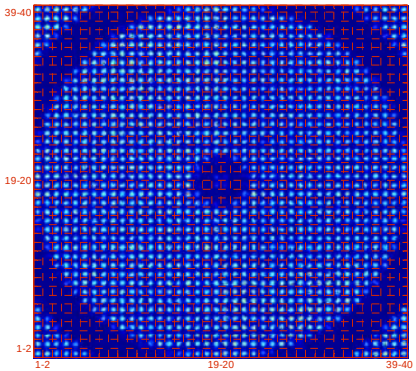
The SPARTA Platform: Design, Status and Perspectives

Marcos Suárez Valles
Adaptive Optics Systems (ESO)
msuarez@eso.org

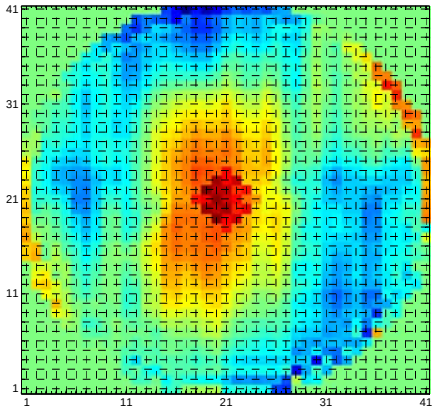
SPARTA Platform Targets

- ESO Standard Platform for Adaptive Optics Real-Time Applications
- SPARTA instrument portfolio:
  - High-order, low-noise SCAO/XAO/GLAO
  - Low-order, low-noise SCAO

	Loop Rate	# WFS	Detector	WFS Size	Subap. Size	# Subap.	Pixel Rate	# DM	# Actuators	Complexity
SPHERE	1.2 kHz	1	CCD220	SH 40x40	6x6 pixel	1,240	66 MPix/s	1	1,377	5.2 GMAC
AOF	1 kHz	4	CCD220	SH 40x40	6x6 pixel	1,240	230 MPix/s	1	1,170	11.8 GMAC
ERIS	1.2 kHz	2	CCD220	PY 40x40	6x6 pixel	1,256	66 MPix/s	1	1,170	
NAOMI	1 kHz	1	CCD60	5x5	6x6 pixel	36	6.8 Mpix/s	1	49	
GRAVITY	500Hz	1	SAPHIRA	9x9	4x4 pixel	62	3.3 Mpix/s	1	60	

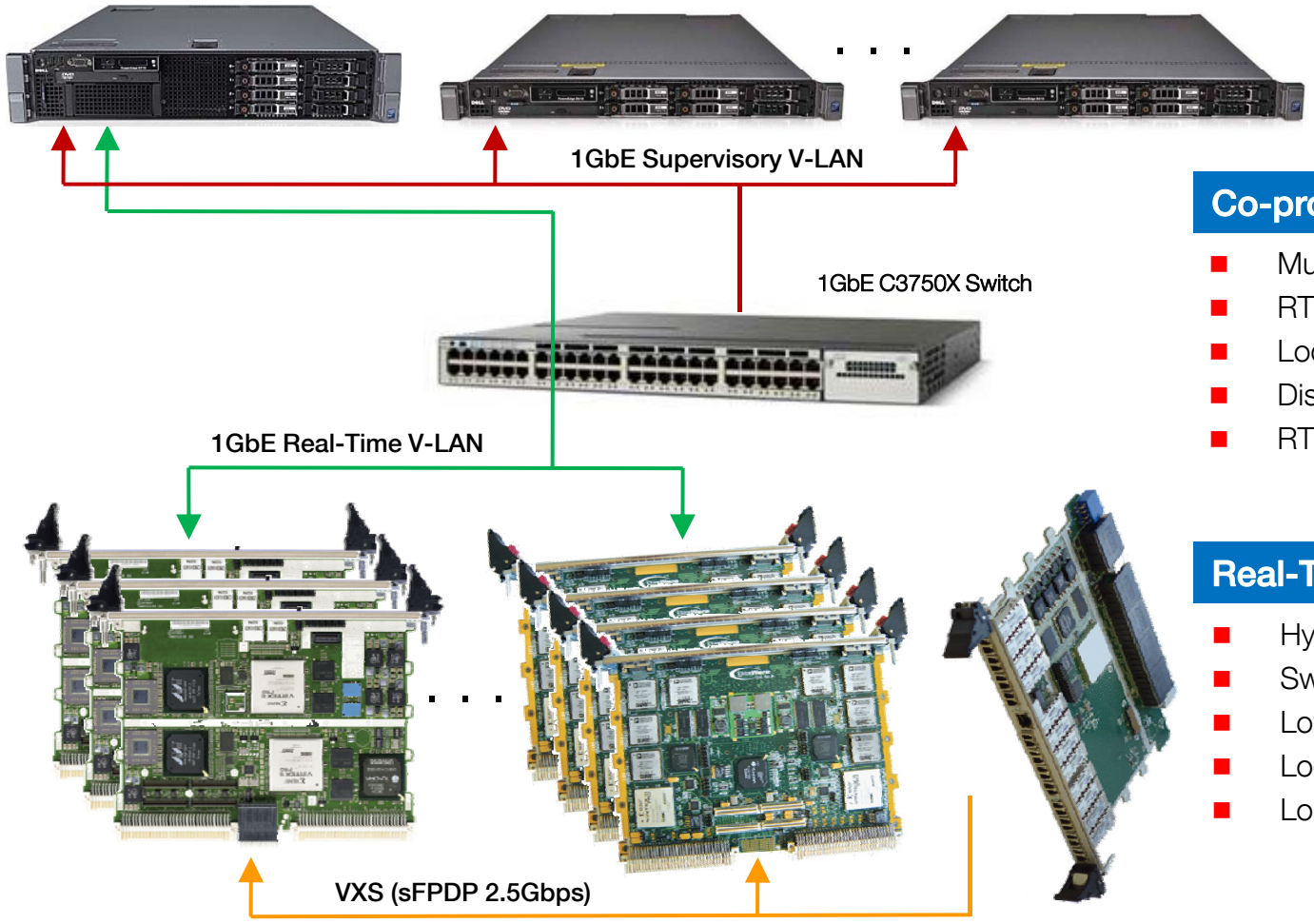


CCD220 240x240 pixel frame:
- 1,240 used subapertures



SPHERE 41x41 HO DM grid:
- 1,377 used actuators

SPARTA Real-Time Computer



Co-processing Cluster

- Multi-CPU, multi-core Linux nodes
- RTC component logic co-ordination
- Loop parameters configuration
- Distributed telemetry data processing
- RTC calibration logic/algorithms

Real-Time Box

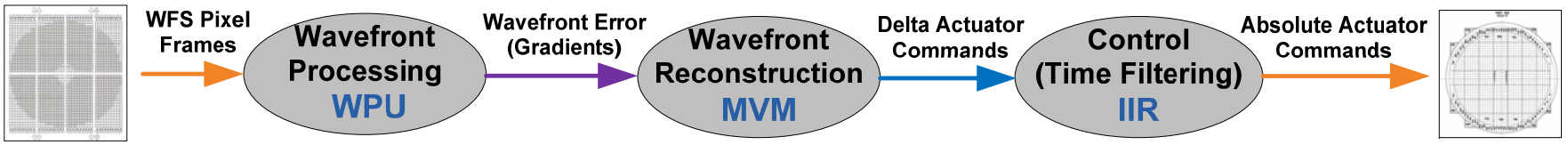
- Hybrid CPU/DSP+FPGA nodes
- Switched VXS communication fabrics
- Low-latency WFS and actuator I/F
- Loop closure/disturbance injection
- Loop-rate telemetry data publication

Curtiss-Wright VPF1:
- 2 x 744x PowerPC CPUs (1GHz)
- 2 x Virtex Pro-II FPGAs (125MHz)

Bittware T2V6:
- 2 x TS201 DSP clusters (600MHz)
- 2 x Virtex-II Pro FPGAs (125MHz)

Curtiss-Wright CSW1:
- Zero-latency circuit switching
- 14 x 3.215Gbps VXS Links (4x)

Real-Time Box Pipeline



$$s_{x/y} = \frac{\sum w_i \cdot p_i \cdot x_i / y_i}{\sum w_i \cdot p_i} \qquad \vec{u}_n = M \cdot \vec{s}_n \qquad \vec{y}_n = b_0 \cdot \vec{u}_n + \vec{K}_{n|n-1}$$

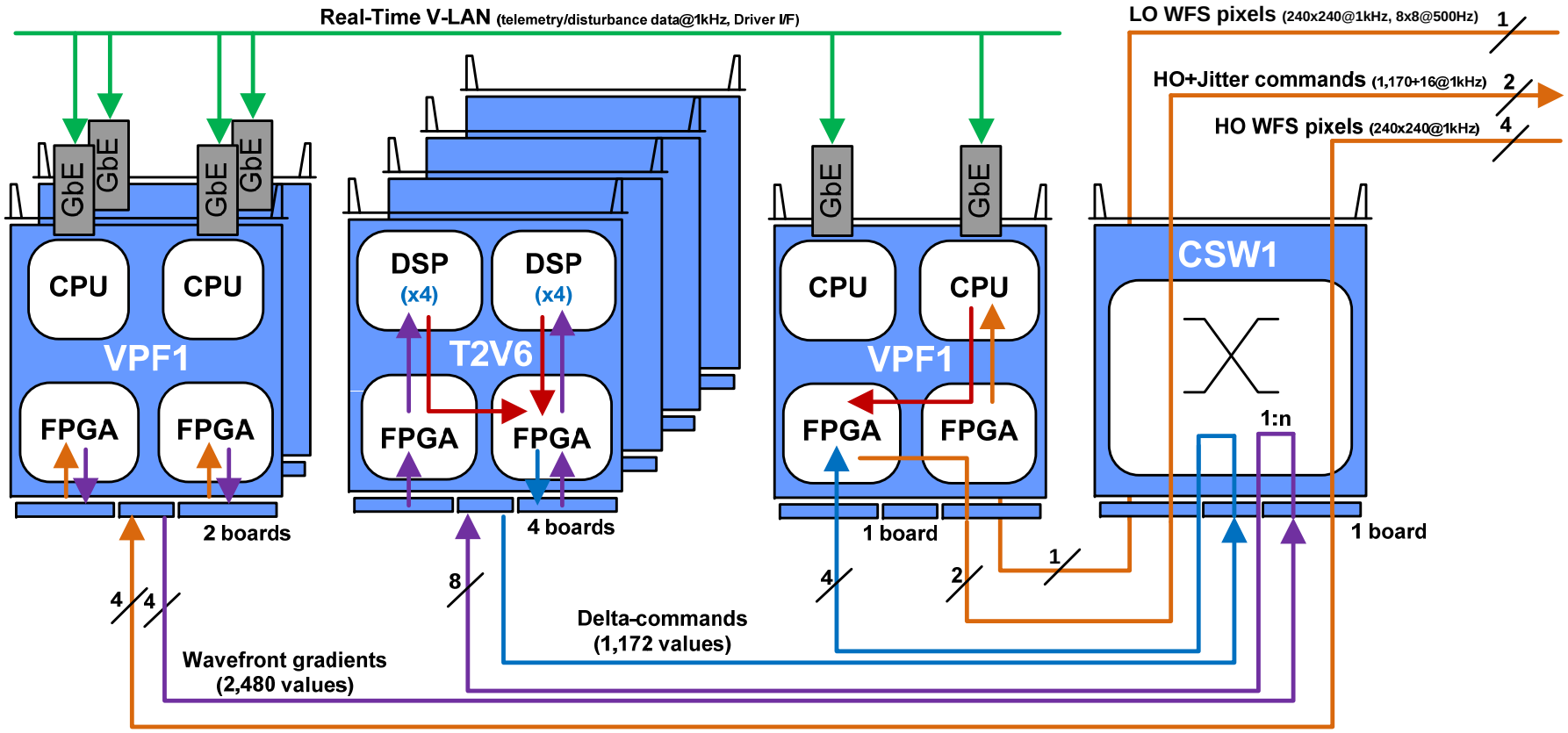
- Wavefront processing:
  - Weighted Centre of Gravity (WCoG)
- Wavefront reconstruction:
  - Matrix-Vector-Multiplication (MVM)
- Command time filtering (control):
  - Infinite-Impulse-response (IIR) filter
  - DM saturation management (anti-windup)
  - DM garbage collection
  - DM disturbance injection

online computation

off-line computation

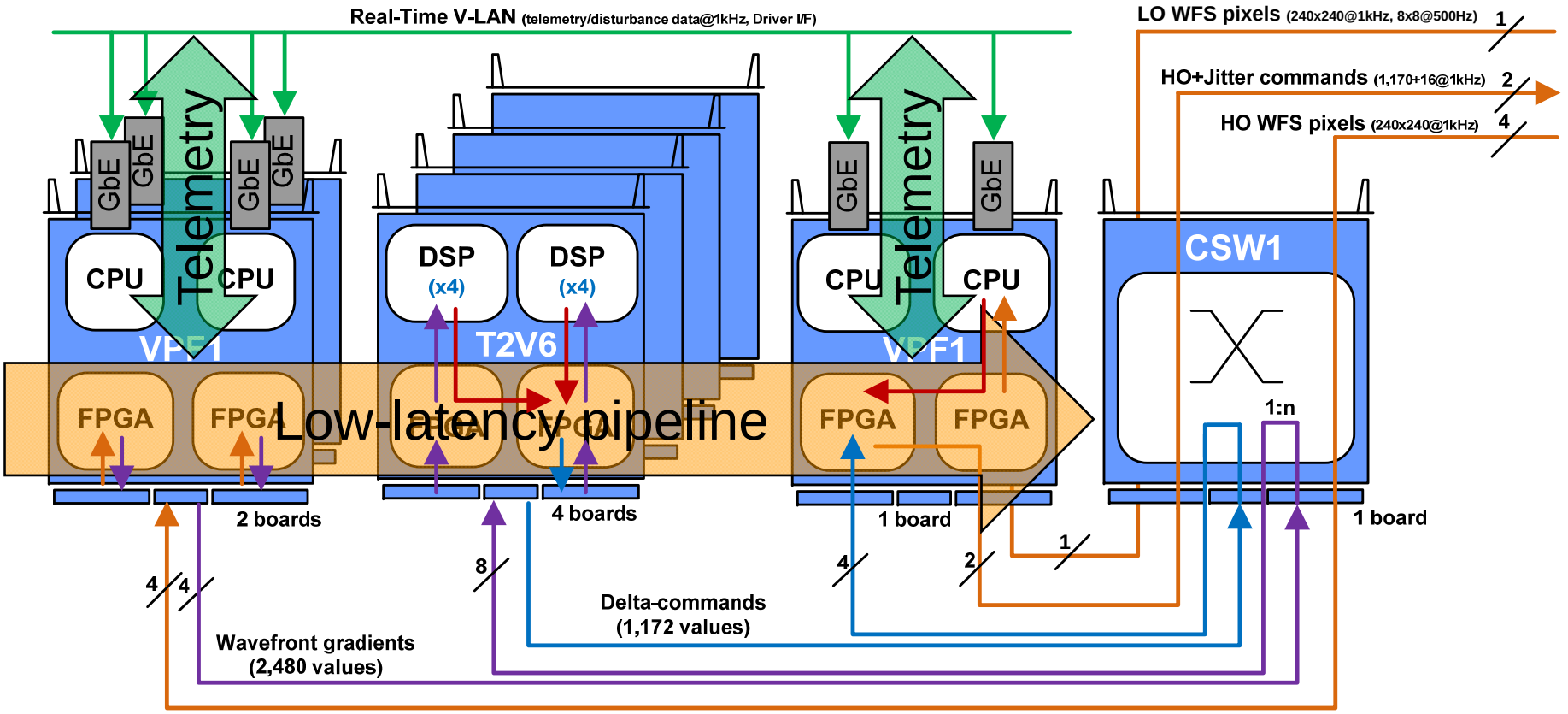
Real-Time Box Processing Nodes

- WCoG in parallel with WFS readout
- Block-wise reconstruction in parallel with WCoG
- Time filtering in parallel with data transfer
- Loop telemetry/disturbance in idle time

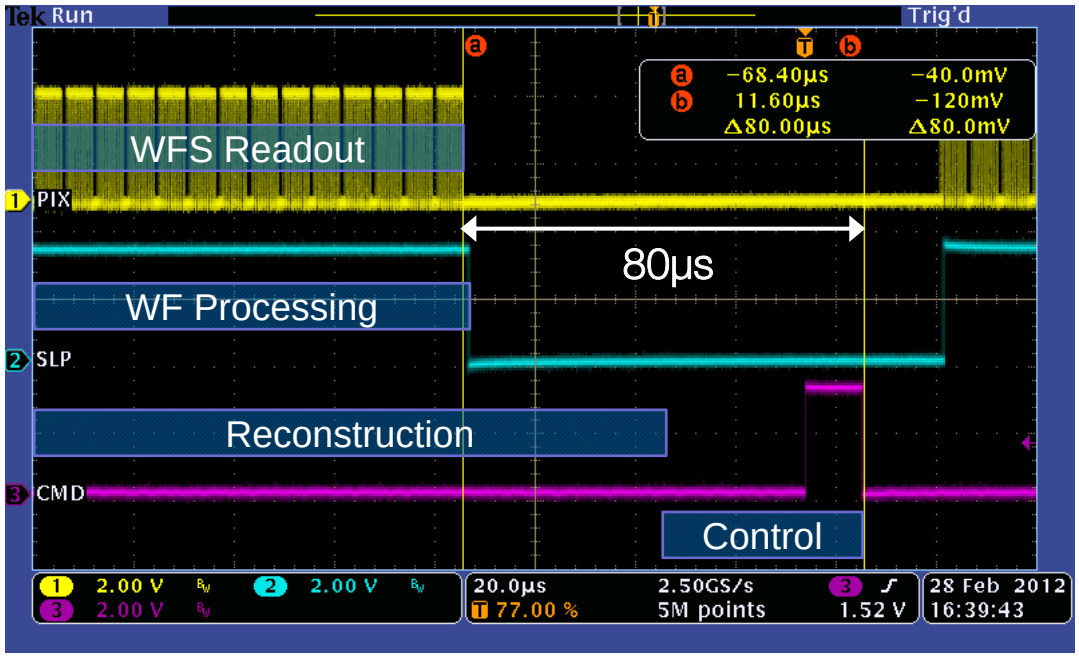


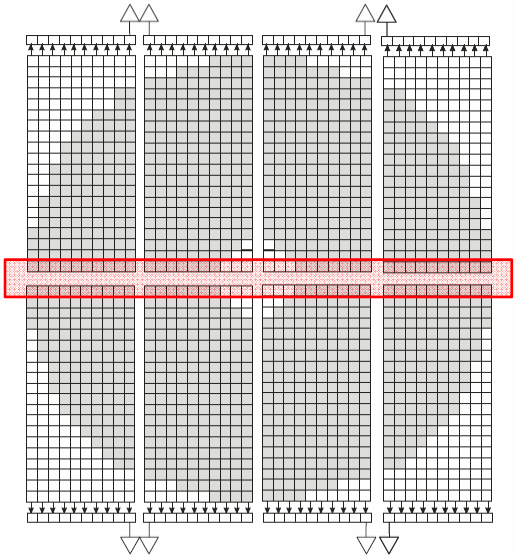
Real-Time Box Processing Nodes

- WCoG in parallel with WFS readout
- Block-wise reconstruction in parallel with WCoG
- Time filtering in parallel with data transfer
- Loop telemetry/disturbance in idle time



Real-Time Box Performance



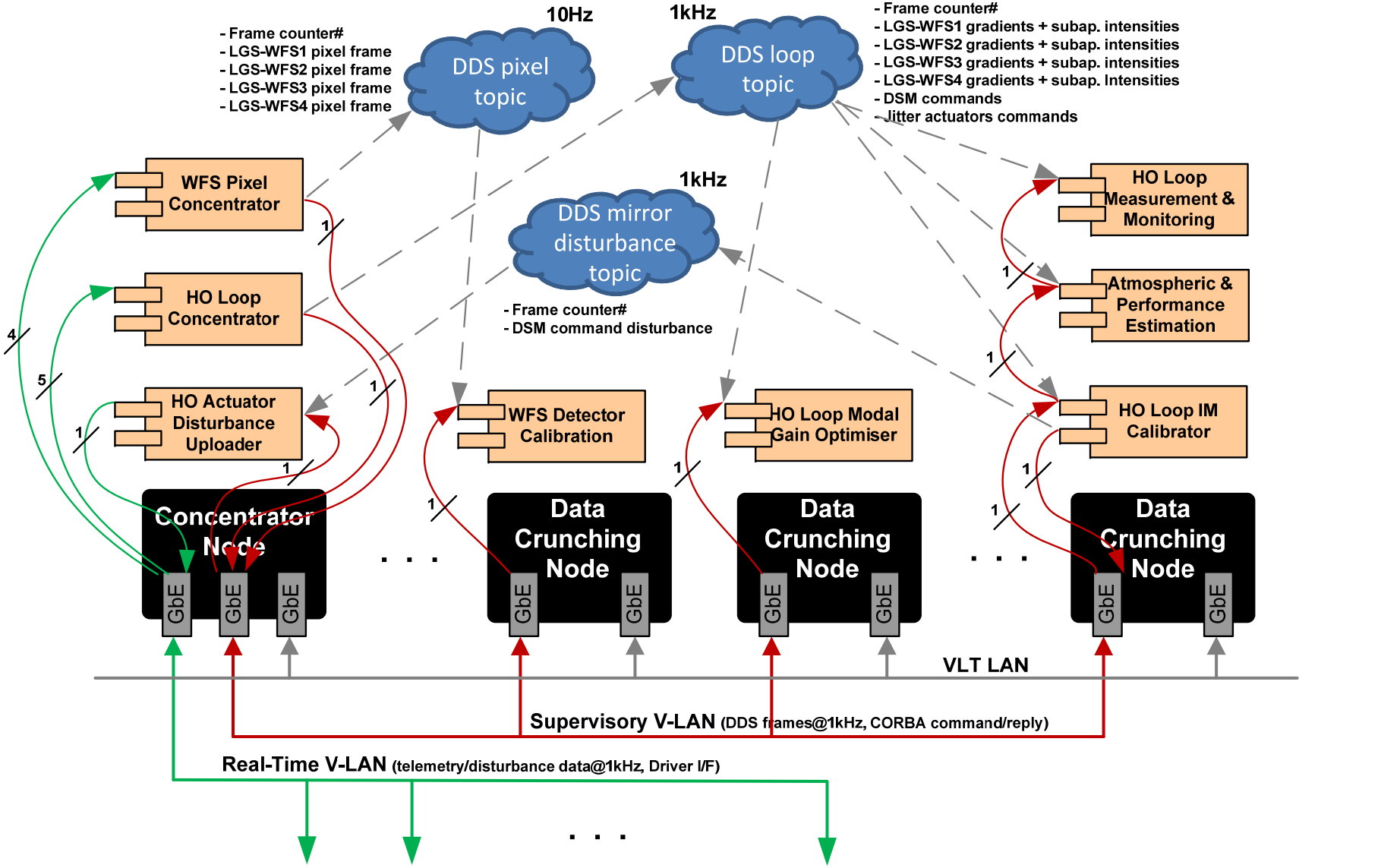


- Last WFS subaperture rows being processed

- SPHERE overall end-to-end latency: ~80µs
  - FPGA wavefront processing latency: ~2 µs
  - DSP reconstruction latency: ~40 µs
  - FPGA HO command filtering: ~17 µs
  - CPU TT command filtering: ~38 µs

} CPU bounded

Co-processing Cluster Nodes



Co-processing Cluster Performance

- DDS confirmed reliability QoS (no packet loss)
- IP multicast over IGMP-enabled switch
- Loop concentrator performance:

	Frequency	Size	Throughput	CPU usage (*)
SPHERE	1.2 kHz	20 kB	~24 MB/s	~60%
AOF	1 kHz	67 kB	~66 MB/s	~55% (mean) / ~160% (peak)

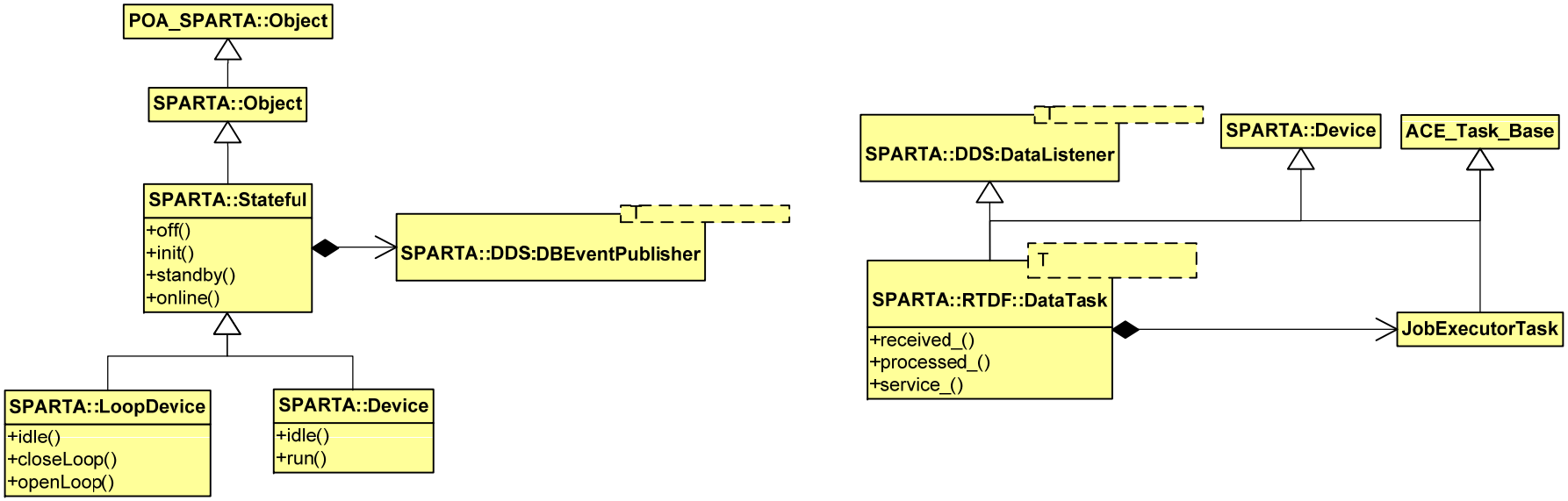
- Number crunching performance:

	Period	# Samples	CPU usage (*)
Modal Gain Optimisation	30 s	4,096	~26%
Atmospheric Statistics	15 s	4,096	~38%

(*) 2 x Xeon 5670 6-core CPU (2.93GHz)
12 GB DDR3 RAM

Co-processing Cluster Software

- Largest part of the SPARTA code base → portability/reusability
  - C++/Linux-based OS (Scientific Linux), no RT patches
  - CORBA (ACE/TAO) for scalable, distributed coordination
  - DDS (RTI) for scalable, distributed data processing
  - Intel MKL for portable, scalable performance
  - MATLAB MCR for complex algorithm portability
  - Minimum number of I/Fs to Real-Time Box, confined in low-level tier

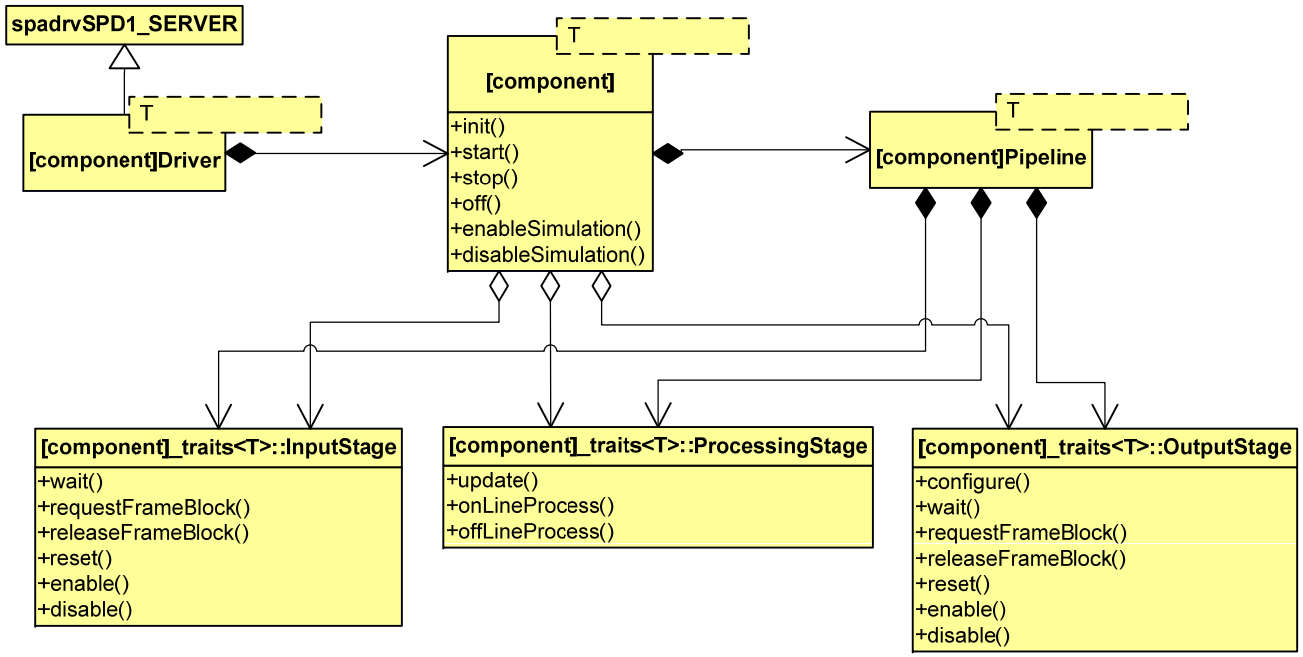


Real-Time Box Software

- Smallest share of the SPARTA code base
- Proportionally largest coding/testing effort

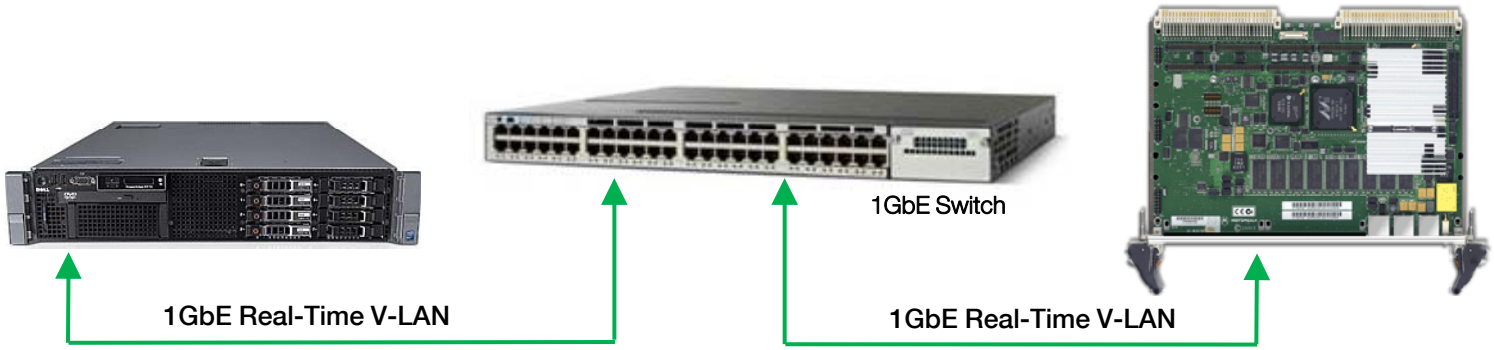
→ performance vs. HW-independence

  - C++/RTOS-based (vxWorks)
  - Strongly vectorised (Altivec), loop-unrolled code
  - Zero-copy, in-place data processing (Transcomm)
  - Pluggable, self-contained I/O and processing stages encapsulate HW-specifics
  - Compile-time stage composition and static polymorphism for maximum decoupling

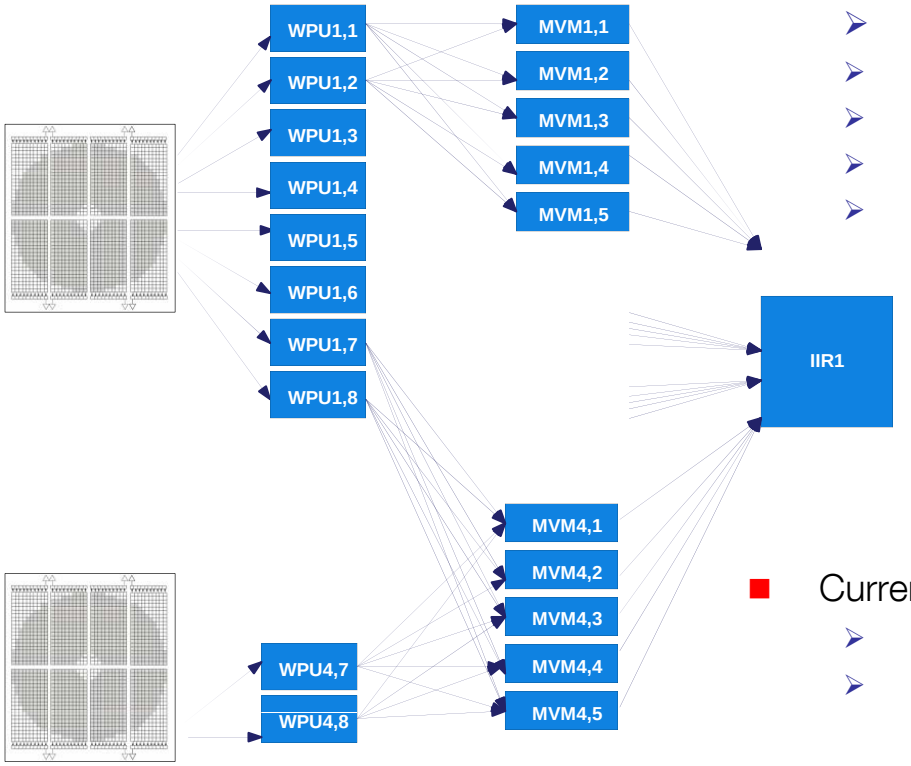


SPARTA Platform Reusability

- Cluster and Real-Time Box strongly decoupled → may be reused separately
  - Low-impact cluster connection to other RTC pipelines (e.g. workstation-based)
- Cluster scalable to great extent:
  - May be collapsed into a single node → SPARTA-Light
  - Or extended to serve bigger E-ELT AO instruments
- Cluster employs standard I/Fs and middleware:
  - Inter-operable with other DDS/CORBA-enabled systems/instruments
- HW-dependencies in Real-Time Box SW strongly encapsulated:
  - Low-impact porting to workstation environment
  - Low-impact porting to non-FPGA architectures → SPARTA-Light



SPARTA Future Plans (I)

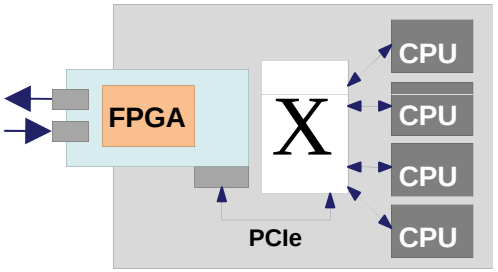


- Switched network RTC interconnects:
  - 10/40/100GbE links
  - UDP protocol potentially running RTPS (DDS interoperablility)
  - Layer 2, cut-thru switch: forwarding and reception in parallel
  - Full-duplex, isolated, P2P connections: no collision domain
  - Optimised switch configurations for 1:n and n:1 topologies

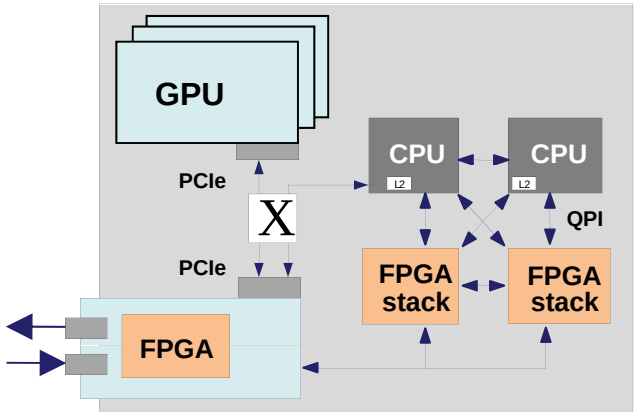
- Current 1:1 scale prototypes demonstrate:
  - Deterministic, low latency end-to-end data propagation
  - Repeatable, out-of-order packet delivery in n:1 topology

SPARTA Future Plans (II)

- Multi-CPU, multi-core Real-Time Box nodes:
  - FPGA-aided I/O: preserve GbE real-time capability



- Block-wise, multi-thread vector computation
- vxWorks for intel IA64 required?



. . .

- Off-line/online GPU computing engine
- Direct FPGA - GPU data transfers possible?
- In-socket FPGA stacks replacing some CPUs?

Thanks for your attention!

Questions?