least partially.
- Telemetry data typically requires reliable network communication. For FPGA-based realizations, this may incur significant processing time for packet acknowledgement, buffering, retransmission, etc. depending on the protocol stack of choice. It remains to be proven that the required overhead can be offloaded to a (likely) single, processor core embedded in the FPGA that, at the same time, performs command and configuration functions.

Telemetry data may also result in a requirement for additional network interfaces in the FPGA board - e.g. it is not immediate that the same network used for deterministic sensor and actuator traffic can, in addition, support significant telemetry traffic subject to retransmissions.

Further refining the recommendation provided by PDR in this respect, it is suggested that the project introduces realistic telemetry data handling in the AO RTC hard real-time prototypes at two levels:

- Actual circulation of telemetry data through critical, hard real-time, shared data paths (e.g. internal buses) and interleaving with sensor/actuator data where applicable.
- Actual propagation of telemetry data via network interfaces (to exercise the protocol stack), even if a comprehensive, dedicated network infrastructure is not built, but data are simply acknowledged by some tests publisher/subscriber components.

By MTR, FPGA technology plays a central role in the portfolio of solutions proposed by Green Flash for the hard real-time AO RTC domain: FPGA boards developed by the project partners are the main building block for the full-FPGA solution, a key core component for the GPU-based solution and the basis for the smart interconnect boards. At the same time, FPGA development inside the project goes in hand with the use of the QuickPlay product by PLDA. Whereas FPGA technology can undoubtedly enhance real-time determinism, it also introduces significant impact in the development process:

- Looking at prototype evolution since PDR and considering the recent MTR discussions, it has not yet been proven that a hard real-time implementation largely based on FPGA will be compatible with the development time constraints involved in the Assembly Integration and Test (AIT) and Commissioning of AO instruments.

Modifications during the AIT process are historically a significant part of the overall AO RTC development effort, which frequently extends into the early commissioning runs. This may become critical for the E-ELT First Light instruments, which rely on actuators embedded in the

telescope, thus requiring a big part of the AIT activities to be done on site. Note that the telescope time during which the dome conditions are suitable for instrument testing is a limited resource and needs to be strongly optimized and used efficiently.

Despite the usage of QuickPlay simulation and high-level programming features, the duration of the Edit-Compile-Test cycle for FPGA technology remains, as of today, in the order of hours until a modification can be deployed and exercised on the final system. This is to be compared with minutes for a CPU-based solution. Similarly, the hourly cost of the Edit-Compile-Test cycle is still much higher than for other technologies, since specialized developer profiles are required. This might pose a real bottleneck for AIT activities.

- The project is putting great effort in making FPGA development more affordable in terms of programmer skills and efficiency. By MTR, a part of this effort is invested in adapting QuickPlay itself to the project requirements and nature of the AO RTC development, with even formal deliverables defined on PLDA side. This introduces the risk of making the development process strongly dependent on a single tool and ecosystem.

Relying on a unique, optimized development environment for a domain where no major standards are currently in force regarding abstract programming, may result in long term obsolescence risks. Should QuickPlay be discontinued but AO RTC maintenance still required, it would be essential to re-synthesize and further modify the existing code with a different tool.

It is recommended that the project mitigates the impact of FPGA technology in the efficiency and cost of the development process by:

- Carefully considering the part of the AO RTC functionality that is mapped onto FPGA devices, taking into account operational aspects/constraints.
- Internally auditing the prototypes development process, quantifying the real time/resource gain derived from the use of QuickPlay and proposing modifications to the classical development cycle to mitigate the increased development time/cost.

Reclaiming some of the recommendations provided at PDR in this respect, it is suggested that the project attempts to minimize the dependency on a single FPGA development tools by:

- Demonstrating the process of continuous integration of HDL code developed using standard Xilinx/Altera tools into the QuickPlay development environment.
- Demonstrating the usability of QuickPlay generated code as part of standard Xilinx/Altera development environments for final project synthesis.

Green Flash Project - A Real-Time Control Computer for the E-ELT

Mid Term Review

Comments and questions by Laura Schreiber
18/03/2017

The main goal of the Green FLASH (GF) project is to design and build a prototype for an AO RTC targeting the E-ELT first-light AO instrumentation addressing an optimized strategy to handle the complex data flows and their interactions in the system and a long term maintainable solution, based on evolving standards.

This goal is realized through the achievement of three main objectives: 1) Real-time HPC using accelerators and smart interconnects; 2) Energy efficient platform based on FPGA for HPC; 3) AO RTC prototyping and performance assessment.

As a general comment, I remark that this report reflects my personal opinion as a scientist involved in RTC dimensioning and not the view of the MAORY consortium. In particular, suggestions on the requirements are based mainly on common sense or have been extracted from the literature. Official rules for the stream of instrument requirements information have not been established yet. The main source of this kind of information should be ESO only.

As already stressed in the first report, the project schedule might not match perfectly the E-ELT schedule and also some basic choice could be not in line with the E-ELT general guidelines, but the project clear goals in terms of performance targets, seem to match well with the E-ELT first light instrumentation interests, even of the most challenging system, MAORY. All the results of the project, even in terms of mid-terms results, are really interesting and it is important for the instrumentation point of view to keep them well monitored.

For this reason, I am glad to be part of the panel and to be informed about the results.

The scope of this mid-term review was to outline the preliminary results from the currently ongoing prototyping phase, get feedback from the community on these results and set a list of down-selection criteria for the design of the final full-scale RTC prototype to be assembled in 2018. The delivered documentation mostly consists of a number of reports.

Almost all the prototyping activities seems to proceed and produce interesting results. It might be very interesting to add to the documentation, maybe in the general prototypes activity report GFD2.1, a graphical representation of the objectives and sub-objectives achievements, to graphically highlight problems and issues and de-scoping.

It might also be useful to recall the specific reference requirement that is reached or not in the specific prototyping activity. Sometimes in fact in the documentation, the results are not clearly summarized in the conclusions. For example, the results reported in GF-D4.3 are very interesting, but not always very easy to find in the text and sometimes they seems to be in contradiction in the text itself. As a last remark, already reported in the previous report as a general comment, I appreciate that the request of a list of selection criteria for the final design review has been considered and a provision of it is now present in the documentation.

I add a small general comment that concern the documents read facility. It could be very useful in fact for a reviewer to have also a small text file with an explanation on how the delivered documentation is organized. The enumeration is not very clear to me, and some of the documents seems to be still in a draft format, but they are named as 'Version 1.0'. What's the number in the document title stands for? Sorry if I missed something.

Feedback to the Green Flash project mid term review
Y. Clénet – February 21th 2017

Being responsible of the development of the MICADO-MAORY SCAO mode, I have been invited by D. Gratadour to be part of the review panel for the Green Flash mid-term review. It took place on February 1st at Meudon Observatory.

Here is a list of comments or questions done during the review, which are driven by my perspective of development of the MICADO-MAORY SCAO mode.

1) In D. Gratadour's presentation for WP1&2, it is mentioned that a Green Flash aims at building a RTC prototype after a technology down-selection process. The latter will be done from different criteria, including "compliance to standards". I was wondering what standards are considered and in particular if they include maintainability. I was also wondering if the ESO schedule for the delivery of the instrument RTC standards was compliant for this down-selection process. From the MICADO-MAORY SCAO perspective and for us to use Green Flash outputs in the development of our RTC, it is important that Green Flash can make use of these ESO standards and comply with them as much as possible.

2) In the same presentation, D. Gratadour mentioned a collaboration with ESO. I asked for details on that collaboration. Indeed, I think it would be a pity that the investment made by *Europe* with the Green Flash project would not find a concrete realization in the *European* instrumental projects for the *European* Extremely Large Telescope. Hence I find important that ESO makes use of the effort provided by the Green Flash project to feed their reflexion in the establishment of the instrument RTC standards

3) In the presentation by Microgate for WP3, it was stated that the hardware and the firmware of their FPGA will be designed to be compatible with PLDA QuickPlay. I asked what was the exact impact of this specification and if this compliance was an additional difficulty.

4) In the presentation by PLDA for WP5, the QuickStore was presented. I asked who is allowed to provide IP cores in this store and more generally how are integrated/distributed developments made for FPGA. I also ask if PLDA would be able to provide enough support if their products are eventually included in the ESO RTC, given that support would be necessary at the same time for 5 AO developments (SCAO MICADO-MAORY, MCAO MAORY, SCAO HARMONI, LTAO HARMONI, SCAO METIS).

5) After the presentation by Durham University about the down-selection process to build the RTC prototype, I asked about the requirements that are driving this down-selection process, if there were only internal to the Green Flash project or if there were requirements coming from ESO. Since ESO is also doing prototyping on their side, I was wondering if the criteria were the same. Discussions between ESO and Green Flash to share these criteria seems important to maximise the output of the project on one hand and to have ESO benefiting from the Green Flash resources on the other hand.

I must say that the work performed by the Green Flash team was impressive and give confidence in the perspective to achieve the goals of the project in schedule. My main concern, from the MICADO-MAORY SCAO perspective but more generally from the European AO community, is to have deep technical discussions between the two and proper information sharing.

Feedback to the Green Flash project mid term review
M. Feldt, 15th March 2017

I participated in the Green-Flash (GF) mid-term review as Co-I and SCAO-responsible of the METIS consortium. From this perspective, I have the following comments and questions to the Green-Flash consortium:

1) ESO is currently leading a process to specify and design the RTCs for the E-ELT instruments. The goal is to arrive at a common specification for all instruments, and have each consortium provide their own RTC fulfilling such specs. It is not fully clear, how the outcome of the specification process, expected before the instruments' PDR dates in 2018, will influence the Green-Flash selection process, and the prototypes developed.

2) It is not fully clear, how the consortia and/or ESO can make use of the prototypes developed within GF. To my understanding, at least part of the development will remain proprietary, which makes it hard to use in an ESO environment. To maintain proprietary software in a critical infrastructure will not be easy.

3) METIS will employ mostly a SCAO-only solution, with the potential LTAO-upgrade being uncertain at least. In this context, METIS is setting up a test system on their own, comparing GPU and CPU based solution which we consider sufficient for the relatively relaxed problem of SCAO. From this perspective, the presentation on the Real-Time Simulator was very interesting. The requirements given do not fully match our own (required frame rate 1kHz instead of 800Hz, jitter requirement not yet fixed), but the result of the simulator development would be most interesting to have access to. Even if the output of GF as a whole will not come in time to impact METIS' design decisions for the RTC, it would be very interesting to collaborate on simulation modes during METIS' AIV phases.

4) Thanks to Damien Gratadour and co-workers, COMPASS is now also in use at the METIS consortium. It is a most useful piece of software. However, while it implements many special E-ELT features it is a bit focused on the numerical real-time side. Imperfections and the error budget of instrument optics such as non-common path aberrations, pixel-misalignment on PWFS sensors, modulation errors for the same etc. are not foreseen to my knowledge. Here we could potentially collaborate to enhance the software and increase its usefulness to the wider community.

5) In the process of defining RTC solutions with ESO, topics of discussion include:

- M4/M5 (DM/TTM) command separation

- Sequential handover versus handover with cascade control between the E-ELT's field-stabilization loop and the RTC's TT commands
- Saturation management

While these will probably not have a major impact on the timing issues examined in GF, I'd be interested in whether such issues are foreseen in the timing budgets within GF.

Overall I'd like to express my respect for the GF team and the great work they have done to date and the achievements that were made. Schedule difficulties due to late or canceled external deliverables have been overcome and the project appears fully on track. I - on behalf of the METIS SCAO team - am looking forward to the upcoming reviews and the final output of GF, as well as on further, closer collaboration on parts of it.