On the Performance Portability of Structured Grid Codes on Many-Core Computer Architectures

Simon McIntosh-Smith

Mike Boulton

Dan Curran

James Price

UOB HPC

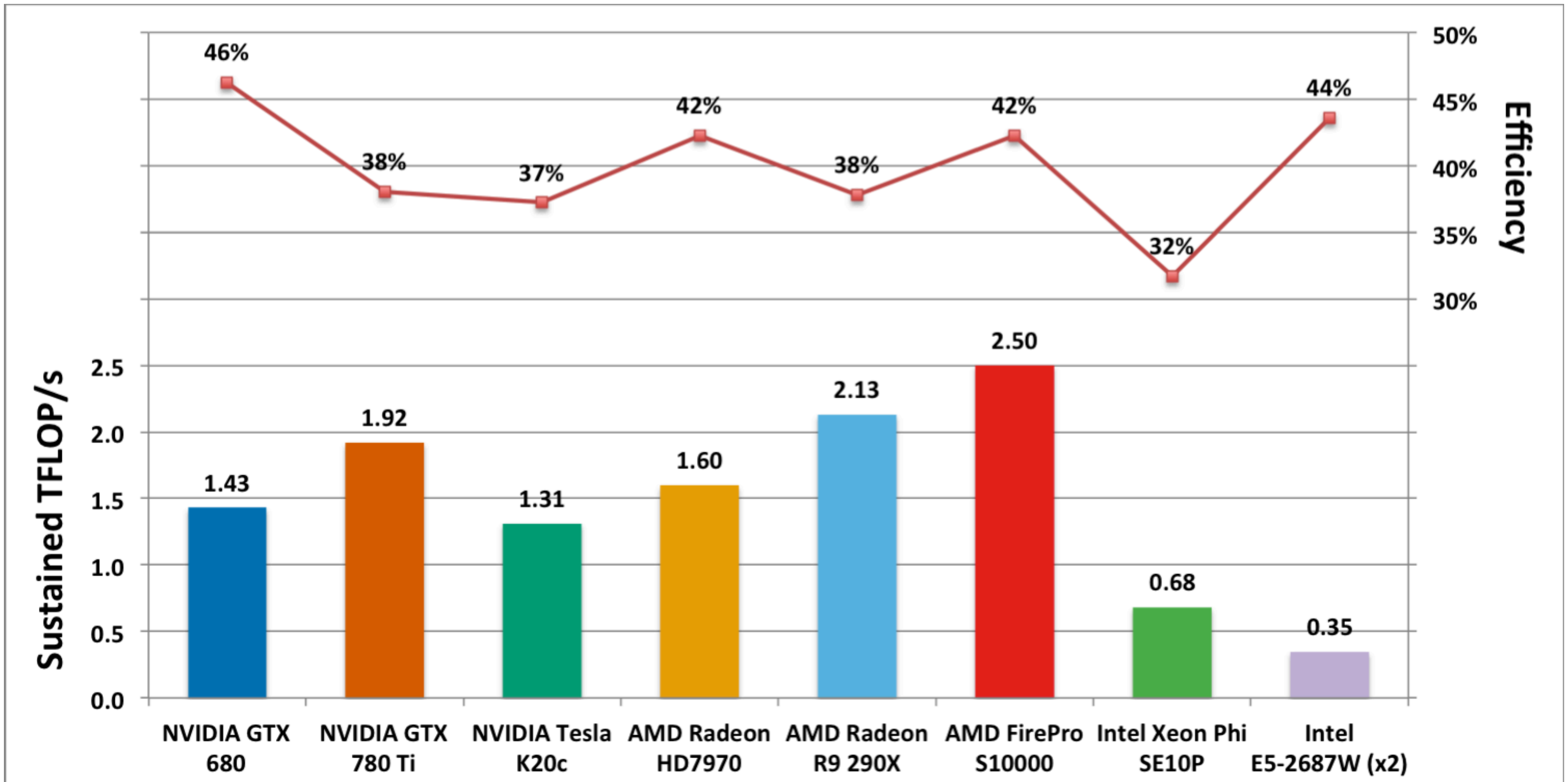
An Intel Parallel Computing Center

University of BRISTOL

Twitter: @simonmcs, @HPCBristol

Motivation

Our BUDE molecular docking OpenCL code showed strong performance portability:

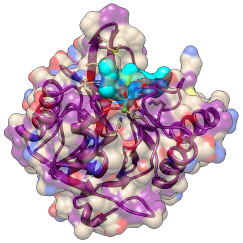


University of BRISTOL

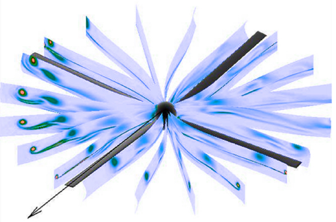
"High Performance *in silico* Virtual Drug Screening on Many-Core Processors", S. McIntosh-Smith, J. Price, R.B. Sessions, A.A. Ibarra, IJHPCA 2014
DOI: 10.1177/1094342014528252

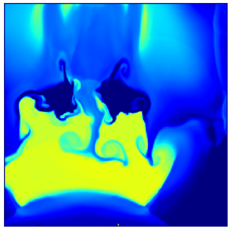
Performance portability

- BUDE was highly performance portable
  - Compute intensive, N-body / Monte Carlo



- Bandwidth intensive codes next
  - Structured grid codes:
    - Lattice Boltzmann
    - CloverLeaf (hydrodynamics)
    - ROTORSIM (CFD)





University of BRISTOL

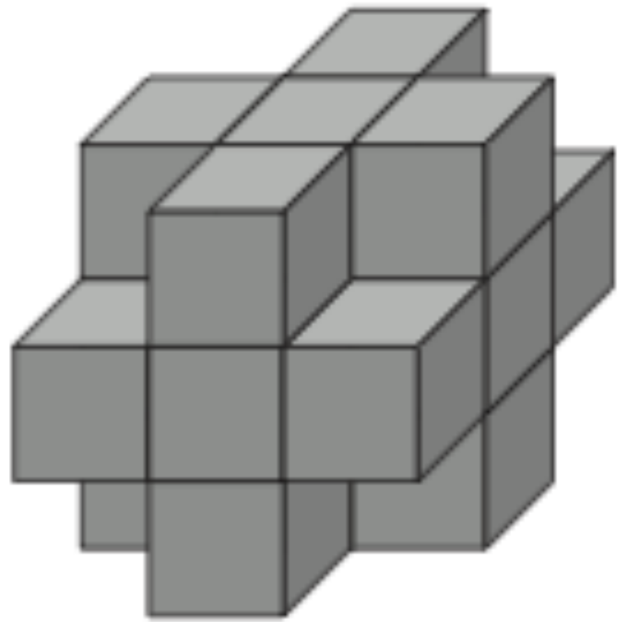
Target hardware

Platform	Clock (GHz)	RAM (GB)	Memory B/W (GB/s)	S.P. TFLOP/s	D.P. TFLOP/s	TDP (W)
AMD FirePro S10000	0.825	6	480	5.91	1.48	375
AMD Radeon HD 7970	0.925	3	264	3.78	0.95	230
AMD Radeon R9 290X	1.000	4	320	5.63	0.70	250
Intel Xeon E5-2687W (x2)	3.100	32	102	0.79	0.40	300
Intel Xeon Phi SE10P	1.100	8	320	2.15	1.07	300
NVIDIA GTX 780 Ti	0.928	3	336	5.05	0.21	250
NVIDIA GTX 680	1.006	2	192	3.00	0.13	195
NVIDIA Tesla K20	0.706	6	208	3.52	1.17	225
NVIDIA Tesla M2090	0.650	6	177	1.33	0.66	225

Lattice Boltzmann (LBM)

- A versatile approach for solving incompressible flows based on a simplified gas-kinetic description of the Boltzmann equation (used for CFD etc)
- Ports well to most parallel architectures
- We targeted one of the most widely used variants, **D3Q19-BGK**

D3Q19-BGK LBM



- To update a cell, need to access 19 of the 27 surrounding cell values in the 3D grid

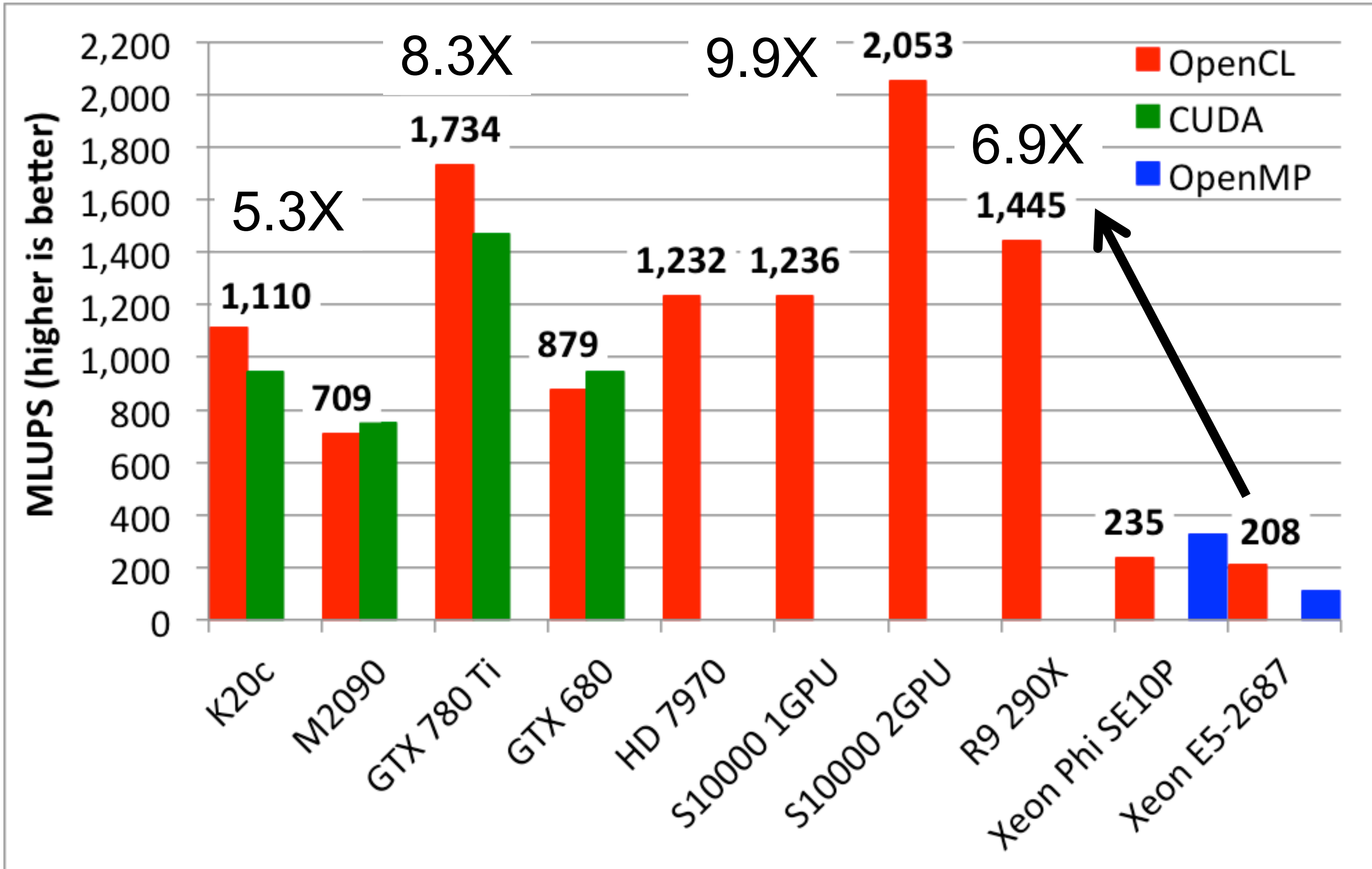
Methodology

- Developed a code that was efficient but not over complicated

- "Identical" versions in OpenCL and CUDA
  - Single precision grid 128^3 (~2m grid points, 304 MBytes)
  - The OpenCL three dimensional work-group size was fixed at (128,1,1) for ***all*** OpenCL runs on ***all*** devices
  - Same arrangement for CUDA version

- The OpenMP code was as close as possible to the OpenCL/ CUDA versions
- Ensured the OpenMP code was being vectorised by the compiler (latest Intel icc)

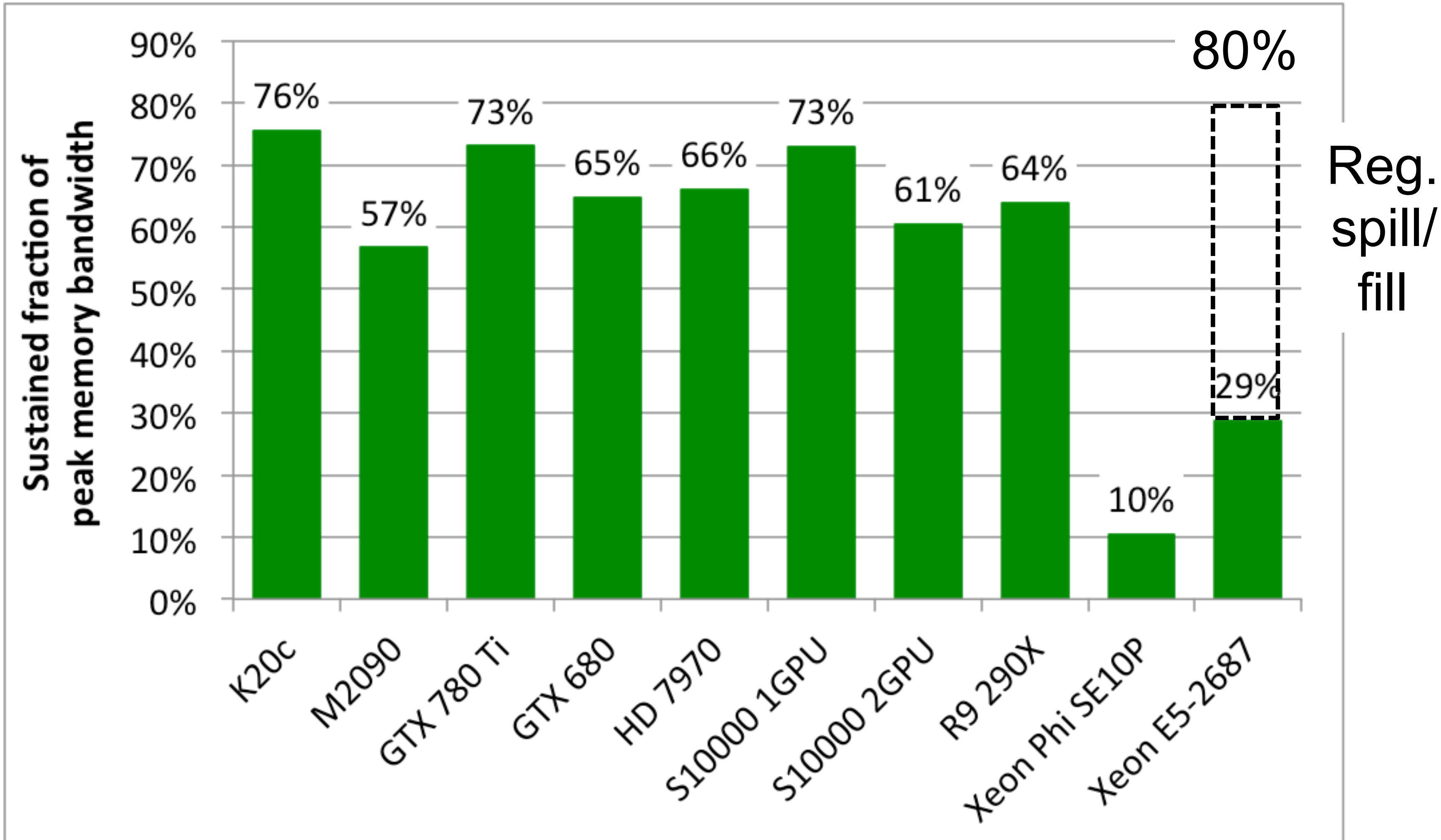
Performance results for 128^3



University of BRISTOL

Single precision results

Performance results for 128^3



OpenCL single precision results

University of BRISTOL

So perf. portable, but is it fast?

- On an Nvidia K20, our best 128^3 single precision performance in OpenCL was **1,110** MLUPS
- In the literature, the fastest quoted results are **~1,000** MLUPS (Januszewski and Kostur's *Sailfish* program) and **982** MLUPS (Mawson and Revell)
- Our results are a **13%** improvement over Mawson-Revell and a **10%** improvement over Januszewski-Kostur

University of BRISTOL

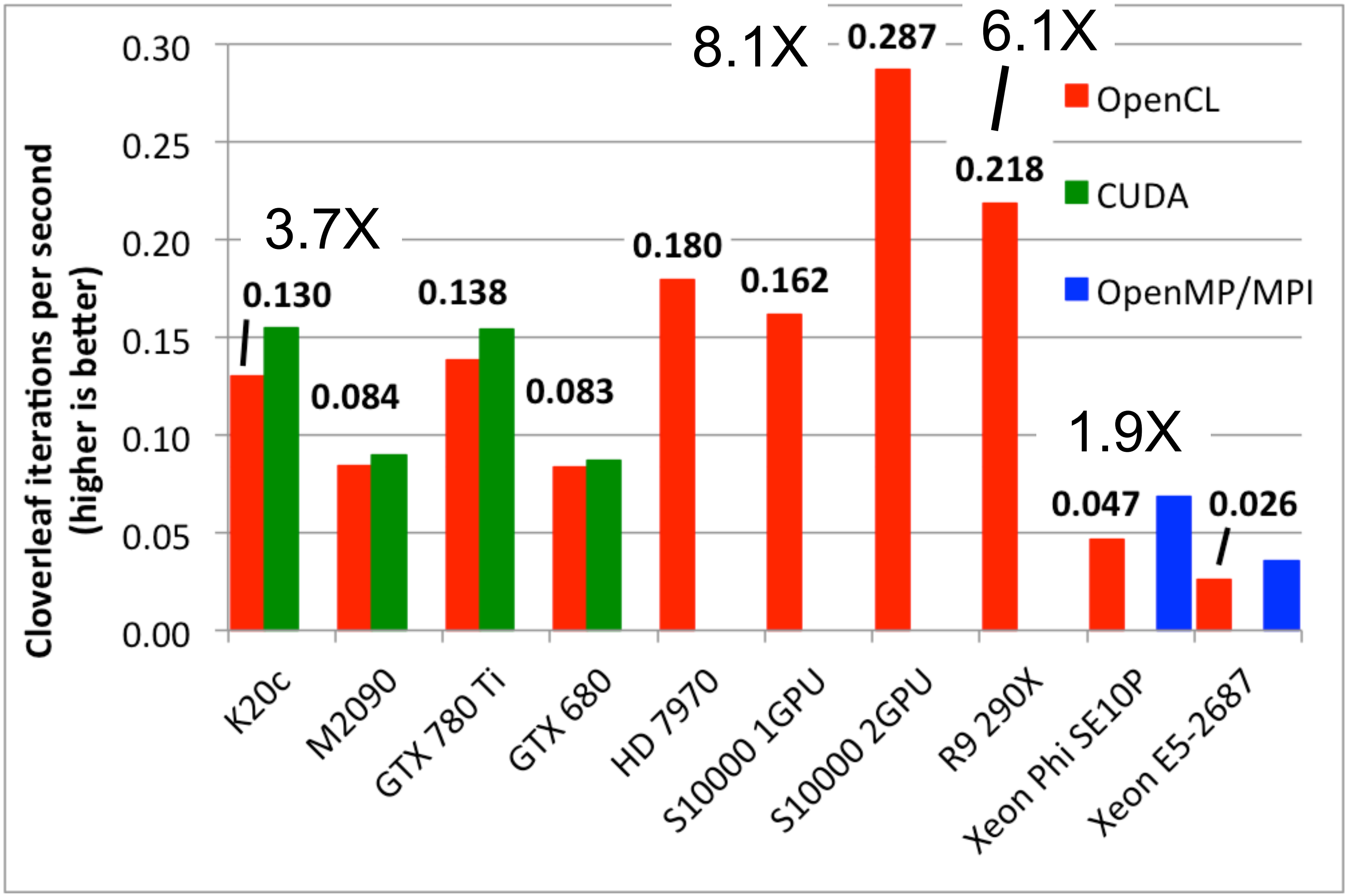
CloverLeaf: A Lagrangian- Eulerian hydrodynamics benchmark

- A collaboration between AWE, Warwick & Bristol
- CloverLeaf is a bandwidth-limited, structured grid code and part of Sandia's "Mantevo" benchmarks.
- Solves the compressible Euler equations, which describe the conservation of energy, mass and momentum in a system.
- These equations are solved on a Cartesian grid in 2D with second-order accuracy, using an explicit finite-volume method.
- Optimised parallel versions exist in OpenMP, MPI, OpenCL, OpenACC, CUDA and Co-Array Fortran.

CloverLeaf benchmark parameters

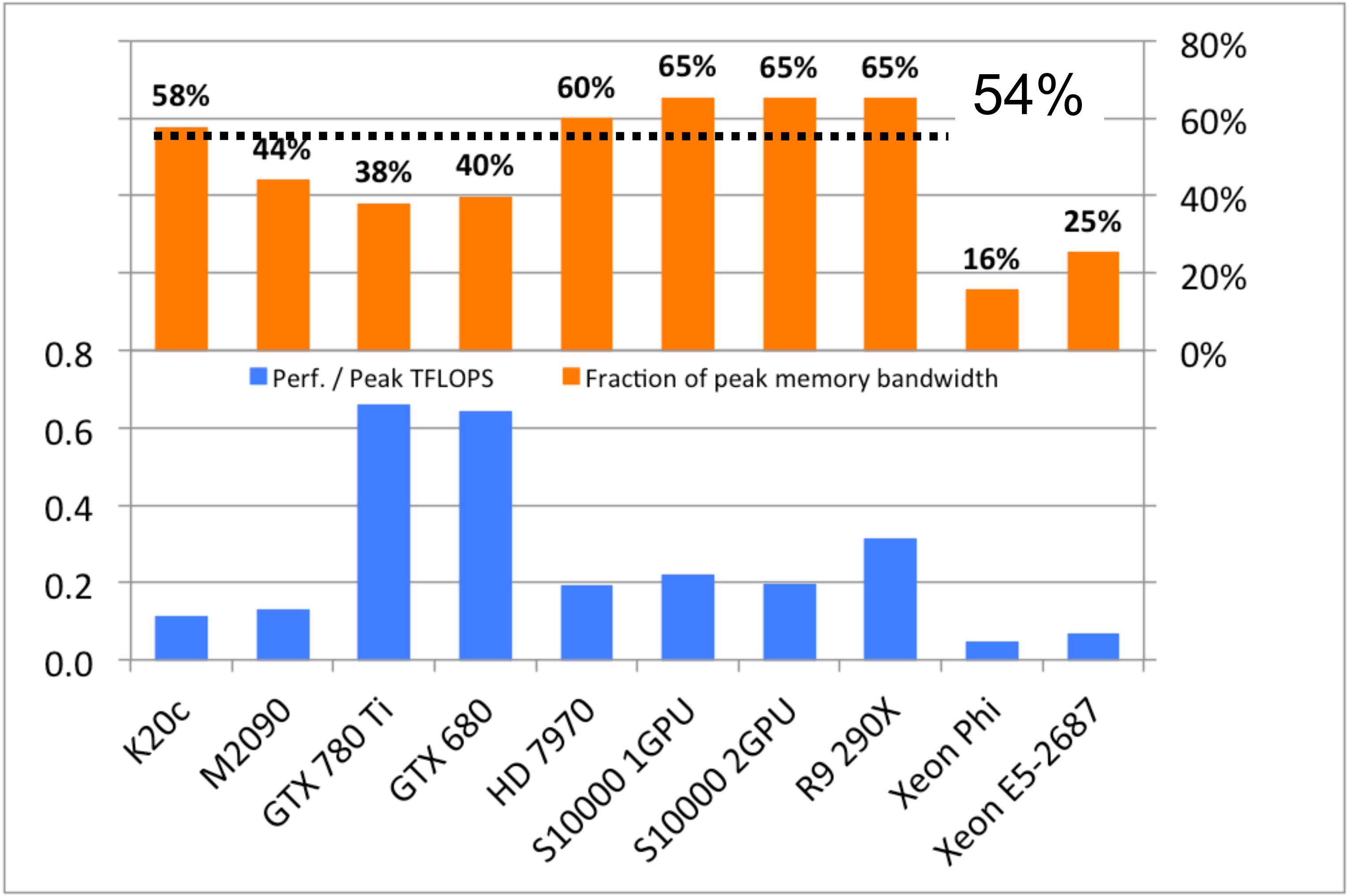
- Double precision grid of size 1920×3840
  - ~7.4m grid points, 25 values per grid point →~1.5 Gbytes in working dataset
- The OpenCL and CUDA parallelisations were performed in an identical manner
  - One work-item/thread for each grid point
  - Identical arrangements for work-group sizes and layouts
  - E.g. 2D work-groups of (128, 1) for OpenCL

Results – performance



S.N. McIntosh-Smith, M. Boulton, D. Curran, & J.R. Price, “On the performance portability of structured grid codes on many-core computer architectures”, ISC, Leipzig, June 2014. DOI: 10.1007/978-3-319-07518-1_4

Results – sustained bandwidth



University of BRISTOL

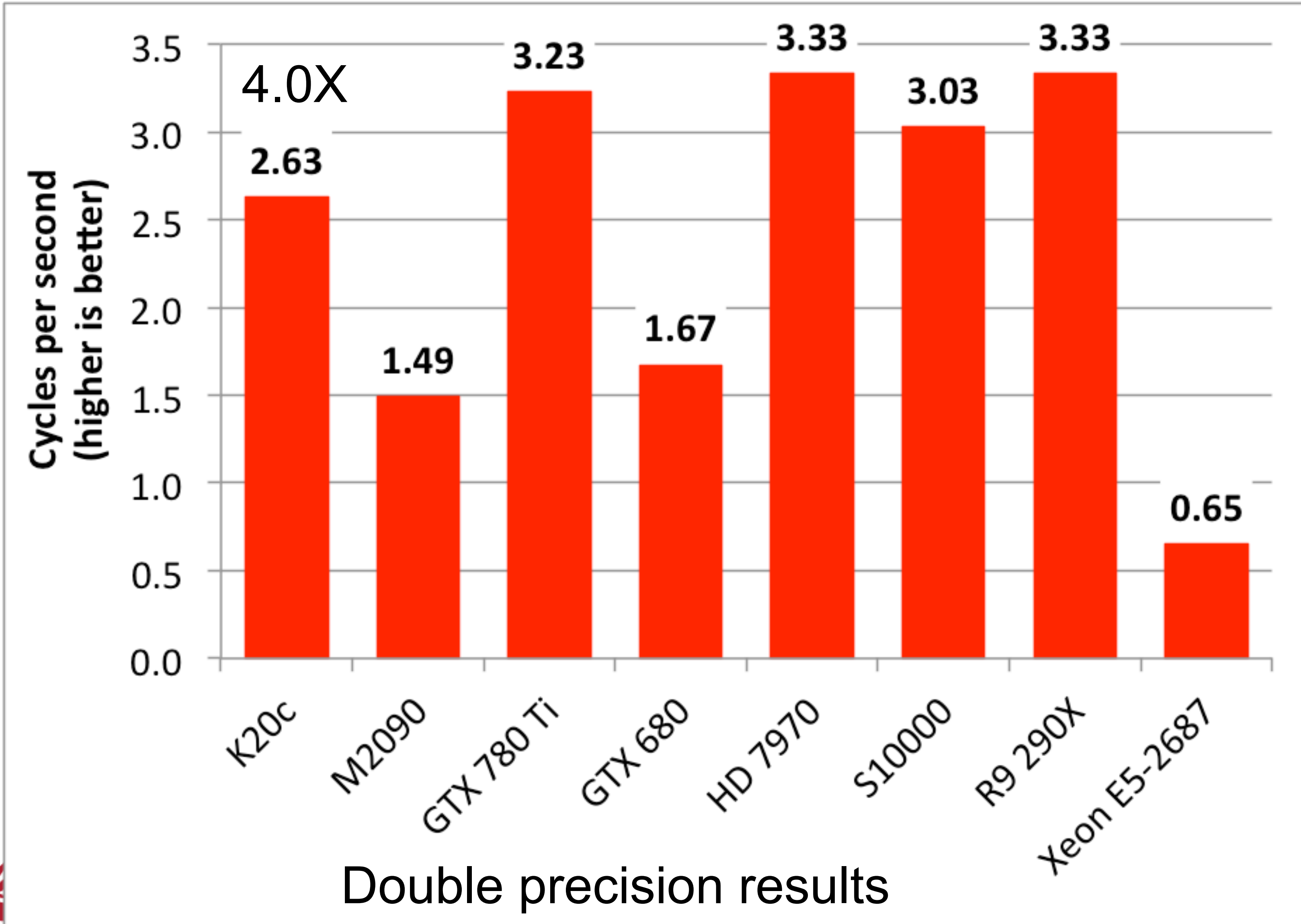
S.N. McIntosh-Smith, M. Boulton, D. Curran, & J.R. Price, “On the performance portability of structured grid codes on many-core computer architectures”, ISC, Leipzig, June 2014. DOI: 10.1007/978-3-319-07518-1_4

ROTORSIM

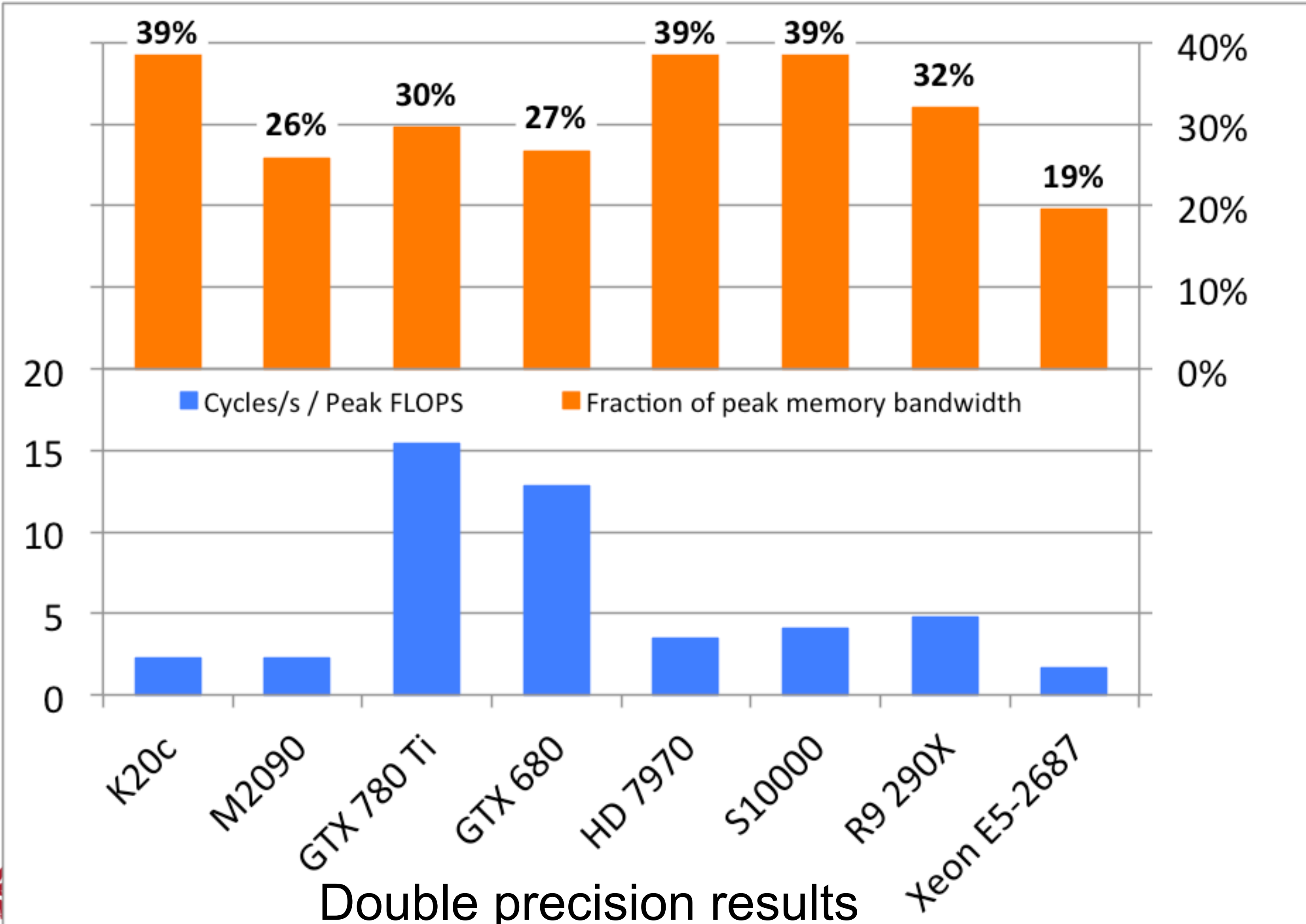
- A production multiblock, compressible finite-volume CFD code
- Developed in Bristol by Prof. Chris Allen
- Upwind, third-order accurate spatial stencil, with an explicit time integration scheme for steady flows
- Implicit dual-time approach for time-accurate calculations
- Optimised versions in MPI and OpenCL

Results – performance

5.1X



Results – sustained bandwidth



Double precision results

Performance portability isn't what we expect

But why not?

Why don't we expect perf. portability?

- Historical reasons
  - Started with immature drivers
  - Started with immature architectures
  - Started with immature applications

- But things have ***changed***
  - Drivers now mature / maturing
  - Architectures now mature / maturing
  - Applications now mature / maturing

Performance portability techniques

- Use a platform portable parallel language
- Aim for 80-90% of optimal
- Avoid platform-specific optimisations
- ***Most*** optimisations make the code faster on ***most*** platforms

Conclusions

- Structured grid codes such as *lattice Boltzmann, CloverLeaf* and *ROTORSIM* can benefit from <u>*significant*</u> *performance improvements on many-core accelerators* such as GPUs and Xeon Phi
- **OpenCL** can straightforwardly enable a much better degree of performance portability than you might expect

University of BRISTOL

Google: **HandsOnOpenCL** on *Github*

Related Publications

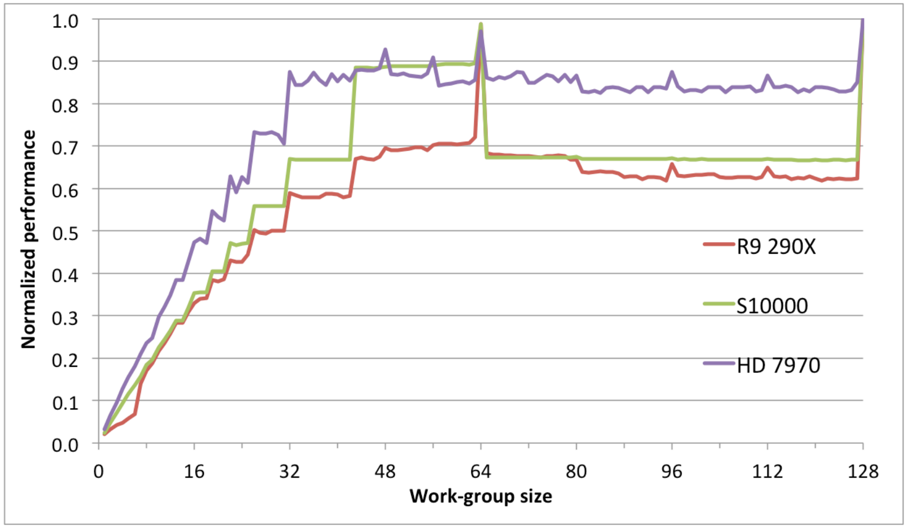
[1] "High Performance *in silico* Virtual Drug Screening on Many-Core Processors", S. McIntosh-Smith, J. Price, R.B. Sessions, A.A. Ibarra, IJHPCA 2014. DOI: 10.1177/1094342014528252

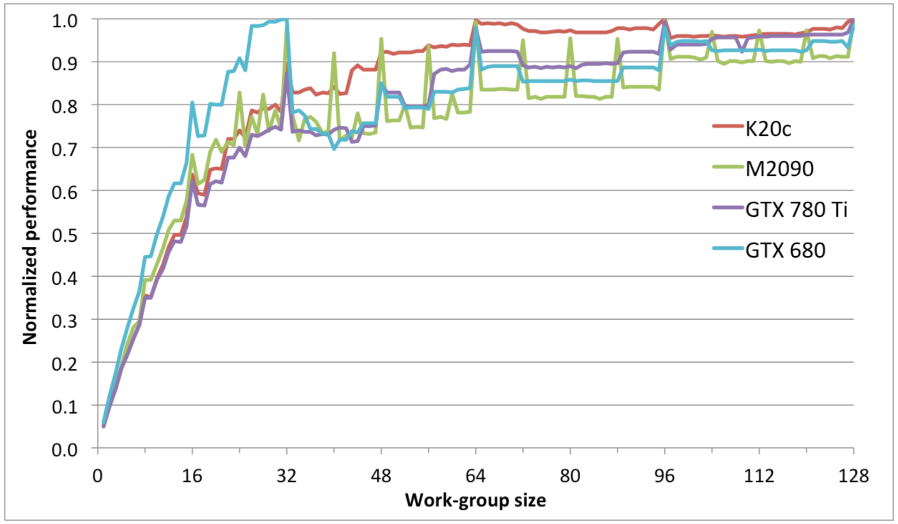
[2] "On the performance portability of structured grid codes on many-core computer architectures", S.N. McIntosh-Smith, M. Boulton, D. Curran and J.R. Price. ISC, Leipzig, June 2014. DOI: 10.1007/978-3-319-07518-1_4

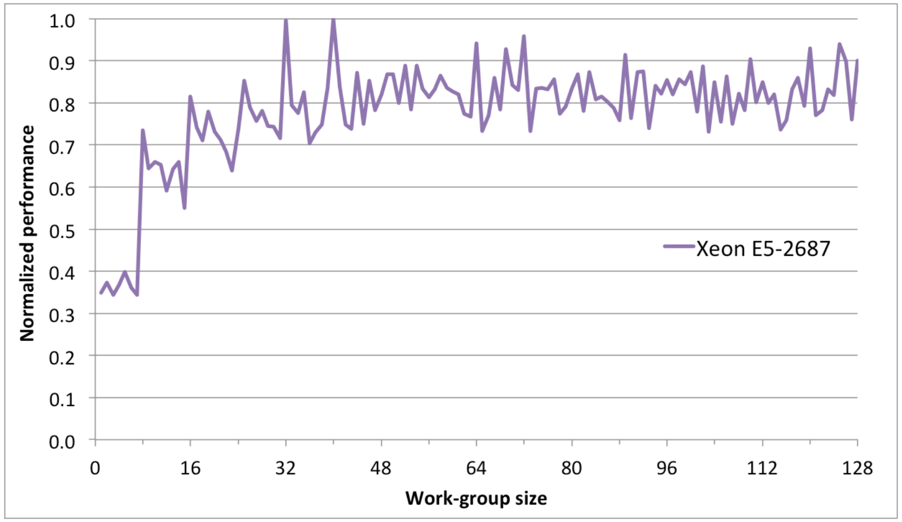
[3] "Accelerating hydrocodes with OpenACC, OpenCL and CUDA", Herdman, J., Gaudin, W., McIntosh-Smith, S., Boulton, M., Beckingsale, D., Mallinson, A., Jarvis, S. In: High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion:. (Nov 2012) 465-471. DOI: 10.1109/SC.Companion.2012.66

BACKUP

Impact of work-group sizes







University of BRISTOL