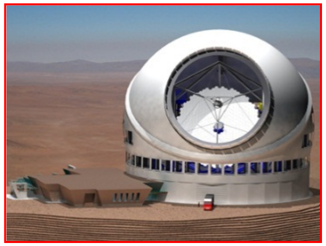


Studying GPU based RTC for TMT NFIRAOS

Lianqi Wang

Thirty Meter Telescope Project

RTC Workshop

Dec 04, 2012

Outline

- Tomography with iterative algorithms on GPUs
- Matrix vector multiply approach
  - Assembling AO control matrix
  - Applying matrix vector multiply
- GPU based RTC
- Benchmarking results
- Conclusion

Minimum Variance Reconstructor

- Minimizing σ^2 over target FoV (9 directions in Φ30")

$$\sigma^2 = \left\langle \left\| H_x x - H_a a \right\|^2 \right\rangle$$

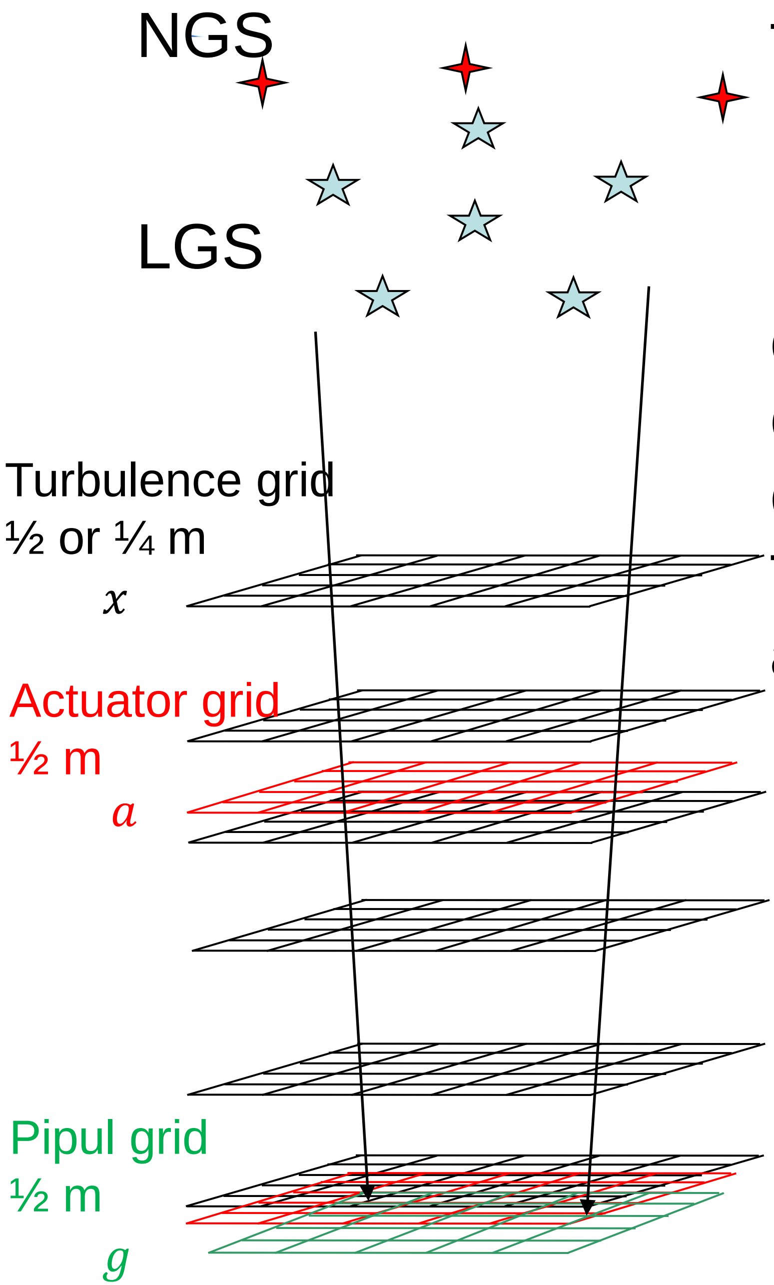
with $g = G_p H_x x + n$

- Gives tomography

$$\widehat{x} = \left(H_x^T G_p^T C_{nn}^{-1} G_p H_x + C_{xx}^{-1} \right)^{-1} H_x^T G_p^T C_{nn}^{-1} g$$

- And DM fitting over target FoV

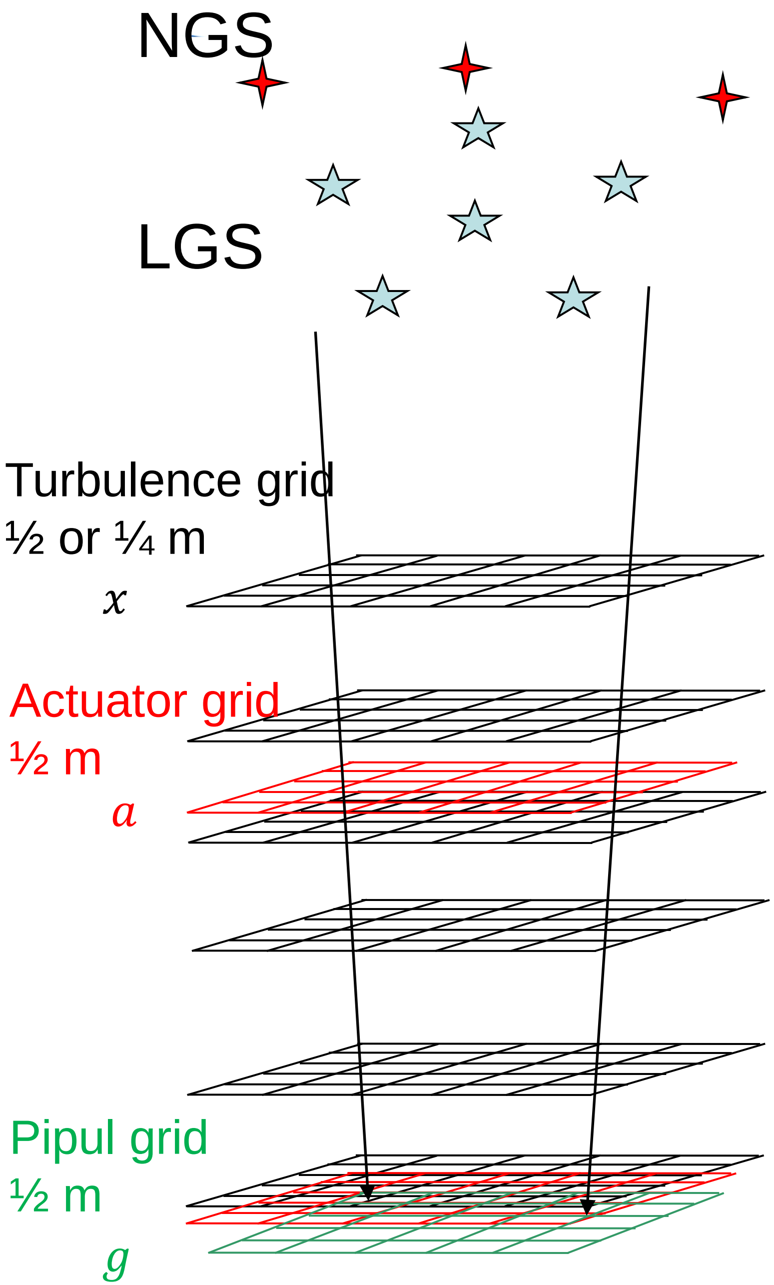
$$a = \left(H_a^T W H_a \right)^{-1} H_a^T W \widetilde{H}_x \widehat{x}$$



Tomography

$$x = (H_x^T G_p^T C_{nn}^{-1} G_p H_x + C_{xx}^{-1})^{-1} H_x^T G_p^T C_{nn}^{-1} g$$

H_x: ray tracing from x to p
G_p: compute gradient from p
C_{nn}: Noise covariance matrix
C_{xx}^{-1}: Using bi-harmonic approximation
The inverse is solved using iterative algorithms like Conjugate Gradients



DM Fitting

$$a = (H_a^T W H_a)^{-1} H_a^T W \tilde{H}_x x$$

Use sparse matrix based operation for the moment.

Benchmarking

- Hardware
  - Single Core i7 3820 @ 3.60 GHz
  - 2 NVIDIA GTX 580 GPU board
    - 3 GB graphics memory with 192GB/s theoretical throughput
    - 512 stream processors with 1.6TFlops theoretical throughput
- Software
  - 64 bit Linux
  - CUDA 4.0 C runtime library with nvcc
  - cublas, cuFFT, cuSparse, cuRand, etc from CUDA package
  - Use single precision floating number

Benchmarking Results of Iterative Algorithms for Tomography

	Timing (ms)	Incr WFE (nm)
CG30OS0	5.17	44.3
CG30OS4	18.20	0
CG30OS6	12.3	11.2
FD1OS0	0.49	52.8
FD1OS6	1.37	33.8
FD2OS0	0.78	42.9
FD2OS6	2.60	-16.9
FD3OS0	1.04	42.6
FD3OS6	3.04	-19.7

CG: Conjugate Gradients

FD: Fourier Domain Preconditioned CG.

OSn: Over sampling n tomography layers (¼ m spacing)

Tomography Detailed Timing

$$x = (H_x^T G_p^T C_{nn}^{-1} G_p H_x + C_{xx}^{-1})^{-1} H_x^T G_p^T C_{nn}^{-1} g$$

Tomo	micro-sec	Flop	Mem	GB/s	GFlops
H_x	74	10616832	15925248	215	143
G_p	45	278856	1921008	43	6
G_P^T	50	402792	2106912	42	8
H_x'	122	10616832	15925248	131	87
C_{xx}^{-1}	48	626688	2064384	43	13
Total	339				

Preconditioner: $Mx = \boldsymbol{\mathcal{F}}^{-1}[A\boldsymbol{\mathcal{F}}[x]]$ where A is block diagonal matrix

FDPCG	micro-sec	Flop	Mem	GB/s	GFlops
$\boldsymbol{\mathcal{F}}$	115	79,531,761	1769472	15	692
A	188	5,308,416	10616832	56	28
$\boldsymbol{\mathcal{F}}^{-1}$	114	79,531,761	1769472	16	698

Total Timing

- DM Fitting uses sparse matrix approach. Haven’t yet optimized. Potential to speed up by a few times

micro-sec	LHS	RHS	Total
Tomography (2 Iterations)	2016	584	2600
DM Fitting (4 iterations)	1641	2862	4503

What limits our performance?

- We are not limited by the steady rate throughput
  - 1581 GFlops of single precision floating point number operation
  - 192 GB/s device memory
- We are limited by latency
  - Kernel launch overhead:
    - ~2.3 micro-second for asynchronous launch,
    - ~6.5 micro-second for synchronization
  - Device memory latency: 600 cycles, ~0.3 micro-second, for intermediate quantities.
    - Sparse matrix vector multiply need to be carefully optimized
  - PCI-E interface (2.0): 8GB/s, 11 micro-second latency, for gradients and actuator commands input/output

Matrix Vector Multiply (MVM)

- Still a long way to go with iterative algorithms for <1.25 ms latency
  - Hard to parallel across GPUs due to low PCIe bandwidth and high latency
- MVM is the easiest to implement in parallel
  - Regular memory access pattern avoids memory latency issue
  - GPU is good at it with ~200 GB/s device memory bandwidth
- Need to obtain the control matrix
  - Update the control matrix every 10 seconds
- Solution: Using iterative algorithms to solve columns of I
  - Update the control matrix with warm restart

Assembling the control matrix in GPUs

- Tomography + fitting can be summarized as $E = F_L^{-1} F_R R_L^{-1} R_R$

With

$$R_L = H_x^T G_p^T C_{nn}^{-1} G_p H_x + C_{xx}^{-1};\quad R_R = H_x^T G_p^T C_{nn}^{-1}\qquad \text{Tomography}$$

$$F_L = H_a^T W H_a;\quad F_R = H_a^T W \widetilde{H}_x\qquad \text{DM Fitting}$$

- Matrix dimensions are

$$(7083 \times 30984) = (7083 \times 7083)^{-1} (7083 \times 62311) \times (62311 \times 62311)^{-1} (62311 \times \mathbf{30984})$$

7083: number of active actuators

30984: number of WFS gradients

62311: number of points in tomography grid

- We assemble E by solving each column one at a time

$$E(:,j) = F_L^{-1} F_R R_L^{-1} R_R e_j$$

- There are **30984 tomography operations** total
  - 1500 seconds to create (FDPCG with 50 iterations)
  - 150 seconds to update (when condition changes. 5 iterations, using warm restart)

Assembling the transpose of control matrix in GPUs

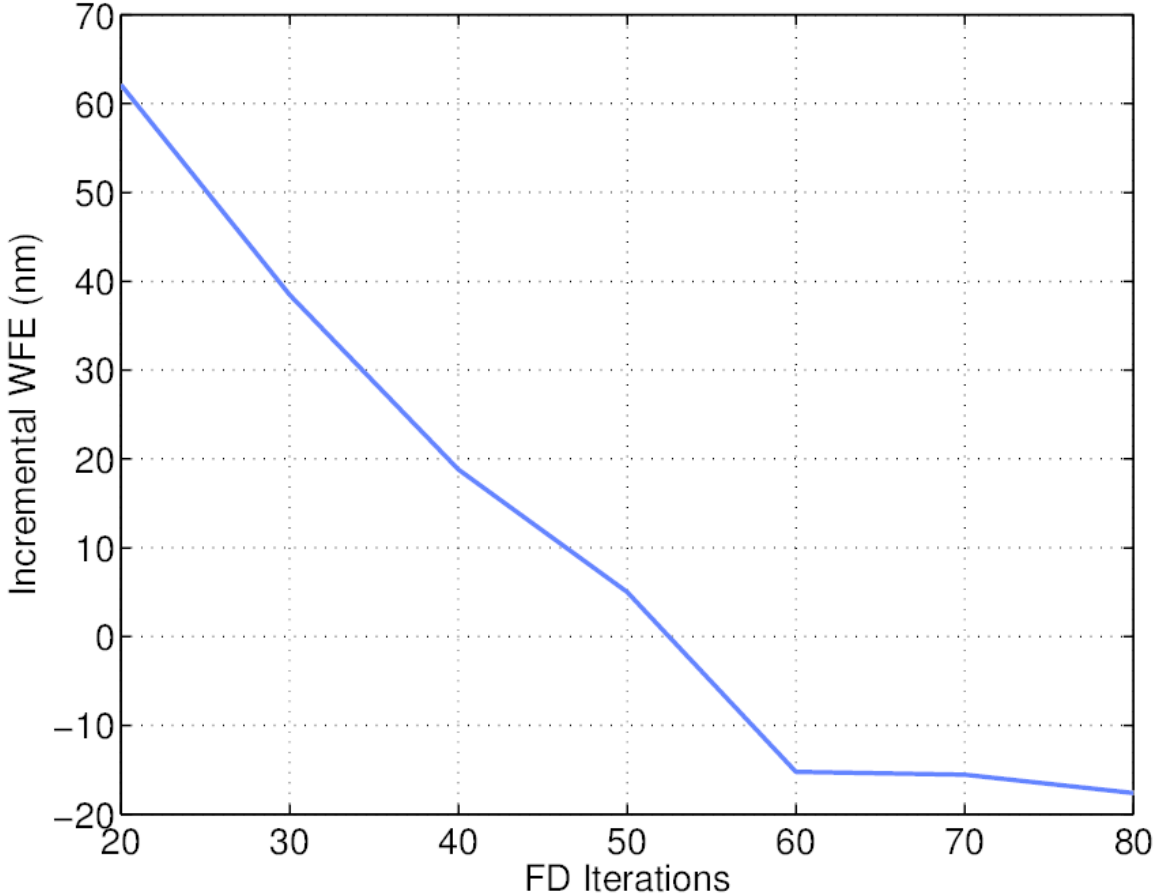
- Solve for the transpose $E^T = R_R^T R_L^{-1} F_R^T F_L^{-1}$
- The dimensions are

$$(30984 \times 7083) = (30984 \times 62311)\,(62311 \times 62311)^{-1} \times (62311 \times 7083)(7083 \times \mathbf{7083})^{-1}$$

- A factor of 4 reduction in number of tomography operations compared to solve E directly
  - F_L^{-1} can be reused
  - 400 seconds to create (50 FD iterations. 2.2ms each step)
  - 40 seconds to update (5 FD iterations)
- With a 8 GPU machine
  - 50 seconds to create (can be avoided by warm warm restart)
  - 5 seconds to update (5 FD iterations, using warm restart)
  - 10 seconds for 10 FD iterations when condition varies significantly
  - NFIRAOS requirement is 10 seconds.

What about Closed Loop Performance?

- RMS wavefront error in science FoV is comparable to baseline algorithm (CG30) with 50 FDPCG iterations (OS6)



GPUs required to apply MVM at 800 Hz for NFIRAOS

Assuming 1.00 ms total time

NGPU	Compute MFLOP	Memory MB	PCI-E kB	Compute GFLOPS	Device Mem GB/s	PCI-E MB/s
1	209	837	149	409	818	145
6	35	140	48	82	136	47
8	26	105	43	51	102	42
Rating		3G		1581	192	8192

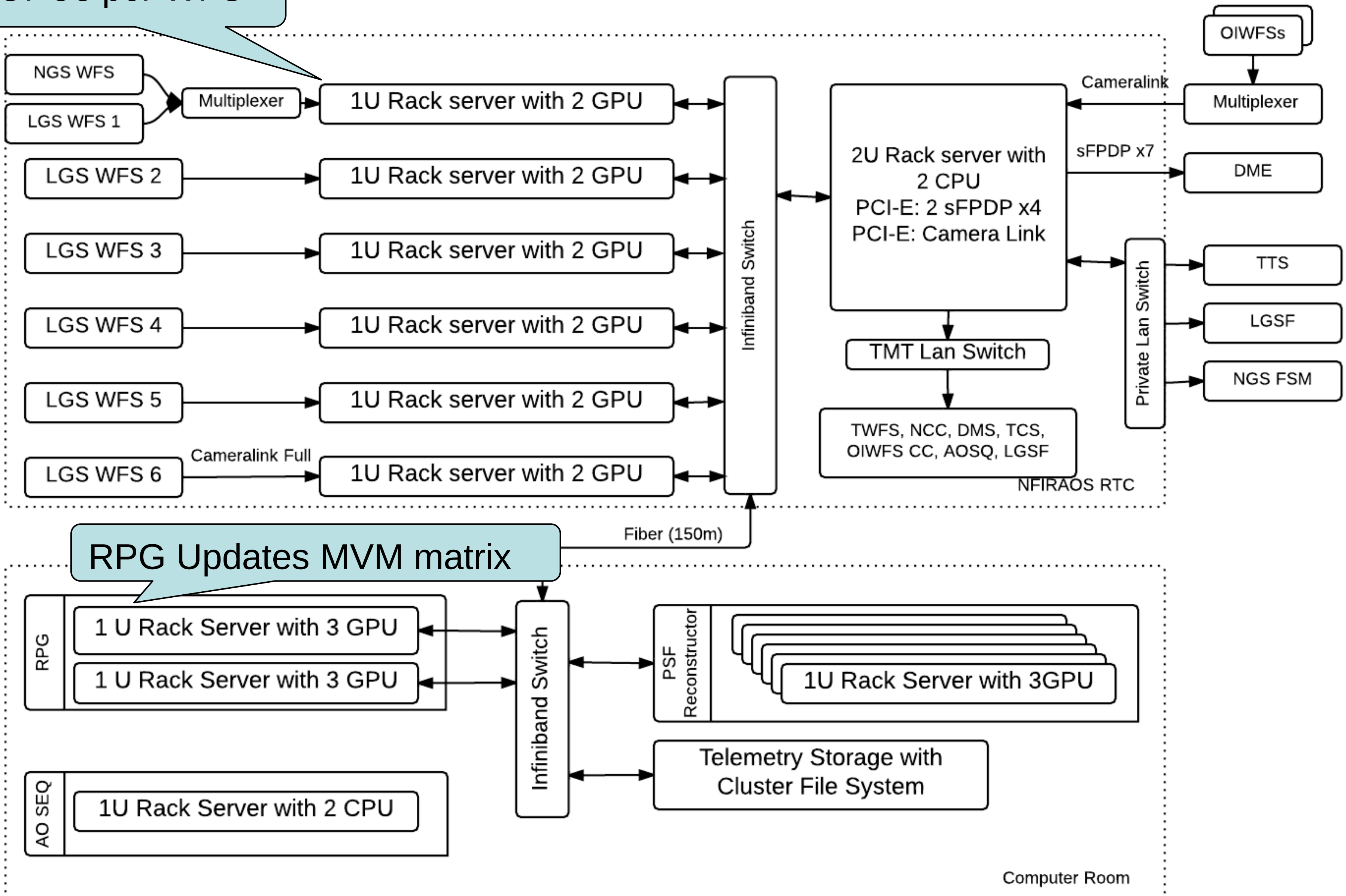
Red	No achievable
Yellow	Nearly achievable
Green	Achievable

A minimum of 6-8 GTX 580 GPU is needed to apply MVM

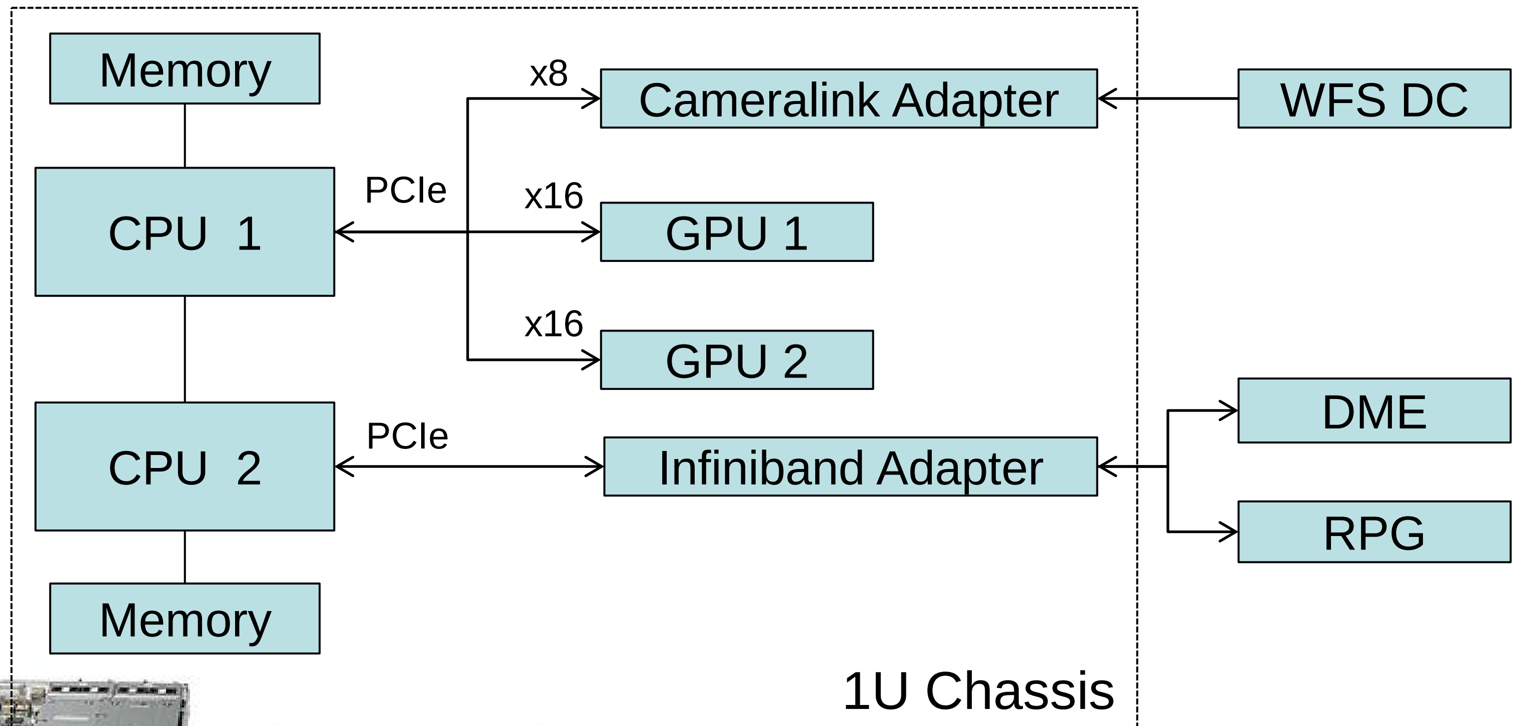
Proposed GPU RTC Architecture

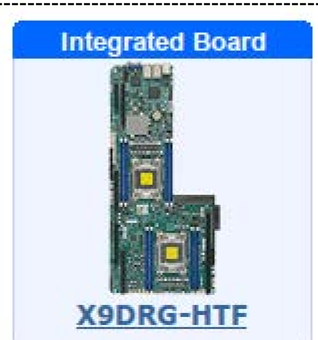
2 GPUs per WFS

RPG Updates MVM matrix



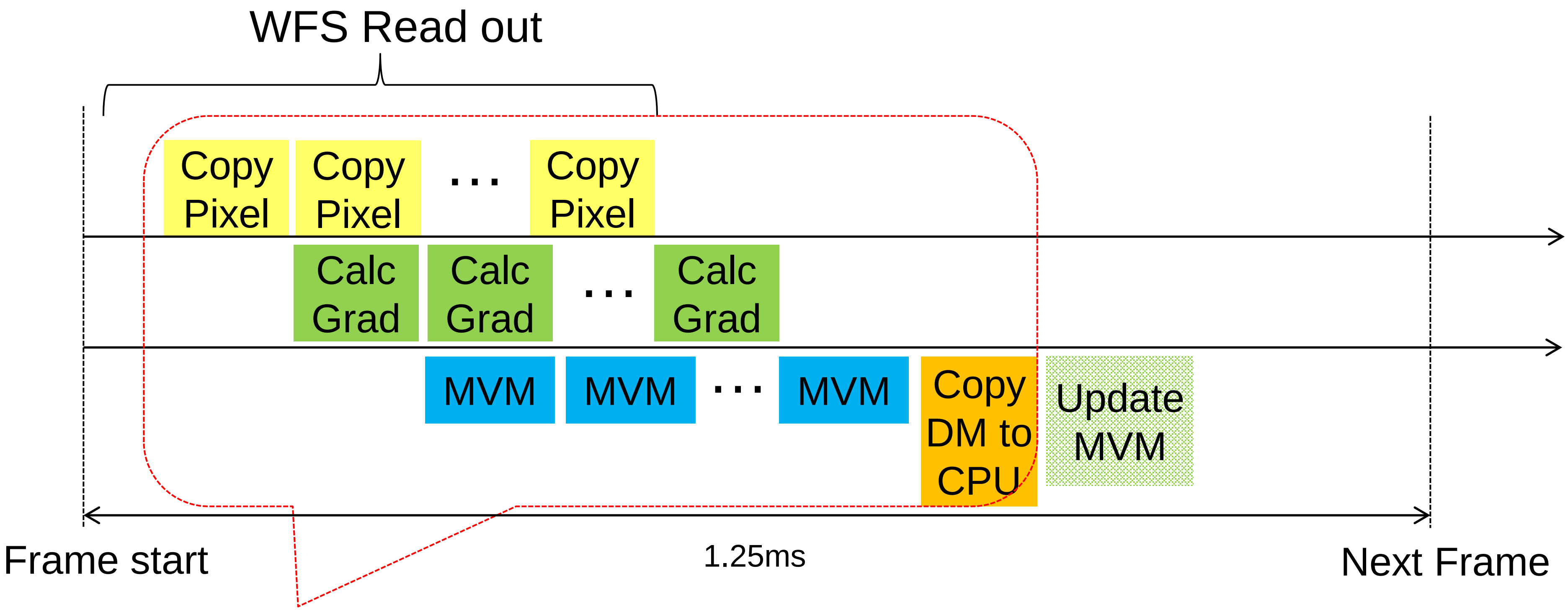
1U Server for Each LGS WFS





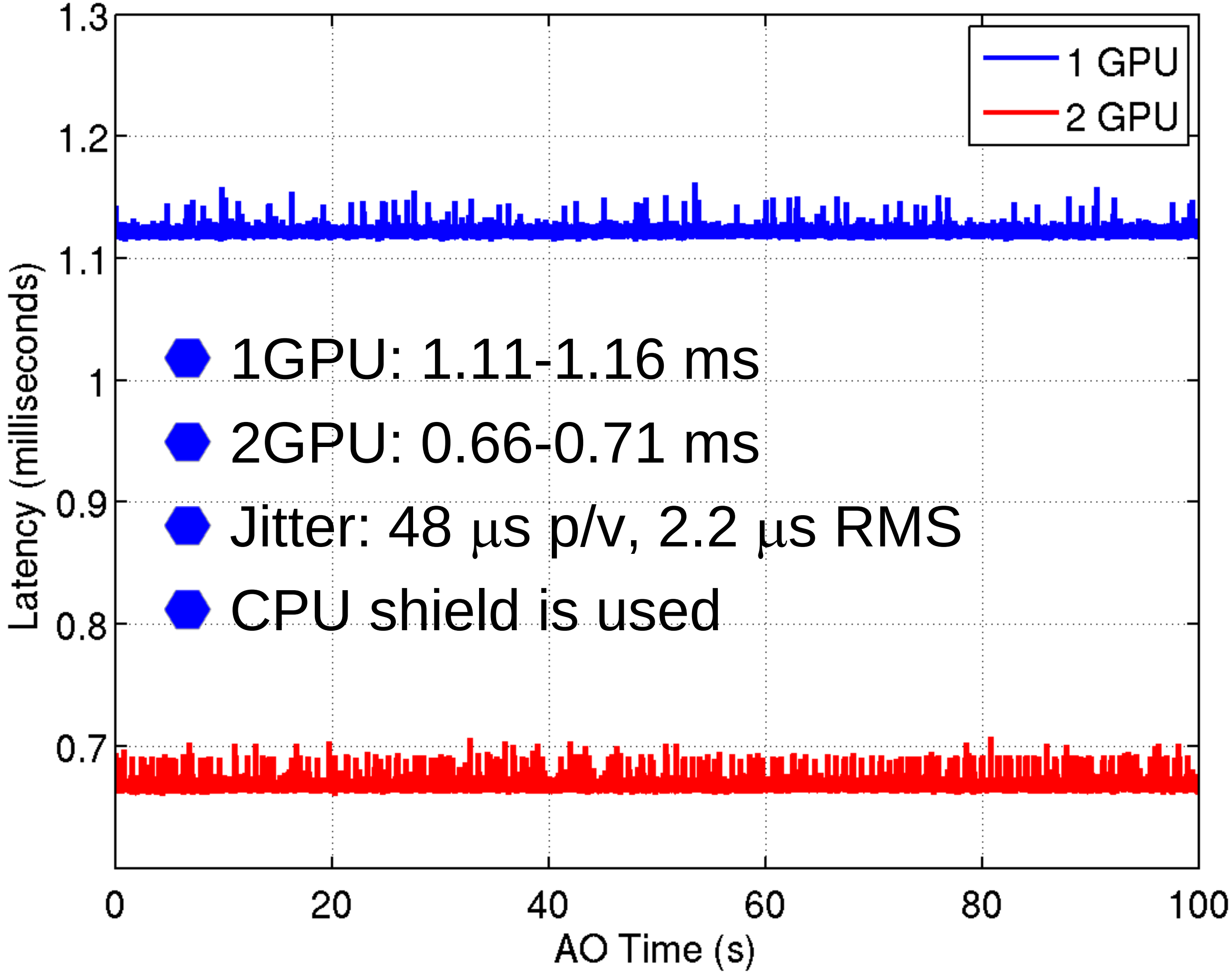
- Infiniband for control matrix and telemetry
- Cameralink (or else?) for WFS pixel data

Pipelining in GPU using 3 streams

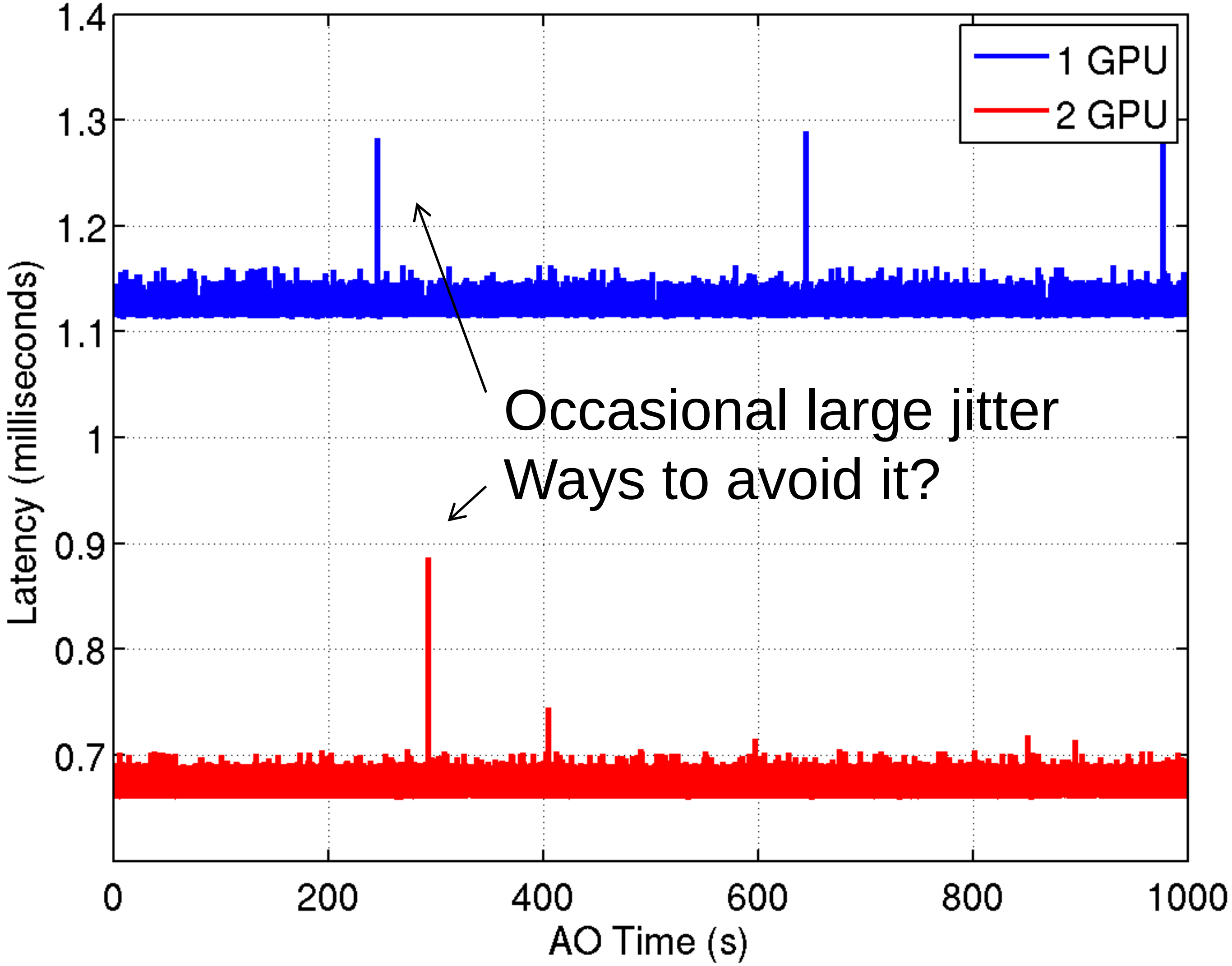


- Benchmarked for an LGS WFS with 2 GTX 580
  - MVM takes most of the time.
  - Memory copying is indeed concurrent with computing

End to End Latency



For 1000 seconds



Other real time tasks

- Copying updated MVM matrix to RTC
  - Do so after DM actuator commands are ready
  - Measured 0.1 ms for 10 columns
  - 519 time steps to copy 5182 columns
- Collect statistics to update matched filter coefficients
  - Do so after DM actuator commands are ready
  - Benchmark next
- Etc
- 0.5 ms to spare

Background process

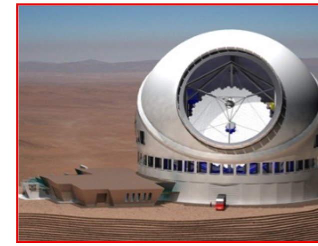
- Updating MVM matrix when condition varies
  - Role of reconstruction parameter generator (RPG).
  - Copy to RTC over Infiniband or ethernet
- etc

Conclusions

- Current gen GPU can handle iterative wavefront reconstruction algorithms in a few ms.
- Control matrix for MVM can be updated every 10 seconds using FDPCG tomography algorithm to cope with varying conditions
- With MVM, A 2 GPU server per LGS WFS can turn pixels into DM actuator commands in 0.7ms, meeting the requirement with good margin

- Any other concerns?

Acknowledgements



- The author gratefully acknowledges the support of the TMT collaborating institutions. They are
  - the Association of Canadian Universities for Research in Astronomy (ACURA),
  - the California Institute of Technology,
  - the University of California,
  - the National Astronomical Observatory of Japan,
  - the National Astronomical Observatories of China and their consortium partners,
  - and the Department of Science and Technology of India and their supported institutes.
- This work was supported as well by
  - the Gordon and Betty Moore Foundation,
  - the Canada Foundation for Innovation,
  - the Ontario Ministry of Research and Innovation,
  - the National Research Council of Canada,
  - the Natural Sciences and Engineering Research Council of Canada,
  - the British Columbia Knowledge Development Fund,
  - the Association of Universities for Research in Astronomy (AURA)
  - and the U.S. National Science Foundation.