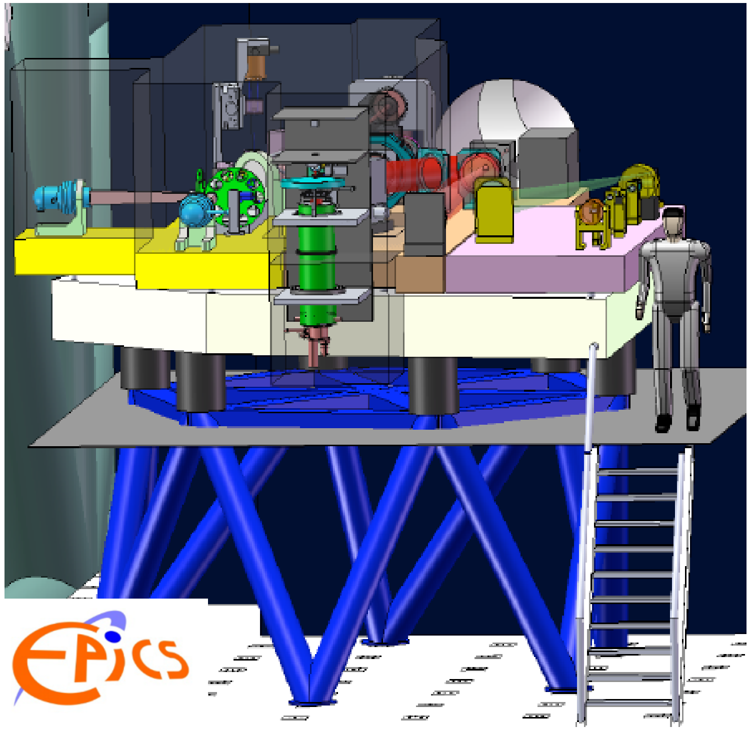
TNO innovation for life

Real-time control architecture for large scale AO systems / the EELT EPICS case

- Remco den Breeje
- Wimar Klop
- Niek Doelman

› In collaboration with Christoph Keller (UL), Michel Verhaegen (TUD).



RTC Workshop, Garching

4-5 December 2012

**Work also sponsored by AgentschapNL.*

Contents

- Problem statement of real-time control for large AO systems
- Possible solutions
- Multi-GPU cluster
- Prospect for EELT-EPICS and other large-scale AO systems


EELT-EPICS in short

- Exo-plant imager and spectrograph with the EELT
- Very high contrast needed: order 10^{-9} at 100 mas.
- Concept baseline* for AO system
  - 4.10^{4} actuators in DM
  - 4.10^{4} "sub-apertures" in WFS
  - 2-3 kHz sampling rate / frame rate

- Most demanding AO RTC system of EELT instruments family, other instruments (EAGLE, MAORY, ..) also challenging

** Numbers as used in real-time analysis here*

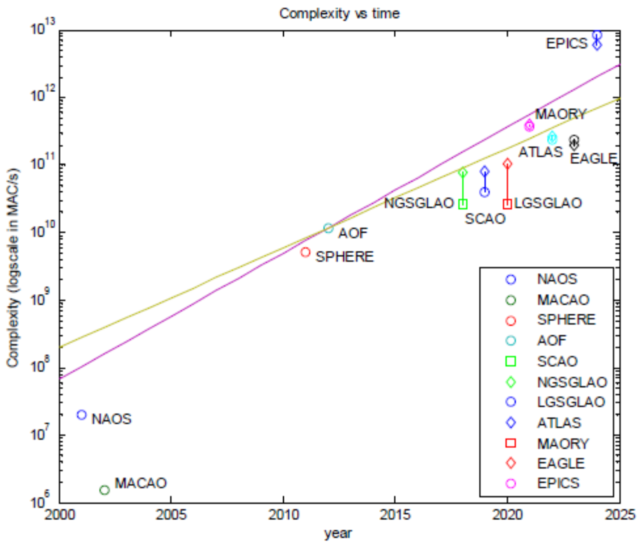
TNO innovation for life

Real-time control demands for EELT- EPICS

› Assuming common wavefront reconstruction (Matrix Vector Multiplication MVM), including integrator control and DM influence matrix compensation

› Matrix access in MVM: 6.10^{12} elements/ s
  - Multiply-add operations: ~ 6 TMAC/s
  - Memory bandwidth: ~ 24 Tbyte/s (32 bits)

› Parallel processing required



ref: Fedrigo, SPIE 2010


Parallel processing or efficient algorithm

1. Computationally more efficient algorithm/ approximation

- Several initiatives; for instance CuRe, FRiM, FTR, D-SABRE, …
- Focal plane image-based reconstruction; Korkiakoski (this workshop)

Important issues with new algorithms (on single CPU)

› Development time;	algorithm design, on-sky validation effort
› Performance;	WF reconstruction error
› Latency;	effective computation time
› Jitter;	convergence time (recursive algorithms)
› Realization;	coding and testing
› Maintenance and calibration time;	

Parallel processing or efficient algorithm

2. Multi-processor real-time computation platform

- MVM algorithm is highly parallelizable
- Fast developments in multi-processor world; CPU, GPU, FPGA

Important issues with multi-processor platforms (with MVM)

- Development time; design parallel RTC cluster
- Performance; e.g. 16 bits vs. 32 bits implementation
- Latency; limited memory bandwidth
- Jitter; communication jitter
- Realization; algorithm parallelization, hardware cost
- Maintenance and calibration time;

Parallel processing or efficient algorithm

2. Multi-processor real-time computation platform

- MVM algorithm is highly parallelizable
- Fast developments in multi-processor world; CPU, GPU, FPGA

<u>Important issues with multi-processor platforms (with MVM)</u>

› Development time;	design parallel RTC cluster
› Performance;	e.g. 16 bits vs. 32 bits implementation
› Latency;	limited memory bandwidth
› Jitter;	communication jitter
› Realization;	algorithm parallelization, hardware cost
› Maintenance and calibration time;	

3. So far only 2 options (?) … How about combinations of the above?

Which type of cluster?

- multi-CPU vs. multi-GPU vs. FPGA cluster
- Criterion trade-off table:

	Criterion	Description	Weighting factor
1.	Processing power	The computational power per node (MAC/s)	5
2.	Memory bandwidth	The speed of data-throughput (MB/s)	5
3.	Flexibility	The ease to modify the implemented control algorithm (software development time)	4
4.	Clustering	The ability to distribute the computations over multiple nodes	4
5.	Roadmap	The prospects of the technology developments	3
6.	Real-time performance	The ability to have equal computational delays in the real-time loop (repeatability)	3
7.	Maintenance	The cost to maintain and upgrade the RTC platform	2
8.	Costs	The development and reproduction cost of an RTC system	2
9.	Scalability	The ease to extend the RTC system with more computational nodes	1

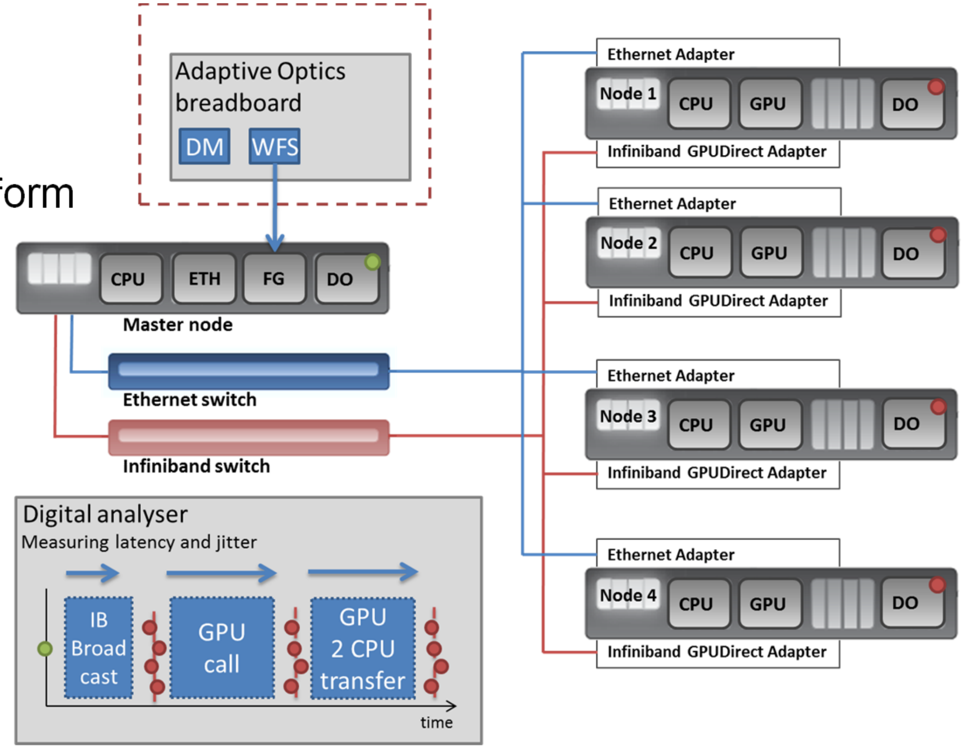

GPU cluster most attractive (year 2012)

- Pro’s: memory bandwidth, clustering, cost
- Con’s: none significant

Criteria	(weight)	CPU	GPU	FPGA
Memory bandwidth	5	0	++	+
Processing power	5	-	+	++
Clustering	4	+	+	0
Flexibility	4	++	+	-
Roadmap	3	0	+	++
Real-time	3	+	0	++
Maintenance	2	++	0	--
Costs	2	+	+	--
Scalability	1	+	0	0
Cumulative weighted result		17	28	15

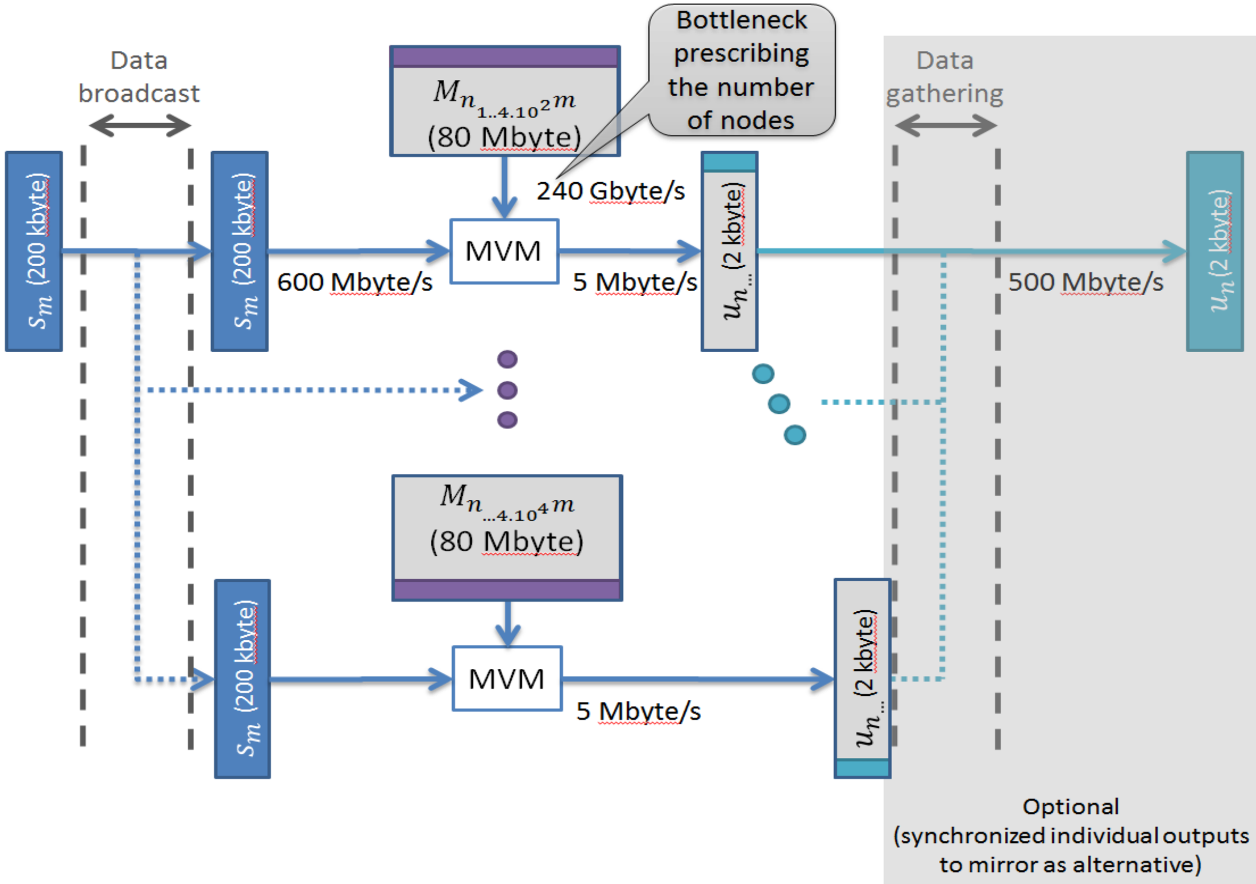
Cluster design – hardware software

- QDR Infiniband network (scalability)
- Nvidia GTX570 GPU (memory bandwidth, 150GB/s)
- Intel i7 3820 CPU
- Open Message Passing Interface (MPI) as communication protocol
- Ubuntu Linux operating system
- Xenomai real-time framework
- CUDA™ parallel computing platform



Cluster Design - Algorithm

- MVM algorithm with row-based parallelization for the nodes and tiling for GPU



Parallel implementation of MVM operation

CUBLAS

- Standard CUDA library provided by NVidia
- Black box as library is closed source
- Degrading performance when matrices get taller

Fujimoto approach

- GPU-based MVM algorithm developed by N. Fujimoto.

  [Proceedings IEEE International Parallel and Distributed Processing Symposium (IPDPS), LSPP-402, 2008]
- Available source allows for customization
- Modified to better fit the extreme tall matrices due to row-based parallelization
- Test with fixed point shorts show a speed-up factor of almost 2.

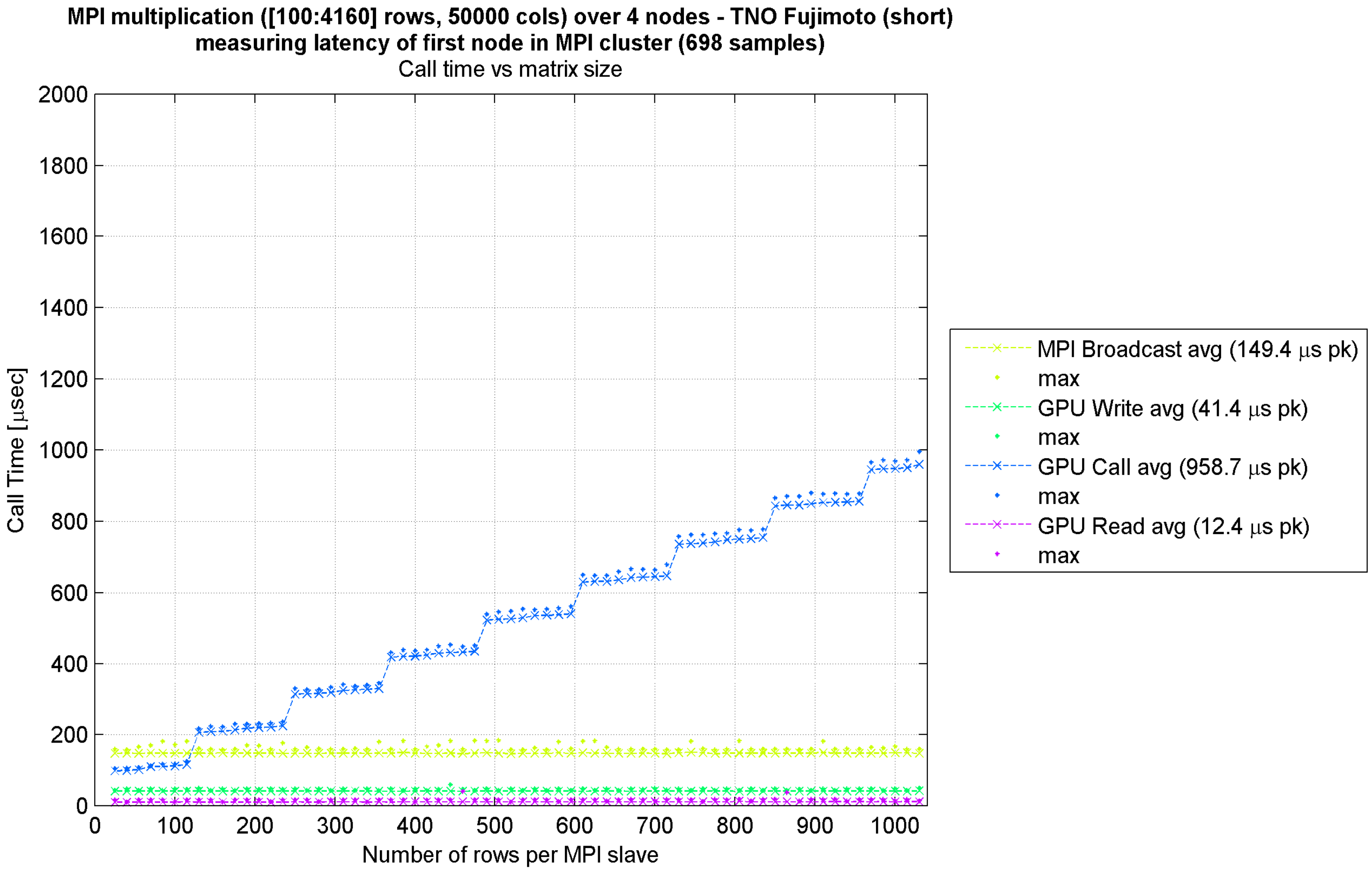

Experimental validation

- 4-GPU GTX 570 cluster
- Measurement of timing for each process step
- Determine maximum achievable number of matrix rows within 1 frame

Results:

- Cluster uses 4% of computational power
- 96% of theoretical memory bandwidth achieved
- Including algorithm overhead: 76% effective memory bandwidth
- Total latency and jitter OK, but could be lower
- Further improvement expected with alternatives for MPI broadcast and MPI barrier and using GPUdirect(?).

Real-time platform results – parallelized MVM algorithm



Experimental validation

- 4-GPU GTX 570 cluster
- Measurement of timing for each process step
- Determine maximum achievable number of matrix rows within 1 frame

Results:

- Cluster uses 4% of computational power
- 96% of theoretical memory bandwidth achieved
- Including algorithm overhead: 76% effective memory bandwidth
- Total latency and jitter OK, but could be lower
- Further improvement expected with alternatives for MPI broadcast and MPI barrier and using GPUdirect(?)

✓ **Cluster with 4 GTX 570 runs EELT-M4 wavefront reconstruction (SCAO) within 0.5 ms.**

What would the EPICS RTC look like, in 2012?

- Extrapolating the experimental results with 4-GPU cluster
- Realizing the cluster with 2012 GPU technology
- Running at 76% effective memory bandwidth

- Fastest GPU currently available; bandwidth 250 GB/s
  - ~ 170 nodes (32 bits implementation)
  - ~ 90 nodes (16 bits implementation)

  - Effective latency: ~ 1 sample period

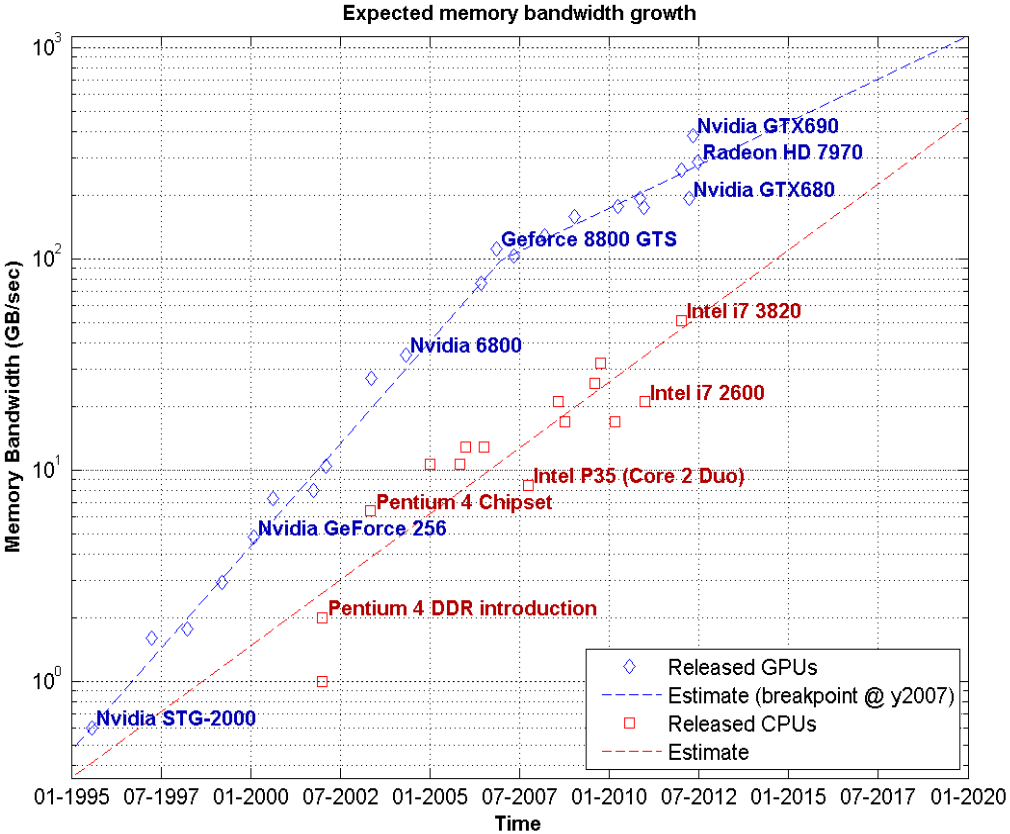
- Still a large cluster, with considerable hardware cost


What would the EPICS RTC look like, when needed?

- Realizing it in 10 years
- Prospect of memory bandwidth improvement: up to 1-2 Tbyte/s

- Number of nodes required:
  - ~30 (32 bits, 1.5 TB/s)
  - ~16 (16 bits, 1.5 TB/s)

- Small cluster of limited hardware cost



Implementation of alternative algorithms

Requirements for a successful implementation:

1. Algorithm should be parallelizable on two levels, on a high level over the nodes and on a low level over the cores inside a single GPU.
2. Global communication should be minimal. Intermediate central processing steps should be avoided.
3. Local communication is possible by using the shared memory or global memory inside the GPU.
4. Preferably no post-processing step on a central level. Local GPU results are communicated directly to designated deformable mirror channels.

5. For high-performance algorithms; LQG, H_∞ -type:
   - Each additional order of controller adds another matrix
   - First approximation: problem scales with controller order
   - Distributed versions of high-performance control required

TNO innovation for life

Conclusions

- Multi-GPU cluster attractive for large scale AO systems based on MVM.
- Successful validation of 4 GPU test set-up
- Test set-up is already sufficient for EELT-M4 SCAO.

- Realizing multi-GPU cluster appeared to be straightforward.
- Parallel programming (MVM) is a minor complication compared to single CPU case.

- EPICS would need a small cluster (16-30 nodes) in 10 years;
  to run the MVM algorithm


Conclusions

- Multi-GPU cluster attractive for large scale AO systems based on MVM.
- Successful validation of 4 GPU test set-up
- Test set-up is already sufficient for EELT-M4 SCAO.

- Realising multi-GPU cluster appeared to be straightforward
- Parallel programming (MVM) is a minor complication compared to single CPU case

- EPICS would need a small cluster (16-30 nodes) in 10 years; to run the MVM algorithm

A multi-processor cluster for large scale AO is not a big step

- Option for high-performance control algorithms?