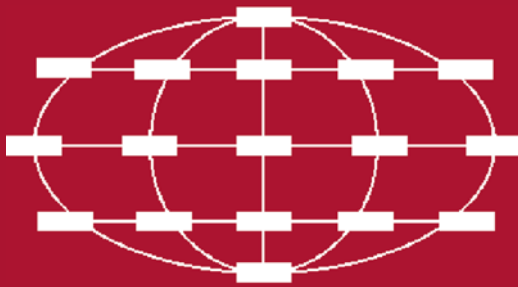


Victor Malyskin (Ed.)

LNCS 9251

# Parallel Computing Technologies

13th International Conference, PaCT 2015  
Petrozavodsk, Russia, August 31 – September 4, 2015  
Proceedings



Springer

*Commenced Publication in 1973*

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

## Editorial Board

David Hutchison

*Lancaster University, Lancaster, UK*

Takeo Kanade

*Carnegie Mellon University, Pittsburgh, PA, USA*

Josef Kittler

*University of Surrey, Guildford, UK*

Jon M. Kleinberg

*Cornell University, Ithaca, NY, USA*

Friedemann Mattern

*ETH Zurich, Zürich, Switzerland*

John C. Mitchell

*Stanford University, Stanford, CA, USA*

Moni Naor

*Weizmann Institute of Science, Rehovot, Israel*

C. Pandu Rangan

*Indian Institute of Technology, Madras, India*

Bernhard Steffen

*TU Dortmund University, Dortmund, Germany*

Demetri Terzopoulos

*University of California, Los Angeles, CA, USA*

Doug Tygar

*University of California, Berkeley, CA, USA*

Gerhard Weikum

*Max Planck Institute for Informatics, Saarbrücken, Germany*

More information about this series at <http://www.springer.com/series/7407>

Victor Malyskin (Ed.)

# Parallel Computing Technologies

13th International Conference, PaCT 2015  
Petrozavodsk, Russia, August 31 – September 4, 2015  
Proceedings

*Editor*  
Victor Malyshkin  
Russian Academy of Sciences  
Novosibirsk  
Russia

ISSN 0302-9743                      ISSN 1611-3349 (electronic)  
Lecture Notes in Computer Science  
ISBN 978-3-319-21908-0              ISBN 978-3-319-21909-7 (eBook)  
DOI 10.1007/978-3-319-21909-7

Library of Congress Control Number: 2015944720

LNCS Sublibrary: SL1 – Theoretical Computer Science and General Issues

Springer Cham Heidelberg New York Dordrecht London  
© Springer International Publishing Switzerland 2015

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media  
([www.springer.com](http://www.springer.com))

## Preface

The PaCT 2015 (Parallel Computing Technologies) conference was a four-day conference held in Petrozavodsk (Russia). This was the 13th international conference in the PaCT series. The conferences are held in Russia every odd year. The first conference, PaCT 1991, was held in Novosibirsk (Academgorodok), September 7–11, 1991. The next PaCT conferences were held in Obninsk (near Moscow), August 30 to September 4, 1993; in St. Petersburg, September 12–15, 1995; in Yaroslavl, September, 9–12, 1997; in Pushkin (near St. Petersburg), September, 6–10, 1999; in Academgorodok (Novosibirsk), September 3–7, 2001; in Nizhni Novgorod, September, 15–19, 2003; in Krasnoyarsk, September 5–9, 2005; in Pereslavl-Zalessky, September 3–7, 2007; in Novosibirsk, August 31 – September 4, 2009; in Kazan, September 19–23, 2011, and in St. Petersburg, September 30 to October 4, 2013. Since 1995 all the PaCT proceedings are published by Springer in the LNCS series. PaCT 2015 was jointly organized by the Institute of Computational Mathematics and Mathematical Geophysics (Russian Academy of Sciences), Novosibirsk State University, Novosibirsk State Technical University, Institute of Applied Mathematical Research (Karelian Research Centre of Russian Academy of Sciences), and Petrozavodsk State University. The aim of the conference is to give an overview of new developments, applications, and trends in parallel computing technologies. We sincerely hope that the conference will help our community to deepen its understanding of parallel computing technologies by providing a forum for an exchange of views between scientists and specialists from all over the world. The conference attracted 87 participants from around the world, with authors from 13 countries submitting papers. Of these, 53 papers were selected for the conference as regular ones; there was also an invited speaker. All the papers were reviewed by at least three referees. Many thanks to our sponsors: the Ministry of Education and Science, Russian Academy of Sciences, and Russian Fund for Basic Research.

September 2015

Victor Malyskin

# HPC Hardware Efficiency for Quantum and Classical Molecular Dynamics

Vladimir V. Stegailov<sup>1,2,3</sup>(✉), Nikita D. Orekhov<sup>1,2</sup>, and Grigory S. Smirnov<sup>1,2</sup>

<sup>1</sup> Joint Institute for High Temperatures of RAS, Moscow, Russia  
stegailov@gmail.com

<sup>2</sup> Moscow Institute of Physics and Technology, Dolgoprudny, Russia

<sup>3</sup> National Research University Higher School of Economics, Moscow, Russia

**Abstract.** Development of new HPC architectures proceeds faster than the corresponding adjustment of the algorithms for such fundamental mathematical models as quantum and classical molecular dynamics. There is the need for clear guiding criteria for the computational efficiency of a particular model on a particular hardware. LINPACK benchmark alone can no longer serve this role. In this work we consider a practical metric of the time-to-solution versus the computational peak performance of a given hardware system. In this metric we compare different hardware for the CP2K and LAMMPS software packages widely used for atomistic modeling. The metric considered can serve as a universal unambiguous scale that ranges different types of supercomputers.

## 1 Introduction

The continuing rapid development of theoretical and computational methods of atomistic simulations during past decades provides a basis of analysis and prediction tools for chemistry, material science, condensed matter physics, molecular biology and nanotechnology. Nowadays molecular dynamics (MD) method that describes motion of individual atoms by the Newton's equations is a research tool of highest importance. The computational speed and the efficiency of parallelization are the main factors that pose limitations on the length and time scales accessible for MD models (the achievable extremes for classical MD are trillions of atoms [4] and milliseconds [12], a typical MD step being 1 fs).

A researcher working in the field of atomistic simulation is an end user of the complex and high performance software and hardware. The main technical question is to find a solution as fast as possible, that is to select appropriate HPC resources and to use them in a most efficient way [14].

In this work we consider a wide-spread type of supercomputer systems comprised of identical nodes and interconnected by a high speed network. Due to the rapid development of hardware, at the moment there is a wide spectrum of node types that can combine several CPUs and accelerators (e.g. GPU, MIC or FPGA). The interconnect architecture spectrum dominated previously by the fat tree and torus topologies has been enriched by the dragonfly and flattened butterfly topologies, the PERCS topology etc.

We can distinguish critical avenues in the development of high performance MD models. Quantum MD (QMD) models demonstrate much higher requirements to the data communication speed and hence to the interconnect properties [5, 7]. The deployment of hybrid architectures for electronic structure calculations and quantum MD is not mature enough. Classical MD (CMD) models are less demanding with respect to data communication. The main limitation in CMD is the computational complexity of interatomic potentials (e.g. [10, 11, 13]) that is determined by the performance of supercomputer nodes. Therefore hybrid architectures of nodes are considered as a major perspective.

## 2 Problem Statement and Benchmarking Metric

Fundamental *mathematical models* (QMD and CMD) are well developed and practically not subjected to changes. HPC *hardware* architectures change quite quickly. Algorithms and *software* couple fundamental *mathematical models* with HPC *hardware*, however they can be adapted to new hardware quite slowly and therefore the role of legacy software is huge. Having in mind the criterion of the “time-to-solution” minimization for particular *mathematical models* we would like to answer the following questions: What *hardware* is more efficient if we use currently available *software*? What is the efficiency of emerging *software* designed for new *hardware*? And how complicated is this *software* development?

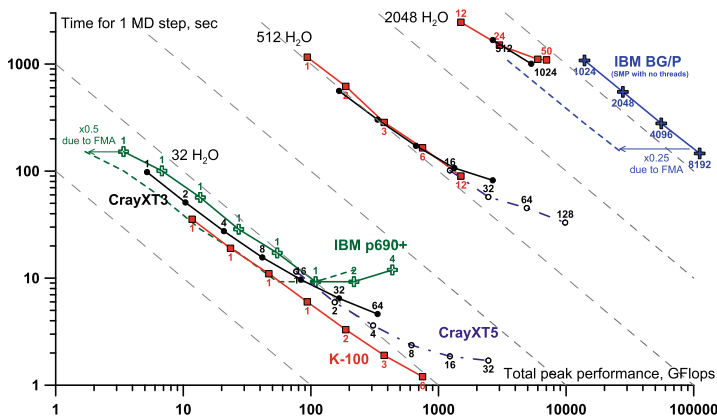
The LINPACK test can not serve as a tool for benchmarking atomistic models. More specialized tests have emerged [1, 6, 9]. Here we use CP2K and LAMMPS codes as representatives of the best HPC atomistic simulation software. Existing benchmarks suites (e.g. [9] and references therein) test the coupling of selected *software* with *hardware* and here we follow this route for QMD. But for CMD we would like to present a wider view: how efficiently *mathematical models* are coupled with *hardware* if we allow *software* to be tuned.

The “time-to-solution” criterion leads us to the evident choice of a time for one MD integration step as one parameter for the metric. The second parameter should characterize the hardware. Usually the number of some abstract processing elements (e.g. cores) is considered. However although this metric serves well in the *weak* and *strong* scaling benchmarks for the given system, it does not allow to compare essentially different hardware. In order to overcome this problem we consider the total peak performance  $R_{peak}$  as a second parameter for the metric that put on equal footing all HPC hardware under consideration. It is in favor of this metric that  $R_{peak}$  is a usual marketing aspect for novel hardware.

## 3 Comparison

Figure 1 shows the comparison for the standard H<sub>2</sub>O benchmark for QMD (CP2K): IBM Regatta 690+ [8], Cray XT3 and XT5 [15], IBM BlueGene/P [2] and K-100 cluster of Keldysh Institute of Applied Mathematics in Moscow (64 nodes connected by Infiniband QDR, each node with 2 six-core Intel Xeon X5670 and 3 NVidia Fermi C2050).





**Fig. 1.** Water model benchmarks with CP2K for various supercomputers (32-2048 water molecules). Numbers show how many nodes are used to run the benchmark. Dashed lines show ideal speed-up  $t \sim R_{peak}^{-1}$ .

For benchmarks with several nodes different supercomputers demonstrate close performance (in seconds per MD step). For large models this agreement is better. In the case of 512 molecules we see that the combination of hardware with compilers provides the same level of efficiency.

The role of the interconnect becomes evident in the multi-node cases where the speed-up worsens. Fat-tree systems show better performance for small model sizes. Torus interconnects of Cray XT3, XT5 and IBM BlueGene/P provides superior strong scaling for large system sizes (in accordance with the detailed analysis for another QMD code SIESTA [3]).

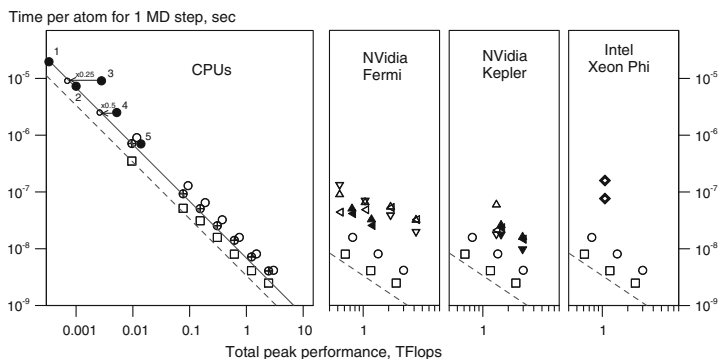
IBM systems show inferior performance in this metric because the fused multiply-add (FMA) operations supported by IBM PowerPC CPUs play no essential role for QMD algorithms.

Figure 2 shows the comparison for the standard Lennard-Jones benchmark for CMD (LAMMPS): pure CPU systems and hybrid systems with NVidia Fermi X5670, NVidia Kepler K40 and Intel Xeon Phi SE10X.

All the data (old benchmarks<sup>1</sup> including) for CPUs without vectorization follow the same trend (with the exception of IBM PowerPC 440 CPU due to the FMA issue mentioned above). Manual vectorization with the USER-INTEL package gives  $\sim 2x$  speed-up. This is the most efficient way among all implemented in LAMMPS to deploy the total peak performance of hardware.

Hybrid nodes with GPUs show inferior timings with respect to CPU-only nodes when compared by the similar  $R_{peak}$ . There are three GPU-oriented versions of MD algorithms in LAMMPS implemented with NVidia CUDA technology (introduced in June 2007). The GPU package is the oldest one introduced in the 1st quarter 2010 and developed up to the 3rd quarter of 2013. The USER-CUDA

<sup>1</sup> <http://lammps.sandia.gov/bench.html>.



**Fig. 2.** Lennard-Jones liquid benchmarks with LAMMPS. Circles show CPU benchmarks without vectorization: open circles and crossed circles show Intel Xeon benchmarks on the “Lomonosov” cluster of Moscow State University and K-100 cluster (their discrepancy illustrate the precision of the metric deployed), black circles are the legacy data: 1 – Pentium II 333 MHz, 2 – DEC Alpha 500 MHz, 3 – PowerPC 440 700 MHz, 4 – Power4 1.3 GHz and 5 – Intel Xeon 3.47 GHz. Boxes correspond to Intel Xeon benchmarks with USER-INTEL. Triangles show the timings from the “Lomonosov” cluster using nodes with NVidia GPUs and different algorithms implemented in LAMMPS:  $\triangle$  – GPU,  $\nabla$  – USER-CUDA,  $\triangleleft$  – KOKKOS. Filled triangles are the benchmarks published on the LAMMPS web-site. The diamonds are the data for Intel Xeon Phi in the native mode (the lower diamond corresponds to the KOKKOS package).

package is a newer one introduced in the 3rd quarter 2011. The KOKKOS package is the most recent introduced in the 2nd quarter 2014 (and it performs essentially better on the novel NVidia Kepler K40).

Nodes with Intel Xeon Phi (an accelerator that became available in 2012–2013) in the *native mode* show more than  $\sim 2x$  speed-up if LAMMPS is used with the KOKKOS package. However Intel Xeon Phi also shows inferior timings with respect to CPU-only nodes when compared by the similar  $R_{peak}$ .

## 4 Conclusions

We introduced a novel metric “time-to-solution (in seconds) vs  $R_{peak}$  (in Flops)” and applied it to representative examples of QMD and CMD. This metric allows us to compare existing HPC hardware, hybrid systems including.

CP2K shows better strong scaling on supercomputers with torus interconnects and especially on IBM BlueGene/P. LAMMPS performs with the best efficiency on Intel Xeon CPUs with manual vectorization of crucial routines. Since MD applications do not use FMA operations IBM PowerPC CPUs perform for these tasks at a fraction of  $R_{peak}$ .

The example of NVidia GPU shows that porting of an existing package on the new hardware takes several years (only after  $\sim 7$  years of development CUDA-based algorithms have approached CPU algorithms efficiency). After  $\sim 3$  years of development classical MD algorithms for Intel Xeon Phi are still not efficient.

**Acknowledgment.** The work is partially supported by the grant No.14-50-00124 of the Russian Science Foundation.

## References

1. Coral benchmark codes. <https://asc.llnl.gov/CORAL-benchmarks/>
2. Bethune, I., Carter, A., Guo, X., Korosoglou, P.: Million atom KS-DFT with CP2K. <http://www.prace-project.eu/IMG/pdf/cp2k.pdf>
3. Corsetti, F.: Performance analysis of electronic structure codes on HPC systems: a case study of SIESTA. *PLoS ONE* **9**(4), e95390 (2014)
4. Eckhardt, W., Heinecke, A., Bader, R., Brehm, M., Hammer, N., Huber, H., Kleinhenz, H.-G., Vrabec, J., Hasse, H., Horsch, M., Bernreuther, M., Glass, C.W., Niethammer, C., Bode, A., Bungartz, H.-J.: 591 TFLOPS multi-trillion particles simulation on SuperMUC. In: Kunkel, J.M., Ludwig, T., Meuer, H.W. (eds.) *ISC 2013*. LNCS, vol. 7905, pp. 1–12. Springer, Heidelberg (2013)
5. Gygi, F.: Large-scale first-principles molecular dynamics: moving from terascale to petascale computing. *J. Phys. Conf. Ser.* **46**(1), 268 (2006)
6. Heroux, M.A., Doerfler, D.W., Crozier, P.S., Willenbring, J.M., Edwards, H.C., Williams, A., Rajan, M., Keiter, E.R., Thornquist, H.K., Numrich, R.W.: Improving performance via mini-applications. Technical report, Sandia Nat. Laboratories (2009)
7. Hutter, J., Curioni, A.: Dual-level parallelism for ab initio molecular dynamics: reaching teraflop performance with the CPMD code. *Parallel Comput.* **31**(1), 1–17 (2005)
8. Krack, M., Parrinello, M.: Quickstep: make the atoms dance. *High Perform. Comput. Chem.* **25**, 29–51 (2004)
9. Muller, M.S., van Waveren, M., Lieberman, R., Whitney, B., Saito, H., Kumaran, K., Baron, J., Brantley, W.C., Parrott, C., Elken, T., Feng, H., Ponder, C.: SPEC MPI2007 – an application benchmark suite for parallel systems using MPI. *Concurrency Comput. Pract. Experience* **22**(2), 191–205 (2010)
10. Orekhov, N.D., Stegailov, V.V.: Graphite melting: atomistic kinetics bridges theory and experiment. *Carbon* **87**, 358–364 (2015)
11. Orekhov, N.D., Stegailov, V.V.: Molecular-dynamics based insights into the problem of graphite melting. *J. Phys.: Conf. Ser.* (2015)
12. Piana, S., Klepeis, J.L., Shaw, D.E.: Assessing the accuracy of physical models used in protein-folding simulations: quantitative evidence from long molecular dynamics simulations. *Curr. Opin. Struct. Biol.* **24**, 98–105 (2014)
13. Smirnov, G.S., Stegailov, V.V.: Toward determination of the new hydrogen hydrate clathrate structures. *J. Phys. Chem. Lett.* **4**(21), 3560–3564 (2013)
14. Stegailov, V.V., Norman, G.E.: Challenges to the supercomputer development in Russia: a HPC user perspective. *Program Systems: Theory and Applications* **5**(1), 111–152 (2014). <http://psta.psiras.ru/read/psta2014.1.111-152.pdf>
15. VandeVondele, J.: CP2K: parallel algorithms. [www.training.prace-ri.eu/uploads/tx\\_pracetmo/cpw09\\_cp2k\\_parallel.pdf](http://www.training.prace-ri.eu/uploads/tx_pracetmo/cpw09_cp2k_parallel.pdf)