
Article:
Wright, Steven A. orcid.org/0000-0001-7133-8533 (2018) Performance Modeling, Benchmarking and Simulation of High Performance Computing Systems. Future generation computer systems. pp. 900-902. ISSN 0167-739X

https://doi.org/10.1016/j.future.2018.11.020

Editorial
Performance Modeling, Benchmarking and Simulation of High Performance Computing Systems

Steven A. Wright*

Department of Computer Science
University of York, UK

Abstract

This special issue of *Future Generation Computer Systems* contains four extended papers selected from the 7th International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computing Systems (PMBS 2016), held as part of the 28th International Conference for High Performance Computing, Networking, Storage and Analysis (SC 2016). These papers represent worldwide programmes of research committed to understanding application and architecture performance to enable post-peta-scale computational science.

Introduction

Scientific discovery has been accelerated enormously in the past century due to the emergence of the field of computational science. For situations where physical experimentation is prohibitively costly, impractical, or dangerous, scientific computing is now extensively used to test theories and further understanding. Computational methods have joined theory and experiment as central pillars of scientific investigation.

Because of the extensive use of computation in science, maximizing computational performance is of paramount importance. Ensuring both hardware and software are operating at their maximum capacity allows scientists to perform increasingly accurate and complex simulations within an acceptable timeframe. This need for high performance, has led to the development of *supercomputers*.

Since 1993, the performance of the world's fastest supercomputers, measured in the number of floating point operations per second (FLOP/s), has been tracked by the twice yearly Top500 list [1]. The Top500 list serves as a valuable resource that allows comparisons to be made between systems and

*Corresponding author
Email address: steven.wright@york.ac.uk [tbc] (Steven A. Wright)

helps identify trends in HPC architectures [2, 3]. The #1 ranked supercomputer in the first list was the Numerical Wind Tunnel in Japan – capable of 124,000,000,000 operations per second, or 124 GFLOP/s.

Supercomputer performance has subsequently increased by 6 orders of magnitude in the decades since. The first major performance barrier, the TeraFLOP, was passed in 1997 by the ASCI Red HPC system built and installed at Sandia National Laboratories. ASCI Red remained at the top of the list for 3 years, with later upgrades taking its performance to 3.1 TFLOP/s.

In 2008, the IBM Roadrunner machine, at Los Alamos National Laboratory, broke the PetaFLOP barrier and assumed the #1 position. Roadrunner was notable for its *hybrid* architecture, where the AMD Opteron CPUs handled the operating system, while computation was performed on an attached accelerator – a common feature in many of today's supercomputers.

As of June 2018, there are 273 PetaFLOP-capable HPC systems in the Top500 list. K Computer, installed at the RIKEN Advanced Institute for Computational Science, was the first machine to eclipse 10 PFLOP/s in November 2011, and consistently remained in the top 10 supercomputers for 6 years.

Sunway TaihuLight, installed in 2017, has a theoretical peak that surpasses 100 PFLOP/s, and achieved 93 PFLOP/s on the LINPACK benchmark. In June 2018, TaihuLight was replaced at the top of the list by Summit, an Oak Ridge National Laboratory based supercomputer with a theoretical peak of 188 PFLOP/s, and an achieved peak of 122 PFLOP/s. Like Roadrunner, Summit is a heterogeneous architecture with each compute node consisting of two IBM POWER9 CPUs and six NVIDIA Tesla V100 GPUs.

The Race to Exascale

The next major milestone for HPC is the ExaFLOP, or 10^{18} floating-point operations per second. It is anticipated that this will be achieved within the next 5 years and that having this level of computation available will have a profound effect on scientific capability.

Around the world there are a number of national and international efforts to deliver an Exascale system along with the supporting infrastructure.

In China, Tianhe-3 is being developed at the National Computer Centre in Tianjin. Tianhe-3 will represent a 11× increase in performance over TaihuLight, and a 33× improvement over the previous Tianhe system (Tianhe-2). The system is planned for 2020 and it is projected to provide an ExaFLOP of performance within 30-40 MW of power.

Japan's FLAGSHIP 2020 project at RIKEN aims to deliver a Post-K Computer in 2021, again within 30-40 MW of power consumption [4]. The design is set to be based on a general-purpose many-core architecture, rather than using a hybrid architecture with accelerators.

The United States has two major projects focussed on the delivery of Exascale computing. The CORAL project (Collaboration of Oak Ridge, Argonne and Livermore) is dedicated to the procurement and installation of three Exascale-class systems between 2021 and 2023 – A21 at Argonne National Laboratory,

Frontier at Oak Ridge National Laboratory and El Capitan at Lawrence Livermore National Laboratory. The Exascale Computing Project (ECP) is the United States' collaborative effort by two Department of Energy organizations – The Office of Science and the National Nuclear Security Administration – with a focus on the delivery of an Exascale-capable computing ecosystem [5].

Similarly, there are a number European Exascale projects – funded through initiatives such as the European Union's Seventh Framework Programme (EU FP7) – investigating innovative approaches to hardware design and programming models as well as supporting scientific application development [6]. The Mont Blanc 2020 project is a three year effort to deliver an Exascale system using a low-power System-on-Chip (SoC) architecture. The DEEP-ER project is working towards the development an heterogeneous modular HPC system capable of an ExaFLOP. Projects such as CRESTA (Collaborative Research into Exascale Systemware, Tools & Applications) and EESI (European Exascale Software Initiative) are developing the software and tools that will be required by Exascale-class systems.

Challenges of Exascale

The anticipated delivery of Exascale-capable systems between 2020 and 2023 will enable new research in some of the grand challenges of computational science [7, 8]. The performance and parallelism available on post-Exascale systems will offer benefits in the fields of weather prediction, astronomy and cosmology, material sciences, biological systems, aerodynamics and theoretical physics, among many others.

Each of the Exascale projects mentioned previously are working towards this common goal, but each is exploring a different approach. Japan's FLAGSHIP 2020 project and Mont Blanc both propose homogeneous many-core architectures based on ARM SoCs; the DEEP-ER and CORAL projects are investigating the use of heterogeneous architectures; and there are projects in China to develop a new custom architecture for use in their Exascale system.

Supercomputing is tending towards a diversification of hardware. In the current Top500, 10 distinct architectures are represented, ranging from Intel's Xeon range of x86 processors and IBM's POWER architecture to custom processors such as the PEZY-SC2 and Matrix-2000 accelerator cards. As of June 2018, there are 110 systems using computational accelerators, the majority of which are NVIDIA GPUs. The trend towards diversification continues beyond just compute, with machines making increasing use of novel memory systems, interconnects and I/O subsystems.

This rapidly changing environment is bringing about significant challenges in HPC [9]. As supercomputing architectures diversify, applications need to remain portable between architectures to avoid potential vendor lock-in or suboptimal performance [10]. Rising component counts mean that resilience mechanisms must evolve to ensure that the effects of hardware and software failures can be mitigated [11]. Each subsystem of the supercomputer must evolve in parallel to ensure that no single part becomes a bottleneck to performance [12–15]. For

supercomputing to remain sustainable, the energy efficiency of applications and systems must be considered carefully [16, 17].

Without Exascale systems available, these challenges are often approached using performance modeling, benchmarking and simulation. Analytical models such as LogGP have traditionally been used to predict performance of current and future systems [18, 19]. Benchmarking has often been used to assess the performance of hardware and identify performance bottlenecks or find opportunities for optimization [13, 20, 21]. Simulators can be used to predict performance at much greater scale than available [22, 23], as well as allow us an opportunity to test hypothetical systems [5, 24]. Each of these approaches to performance engineering is helping to prepare for the availability of Exascale computing.

Performance Modeling, Benchmarking and Simulation

This issue of *Future Generation Computer Systems* contains four extended papers from the 7th Performance Modeling, Benchmarking and Simulation of High Performance Computing Systems Workshop (PMBS 2016), which was held as part of the 28th International Conference for High Performance Computing, Networking, Storage, and Analysis (often simply refered to as Supercomputing (SC)) in 2016. The SC conference is the premier international forum for research that focusses on addressing some of these challenges.

The SC conference offers a vibrant technical program, which includes technical papers, tutorials in advanced areas, birds-of-a-feather sessions, panel debates, a doctoral showcase and a number of technical workshops in specialist areas. The SC conference hosts a wide range of international participants from academia, national laboratories and industry, and regularly features over 350 exhibitors in the industry's largest annual HPC technology fair.

The PMBS workshop began at the 2010 SC conference in New Orleans and has been a fixture of the workshop programme ever since [25–30]. The focus of the workshop is in comparing high performance computing systems through performance modeling, benchmarking or the use of tools such as simulators. In recent years, we have been particularly interested in receiving research papers which report the ability to measure and make trade-offs in hardware/software co-design to improve sustained application performance. We have also been keen to capture the assessment of future systems, for example through work that ensures continued application scalability through to Exascale systems.

Future Generation Computer Systems

Following the 2016 PMBS workshop, selected authors were invited to submit extended versions of their papers for consideration to FGCS. Four of these submissions were accepted following a subsequent round of reviews to ensure that they were of the highest quality.

The first of the four papers is concerned with the use of memory tiling techniques to improve the performance of scientific applications running on many-core architectures [31]. Yount et al. demonstrate how rewriting finite-difference

numerical simulations using techniques such as vector-folding and spatial tiling can improve cache resource utilisation. Further, the authors show how temporal wave-front tiling can be applied when stencil problem sizes exceed the capacity of a shared cache, leading to speedups ranging from 1.9× to 3.3×.

Proxy applications are increasingly being developed to investigate potential performance issues in scientific simulations. The second paper in this special issue, by Pearce et al., uses the CoMD proxy application to explore the issue of load imbalance at scale [32]. In the paper, CoMD is extended to allow users to control initial load imbalance and to enable work migration. Using their extended application, the authors are able to analyse the negative impact of load imbalance, and using dynamic rebalancing, can significantly improve its performance.

The paper by Guerrera et al., outlines the use of Prova! to record and generate reproducible experimental environments to explore application performance on a variety of different HPC systems under different configurations [33]. In the paper, the authors demonstrate the use of Prova! on stencil kernels, providing a comparative analysis across four different systems using a variety of parallelisation techniques (MPI, OpenMP and CUDA).

The final paper in this special issue focuses on *performance portability*. Pennycook et al. introduce a new definition and a novel metric for characterizing performance portability [34]. They apply their metric to a number of published application studies to highlight the use of a shared metric for comparing approaches to portability, and suggest a number of techniques and tools to aid application developers in future.

The four papers contained in this special issue represent worldwide programmes of research focussed on the performance of scientific applications on HPC systems. Two of the papers focus on improving the performance of applications at scale, through exploiting cache reuse on many core architectures and through dynamically applying load balancing to alleviate the issue caused by imbalance. The final two papers cover the topics of reproducibility and portability. With the increasing diversification of HPC architectures as we approach Exascale, these topics will only grow in importance.

Acknowledgements

I would like to express my thanks to the international review committee who have supported this special issue. I am grateful to Hilda Xu (at the Elsevier Editorial Office) and Peter Sloot (Editor-in-Chief) for assisting with the production of this issue of Future Generation Computer Systems.

The PMBS workshop is made possible thanks to significant input from Atomic Weapons Establishment (AWE) in the UK, and from Sandia National Laboratories and the Lawrence Livermore National Laboratory in the US. We acknowledge the support of the AWE Technical Outreach Programme (project CDK0724).

References

[1] H. Meuer, E. Strohmaier, J. J. Dongarra, H. D. Simon, M. Meuer, TOP500 Supercomputer Sites, `http://top500.org` (accessed June 29, 2018) (2018).

[2] J. J. Dongarra, H. W. Meuer, H. D. Simon, E. Strohmaier, Changing Technologies of HPC, Future Generation Computer Systems 12 (5) (1997) 461 – 474.

[3] J. J. Dongarra, P. Luszczek, A. Petitet, The LINPACK Benchmark: Past, Present and Future, Concurrency and Computation: Practice and Experience 15 (9) (2003) 803–820.

[4] Y. Ishikawa, System Software in Post K Supercomputer, in: Proceedings of the 27th ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis (SC'15), ACM, 2015.

[5] S. S. Dosanjh, R. F. Barrett, D. W. Doerfler, S. D. Hammond, K. S. Hemmert, M. A. Heroux, P. T. Lin, K. T. Pedretti, A. F. Rodrigues, T. G. Trucano, J. P. Luitjens, Exascale Design Space Exploration and Co-design, Future Generation Computer Systems 30 (2014) 46–58.

[6] N. Attig, P. Gibbon, T. Lippert, Trends in Supercomputing: The European Path to Exascale, Computer Physics Communications 182 (9) (2011) 2041–2046.

[7] K. G. Wilson, Grand Challenges to Computational Science, Future Generation Computer Systems 5 (2) (1989) 171–189.

[8] Advisory Committee for Cyberinfrastructure, Task Force on Grand Challenges, Tech. rep., National Science Foundation (2011).

[9] Advanced Scientific Computing Advisory Committee, The Opportunities and Challenges of Exascale Computing, Tech. rep., U.S. Department of Energy, Office of Science (2010).

[10] U. Gärtel, W. Joppich, A. Schüller, H. Schwichtenberg, U. Trottenberg, G. Winter, Two Strategies in Parallel Computing: Porting Existing Software Versus Developing New Parallel Algorithms – Two Examples, Future Generation Computer Systems 10 (2) (1994) 257–262.

[11] D. A. Reed, C. da Lu, C. L. Mendes, Reliability Challenges in Large Systems, Future Generation Computer Systems 22 (3) (2006) 293–302.

[12] S. A. Wright, S. A. Jarvis, Quantifying the Effects of Contention on Parallel File Systems, in: Proceedings of the 29th IEEE International Parallel & Distributed Processing Symposium Workshops & PhD Forum (IPDPSW'15), IEEE Computer Society, Washington, DC, Hyderabad, India, 2015, pp. 932–940.

[13] D. J. Kerbyson, K. J. Barker, A. Vishnu, A. Hoisie, A Performance Comparison of Current HPC Systems: Blue Gene/Q, Cray XE6 and InfiniBand Systems, Future Generation Computer Systems 30 (2014) 291–304.

[14] C. S. Daley, D. Ghoshal, G. K. Lockwood, S. Dosanjh, L. Ramakrishnan, N. J. Wright, Performance Characterization of Scientific Workflows for the Optimal use of Burst Buffers, Future Generation Computer Systems.

[15] H. Meyer, J. C. Sancho, M. Mrdakovic, W. Miao, N. Calabretta, Optical Packet Switching in HPC. An Analysis of Applications Performance, Future Generation Computer Systems 82 (2018) 606–616.

[16] S. I. Roberts, S. A. Wright, D. Lecomber, C. January, J. Byrd, X. Oro, S. A. Jarvis, POSE : A Mathematical and Visual Modelling Tool to Guide Energy Aware Code Optimisation, in: Proceedings of the 6th International Green and Sustainable Computing Conference (IGSC'15), IEEE, 2015, pp. 1–8.

[17] G. T. Chetsa, L. Lefèvre, J. Pierson, P. Stolf, G. D. Costa, Exploiting Performance Counters to Predict and Improve Energy Performance of HPC Systems, Future Generation Computer Systems 36 (2014) 287–298.

[18] A. Alexandrov, M. F. Ionescu, K. E. Schauser, C. Scheiman, LogGP: Incorporating Long Messages into the LogP Model – One Step Closer Towards a Realistic Model for Parallel Computation, in: Proceedings of the 7th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA'95), ACM, New York, NY, Santa Barbara, CA, 1995, pp. 95–105.

[19] R. A. Bunt, S. A. Wright, S. A. Jarvis, M. Street, Y. K. Ho, Predictive Evaluation of Partitioning Algorithms Through Runtime Modelling, in: Proceedings of the 23rd High Performance Computing, Data, and Analytics (HiPC'16), IEEE, 2016, pp. 351–361.

[20] M. Martineau, S. McIntosh-Smith, W. Gaudin, Assessing the Performance Portability of Modern Parallel Programming Models using TeaLeaf, Concurrency and Computation: Practice and Experience 29 (15) (2017) e4117.

[21] J. A. Herdman, W. P. Gaudin, S. McIntosh-Smith, M. Boulton, D. A. Beckingsale, A. C. Mallinson, S. A. Jarvis, Accelerating Hydrocodes with OpenACC, OpenCL and CUDA, in: 2012 SC Companion: High Performance Computing, Networking, Storage and Analysis (SCC), IEEE, 2012, pp. 465–471.

[22] S. A. Jarvis, D. P. Spooner, H. N. L. C. Keung, J. Cao, S. Saini, G. R. Nudd, Performance Prediction and its use in Parallel and Distributed Computing Systems, Future Generation Computer Systems 22 (7) (2006) 745–754.

[23] C. Engelmann, Scaling to a Million Cores and Beyond: Using Light-weight Simulation to Understand the Challenges Ahead on the Road to Exascale, Future Generation Computer Systems 30 (2014) 59–65.

[24] A. F. Rodrigues, K. S. Hemmert, B. W. Barrett, C. Kersey, R. Oldfield, M. I. Weston, R. Riesen, J. Cook, P. Rosenfeld, E. Cooper-Balis, B. Jacob, The Structural Simulation Toolkit, SIGMETRICS Performance Evaluation Review 38 (4) (2011) 37–42.

[25] S. A. Jarvis (Ed.), Special Issue on the 1st International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computing Systems (PMBS'10), Vol. 38 of SIGMETRICS Performance Evaluation Review, ACM New York, NY, New Orleans, LA, 2011.

[26] S. A. Jarvis (Ed.), Special Issue on the 2nd International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computing Systems (PMBS'11), Vol. 40 of SIGMETRICS Performance Evaluation Review, ACM New York, NY, Seattle, WA, 2012.

[27] S. A. Jarvis, S. A. Wright, S. D. Hammond (Eds.), High Performance Computing Systems. Proceedings of the 4th Internation Performance Modeling, Benchmarking, and Simulation of High Performance Computing Systems Workshop (PMBS'13), Denver, CO, USA, November 18, 2013, Vol. 8551 of Lecture Notes in Computer Science (LNCS), Springer, Berlin, 2014.

[28] S. A. Jarvis, S. A. Wright, S. D. Hammond (Eds.), High Performance Computing Systems. Proceedings of the 5th Internation Performance Modeling, Benchmarking, and Simulation of High Performance Computing Systems Workshop (PMBS'14), New Orleans, LA, USA, November 16, 2014, Vol. 8966 of Lecture Notes in Computer Science (LNCS), Springer, Berlin, 2015.

[29] S. A. Jarvis, S. A. Wright, S. D. Hammond (Eds.), Proceedings of the 6th International Workshop on Performance Modeling, Benchmarking, and Simulation of High Performance Computing Systems (PMBS'15), ACM Special Interest Group on High Performance Computing (SIGHPC), ACM New York, NY, 2015.

[30] S. A. Jarvis, S. A. Wright, S. D. Hammond (Eds.), Proceedings of the 7th International Workshop on Performance Modeling, Benchmarking, and Simulation of High Performance Computing Systems (PMBS'16), ACM Special Interest Group on High Performance Computing (SIGHPC), IEEE, 2016.

[31] C. Yount, A. Duran, J. Tobin, Multi-level Spatial and Temporal Tiling for Efficient HPC Stencil Computation on Many-Core Processors with Large Shared Caches, Future Generation Computer Systems (2017) 1–14.

[32] O. Pearce, H. Ahmed, R. W. Larsen, P. Pirkelbauer, D. F. Richards, Exploring Dynamic Load Imbalance Solutions with the CoMD Proxy Application, Future Generation Computer Systems (2017) 1–14.

[33] D. Guerrera, A. Maffia, H. Burkhart, Reproducible Stencil Compiler Benchmarks Using PROVA!, Future Generation Computer Systems (2017) 1–14.

[34] S. J. Pennycook, J. D. Sewall, V. W. Lee, Implication of a Metric for Performance Portability, Future Generation Computer Systems (2017) 1–14.