Work Package 4

4.3: Xeon Phi MCAO tests + EPYC server results

l'Observatoire de Paris PLDA MICROGATE Green Flash Durham University

Outline

- The Xeon Phi
- DARC
- Optimisations of DARC for Xeon Phi
  - Thread Synchronisation
  - Subaperture allocation
  - OS, BIOS, kernel settings etc.
- Current SCAO results
- MCAO results
- Conclusion

The Xeon Phi


l'Observatoire
de Paris

PLDA

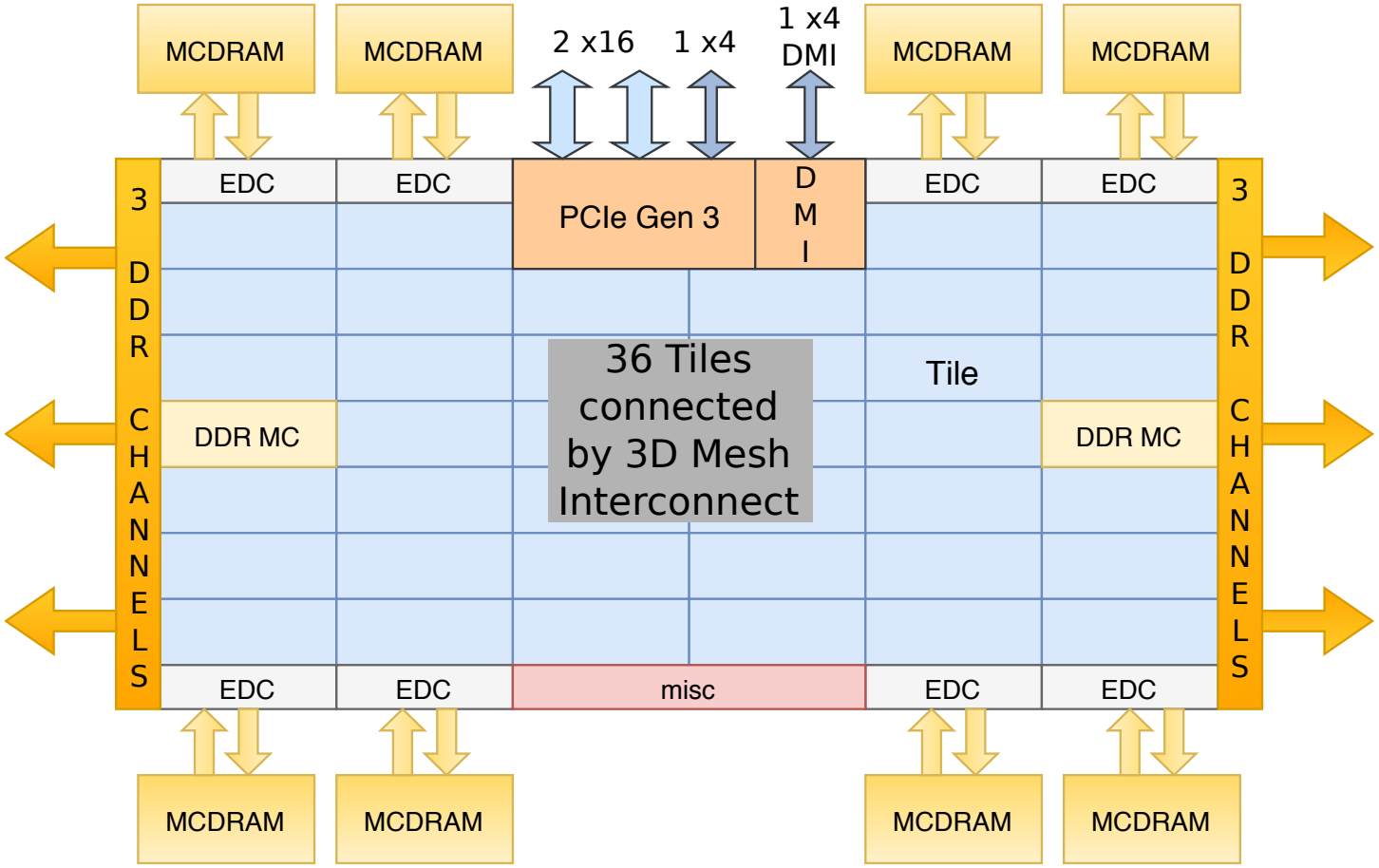
MICROGATE

Green
Flash

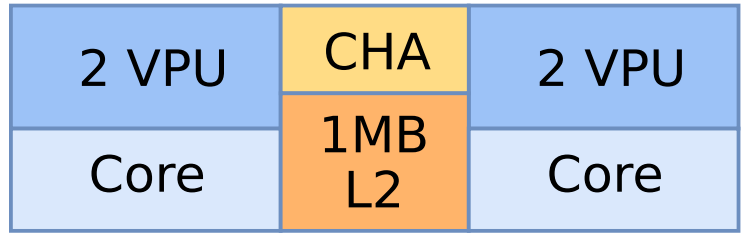
Durham
University

Xeon Phi Overview

- The Xeon is a range of Intel Many Integrated Core architecture CPUs
- Standard socketed CPU with a standard Linux OS and software packages
- Up to 72 CPU cores with mesh topology
- Measured memory bandwidth of ~480GB/s
- 512bit vector registers allow up to 16 SP FLOPs per instruction cycle
- Eliminates data transfer of accelerator cards and provides a unified software environment



1 Tile



A single tile composed of two cores and 2 VPUs per core

Mesh interconnect of Xeon Phi CPU, showing high bandwidth MCDRAM and PCIe interconnect

MCDRAM = Multi-channel DRAM, DDR MC = DDR memory controller, DMI = Direct Media Interface, EDC = MCDRAM controllers, MCDRAM = Multi-Channel DRAM, CHA = Caching and Home Agent

DARC

- The Durham Adaptive optics real-time controller
  - Fully modular on-sky tested AO RTC
  - Has been used extensively for the CANARY project
  - Many publications available outlining it's functionality, on-sky tested modes and results
  - Multi-threaded for improved performance, AO RTC can be very parallel
  - Has been optimised for the Xeon Phi's many cores

Optimisations

Optimisations

- Main optimisations of DARC for the Xeon Phis Many Integrated Core architecture
  - Replace the thread synchronisation mutexes with spinlocks
  - Allocate the correct number of threads for the problem size
  - Assign the correct number of subapertures per thread
  - Use optimum bios, kernel and OS settings
  - Enable autovectorisation and use manual vectorisation for performance critical parts
  - Parallelise the reduction operation for assembling the partial DM vectors

Synchronisation

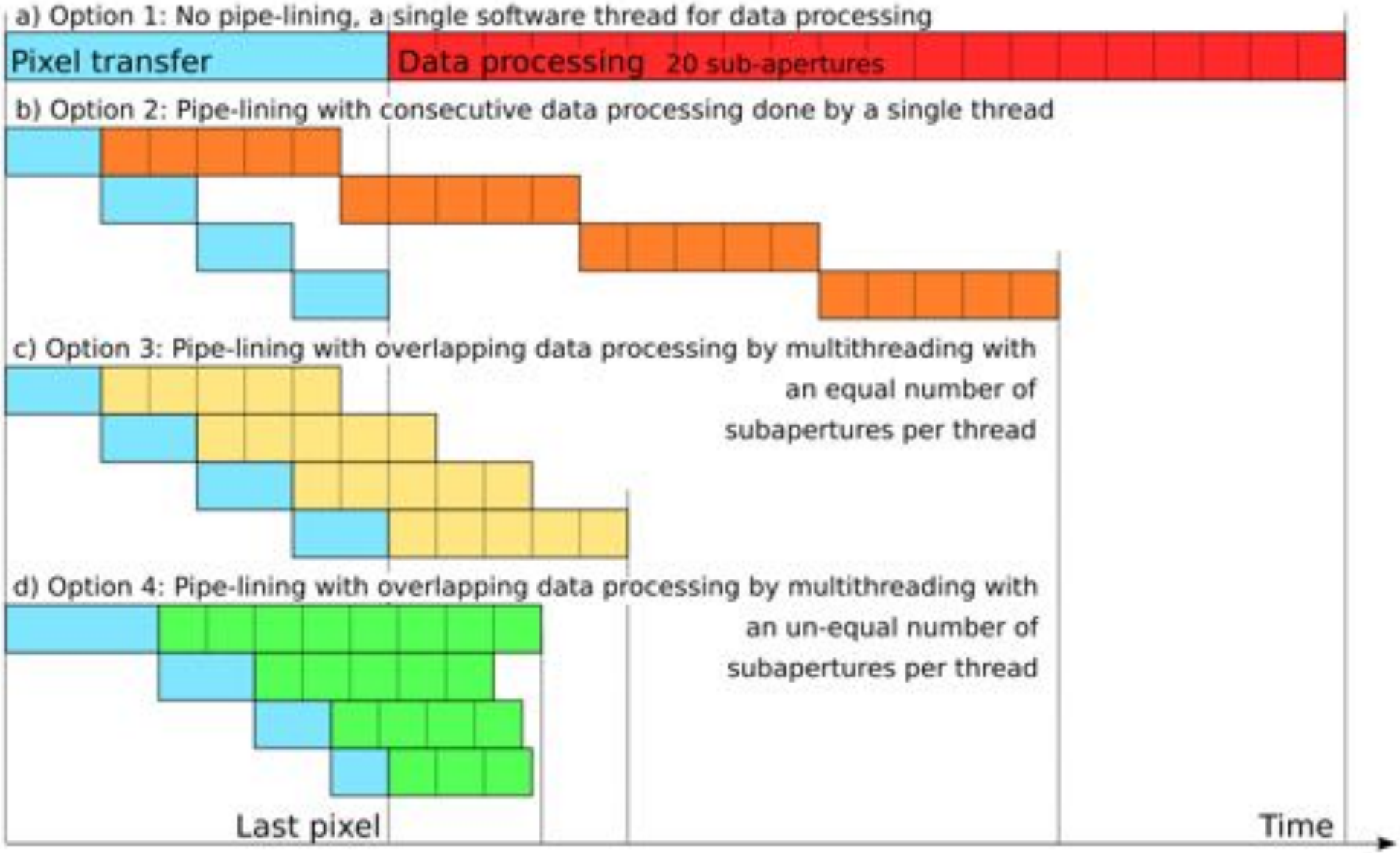
- Mutexes, mutual exclusion locks, are used to protect memory accesses in parallel sections of code
- Only one thread can acquire a mutex while others must wait
- Waiting threads can be put to sleep and descheduled
- Spinlocks similarly protect memory accesses in parallel sections of code
- Waiting threads instead block using a ”busy-wait”, simply consuming CPU cycles until they can proceed
- This is much more efficient for large numbers of waiting threads by reducing the wake latency

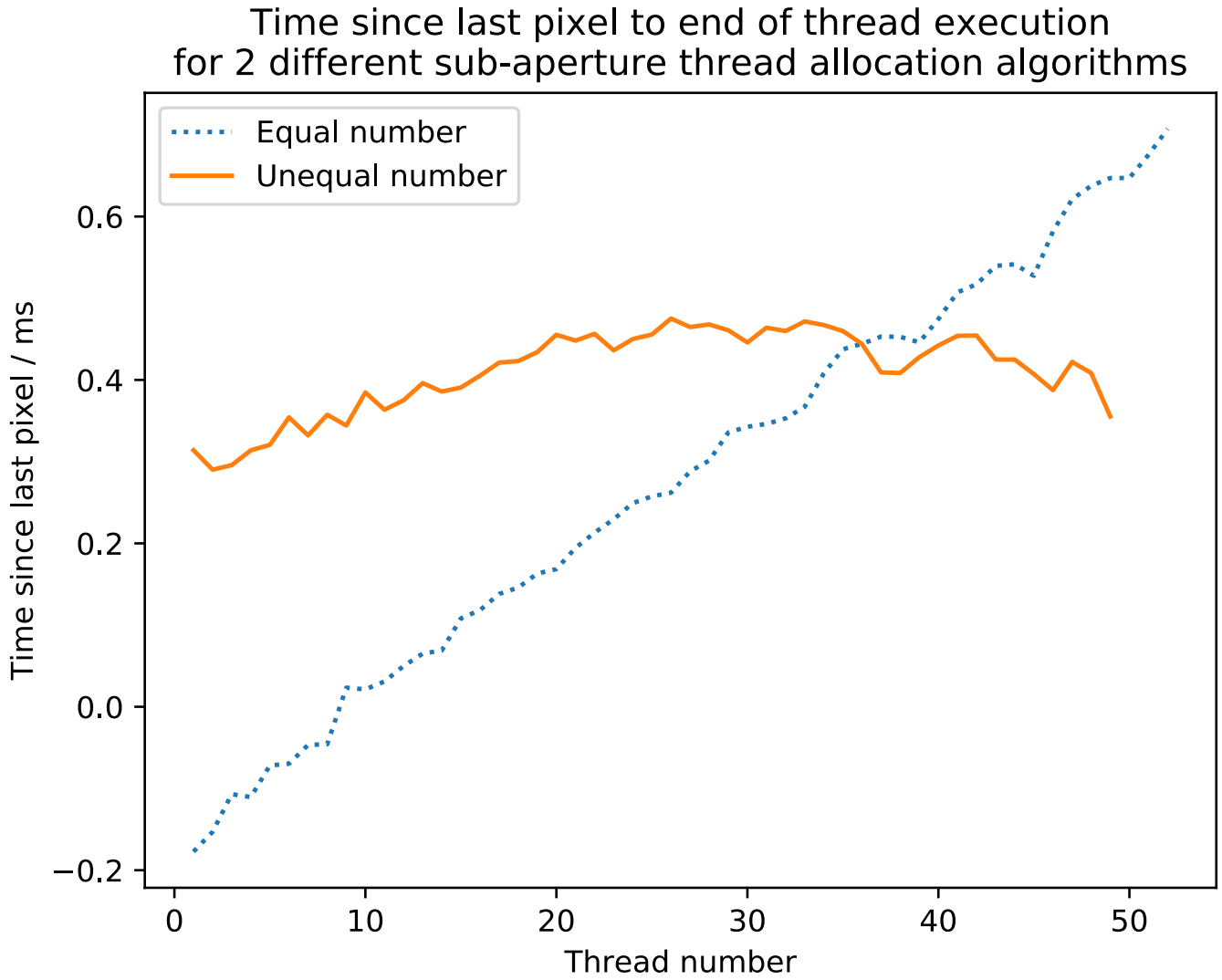
Thread Allocation

- Isolate enough CPU cores for the reconstruction threads, control thread and pixel grabbing thread
- All software threads are pinned to a unique CPU core to eliminate context switch latency
- Optimal reconstruction thread count for ELT-scale is 54-56 on a 64 core CPU
- Leaving 8-10 threads for the OS and other RTC functions
- Interrupts are reduced as much as possible by disabling irqbalance
- Camera network interrupts are pinned to specific isolated CPUs

Subaperture Allocation

- Previously the subapertures allocated equally to all threads so all threads have roughly equal workload
- To aid in pipe-lining of pixel transfer, threads which process the early arriving subapertures can be allocated more work
- This allows threads to finish at roughly the same time, decreasing latency and thread down-time


a) Option 1: No pipe-lining, a single software thread for data processing
Pixel transfer
Data processing 20 sub-apertures
b) Option 2: Pipe-lining with consecutive data processing done by a single thread
c) Option 3: Pipe-lining with overlapping data processing by multithreading with an equal number of subapertures per thread
d) Option 4: Pipe-lining with overlapping data processing by multithreading with an un-equal number of subapertures per thread
Last pixel
Time


Time since last pixel to end of thread execution
for 2 different sub-aperture thread allocation algorithms
Equal number
Unequal number
Time since last pixel / ms
0.6
0.4
0.2
0.0
−0.2
0
10
20
30
40
50
Thread number

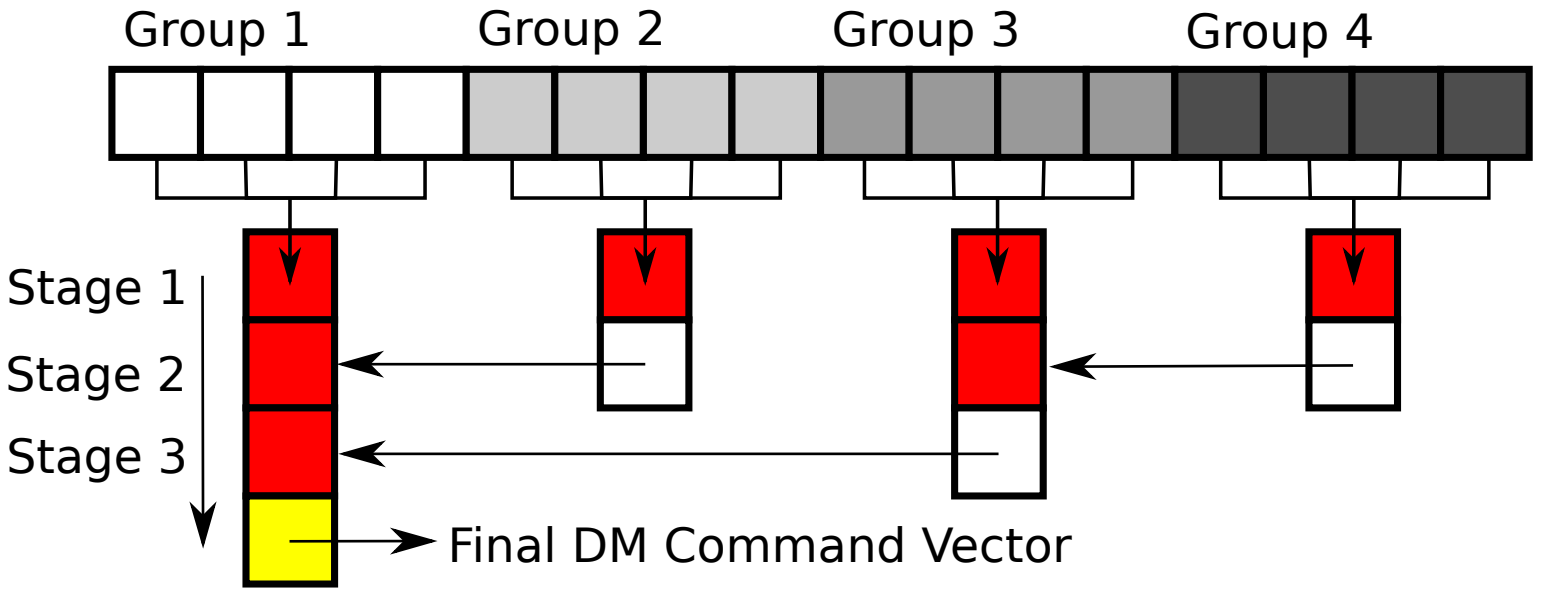
BIOS, kernel and OS setup

- BIOS setup
  - Hyperthreading off, reduces latency and jitter
  - CPU set to Quadrant mode, best default
  - Memory mode set to flat, all MCDRAM as user accessible RAM
- Kernel setup
  - isolcpus used to isolate the CPUs
  - irqaffinity set to the isolated cores
  - nohz mode set to on; to reduce jitter
- Linux OS setup
  - sysctl options optimised for best network performance
  - irqbalance system service disabled
  - CPU power settings set to performance

Vectorisation

- Vector operations allow the same operations to be performed for multiple data values simultanesouly; SIMD
- The Xeon Phis 512bit vector registers can allow up to 16 single precision operations to execute concurrently
- Some parallelisable structures such as for loops can be autovectorised using compiler optimisation
- Other more critical functions, e.g. MVM, can be vectorised manually using intrinsic instructions
  - Intrinsic instructions allow direct access to the vector data structures and the functions that operate on them
  - For loops can be partially unrolled to improve cache use
  - Can also investigate storing control matrix as 16bit half-floats

Reduction of partial DM vectors



Branching reduction algorithm for the summation of partial DM vectors.
Each groups adds into a temporary vector in parallel.

Generality of Optimisations

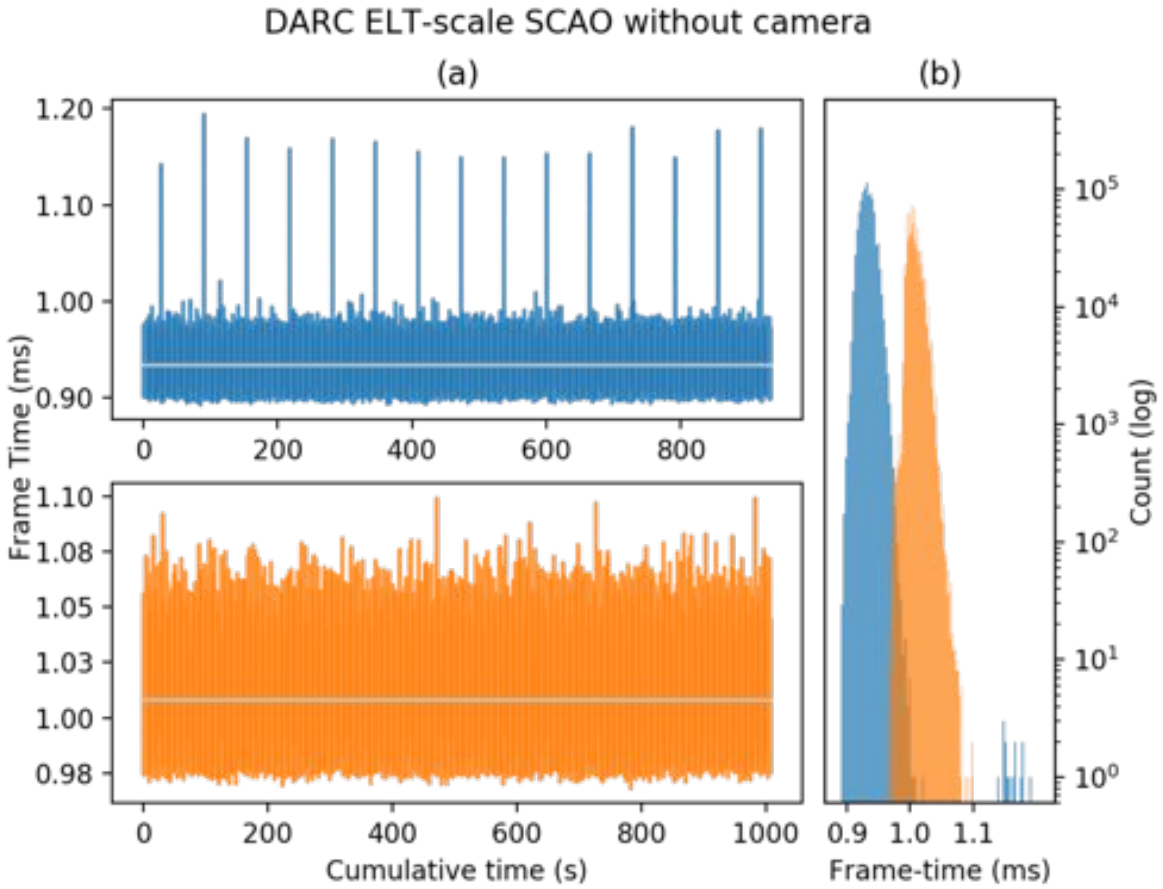
- Most of work carried out is not Xeon Phi specific
- Applicable to all modern many-core CPU systems
- Current and future systems from both Intel and AMD are compatible
- High memory bandwidth is critical for AO RTC
- Intel Xeon Exascale Architecture will continue the many core/high memory bandwidth idea behind the Xeon Phi
- Multi-socket CPU systems are capable of providing the required memory-bandwidth

SCAO results

Overview

- Testing DARC on Xeon Phi with an ELT-scale SCAO RTC configuration
- We test with a real camera and pipelining
- Can also test pure computation, no camera and no pipelining
- System specification = 80x80 subapertures, a total of 4632
- 5318 actuators for M4 + M5 tip-tilt
- >1kHz computational performance with ~10µs RMS jitter
  - and ~965Hz with a real camera attached (max camera framerate)

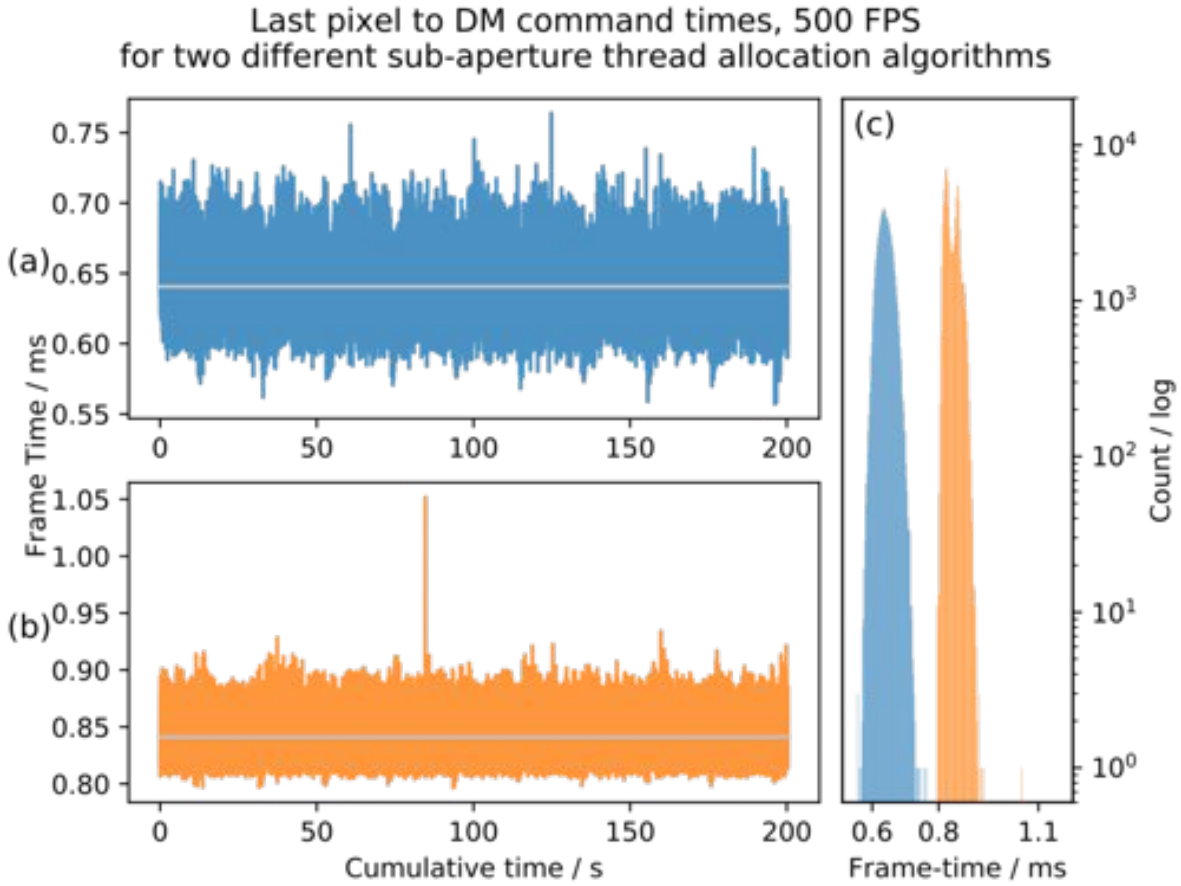
Motherboard Latency Spikes



a) Frametimes for DARC SCAO on an Intel Motherboard with no camera for 10^6 frames with an average frametime of (0.93 ± 0.01) ms which corresponds to an average framerate of (1070 ± 10) Hz. b) As for (a), except for a Supermicro motherboard with an average frame- time of (1.01 ± 0.01) ms which corresponds to an average framerate of (990 ± 10) Hz. c) Histograms of the frametimes presented in (a) and (b)

Jenkins et. al. submitted MNRAS 02/18

RTC Latency



Latency measurements (time between last pixel received to DM demand ready) for DARC operation with a real wavefront sensor camera at ELT-scale at 500 Hz. a) Shows results when using the unequal subaperture thread allocation. b) Shows results for the equal subaperture thread allocation. c) Shows the respective histograms

Jenkins et. al. submitted MNRAS 02/18

MCAO


l'Observatoire
de Paris

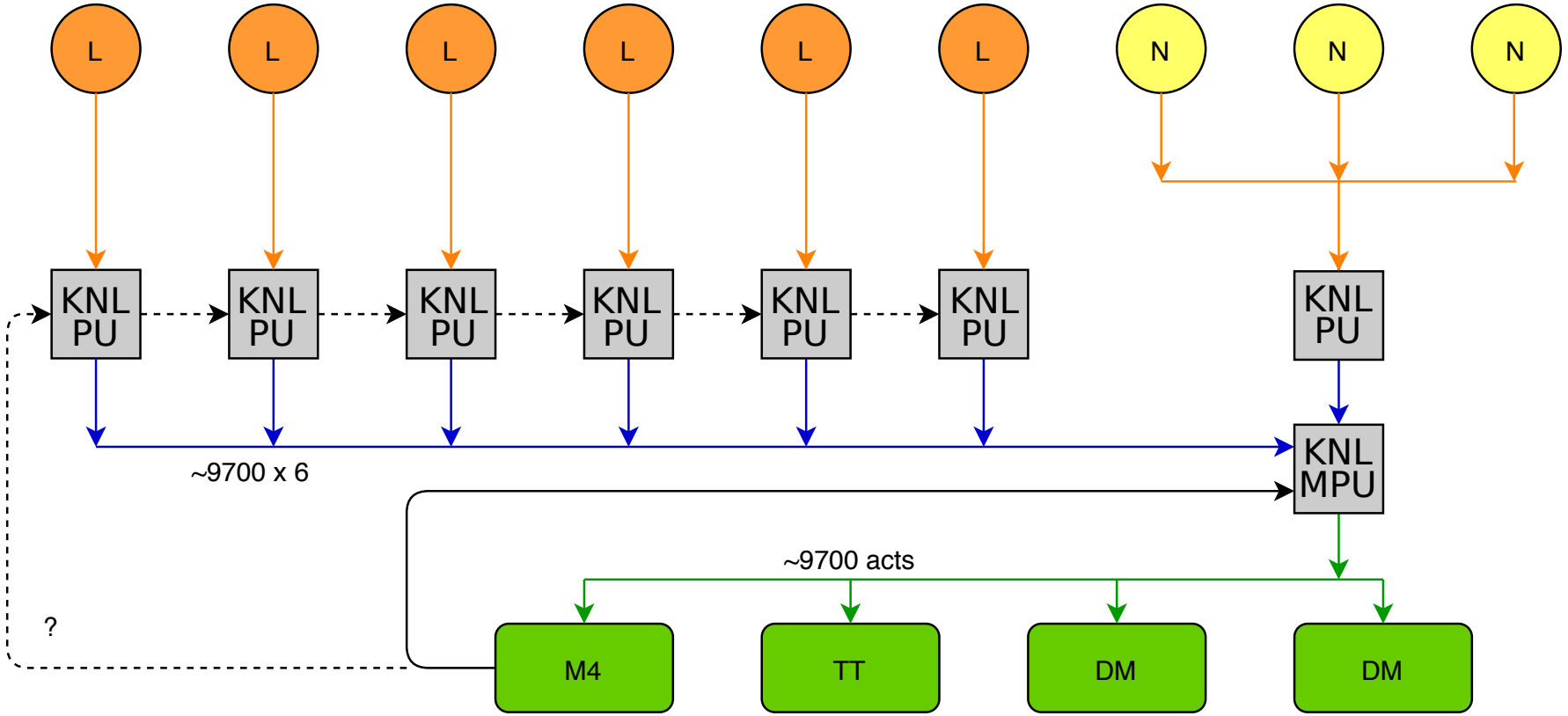
PLDA

MICROGATE

Green
Flash

Durham
University

Setup


L
L
L
L
L
L
N
N
N
KNL PU
KNL PU
KNL PU
KNL PU
KNL PU
KNL PU
KNL PU
~9700 x 6
KNL MPU
~9700 acts
?
M4
TT
DM
DM

Setup

- Use a MAORY-like specification for testing
- 6 80x80 LGS, 3 10x10 NGS, 4 DMs: M4, M5 + 2x 1700act DM
- Each LGS processed from images to partial DM commands by 1 KNL node
- 3 NGS processed by a single node
- Master combines partial DM commands
- Feedback from M4 to master node and potentially the LGS processing nodes

Setup

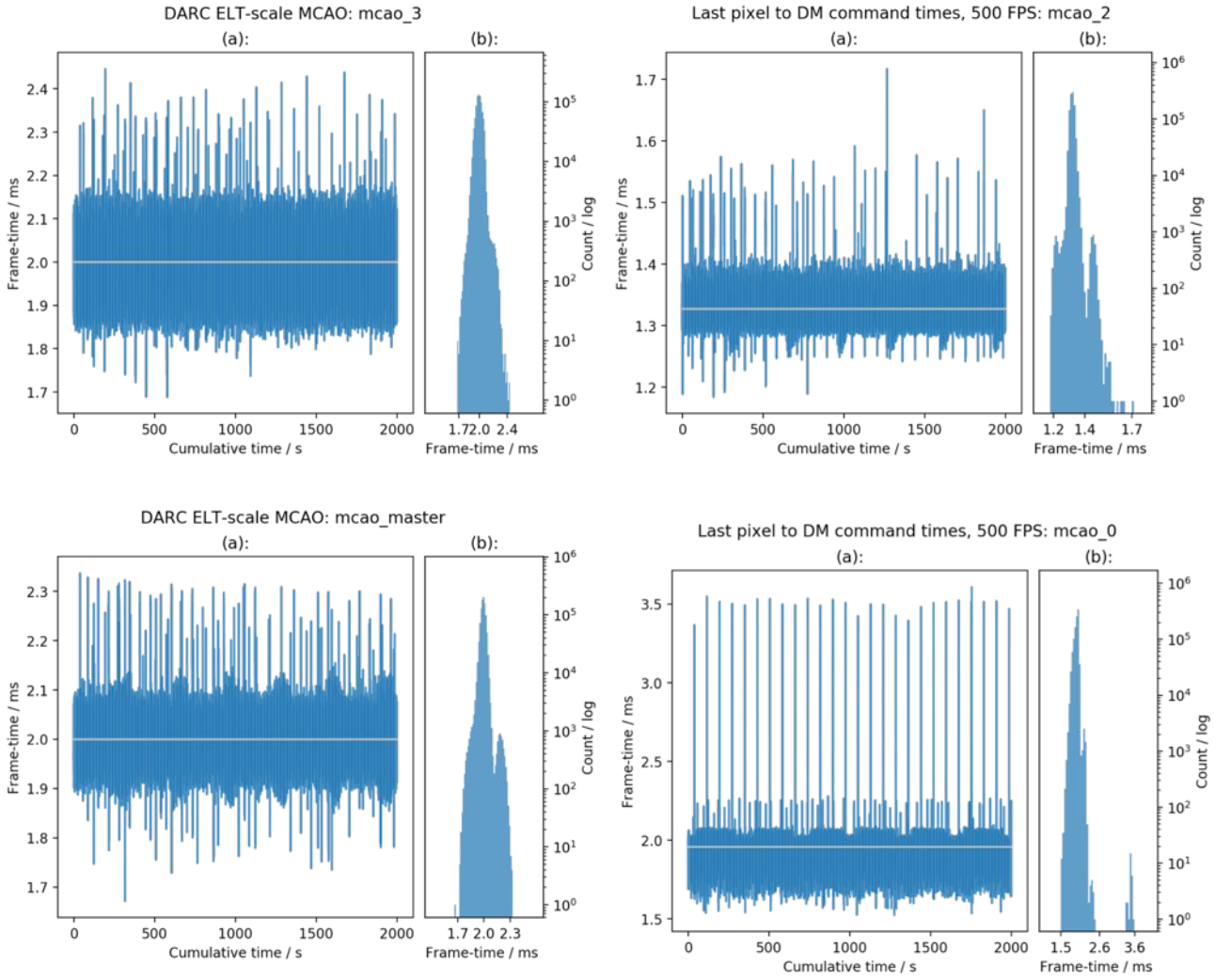
- Simulated camera is multicast to all processing nodes
  - Using Aravis camera library, introduces some overhead
  - Not deterministic, camera has jitter, data shaper required
  - Threads are run on isolated cores to improve jitter
- Camera outputs 800x800 pixels, 16 bits per pixel @ 500Hz
- Corresponds to ~4.8Gbps data rate per node
  - 1600x1600 would require ~19Gbps per node...
- Nodes communicate over 40G network

Results

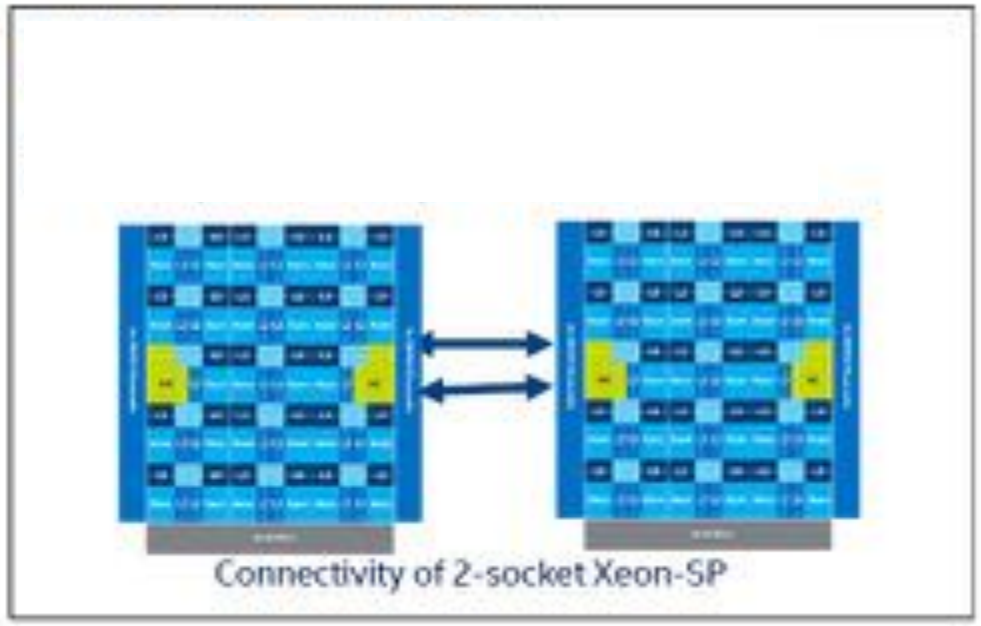
- Have demonstrated the idea with all nodes working together and processing a MAORY like system at 500Hz
- No data shaper so no external timing yet possible
- Timing is done on the machines, synchronisation is a challenge
- Regular latency-spikes similar to a frame drop every ~78s
  - We think this is due to the simulator/network, consistent on nodes
  - Needs investigation

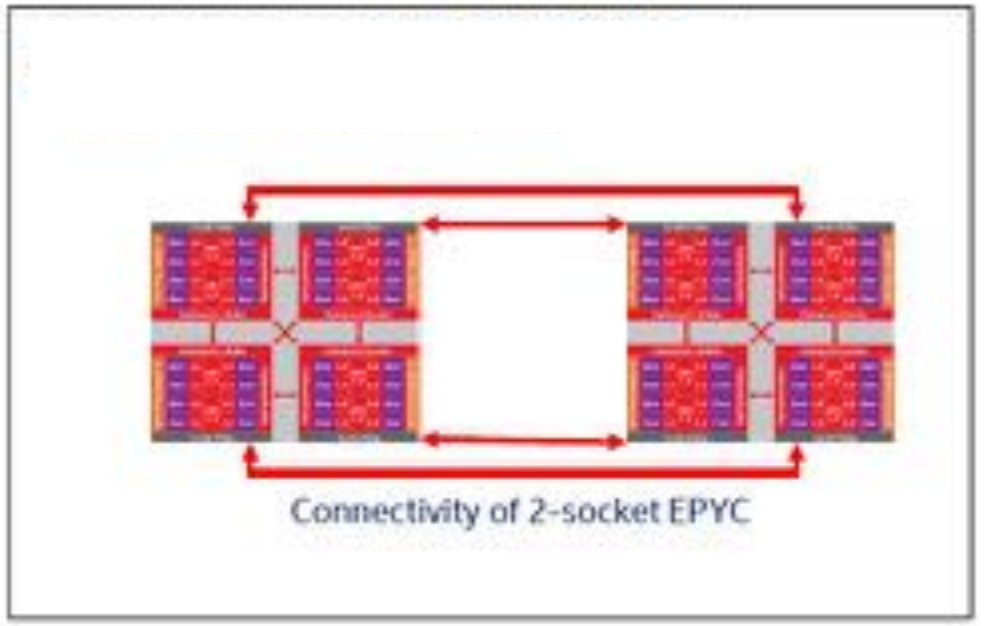
Results

- On-node timings show 2ms frametimes with ~50µs RMS jitter
- RTC latency for LGS nodes show <1.5ms timing with <20µs RMS jitter
- Camera simulator is believed to be the source of extra jitter
- Timing between nodes not possible without data shaper



AMD dual-socket EPYC


Connectivity of 2-socket Xeon-SP


Connectivity of 2-socket EPYC

NUMA-aware

- NUMA: Non-Uniform Memory Access
  - Latency and bandwidth between memory and cores can vary
  - Can be critical for memory-bound applications
    - e.g. AO RTC!
  - Understanding of the underlying hardware
- WFS/DM Hardware access
  - Specific cores will be closely tied to the WFS/DMs
- Core locality
  - Group similar tasks together when data sharing is necessary

Current Memory Performance

- ELT SCAO/LTAO/GLAO
  - Approx 100GB/s/WFS at 500Hz
- Intel Scalable processors
  - 6x memory channels per CPU (128GB/s)
  - 4 or 8 processors = 512-1024GB/s (~£20-40k)
  - 1 NUMA node per processor
- AMD EPYC processors
  - 8x memory channels per CPU (170GB/s)
  - 2 processors = 341GB/s (~£5k)
  - 4 NUMA nodes per processor
- IBM POWER8/9 processors
  - 128GB/s/CPU
  - 4-processor nodes £££~80

Recent Updates

- NUMA awareness:
  - New double-buffered parameter region for each NUMA node
    - Synchronous with main parameter buffers
  - Assign sub-apertures to cores
- Place relevant part of control matrix in correct NUMA regions
  - Significant performance improvements
- Dual Xeon E5-2630v3 CPU – 4.5x reduction in computation time
  - ELT SCAO at 330Hz (75% of theoretical, no tuning)
- Dual EPYC 7351 CPU – 10x reduction in computation time
  - ELT SCAO at 800Hz (50% of theoretical, no tuning)
- Note – not required for a Xeon Phi CPU (>1kHz ELT SCAO)

Notes:

- Very early results
- Uses Xeon Phi Optimised code but NUMA aware
- NUMA awareness is key to ELT-scale AO RTC on multi-socket systems
- No performance tuning yet
- An EPYC specific config file would give optimal performance

Summary

- ELT-scale SCAO is achievable on a single Xeon Phi node at >1kHz
- An 8 node Xeon Phi cluster has demonstrated ELT-scale MCAO with a multi-casted simulated camera at 500Hz
- There is potential for multi-socket standard CPU systems to achieve similar results with NUMA-aware DARC and tuning

WP4: Supervisor (CPU)

Task 4.3: Xeon Phis as alternative solution

l'Observatoire de Paris
PLDA
MICROGATE
Green Flash
Durham University

Soft Real-Time Core Requirements

- Consume real-time telemetry data from the hard-RT pipeline in real time
- Perform matrix inversion to compute control matrices in 5 minutes
- Compute phase covariance within 1 minute
- Compute reference slopes at 0.5-10Hz

Approach

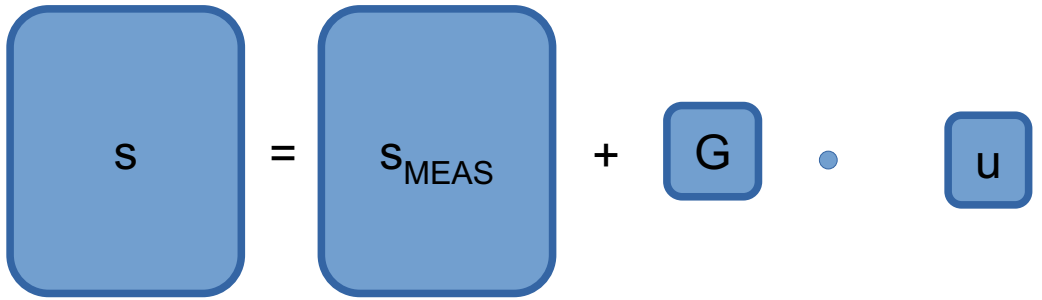
- GreenFlash supervisor concentrates on largest computational load for ELT instrument

GreenFlash	Non-GreenFlash
Pseudo Open-Loop Slopes?	M4 Position Monitoring
Parameter Estimation	Projection Matrices
Covariance Matrix Generation	Pupil Position
Matrix Inversion	LGS/NGS Centroid Gains
Reference Slope Updates	PSF Reconstruction

- For GreenFlash case and individual ELT instruments
  - Resolution of parameters
  - Matrix sizes

Pseudo Open Loop Slopes

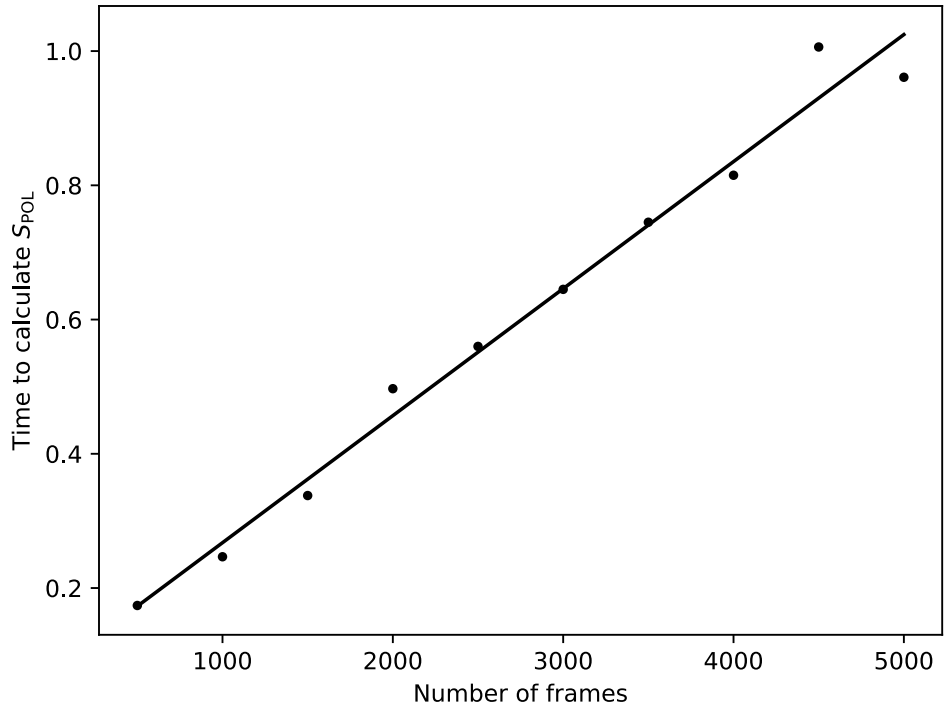
- Can be computed as a batch process
  - Many frames at once
  - Matrix-matrix multiplication
- Compute bound, not memory bound
  - (one, not per WFS)
  - 1-2s delay – not significant



Pseudo Open Loop Slopes

$$\overrightarrow{S}_k^{\mathrm{POL}} = \overrightarrow{S}_k + \hat{G}\overrightarrow{u}_{k-1}$$

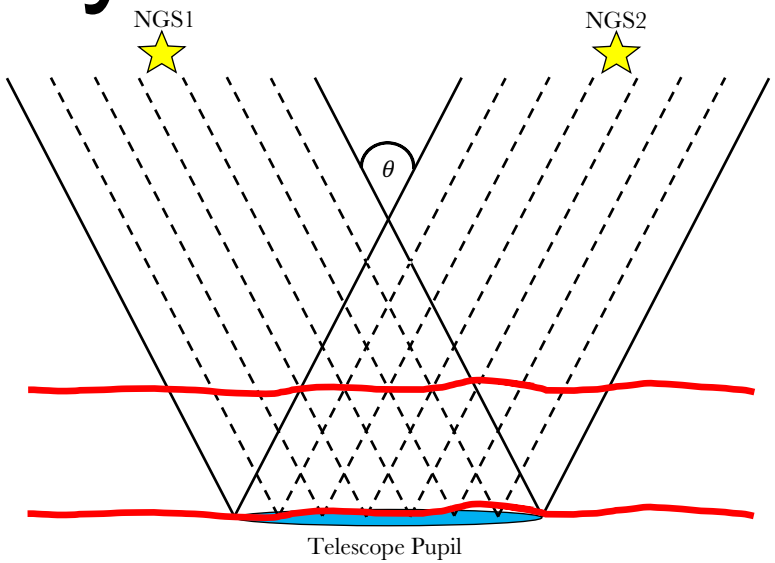
$$\overrightarrow{S}_k^{\mathrm{POL}} : [10k \times N_{\mathrm{WFS}}, N_{\mathrm{frames}}]$$
$$\overrightarrow{S}_k : [10k \times N_{\mathrm{WFS}}, N_{\mathrm{frames}}]$$
$$\hat{G} : [10k \times N_{\mathrm{WFS}}, 5k]$$
$$\overrightarrow{u}_{k-1} : [5k, N_{\mathrm{frames}}]$$



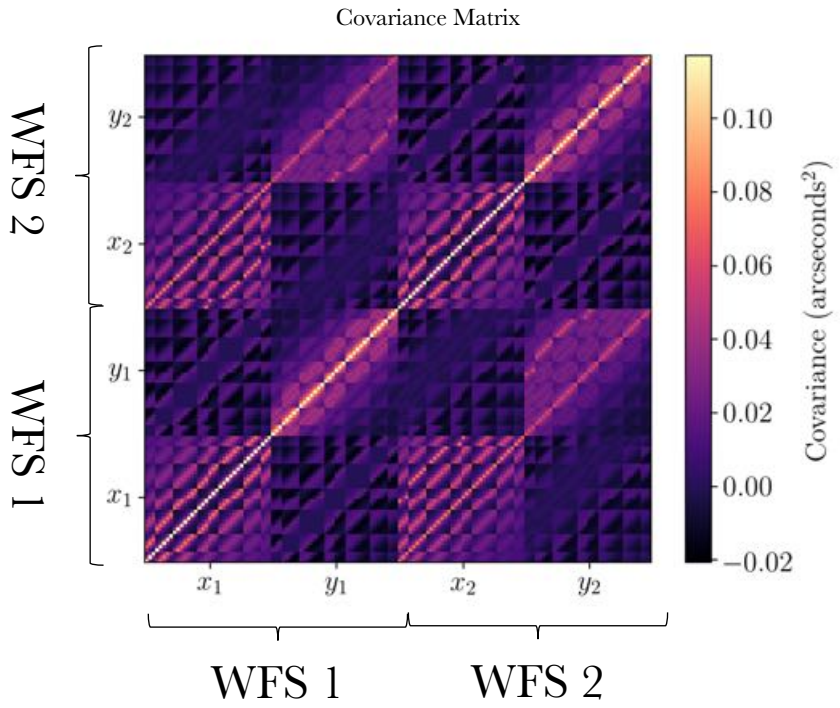
- Faster than real-time
  - For 2 WFSs
  - Done in “batch” mode

l'Observatoire de Paris PLDA MICROGATE Green Flash Durham University

Characterising optical turbulence with AO telemetry

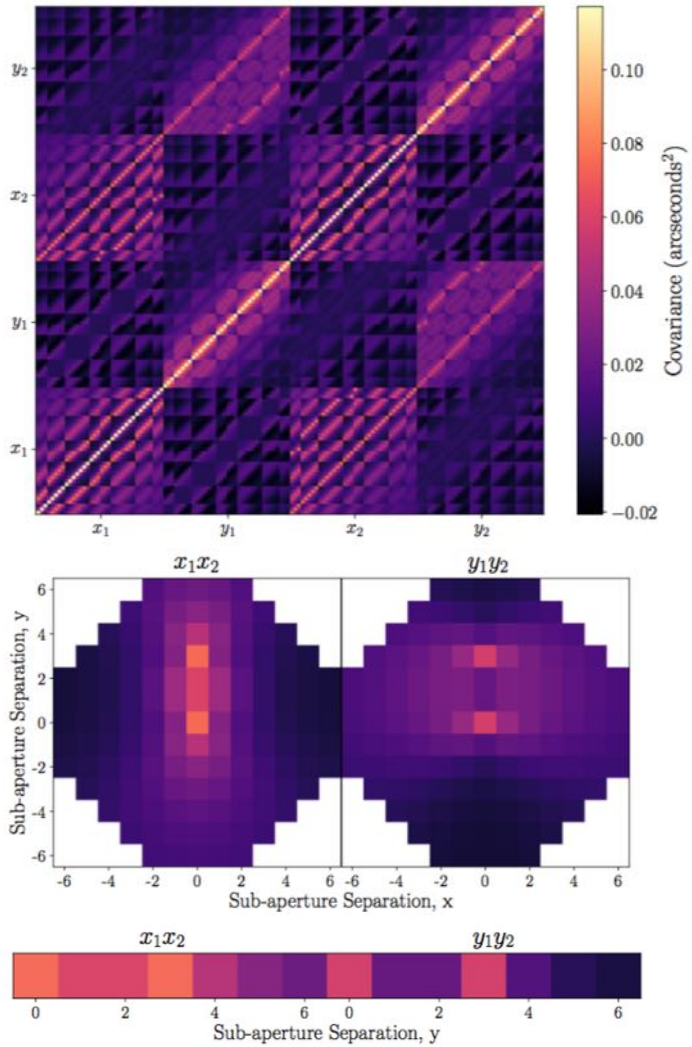


- Turbulence profile
- ”Outer scale” profile
- Wind velocity profile
- Instrument misalignments



Parameter Fitting

- Turbulence profile
- Mis-alignments
  - Offsets
  - Rotations
- Fitting sub-aperture covariance
  - Covariance Matrix
  - Covariance Map
  - Covariance Map ROI



Parameter estimation

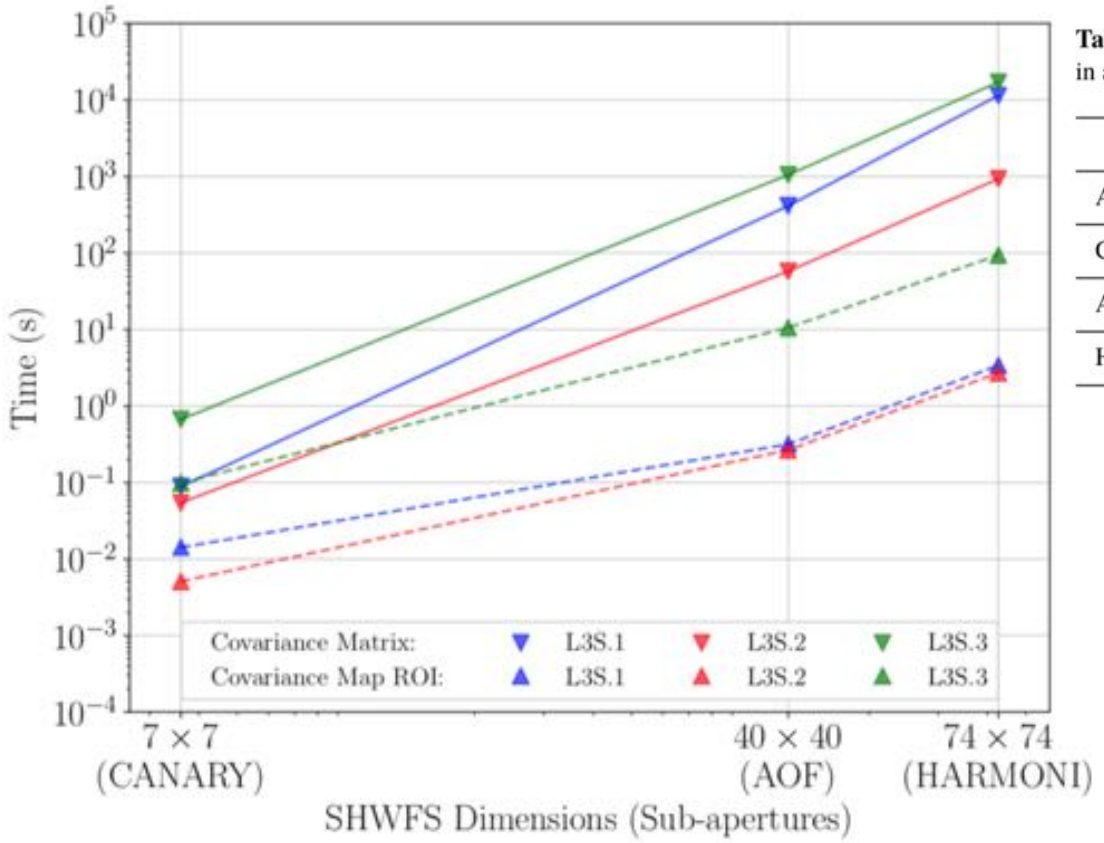
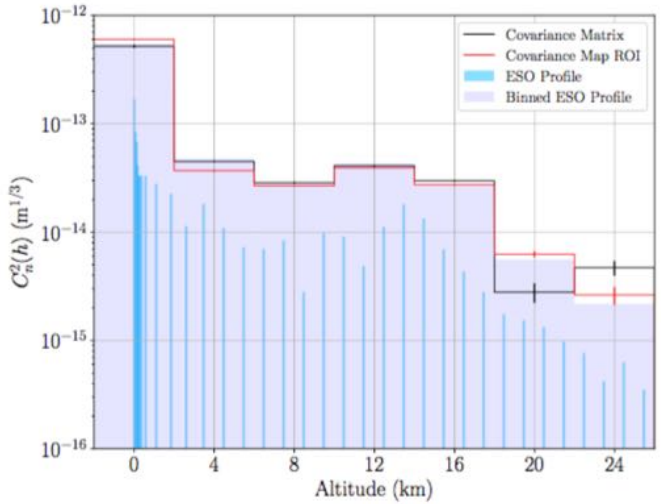


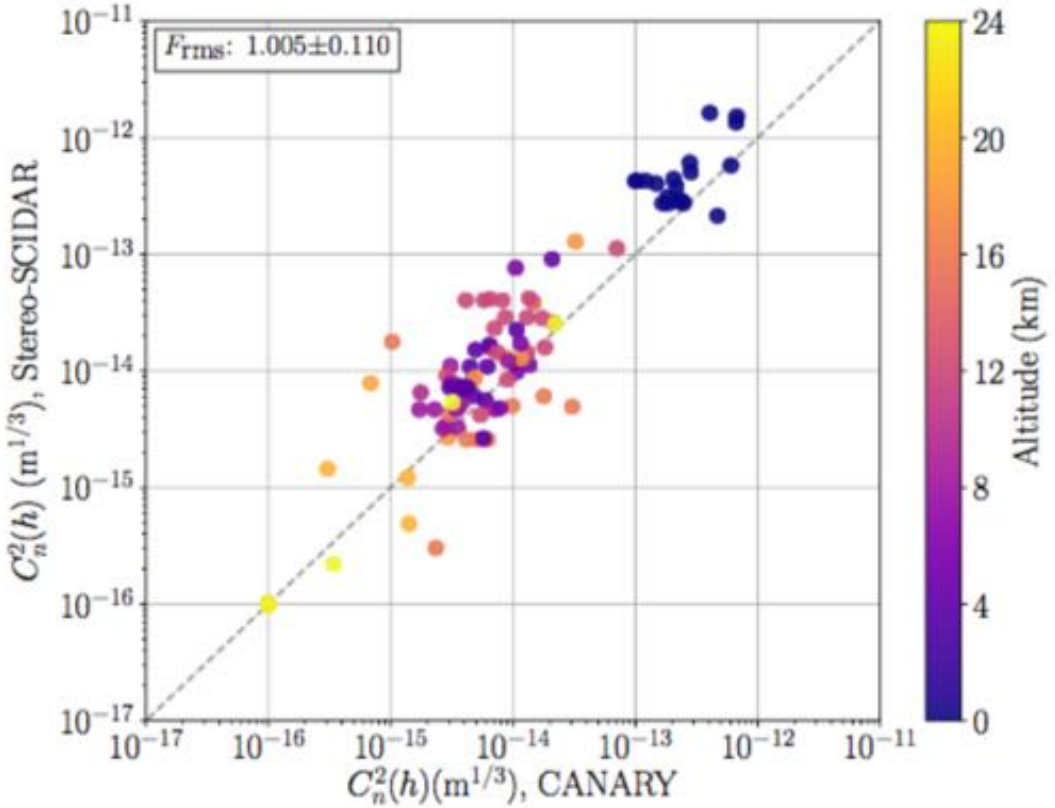
Table 1. Total L3S fitting time for covariance matrix and map ROI algorithms in a 2-NGS system where $N_L = 7$.

	Time Taken to Complete L3S (s)	
AO System	Covariance Matrix	Covariance Map ROI
CANARY	0.81 ± 0.01	0.12 ± 0.00
AOF	$1.52 \pm 0.00 \times 10^3$ (42 mins)	11.16 ± 0.02
HARMONI	$2.94 \pm 0.00 \times 10^4$ (8.17 hours)	99.71 ± 0.58

Parameter Fitting

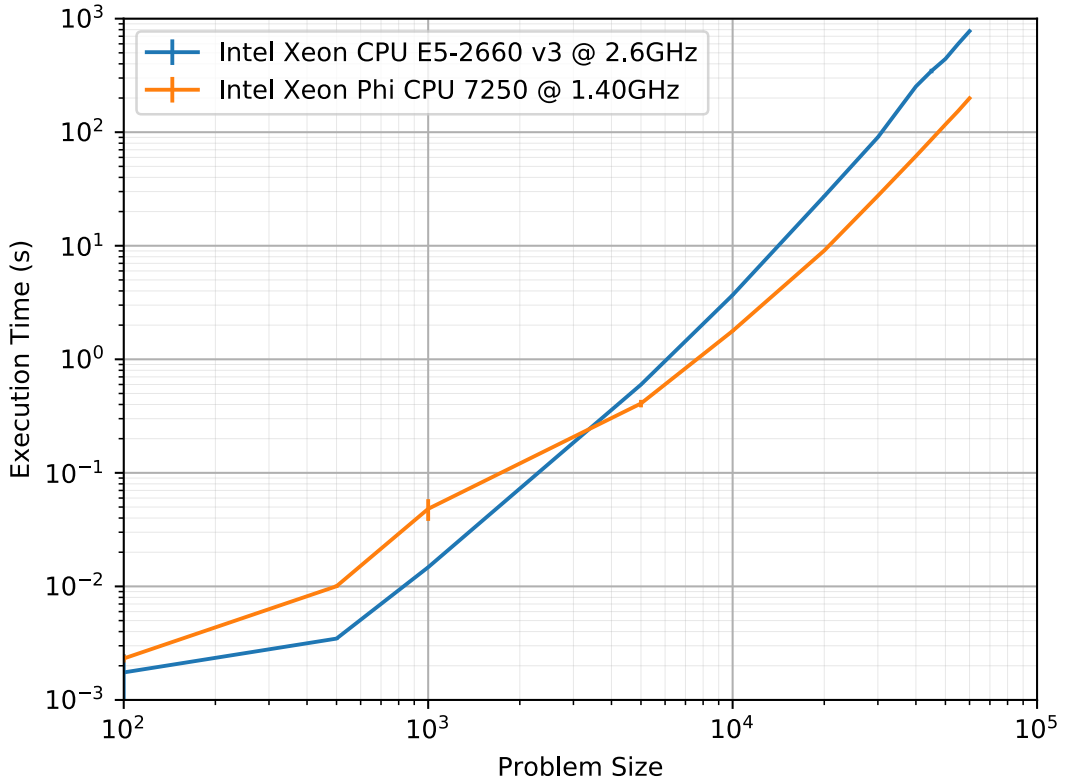
- Profile can be recovered equally well using all methods
  - Significantly less computational effort to compute using covariance maps





Ongoing Work

- Generate final reconstructor
- Generate Covariance Matrix
- Inverting Covariance Matrix
  - ~150s for 60kx60k matrix



END


l'Observatoire
de Paris

PLDA

MICROGATE

Green
Flash

Durham
University