A GPU based brute force de-dispersion algorithm for LOFAR

W. Armour, M. Giles, A. Karastergiou and C. Williams.
University of Oxford.

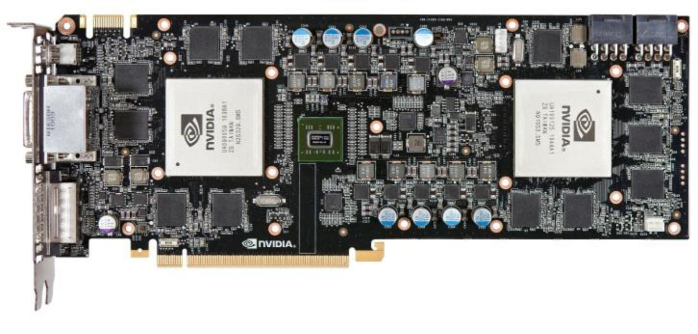
8th May 2012

GPUs

Why use GPUs ?

Latest Kepler/Fermi based cards

- GeForce GTX 690 – 2x GK104 GPUs
- 18.7 GFLOPS / watt
- Peak performance ~6 TFLOPS
- Price point ~ € 1K



- M2090 – 1x GF110 GPU 512 cores.
- Peak performance 1.3 TFLOPS
- 6.8 GFLOPS / watt
- Price point ~ € 3K

Influence and Take-up

TOP 500 – 3 out of top 5 utilise Fermi/Tesla

- Tianhe-1A 2.5 petaflops
- Based on 14336 Xeon and 7168 M2050
- To achieve same performance using only CPUs 50000 CPUs, 2x floor space and 3x power (Estimates made by NVIDIA)
- Uses Lustre :-s

OXFORD e-Research CENTRE

The MOTIVATE project…

- MOTIVATE is a pathfinder project and aims to investigate the latest many-core technologies with the aim of delivering energy and cost efficiency in the area of radio astronomy HPC.

- MOTIVATE stands for Many-cOre Technology Investigating Value, Application, deploymenT and Efficiency.

- The MOTIVATE project is funded by the Oxford-Martin School through the Institute for the Future of Computing

OXFORD MARTIN SCHOOL

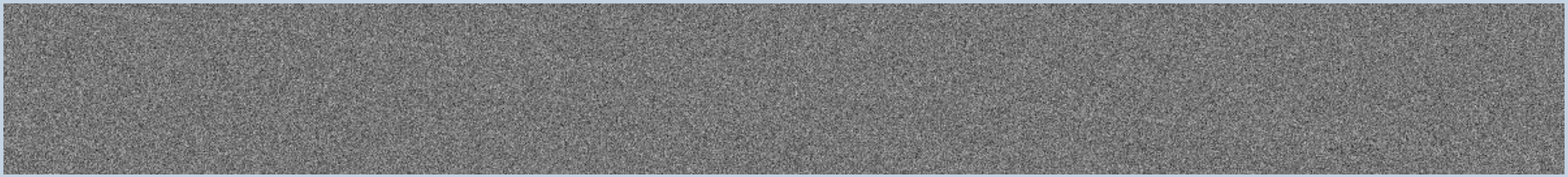
UNIVERSITY OF OXFORD

http://www.oerc.ox.ac.uk/research/many-core

De-dispersion

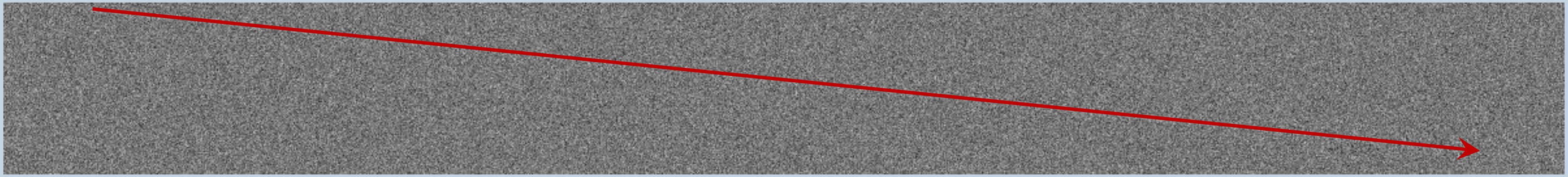
Experimental data

Most of the measured signals live in the noise of the apparatus.

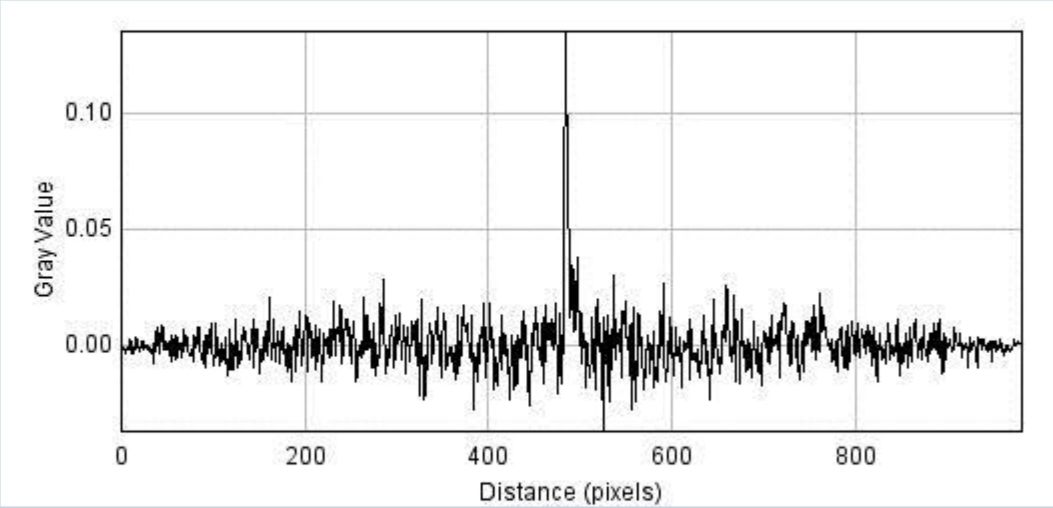


Experimental data

Most of the measured signals live in the noise of the apparatus.

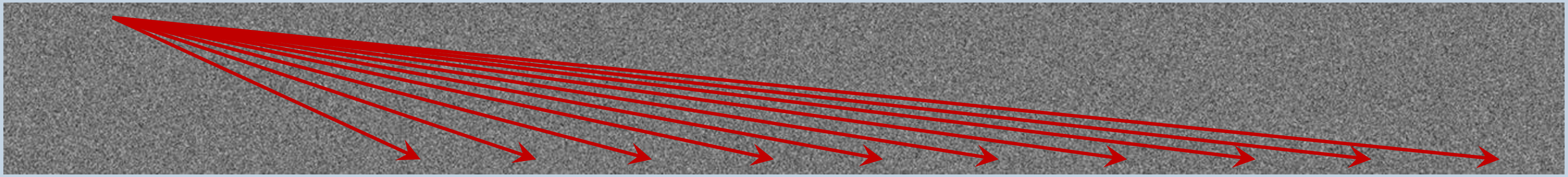


Hence frequency channels have to be “folded”



Brute force algorithm

Every DM is calculated to see if a signal is present.



- In a blind search for a signal many different dispersion measures are calculated.

- This results in many data points in the (f,t) domain being used multiple times for different dispersion searches.

- This allows for data reuse in a GPU algorithm.

OXFORD e-Research CENTRE

A new GPU brute force algorithm

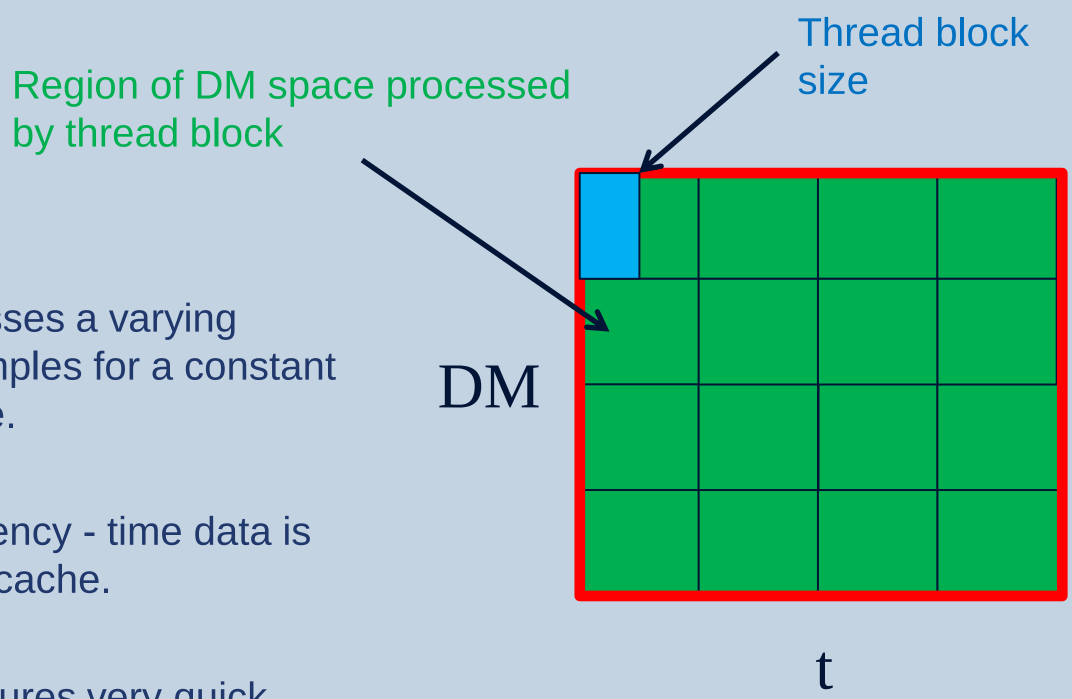
Three key features…

- Each thread processes a variable number of dispersion measures in local registers.
- Exploit the L1 Cache / Shared Memory present on the Fermi architecture.
- Optimise the region of dispersion space being processed (thread blocksize).

Web page : http://www.oerc.ox.ac.uk/research/wes

Processing several DM's per thread

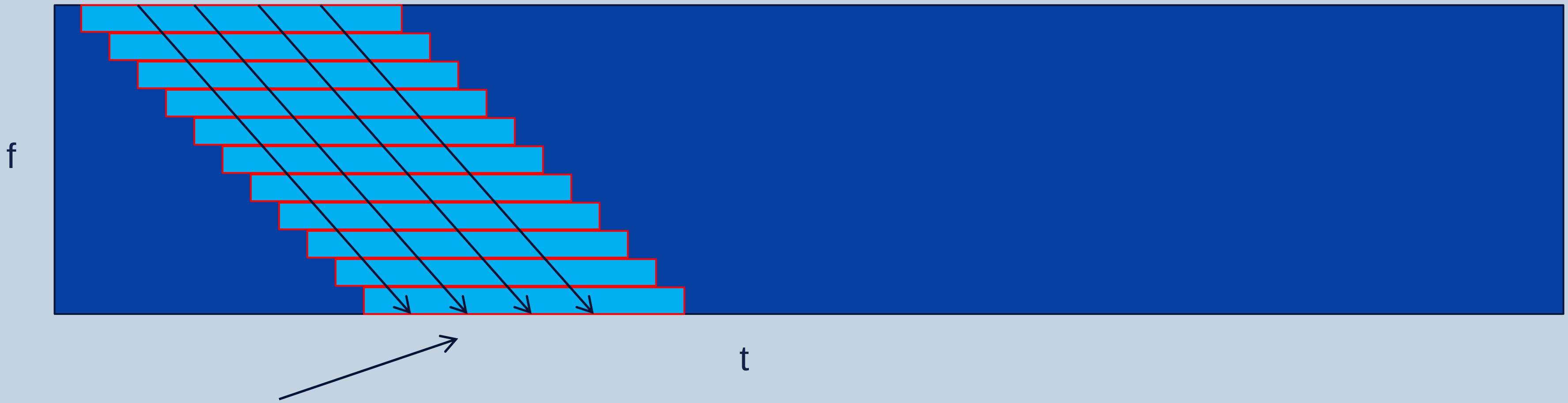
New Algorithm works in the DM - t space rather than frequency – time space.



- Each thread processes a varying number of time samples for a constant dispersion measure.
- This ensures frequency - time data is loaded into fast L1 cache.
- Using registers ensures very quick memory access.

Exploiting the L1 cache / Shared Memory…

Each dispersion measure for a given frequency channel needs a shifted time value.

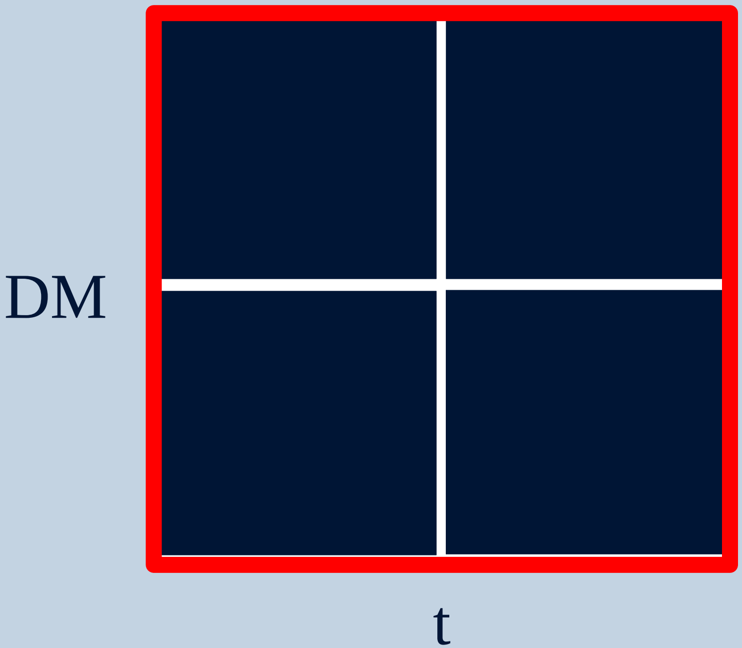


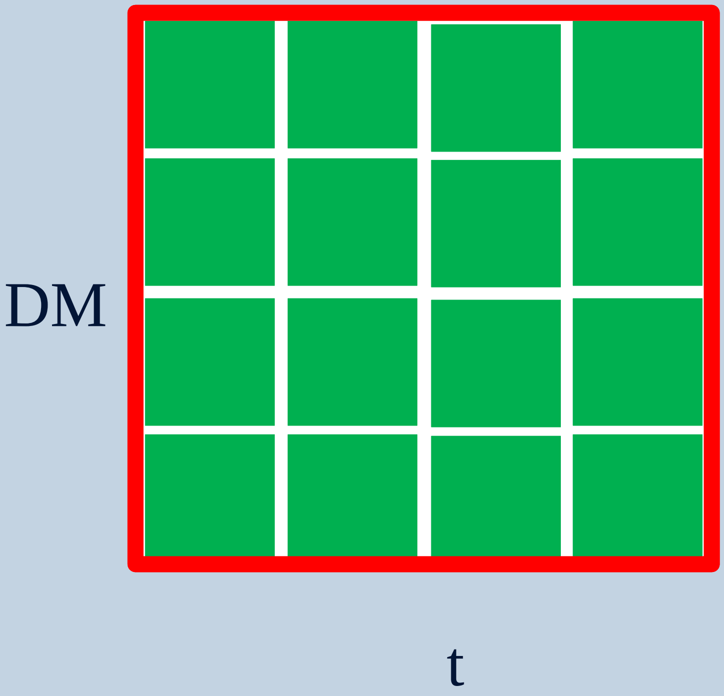
Constant DM's with varying time.

Incrementing all of the registers at every frequency step ensures a high data reuse of the stored frequency time data in the L1 cache.

Optimising the parameterisation.

The GPU block size of the new algorithm can take on any size that is integer multiples of the size of a “data chunk”…





A CPU brute force algorithm

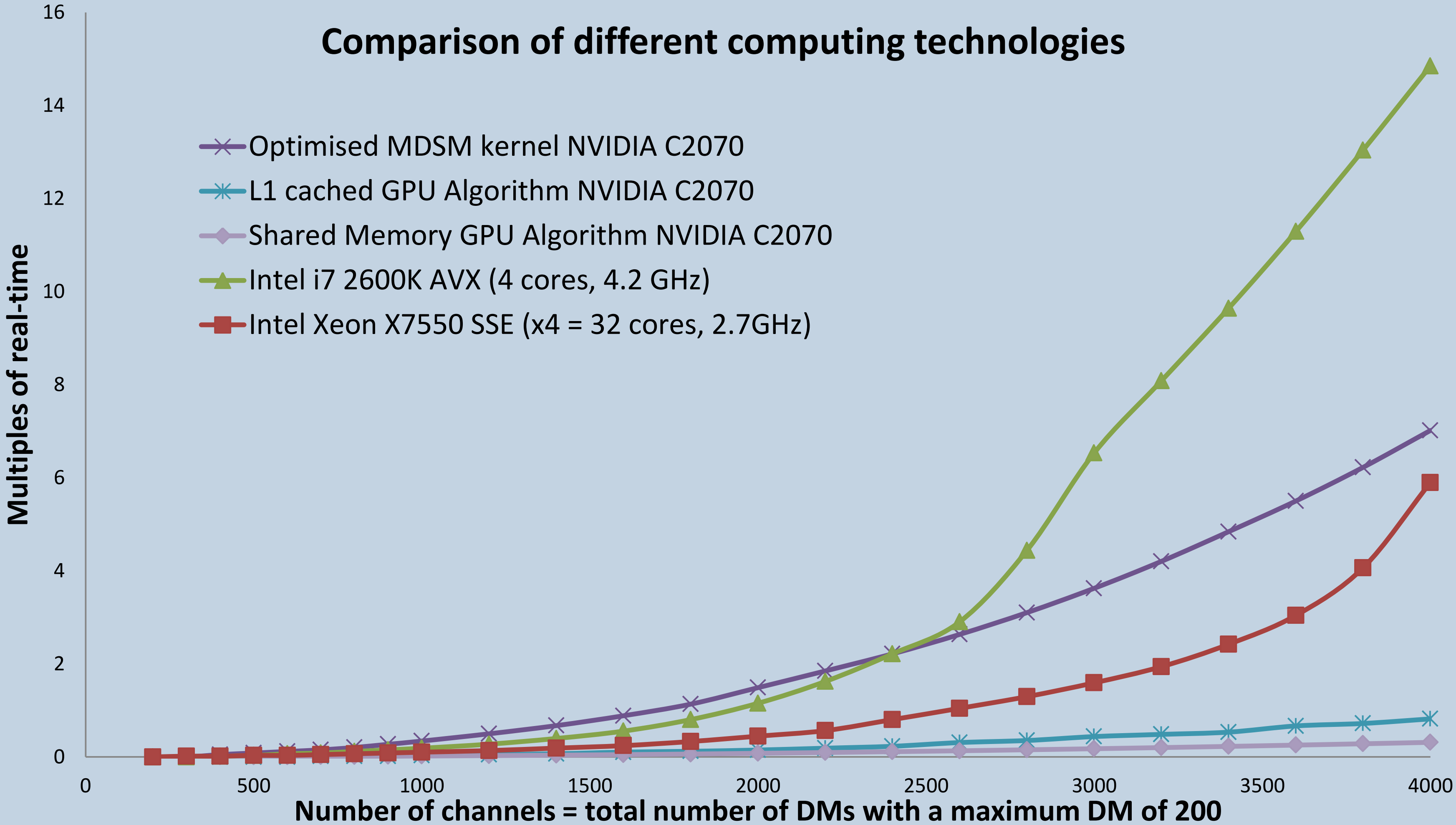
Four key features…

- Aim to achieve full cache line utilization.
- Exploit the large (~375 GB/s) LLC bandwidth present on the new Intel Sand Bridge CPUs.
- Use the Intel Intrinsics to exploit the 16 AVX/SSE (YMM/XMM) SIMD registers (don’t rely on the Intel auto-vectorizer!)
- Use OpenMP to share work across the CPU cores.

Brute force results…

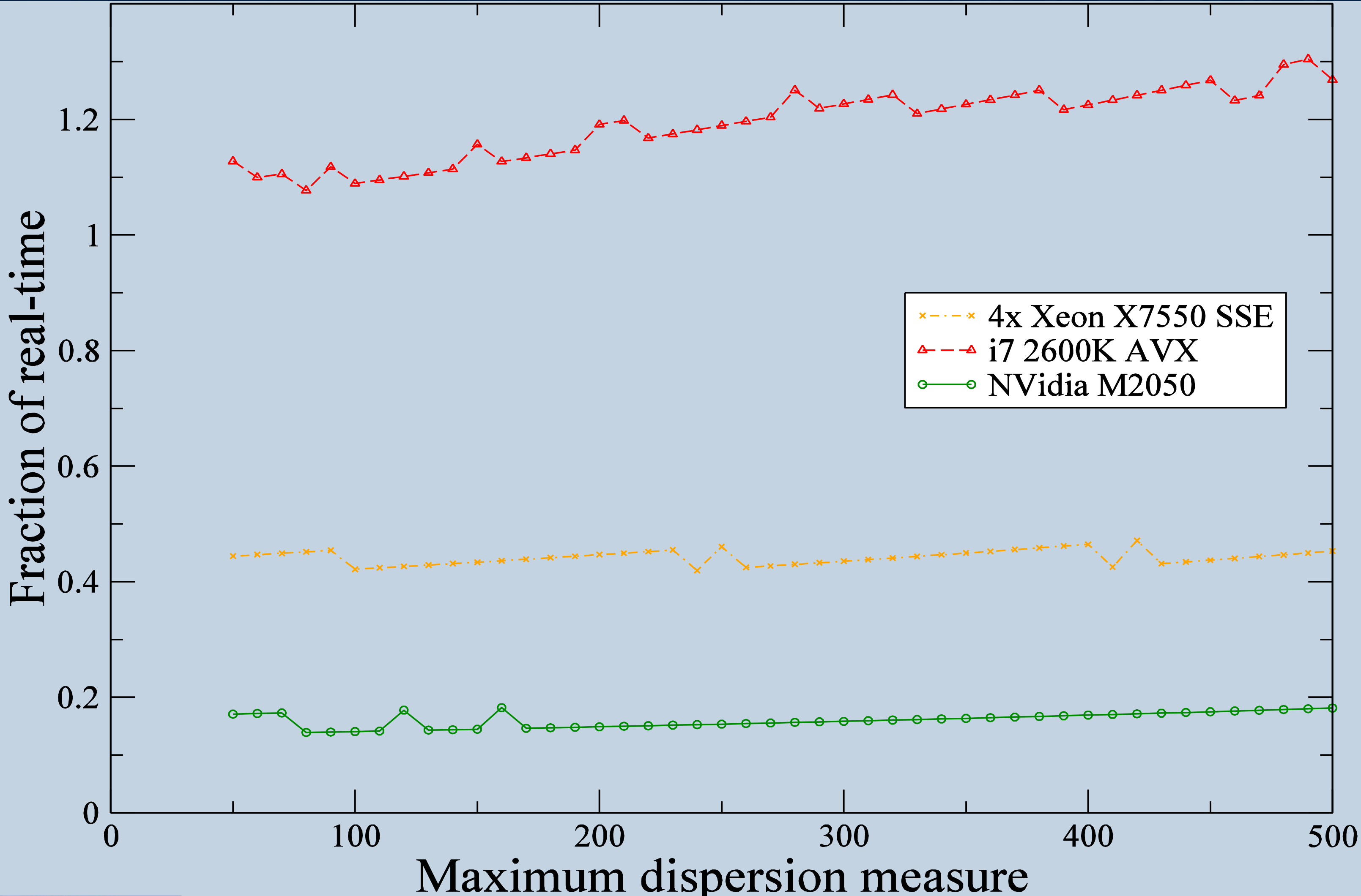
Results

Results…



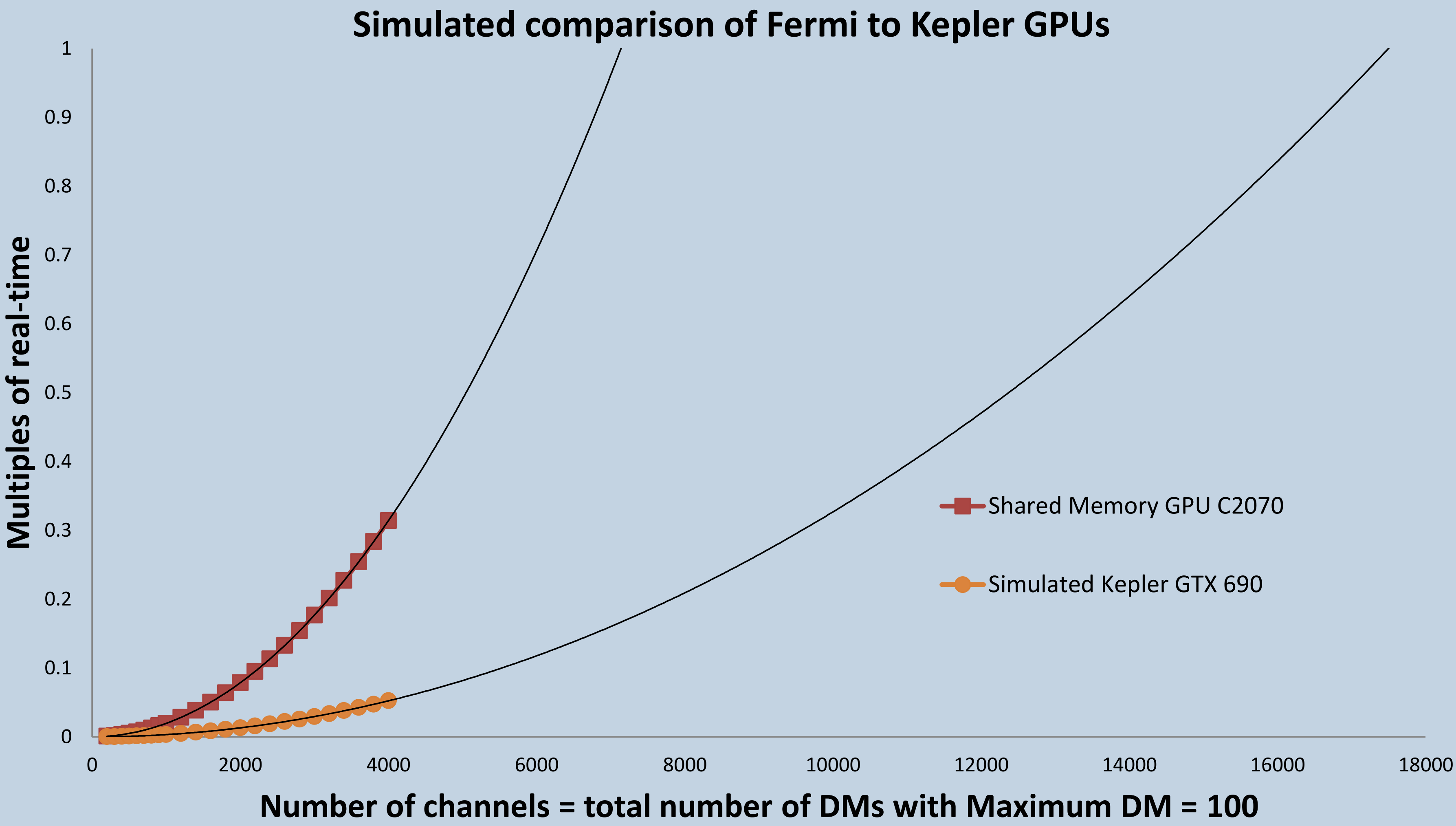
OXFORD e-Research CENTRE

Results…



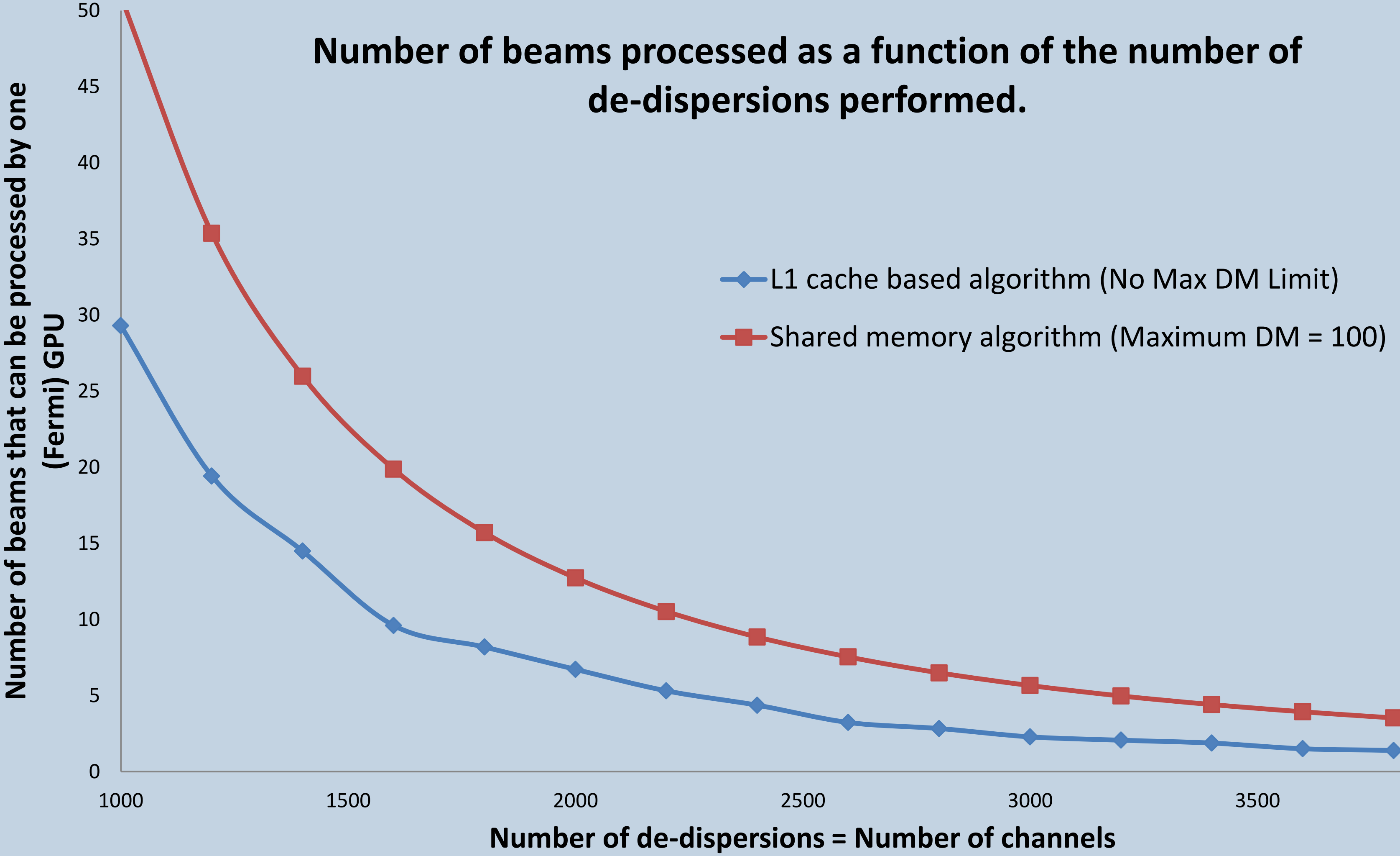
Fraction of real-time
0
0.2
0.4
0.6
0.8
1
1.2
0
100
200
300
400
500
Maximum dispersion measure
4x Xeon X7550 SSE
i7 2600K AVX
NVidia M2050

Results…



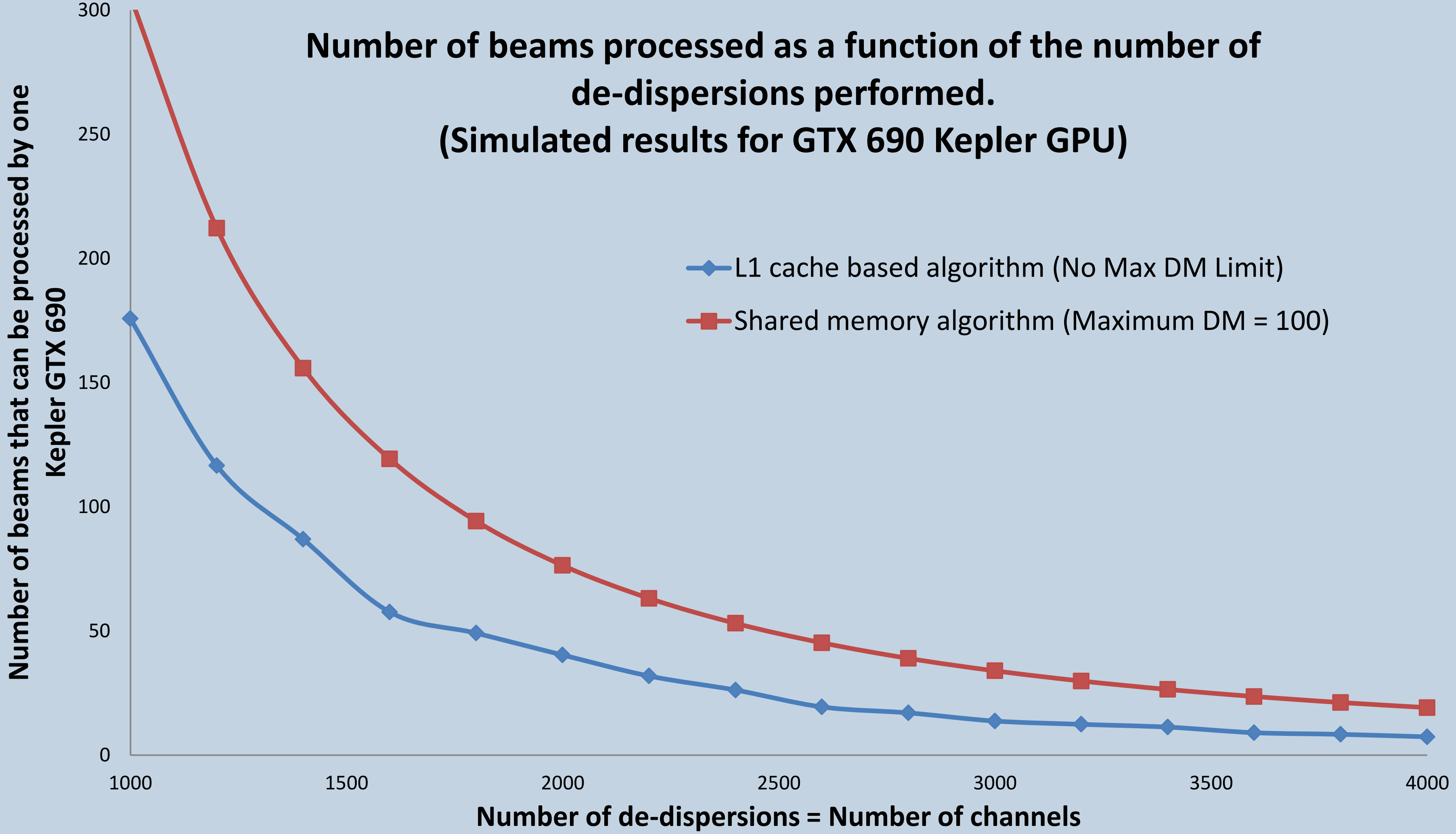
OXFORD e-Research CENTRE

Results…



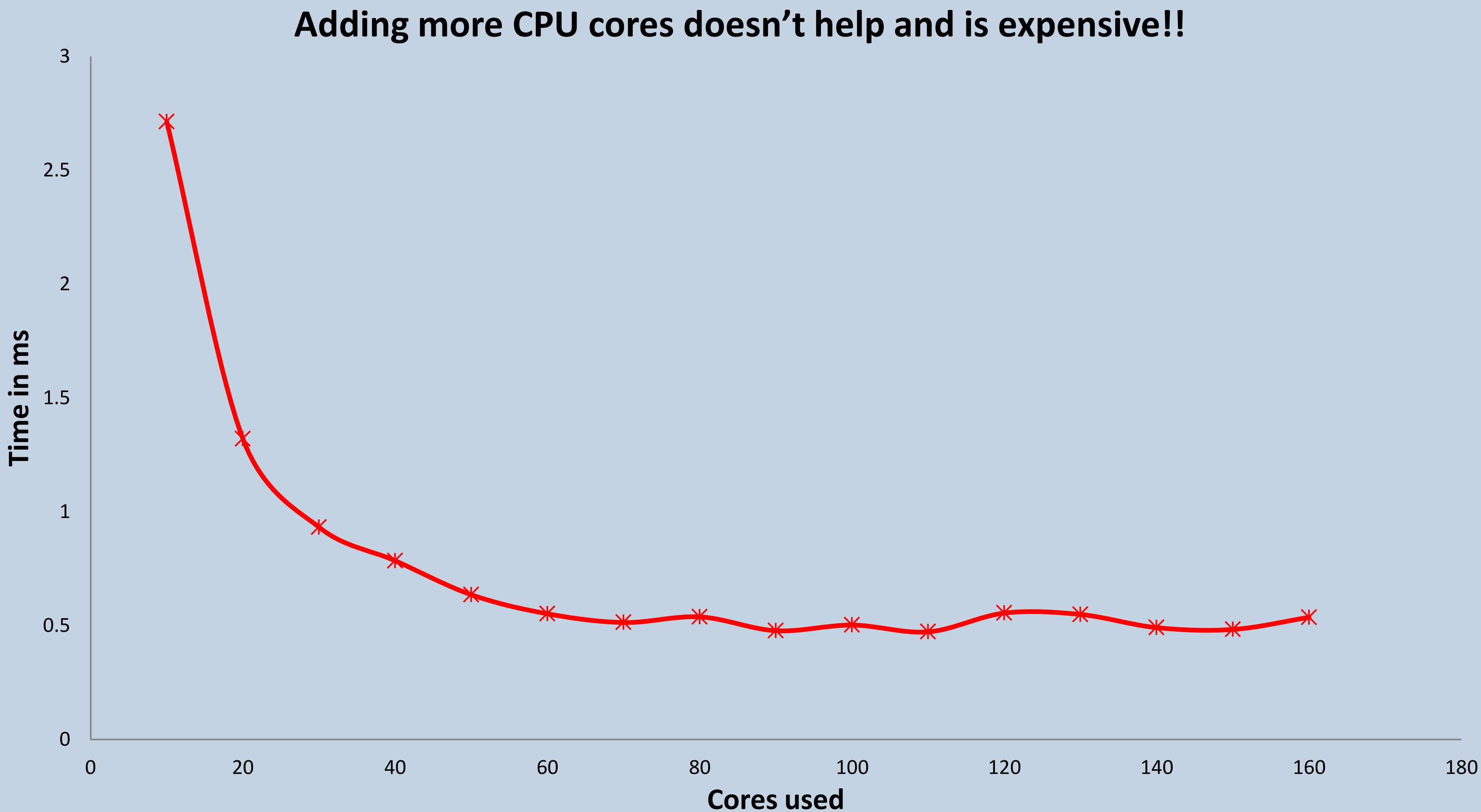
OXFORD e-Research CENTRE

Results…



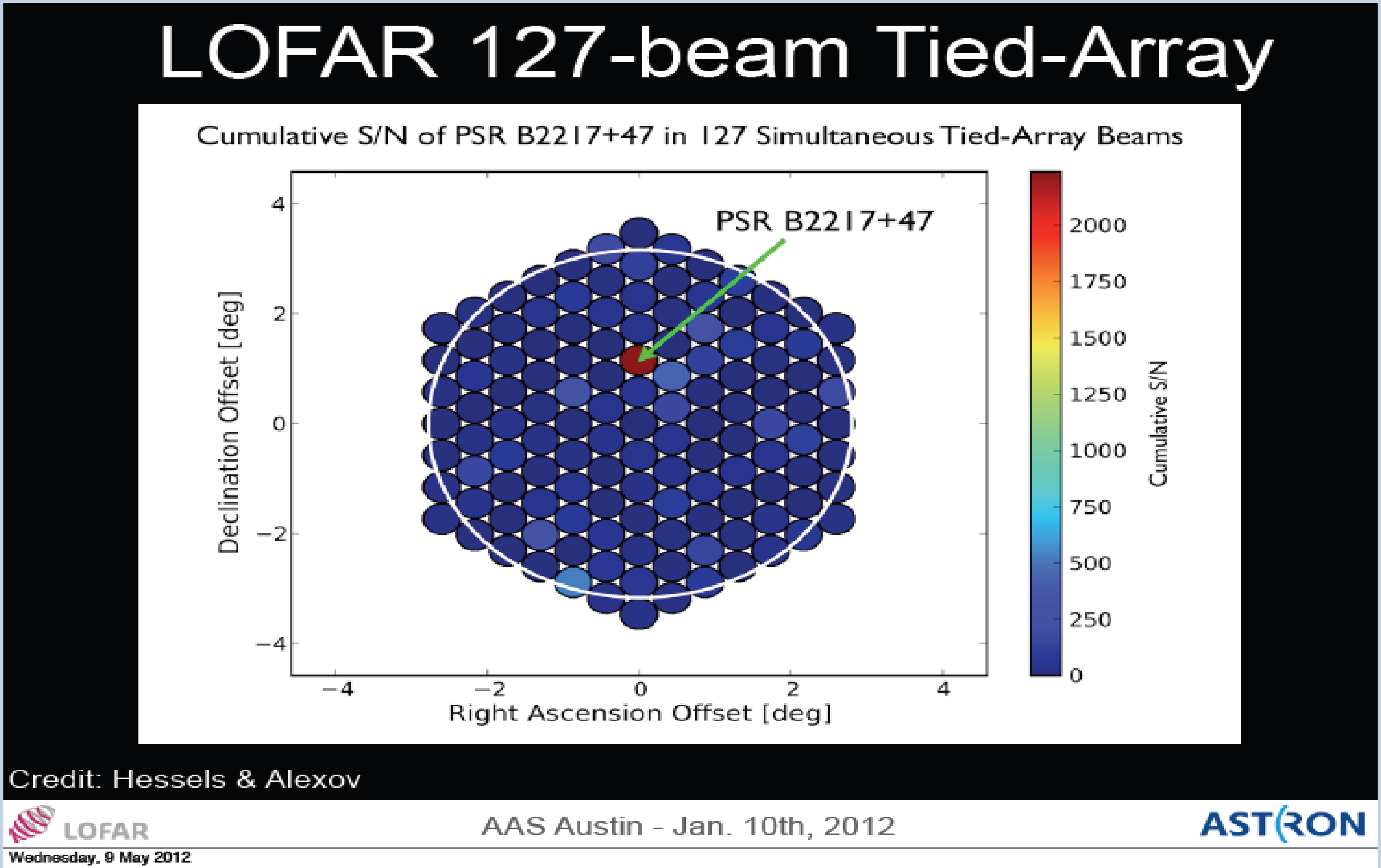
OXFORD e-Research CENTRE

Results…

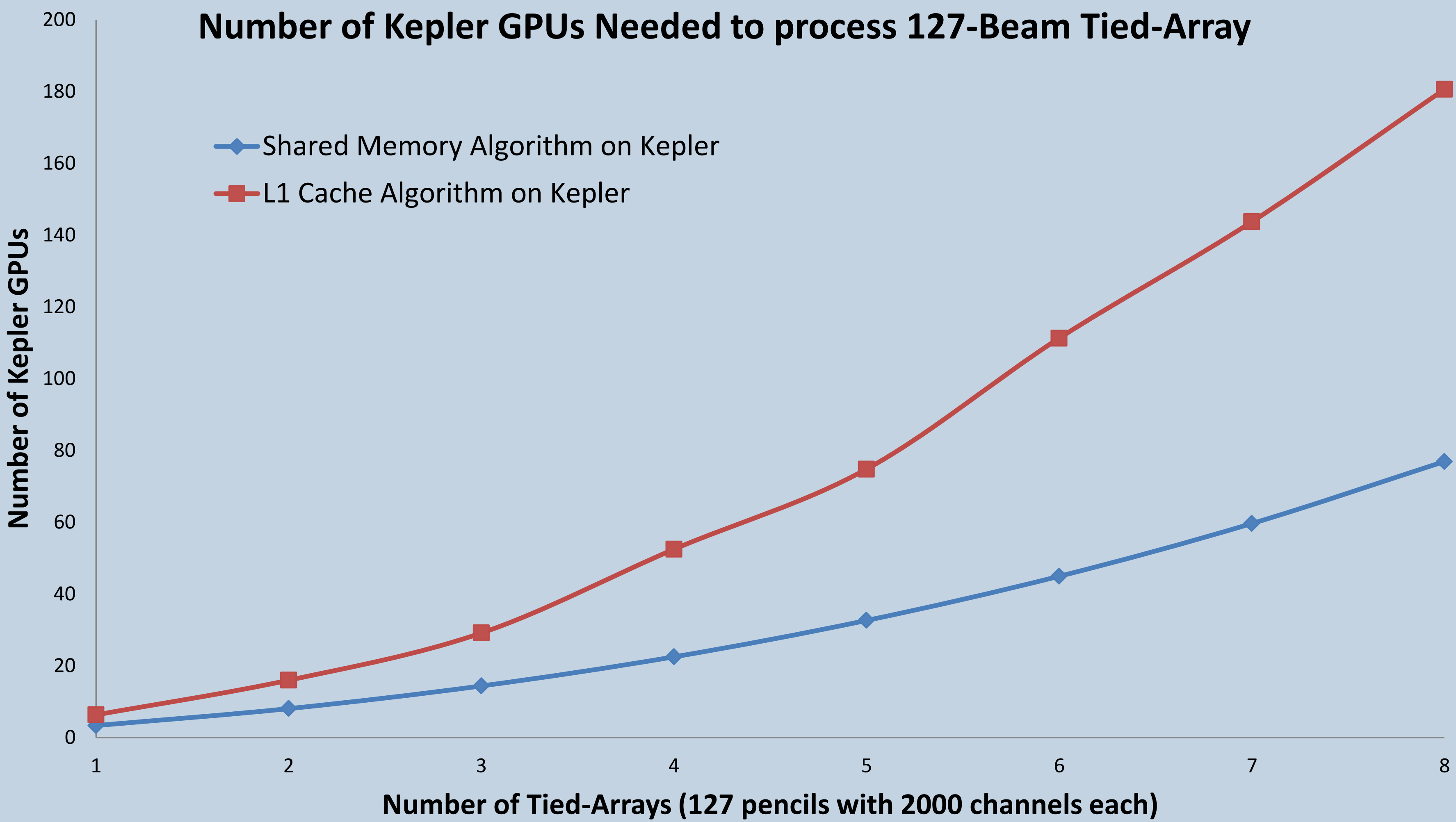


Adding more CPU cores doesn't help and is expensive!!
Time in ms
3
2.5
2
1.5
1
0.5
0
0
20
40
60
80
100
120
140
160
180
Cores used

Conclusions and Future Work



Conclusions and Future Work



OXFORD e-Research CENTRE

Conclusions and Future Work

- GPU wins hands-down. **At the moment (and for the foreseeable future)!**
- ***Shared Memory Algorithm achieves between 60-70% of peak performance.***
- AVX puts up a good fight.
- Watch out for Intels MIC (Many Integrated Core) chip – 32 in-order cores, 4 threads per core 512 bit SIMD units running a 1024 bit ring bus.
- OpenCL Algorithm, Dan Curran / Simon McIntosh-Smith (Bristol): Initial results are currently 2x slower than NVIDIA CUDA Code.

Is a GPU based back-end to the Blue Gene/P at the Central Processing (CEP) facility feasible??

Acknowledgments and Collaborators

GPU de-dispersion : http://www.oerc.ox.ac.uk/research/wes

ARTEMIS : http://www.oerc.ox.ac.uk/research/artemis

University of Malta

Alessio Magro – MDSM

University of Oxford

Mike Giles (Maths) – Cuda, GPU algorithms.

Aris Karastergiou (Physics) – ARTEMIS, Astrophysics, Experimental Work.

Kimon Zagkouris (Physics) – Astrophysics, Experimental Work.

Chris Williams (OeRC) – RFI Clipper, Data pipeline.

Ben Mort (OeRC) – Data Pipeline, pelican.

Fred Dulwich (OeRC) – Data Pipeline, pelican.

Stef Salvini (OeRC) – Data Pipeline, pelican.

Steve Roberts (Engineering) – Signal processing/detection algorithms.

University of Bristol

Dan Curran (Electrical Engineering) – OpenCL work.

Simon McIntosh Smith (Electrical Engineering) – OpenCL work.