GPU-Accelerated Molecular Dynamics: Energy Consumption and Performance

Vyacheslav Vecher[1,2(✉)], Vsevolod Nikolskii[1,3], and Vladimir Stegailov[1]

[1] Joint Institute for High Temperatures of RAS, Moscow, Russia
vecher@phystech.edu,thevsevak@gmail.com,v.stegailov@hse.ru
[2] Moscow Institute of Physics and Technology (State University), Dolgoprudny, Russia
[3] National Research University Higher School of Economics, Moscow, Russia

Abstract. Energy consumption of hybrid systems is an actual problem of modern high-performance computing. The trade-off between power consumption and performance becomes more and more prominent. In this paper, we discuss the energy and power efficiency of two modern hybrid minicomputers Jetson TK1 and TX1. We use the Empirical Roofline Tool to obtain peak performance data and the molecular dynamics package LAMMPS as an example of a real-world benchmark. Using the precise wattmeter, we measure Jetsons power consumption profiles. The effectiveness of DVFS is examined as well. We determine the optimal GPU and DRAM frequencies that give the minimum energy-to-solution value.

Keywords: Nvidia Jetson · LAMMPS · Energy efficiency

1 Introduction

The method of molecular dynamics (MD) is a modern and powerful method of computer simulations allowing to calculate the behavior of millions of atoms. Based on the integration of classical Newton's equation of motion, this method allows one to estimate the evolution of systems consisted of particles that obey the laws of classical mechanics. With the increase in available computation power the size and complexity of models increase as well. Usually, it leads to significant improvements in the accuracy of results. However, the growth of the computational demands has led to a situation when MD simulations are considered among the main tasks for parallel computing.

The technological progress in the development of graphical accelerators makes them quite powerful computation devices with relatively low price. GPUs outperform the conventional processors in the constantly increasing number of computational tasks in the price-to-performance ratio. This trend leads to the situation when the use of graphical accelerators for MD computation has become a common practice [1–9].

However, the increase of power consumption and heat generation of computing platforms is also a very significant problem, especially in connection

V. Voevodin and S. Sobolev (Eds.): RuSCDays 2016, CCIS 687, pp. 78–90, 2016.
DOI: 10.1007/978-3-319-55669-7_7

with the development of exascale systems. Measurement and presentation of the results of performance tests of parallel computer systems become more and more often evidence-based [10], including the measurement of energy consumption [11], which is crucial for the development of exascale supercomputers [12].

The purpose of this work is to evaluate the efficiency of MD algorithms in terms of power consumption on Nvidia Tegra K1 and X1 systems-on-chip (SoCs).

2 Related Work

The power and energy consumption have been under consideration for a long time. For example, we could mention the work [13] that showed the way of lowering the energy consumption of processors by reducing the number of switching operations. Joseph and Martonosi [14] investigated the problem of energy consumption in its relationship with code optimization for 32-bit embedded RISC processors. Russel and Jacome [15] discussed a more complex model of evaluation of power consumption in real-time. The work [16] showed the evaluation of energy consumption at the OS level.

The development of portable devices gave additional impulse to this filed (e.g. see Zhang et al. [17]).

An important aspect of GPGPU technologies that makes them beneficial is the energy efficiency. A lot of efforts have been invested into the low-level models for modeling the energy consumption of GPUs. In recent review [18], the key aspects of accelerator-based systems performance modeling have been discussed. The McPAT model (Multicore Power, Area and Timing) [19] is considered as one of the cornerstone ideas in this area. Another approach called GPUWattch [20] is aimed at the prediction of energy consumption and its optimization through careful tuning on the basis of series of microtests. These approaches make possible to accurately predict the power consumption of CPU and/or GPU with accuracy of the order of 5–10%. However, the use of a specific model of energy consumption (like McPAT or GPUWattch) for new types of hardware and software is a very significant effort to be undertaken. Therefore, direct experimental measurements of power consumption and energy usage are instructive.

The work of Calore et al. [21] discloses some aspects of relations between power consumption and performance for Tegra K1 device running the Lattice Boltzmann method algorithms. Our preliminary results on energy consumption for minicomputers running MD benchmarks have been published previously for Odroid C1 [22] and Nvidia Jetson TK1 and TX1 [23]. In the work of Gallardo et al. [24] one can find the performance analysis and comparison of Nvidia Kepler and Maxwell and Intel Xeon Phi accelerators for the hydrodynamic benchmark LULESH. An energy-aware task management mechanism for the MPDATA algorithms on multicore CPUs was proposed by Rojek et al. [25].

The method of determining the power consumption of large computer systems are constantly improving [12]. The work of Rajovic et al. [26] shows the possible way of designing HPC systems from modern commodity SoCs and presents the energy consumption analysis of the prototype cluster created.

3 Software and Algorithms

3.1 A Peak Load Benchmark: Empirical Roofline Toolkit

The performance of heterogeneous systems can be evaluated in different ways. The consideration of only the theoretical peak performance can be instructive (e.g. see [27,28]) but is not sensitive to details of algorithms. This approach is justified for compute-bound algorithms only. For memory-bound algorithms, the memory bandwidth is to be addressed.

This idea has led to the creation of the Roofline model [29]. The model introduces a special characteristic called "arithmetic intensity". It quantifies the ratio of the number of arithmetic operations to the amount of data transferred. Obviously, the limiting factor for algorithms with large arithmetic intensity is the peak performance of a processor, while the memory bandwidth limits the performance of algorithms with intensive data transfers.

The main outcome of the Roofline model is a graph of performance restrictions for algorithms with different arithmetic intensities. One can estimate the performance of a system under consideration for a particular algorithm from such a roofline plot. For example, the typical arithmetic intensity for Lattice Boltzmann methods is less than one Flops/byte, whether for particle methods this parameter is usually around ten Flops/byte.

One can use the Empirical Roofline Toolkit (ERT) [30] for evaluation of memory bandwidth and peak computing power taking into account memory hierarchy of today's complex heterogeneous computing systems. The core idea of the ERT algorithm consists in performing cycles of simple arithmetic operations on the elements of an array of specified length. The algorithm varies the size of the array and the number of operations on the same element of the array (ERT_FLOPS) in nested loops (Fig. 1). The change of the data array size helps

```
for(int i=0; i<n; ++i){
        if (ERT_FLOPS==1){
                b1 = a[i] + alpha;
        }
        if (ERT_FLOPS==2){
                b1 = a[i]*b1 + alpha;
        }
        if (ERT_FLOPS==4){
                b1 = a[i]*b1 + alpha;
                b2 = a[i]*b2 + alpha;
        }
...
};
```

Fig. 1. An illustration for the ERT_FLOPS parameter

to detect the presence of caches. The change of operations number on one element of array helps to identify automatic vectorization effects.

3.2 Classical Molecular Dynamics: LAMMPS

The LAMMPS package [31] is used in this work as an example of a real-life application. It is a flexible tool for building models of classical MD in materials science, chemistry and biology. LAMMPS is not the only MD package that is ported to the hybrid architecture (for example HOOMD [32] was originally designed with the perspective to run it on GPU accelerators). Two GPU MD algorithms implemented in LAMMPS are considered in this work: USER-CUDA [33] and GPU [34,35].

To evaluate the performance of the hardware available, we use the Lennard-Jones fluid model (108,000 atoms, the density $0.8442\sigma^{-3}$, the cut-off radius 2.5σ, NVE-ensemble, 250 timesteps).

4 Hardware

4.1 Tested Platforms

We consider two different generation of Nvidia Tegra SoCs installed in Jetson TK1 and TX1 platforms. These SoCs consist of several ARM-cores and GPU on a single chip. These platforms are designed for embedded applications (robots, drones) and optimized for minimum power consumption with relatively high performance.

These devices are aimed to be energy effective and usually operate in the dynamic voltage and frequency scaling (DVFS) mode. In this mode, the GPU memory and core frequencies change during the runtime that allows to reduce the power consumption of hardware significantly. In the measurements, we disable the DVFS mode by setting manually the GPU and DRAM frequencies. However, we make several measurements with the DVFS mode enabled.

Nvidia Jetson TK1. Nvidia Jetson TK1 is a developer board based on the 32-bit Tegra K1 SoC with LPDDR3 (930 MHz). Tegra K1 CPU complex includes 4 Cortex-A15 cores running at 2.3 GHz, the 5-th low power companion Cortex core designed to replace the basic cores in the low load mode to reduce power consumption and heat generation. The chip includes one GPU Kepler streaming multiprocessor (SM) running at 852 MHz (128 CUDA cores). Each Cortex-A15 core has 32 KB L1 instruction and 32 KB L1 data caches. 4-core cluster has 2 MB of shared L2 cache.

The program environment of the device consists of Linux Ubuntu 14.04.1 LTS (GNU/Linux 3.10.40-gdacac96 armv7l). The toolchain includes GCC ver. 4.8.4 and CUDA Toolkit 6.5.

Nvidia Jetson TX1. Jetson TX1 is based on the 64-bit Tegra X1 SoC with LPDDR4 memory (1600 MHz). Tegra X1 includes 4 Cortex-A57 cores running at 1.9 GHz, 4 slower Cortex-A53 in big.LITTLE configuration and two GPU Maxwell SMs running at 998 MHz (256 CUDA cores). Each Cortex-A57 core has 48 KB L1 instruction cache, 32 KB L1 data cache and 2 MB of shared L2 cache.

The operation system is Linux Ubuntu 14.01.1 LTS with 64-bit core built for aarch64. Nevertheless we use the 32-bit toolchain and software environment (same as for Nvidia Jetson TK1), except for the newer CUDA Toolkit 7.0.

In summer 2016, the 64-bit userspace and toolchain have been released. Preliminary tests show that the new 64-bit software can be noticeably faster in some rare cases only.

4.2 Energy Consumption Measurement Technique

We use the SmartPower digital wattmeters with the integrated DC source to measure the energy consumption of the Jetson boards. The wattmeter provides voltage in the range from 3 to 5.25 V and measures the current and power consumption every 0.2 s with a nominal error of less than 0.01 V. The wattmeter shows the data on the display in real time and allows to transfer the data via USB to the PC for further analysis.

Because both Jetson platforms have nominal voltage values higher than 5.25 V, we connect several SmartPower wattmeters in a sequential way to achieve higher voltage (see Fig. 2). To confirm the accuracy of the achieved output voltage, we use the precise Tektronix TDS2014C oscilloscope. In this way we show that the average error in the power consumption measurements are about 1%.

The nominal voltages are 12 V for Jetson TK1 and 19 V for Jetson TX1. However, we discover that both devices can operate at much lower voltages: down to 6 V for TK1 and 8 V for TX1.

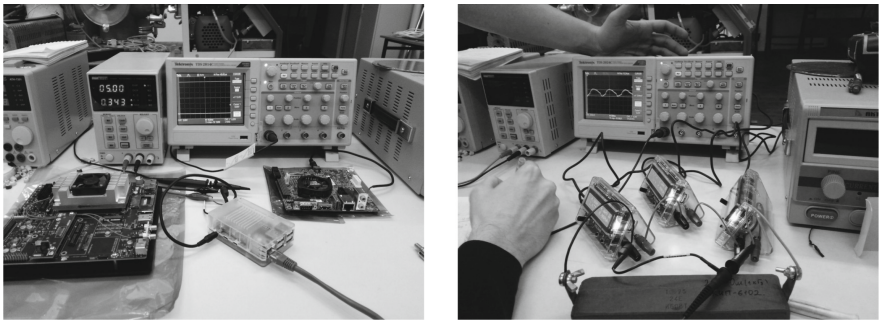

Fig. 2. The Jetson TX1 and TK1 boards with the ODROID-C1 in the center (left photo). The sequentially connected SmartPower wattmeters with the oscilloscope (right photo)

The measurement of energy consumption in a particular test consists in the simultaneous execution of the logging program on the PC as well as the necessary tests on the Jetson. The Jetson boards do not have any connected peripherals except for the LAN cable.

We should note that other methods of measuring power consumption exist as well. For example, built-in hardware counters can be used that allow a user to accurately determine the chip power consumption. However, we are not aware of the similar counters in Tegra SoCs and therefore measure directly the power consumption of the entire development board.

5 Measurements Results

5.1 Energy Consumption for the ERT Benchmark

We use the results of the ERT launches to determine the ratio of the peak performance (in GFlops) to the average power consumption during the benchmark.

On Fig. 3 one can see an example of the power consumption profiles for both Jetson boards. Using the measured value of energy consumed for the benchmark launch and the peak performance obtained, one can determine the energy efficiency of systems considered.

Since the first 10 s of the ERT benchmark are spent on rebuilding the binary, this part of the log is not included in consideration.

The total energy consumption for the GPU single precision ERT benchmark for TK1 is 5.9 W, with the maximum achieved performance in single precision of 209.9 GFlops. This gives us the ratio of 35.5 GFlops/W. For TX1, the energy consumption is 6.28 W, which is slightly above TK1. However, the newer device is significantly superior in terms of achieved maximum performance (485.1 GFlops) that gives a higher performance of 77.2 GFlops/W in single precision.

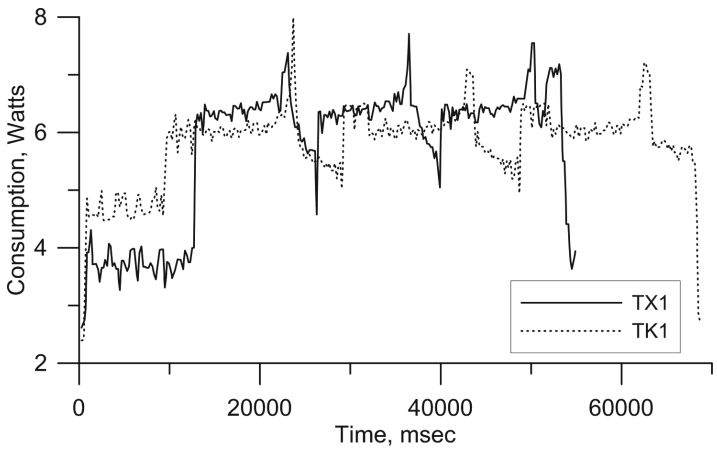

Fig. 3. An example of Jetsons power consumption profiles during ERT launches

Both Jetson minicomputers demonstrate not very impressive results in double precision: 2.1 GFlops/W for TK1 and 2.7 GFlops/W for TX1. The reason for this is the significantly lower double precision performance with the energy consumption level comparable to the single precision case.

On the other hand, the ERT launches on the ARM cores of TX1 show that the Cortex-A57 core has much lower efficiency: 0.8 GFlops/W for double precision and 4 GFlops/W for single precision.

5.2 Energy Consumption for the LAMMPS Benchmark

On Fig. 4, one can see typical power consumption profiles during LAMMPS launches. The area under the graph represents the amount of energy spent for the calculation.

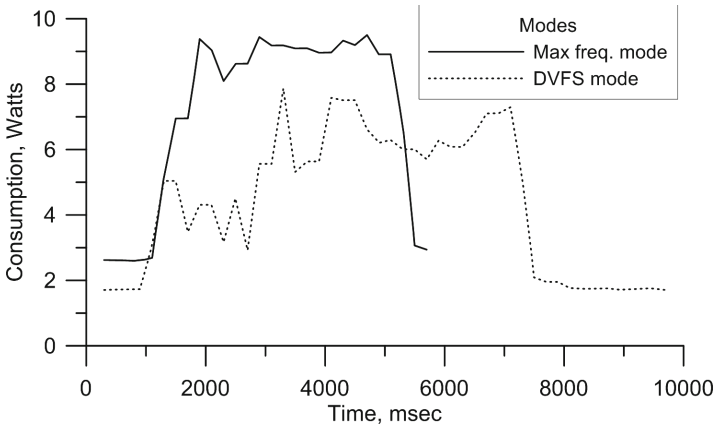

Fig. 4. Power consumption profiles for the launches of LAMMPS with USER-CUDA for TX1 in the maximum frequency mode and in the DVFS mode

As noted above, both Jetson systems support the DVFS energy optimization mode. Therefore, we consider how the results change in the case of the activated DVFS mode. Thus, for each launch of the standard Lennard-Jones benchmark in the fixed maximum frequency mode, we have performed the same launch but with the DVFS mode enabled.

Figure 4 shows the comparison of the LAMMPS energy consumption in DVFS and fixed maximum frequencies modes. One can see that the DVFS-enabled launch takes more time with clearly lower power consumption level.

To answer the question whether or not DVFS is beneficial, we calculate the consumed energy per one LAMMPS launch with and without DFVS.

The results presented on Fig. 5 show that the total energy consumption values for the LAMMPS calculations with the DVFS mode enabled are roughly equal or higher than the corresponding values in the case of the DVFS mode disabled. However, the times-to-solution with DVFS are much higher than in the case

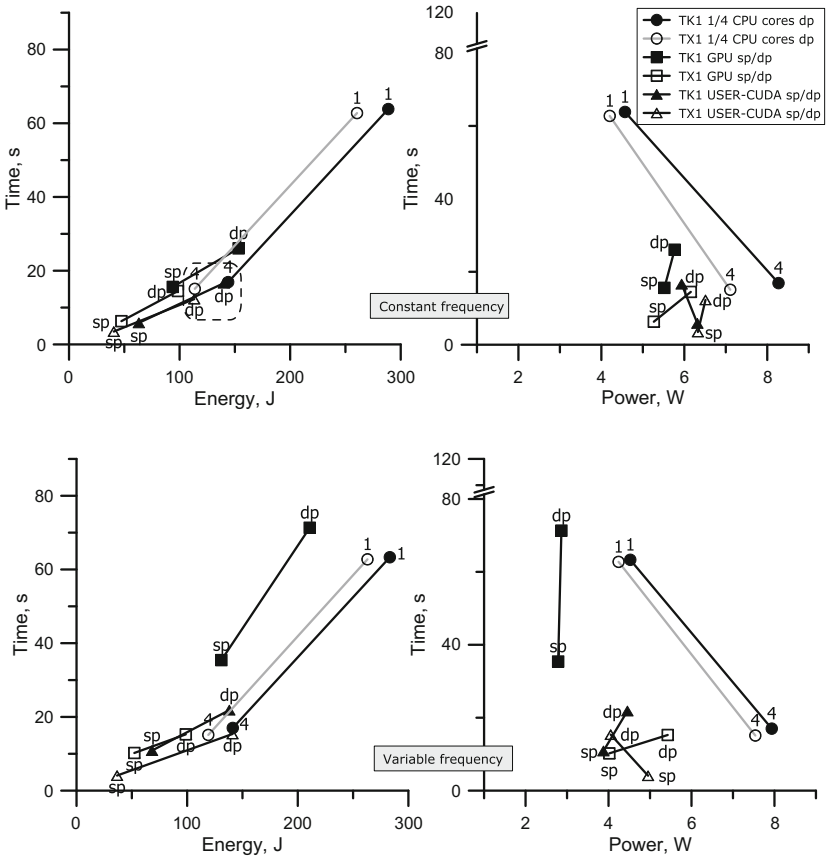

Fig. 5. Time-to-solution and the corresponding energy consumption values for the CPU MD algorithm, GPU and USER-CUDA variants in the maximum frequency mode and in the DVFS mode (single and double precision)

of the fixed maximum frequency. Therefore, the usage of DVFS in the cases considered does not improve energy efficiency.

6 GPU and DRAM Frequencies Variation Effect

We use the USER-CUDA accelerated LAMMPS variant as a benchmark that shows the effect of GPU and DRAM frequencies variation on the execution time and the total energy consumption. The frequency of the GPU are changed from one launch to another and the DRAM frequency is fixed for the whole group of launches. For each group of experiments, we set the Jetson TK1 GPU frequency to the following values (in MHz): 72, 108, 180, 252, 324, 396, 468, 540, 612, 648, 684, 708, 756, 804 and 852. The DRAM frequencies are fixed for each group at the values of 924, 396, 204 and 102 MHz. For each launch, the power consumption is measured.

One can find the measurement results in Fig. 6. Initially, the increase of GPU frequency is accompanied by the decrease in times-to-solution. In terms of energy consumption, the situation is different. With the increase of GPU frequency, the TK1 energy consumption decreases down to a certain limit and reaches its minimum. After that, any increase of the GPU frequency leads to the increase of energy consumption.

This minimum of power consumption is associated with the transition of the LAMMPS USER-CUDA algorithm from the compute-bound mode to the memory-bound mode. The GPU computational speed limits the total performance of the system at low GPU frequencies. This situation corresponds to the compute-bound mode. On the other hand, the DRAM memory bandwidth limits the total performance at high GPU frequencies. This situation corresponds to the memory-bound mode. An increased energy consumption in the latter case is

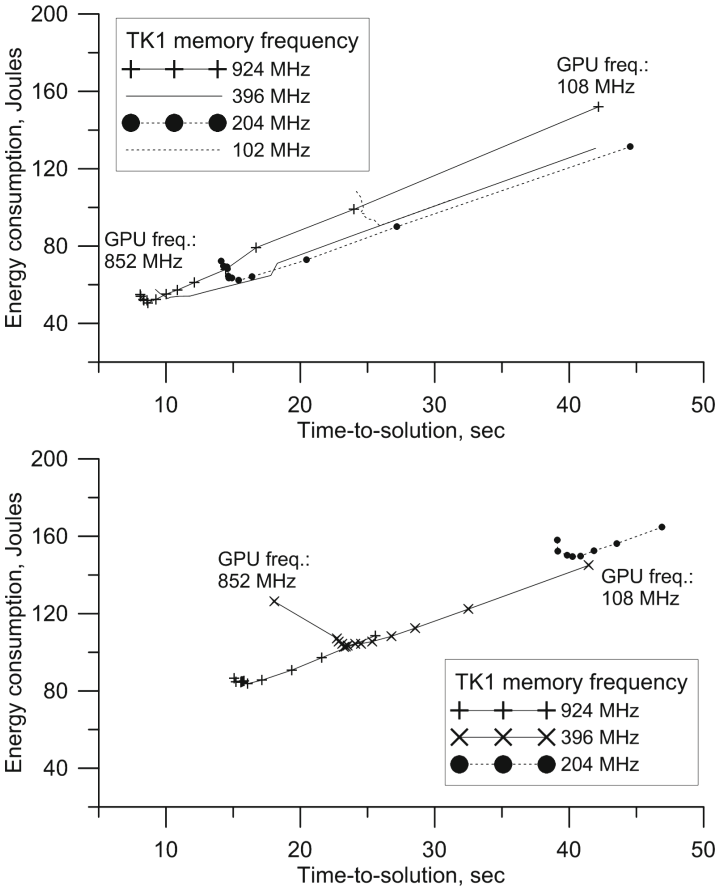

Fig. 6. LAMMPS power consumption on different TK1 memory frequencies with USER-CUDA (upper plot) and GPU (lower plot) packages.

not a desirable effect because this growth of consumption is not associated with any significant speedup of the calculation.

Also, we notice that lowering the DRAM memory frequency shifts the point of minimum consumed energy to lower GPU frequencies, as it might be expected.

7 Summary

We have described the energy consumption of the minicomputers Nvidia Jetson TK1 and TX1 based on the hybrid systems-on-chip Nvidia Tegra K1 and X1.

The peak load benchmarks have been performed with the Empirical Roofline Toolkit (with the CPU and GPU versions). The CPU version has shown 4 GFlops/W for single precision and 0.8 GFlops/W for double precision for one Cortex-A57 core of Jetson TX1. The GPU version for Kepler TK1 and Maxwell TX1 has shown 35.5 GFlops/W and 77.2 GFlops/W respectively for single precision and 2.1 GFlops/W and 2.7 GFlops/W for double precision.

Two CUDA-accelerated MD algorithms implemented in LAMMPS have been used for energy consumption benchmarks (in single and double precision). DVFS has been found inefficient for energy efficiency improvement in the cases considered.

By changing GPU and DRAM frequencies on TK1, we have shown the transition of the both CUDA-based MD algorithms from the compute-bound to memory-bound mode. We have located the minima of the energy-to solution with respect to the set of GPU and DRAM frequencies considered.

In the future, we plan to conduct a similar analysis for systems with desktop or server GPU accelerators. In addition, we are going to consider the case of more complex molecular dynamics models, e.g. with the Coulomb interaction.

Acknowledgments. HSE and MIPT provided funds for purchasing the hardware used in this study. The work was supported by the grant No. 14-50-00124 of the Russian Science Foundation.

References

1. Morozov, I., Kazennov, A., Bystryi, R., Norman, G., Pisarev, V., Stegailov, V.: Molecular dynamics simulations of the relaxation processes in the condensed matter on GPUs. Comput. Phys. Commun. **182**(9), 1974–1978 (2011). doi:10.1016/j.cpc.2010.12.026
2. Budea, A., Derzsi, A., Hartmann, P., Donko, Z.: Shear viscosity of liquid-phase yukawa plasmas from molecular dynamics simulations on graphics processing units. Contrib. Plasma Phys. **52**(3), 194–198 (2012). doi:10.1002/ctpp.201100083
3. French, W.R., Pervaje, A.K., Santos, A.P., Iacovella, C.R., Cummings, P.T.: Probing the statistical validity of the ductile-to-brittle transition in metallic nanowires using GPU computing. J. Chem. Theory Comput. **9**(12), 5558–5566 (2013). doi:10.1021/ct400885z

4. Fu, H., Zheng, L., Yang, M.: Accelerating modified shepard interpolated potential energy calculations using graphics processing units. Comput. Phys. Commun. **184**(4), 1150–1154 (2013). doi:10.1016/j.cpc.2012.12.005
5. Wu, Q., Yang, C., Tang, T., Xiao, L.: MIC acceleration of short-range molecular dynamics simulations. In: Proceedings of the First International Workshop on Code OptimiSation for MultI and Many Cores, COSMIC 2013, pp. 2:1–2:8. ACM, New York (2013). doi:10.1145/2446920.2446922
6. Wu, Q., Yang, C., Tang, T., Xiao, L.: Exploiting hierarchy parallelism for molecular dynamics on a petascale heterogeneous system. J. Parallel Distrib. Comput. **73**(12), 1592–1604 (2013). doi:10.1016/j.jpdc.2013.07.015
7. Filho, T.M.R.: Molecular dynamics for long-range interacting systems on graphic processing units. Comput. Phys. Commun. **185**(5), 1364–1369 (2014). doi:10.1016/j.cpc.2014.01.008
8. Minkin, A.S., Knizhnik, A.A., Potapkin, B.V.: GPU implementations of some many-body potentials for molecular dynamics simulations. Adv. Eng. Softw. (2016). doi:10.1016/j.advengsoft.2016.05.013
9. Nguyen, T.D.: GPU-accelerated Tersoff potentials for massively parallel molecular dynamics simulations. Comput. Phys. Commun. **212**, 113–122 (2017). doi:10.1016/j.cpc.2016.10.020
10. Hoefler, T., Belli, R.: Scientific benchmarking of parallel computing systems: twelve ways to tell the masses when reporting performance results. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2015, pp. 73:1–73:12. ACM, New York (2015). doi:10.1145/2807591.2807644
11. Pruitt, D.D., Freudenthal, E.A.: Preliminary investigation of mobile system features potentially relevant to HPC. In: Proceedings of the 4th International Workshop on Energy Efficient Supercomputing, E2SC 2016, pp. 54–60. IEEE Press, Piscataway (2016). doi:10.1109/E2SC.2016.13
12. Scogland, T., Azose, J., Rohr, D., Rivoire, S., Bates, N., Hackenberg, D.: Node variability in large-scale power measurements: perspectives from the Green500, Top500 and EEHPCWG. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2015, pp. 74:1–74:11. ACM, New York (2015). doi:10.1145/2807591.2807653
13. Su, C.L., Tsui, C.Y., Despain, A.M.: Low power architecture design and compilation techniques for high-performance processors. In: Compcon Spring 1994, Digest of Papers, pp. 489–498 (1994). doi:10.1109/CMPCON.1994.282878
14. Joseph, R., Martonosi, M.: Run-time power estimation in high performance microprocessors. In: Proceedings of the 2001 International Symposium on Low Power Electronics and Design, ISLPED 2001, pp. 135–140. ACM, New York (2001). doi:10.1145/383082.383119
15. Russell, J.T., Jacome, M.F.: Software power estimation and optimization for high performance, 32-bit embedded processors. In: Proceedings International Conference on Computer Design. VLSI in Computers and Processors (Cat. No. 98CB36273), pp. 328–333 (1998). doi:10.1109/ICCD.1998.727070
16. Li, T., John, L.K.: Run-time modeling and estimation of operating system power consumption. SIGMETRICS Perform. Eval. Rev. **31**(1), 160–171 (2003). doi:10.1145/885651.781048

17. Zhang, L., Tiwana, B., Qian, Z., Wang, Z., Dick, R.P., Mao, Z.M., Yang, L.: Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In: Proceedings of the Eighth IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, CODES/ISSS 2010, pp. 105–114. ACM, New York (2010). doi:10.1145/1878961.1878982
18. Lopez-Novoa, U., Mendiburu, A., Miguel-Alonso, J.: A survey of performance modeling and simulation techniques for accelerator-based computing. IEEE Trans. Parallel Distrib. Syst. **26**(1), 272–281 (2015). doi:10.1109/TPDS.2014.2308216
19. Li, S., Ahn, J.H., Strong, R.D., Brockman, J.B., Tullsen, D.M., Jouppi, N.P.: McPat: an integrated power, area, and timing modeling framework for multicore and manycore architectures. In: Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture, MICRO 42, pp. 469–480. ACM, New York (2009). doi:10.1145/1669112.1669172
20. Leng, J., Hetherington, T., ElTantawy, A., Gilani, S., Kim, N.S., Aamodt, T.M., Reddi, V.J.: GPUWattch: enabling energy optimizations in GPGPUs. SIGARCH Comput. Archit. News **41**(3), 487–498 (2013). doi:10.1145/2508148.2485964
21. Calore, E., Schifano, S.F., Tripiccione, R.: Energy-performance tradeoffs for HPC applications on low power processors. In: Hunold, S., et al. (eds.) Euro-Par 2015. LNCS, vol. 9523, pp. 737–748. Springer, Heidelberg (2015). doi:10.1007/978-3-319-27308-2_59
22. Nikolskiy, V., Stegailov, V.: Floating-point performance of ARM cores and their efficiency in classical molecular dynamics. J. Phys.: Conf. Ser. **681**(1), 012049 (2016). http://stacks.iop.org/1742-6596/681/i=1/a=012049
23. Nikolskiy, V.P., Stegailov, V.V., Vecher, V.S.: Efficiency of the Tegra K1 and X1 systems-on-chip for classical molecular dynamics. In: 2016 International Conference on High Performance Computing Simulation (HPCS), pp. 682–689 (2016). doi:10.1109/HPCSim.2016.7568401
24. Gallardo, E., Teller, P.J., Argueta, A., Jaloma, J.: Cross-accelerator performance profiling. In: Proceedings of the XSEDE16 Conference on Diversity, Big Data, and Science at Scale, XSEDE16, pp. 19:1–19:8. ACM, New York (2016). doi:10.1145/2949550.2949567
25. Rojek, K., Ilic, A., Wyrzykowski, R., Sousa, L.: Energy-aware mechanism for stencil-based MPDATA algorithm with constraints. Concurr. Comput.: Pract. Exp. (2016). doi:10.1002/cpe.4016
26. Rajovic, N., Rico, A., Mantovani, F., Ruiz, D., Vilarrubi, J.O., Gomez, C., Backes, L., Nieto, D., Servat, H., Martorell, X., Labarta, J., Ayguade, E., Adeniyi-Jones, C., Derradji, S., Gloaguen, H., Lanucara, P., Sanna, N., Mehaut, J.F., Pouget, K., Videau, B., Boyer, E., Allalen, M., Auweter, A., Brayford, D., Tafani, D., Weinberg, V., Brömmel, D., Halver, R., Meinke, J.H., Beivide, R., Benito, M., Vallejo, E., Valero, M., Ramirez, A.: The Mont-Blanc prototype: an alternative approach for HPC systems. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2016, pp. 38:1–38:12. IEEE Press, Piscataway (2016). http://dl.acm.org/citation.cfm?id=3014904.3014955
27. Stegailov, V.V., Orekhov, N.D., Smirnov, G.S.: HPC hardware efficiency for quantum and classical molecular dynamics. In: Malyshkin, V. (ed.) PaCT 2015. LNCS, vol. 9251, pp. 469–473. Springer, Heidelberg (2015). doi:10.1007/978-3-319-21909-7_45

28. Smirnov, G.S., Stegailov, V.V.: Efficiency of classical molecular dynamics algorithms on supercomputers. Math. Models Comput. Simul. **8**(6), 734–743 (2016). doi:10.1134/S2070048216060156
29. Williams, S., Waterman, A., Patterson, D.: Roofline: an insightful visual performance model for multicore architectures. Commun. ACM **52**(4), 65–76 (2009). doi:10.1145/1498765.1498785
30. Lo, Y.J., Williams, S., Straalen, B., Ligocki, T.J., Cordery, M.J., Wright, N.J., Hall, M.W., Oliker, L.: Roofline model toolkit: a practical tool for architectural and program analysis. In: Jarvis, S.A., Wright, S.A., Hammond, S.D. (eds.) PMBS 2014. LNCS, vol. 8966, pp. 129–148. Springer, Heidelberg (2015). doi:10.1007/978-3-319-17248-4_7
31. Plimpton, S.: Fast parallel algorithms for short-range molecular dynamics. J. Comput. Phys. **117**(1), 1–19 (1995). doi:10.1006/jcph.1995.1039
32. Glaser, J., Nguyen, T.D., Anderson, J.A., Lui, P., Spiga, F., Millan, J.A., Morse, D.C., Glotzer, S.C.: Strong scaling of general-purpose molecular dynamics simulations on GPUs. Comput. Phys. Commun. **192**, 97–107 (2015). doi:10.1016/j.cpc.2015.02.028
33. Trott, C.R., Winterfeld, L., Crozier, P.S.: General-purpose molecular dynamics simulations on GPU-based clusters. arXiv e-prints (2010). http://arxiv.org/abs/1009.4330
34. Brown, W.M., Wang, P., Plimpton, S.J., Tharrington, A.N.: Implementing molecular dynamics on hybrid high performance computers – short range forces. Comput. Phys. Commun. **182**(4), 898–911 (2011). doi:10.1016/j.cpc.2010.12.021
35. Brown, W.M., Kohlmeyer, A., Plimpton, S.J., Tharrington, A.N.: Implementing molecular dynamics on hybrid high performance computers – particle-particle particle-mesh. Comput. Phys. Commun. **183**(3), 449–459 (2012). doi:10.1016/j.cpc.2011.10.012