PAPER • OPEN ACCESS

Floating-point performance of ARM cores and their efficiency in classical molecular dynamics

To cite this article: V Nikolskiy and V Stegailov 2016 J. Phys.: Conf. Ser. 681 012049

View the article online for updates and enhancements.

You may also like

- Roadmap on electronic structure codes in the exascale era Vikram Gavini, Stefano Baroni, Volker Blum et al.
- Evidence for Asteroid Scattering and Distal Solar System Solids From Meteorite Paleomagnetism James F. J. Bryson, Benjamin P. Weiss, Eduardo A. Lima et al.
- <u>Opportunities and choice in a new vector</u> era A Nowak





DISCOVER how sustainability intersects with electrochemistry & solid state science research



This content was downloaded from IP address 3.137.185.180 on 29/04/2024 at 02:36

Floating-point performance of ARM cores and their efficiency in classical molecular dynamics

V Nikolskiv^{1,2}, V Stegailov^{2,1}

¹National Research University Higher School of Economics, 20 Myasnitskaya ulitsa, Moscow 101000, Russia

²Joint Institute for High Temperatures of Russian Academy of Sciences (JIHT RAS), Izhorskaya st. 13 Bd. 2, Moscow 125412, Russia

Abstract. Supercomputing of the exascale era is going to be inevitably limited by power efficiency. Nowadays different possible variants of CPU architectures are considered. Recently the development of ARM processors has come to the point when their floating point performance can be seriously considered for a range of scientific applications. In this work we present the analysis of the floating point performance of the latest ARM cores and their efficiency for the algorithms of classical molecular dynamics.

1. Introduction

The development of computer hardware of the last decades followed the Moore's law that the number of transistors in a dense integrated circuit doubled every one and a half or every two years. Nowadays the development of the circuit technology has nearly reached its physical limits and the further increase of performance is not based on the growth of integrated circuits, but on the supercomputers that combine huge numbers of processing elements. It is estimated that supercomputers of the exaflops era are going to consist of millions of computing elements. Energy efficiency is essential for creation of such systems. Priorities in hardware development are shifted from creating faster computing elements to creating more energy efficient elements that can serve as building blocks for large systems and to creating an appropriate interconnect. A typical example is the development of IBM Blue Gene systems, the first of which was based on a notoriously weak processor with a clock speed of only 700 MHz [1].

GPU accelerators are an important current hardware trend in high performance computing (HPC). Initially this technology was focused on the mass market (video games accelerators). An attractive "price-to-performance" ratio ensured the adaptation of such hardware to HPC.

Another similar trend is a mass use of ARM processors in small-scale computing machinery and especially in smartphones and tablets [2]. Market demands determine the growth of performance of ARM CPUs while maintaining the high level of energy efficiency. In the latest family of ARM processors it became possible to use a special module for floating point calculations that opened a way to HPC related tasks [3]. In this paper, we analyze the performance of several examples of ARM Cortex-A processors family both in the terms of peak and Linpack performance and in a test case of classical molecular dynamics (MD).

Nowadays the MD method has an important place among the high-performance computing applications [4]. For example, the projects connected with classical MD correspond to about

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution $(\mathbf{\hat{H}})$ (cc of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI. Published under licence by IOP Publishing Ltd 1

1/5 of the total number of projects and of the allotted computational time in the Innovative and Novel Computational Impact on Theory and Experiment (INCITE) program of the US Department of Energy [1].

Along with the development of theoretical foundations of the MD method [5], there is a rapid growth of the number of its applications in modern multiscale models based on the description of processes at atomistic level in physics, chemistry, biology, materials science and other areas. Examples of new parallel programming approaches for MD appear constantly (e.g. [6]). However the modern record model size of trillions of particles [7] corresponds, for example, to only a few μm^3 of metal at normal density. The increase of the maximum MD simulation time is even a more complicated problem [8]. Expanding the limitations of MD methods is inextricably connected with advancement in HPC technology.

The plan of this paper is as follows. The second section describes the hardware and the software used to test the performance of ARM systems. In the third section we consider the Linpack-like performance using the RGbenchMM test and the peak performance of ARM cores. The fourth section brings together the results of a test MD problem for different CPU types, including our results for ARM Cortex-A5 CPU. In the fifth section we compare power consumption of Intel-based and ARM-based systems on the example of an MD test.

2. Hardware and software details

The ARM architecture is a microprocessor architecture with a reduced instructions set (Advanced RISC Machine). This architecture is subdivided into different types of cores. We consider the Cortex-A family and Cortex-A5, Cortex-A9 and Cortex-A15 cores. These names do not defined CPUs types strictly, but they are licensed patterns used by different vendors for (often customized) CPU designs (e.g., Scorpion and Krait cores). All types of cores considered in this work include the floating point module (VFPv3 or VFPv4).

The main results of the work are obtained with ODROID-C1. This is a portable minicomputer made by Hardkernel co. Ltd., which contains a quad core processor Amlogic S805 Cortex-A5 with VFPv4 modules on each core and a graphics accelerator Mali-450 MP2 that we do not use. The processor operates at 1.5 GHz with 1 GB of DDR3 SDRAM. The device has eMMC and microSD slots, 4 USB ports and 10/100/1000 Mbps Ethernet with a RJ-45 port. Power is connected via a special adapter. The monitor is connected via microHDMI port. We use ODROID-C1 either with Linux Ubuntu 14.05.1 LTS OS (odroid 3.10.67-55 #1 SMP) with the LXDE lightweight graphical environment, or with Android 4.4.2 OS. Several different smartphones running Android OS are used to test the performance of other types of ARM Cortex cores. Analysis of energy consumption of the minicomputer ODROID-C1 is carried out with a digital watt-meter "Smart Power" made by Hardkernel co. Ltd. The previously benchmarks and energy consumption tests are used for the analysis as well.

The tests of the performance of microprocessors have a long history of development [9]. The *de facto* standard unit for describing the performance in the field of scientific computing and mathematical modeling is the number of floating-point operations per second (flop per second or flops). The measurement of this property is usually associated with the Linpack test. However the ARM architecture was originally focused on integer operations, and the peak performance of ARM cores in terms of flops is not declared by vendors (however, as shown below, it can be estimated by specific tests).

LAMMPS [10] and the Lennard-Jones fluid model are used for CPU performance tests in classical MD algorithms (32K atoms at the density 0.8442 σ^{-3} , the potential cut-off radius 2.5 σ that gives about 55 neighbors per atom, 100 time steps with the NVE integration scheme).

The GCC compiler ver. 4.9 is used for the compilation of the LAMMPS under Linux OS. The -mcpu-cortexa5 -mfpu=vfpv4-d16 -mfloat-abi=softfp compiler options are used (performance differences from -mfloat-abi=hard are negligible). The serial compilation without

 R_{max} , MFlops/core Flop/cycle ARM-core Source ODROID-C1 (this work) Cortex-A5 2840.20.1[11] (Google Nexus S) Cortex-A8 66 Cortex-A9 3720.3Google Nexus 7 Cortex-A9 460, 614 0.4Samsung Galaxy S II X, Note II Cortex-A9 352 - 6550.2 - 0.4[11] (8 devices) 575, 588 Scorpion Samsung Galaxy S Plus, S II 0.4Scorpion 446 - 772 [11] (4 devices) 0.4 - 0.5Krait 613 - 896 0.4 - 0.5[11] (14 devices) 1073 Sony Xperia Z2 Krait 400 0.5Krait 400 1038 - 1197 0.5[11] (11 devices) Cortex-A15 11250.7Samsung Nexus 10 Cortex-A15 1164 - 1502 0.6 - 0.7 [11] (4 devices)

multithreading is used for the execution on a single Cortex-A5 core.

Table 1. The results of the RGBenchMM test on different devices with ARM processors

3. Estimation of peak performance for ARM cores

Unlike Intel, AMD and IBM, the developers and vendors of ARM processors do not publish the information about the peak performance in flops. According to recent studies [12] it takes 1-2 cycles for the Cortex-A9 core to perform one flop, while Cortex-A15 cores can make in 1 cycle all operations including fused multiply-add or fused multiply-accumulate (FMA) operations.

The RGBenchMM test application [13], based on the implementation of the DGEMM-type procedures with the possibility of parallelizing into 2 or 4 threads is available for Android OS. This application was written in C++ in the Android NDK and is devoid of the understated performance inherent to Java programs running Android OS. In particular, the executable code uses special FMA instructions in the key operations [14]. The data on the absolute performance of R_{max} in MFlops at a single core and on the relative performance in flop units per one processor cycle is presented in the table 1 (with respect to the CPU clock frequency).

The table shows that, unlike the data mentioned above [12] obtained for peak performance R_{peak} in specialized tests, the RGBenchMM test performance R_{max} of Cortex-A9 and Cortex-A15 cores is about half lower than R_{peak} . This is due to the fact that about one half of the cycles in the RGBenchMM test is used for data transfers. We see that these effects are even more pronounced for "weaker" cores like Cortex-A5 and A8. Based on these data, it seems reasonable to assume that the peak performance of the Cortex-A5 core equals 2-3 cycles per flop (taking into account the FMA operations) or 4-6 cycles per flop (excluding FMA).

4. Comparison of CPU cores using the LAMMPS MD test

From the perspective of HPC hardware end-users processors differ in peak performance R_{peak} measured in flops. Broadly speaking, the consumer expects a time-to-solution to be inversely proportional to R_{peak} :

$$t \sim \frac{1}{R_{peak}}.$$
(1)

In parallel computing the possibility of decreasing the calculation time is limited by many factors (algorithm parallelization, communication etc.) However, even for a sequential algorithm the





Figure 1. The comparison of LAMMPS performance for the Lennard-Jones liquid model on different CPUs.

The ODROID-C1 power Figure 2. consumption when running 4 copies of LAMMPS with 4 identical MD tasks.

relation (1) may not hold for a particular task and different CPU types because of different data transfer rates and other factors (cache, compiler, operating system) as well as the use of a special kind of operations like FMA that make the peak performance value "task specific" [4].

A reasonable "common denominator" is needed for the comparison of different types of CPUs (as well as heterogeneous computing elements). This role can be naturally played by the total peak performance R_{peak} [4]. Figure 1 shows the times for 1 MD time step per 1 atom of the Lennard-Jones fluid model.

The benchmark results for one core (black circles) from the LAMMPS website [10] are shown for the Intel Pentium II Over-Drive 333 MHz, DEC Alpha 500 MHz, PowerPC 440 700 MHz, Power4 1.3 GHz and Intel Xeon 3.47 GHz processors. In this case, the calculation time is determined by the structure of the microprocessor core and the ability of the compiler to create an efficient executable code. All the points (except for a square) correspond to the same LAMMPS code written in C++, but to different compilers. The results for the Intel and DEC processors are in a good agreement (1). If we scale the peak performance of IBM Power CPUs to exclude the FMA operations that are not used in the classical MD algorithm then the IBM Power points move to the common dependence $6.84 \cdot 10^3$ Flop/ R_{peak} pretty well (the solid line at figure 1).

The results of the tests [4] made on the supercomputer "Lomonosov" of the Lomonosov Moscow State University without vectorization of the code (a purple circle) and on the supercomputer MVS-10P of JSCC RAS without (an orange circle) and with (an orange square) vectorization of the code are presented. In this case LAMMPS is compiled by Intel C++ compiler with manual vectorization, which is oriented at the Intel Xeon architecture (the USER-INTEL module in LAMMPS). This kind of code optimization results in acceleration of calculations in 2 times corresponding to $3.36 \cdot 10^3$ Flop/ R_{peak} (the dotted line in figure 1).

As mentioned above the ARM Cortex-A5 R_{peak} value is not declared by the manufacturer. Rectangles at figure 1 show the results of the LAMMPS test with one Cortex-A5 core. Two rectangles illustrate the uncertainty of its peak performance (either corrected for FMA operations or not), and the uncertainty of time values (the 4-core Amlogic S805 CPU with 1 copy of

International Conference on Computer Simulation in Physics and Beyon	nd 2015	IOP Publishing
Journal of Physics: Conference Series 681 (2016) 012049	doi:10.1088/1	742-6596/681/1/012049

LAMMPS running gives 5.95 μ s/atom/step and 7.8 μ s/atom/step with 4 copies).

It is evident that the Cortex-A5 architecture together with the compiler provides a highly efficient use of hardware resources for floating-point operations. The calculation time values at the given peak performance follow the same trend as Intel Xeon CPUs with manual vectorization (the USER-INTEL module in LAMMPS). However, in the case of Cortex-A5 no manual tuning of the LAMMPS code is required!

The value of the Cortex-A15 peak performance is known better [12]. An estimate of the calculation time for the LAMMPS test considered and a system similar to Samsung Nexus 10 can be performed by rescaling the results for ODROID-C1 using the R_{max} from table 1. The figure 1 shows that we can expect a slightly higher performance for Cortex-A15 in comparison with IBM Power4.

Table 2. The energy consumption in the Lennard-Jones liquid test calculations (for 1 atom and 1 MD integration step) at the full load of 4 cores of a server/minicomputer. The benchmark data for Intel Xeon servers are used (for Ivy Bridge and Haswell).

System Power, W	Dowor W	Time per atom	Energy per atom
	Tower, w	per step, $\mu \mathbf{s}$	$\mathbf{per} \ \mathbf{step}, \ \mu \mathbf{J}$
Ivy Bridge Server	316 [15]	0.053	16.8
Ivy Bridge Server	451.8 [16]	0.029	13.1
Haswell Server	462.9 [16]	0.027	12.5
ODROID-C1	2.4	1.75	4.2
Equivalent of Samsung Nexus 10	$\sim 8 \ [17]$	0.4	3.2

5. Comparison of energy efficiency

Comparison of different computer systems by energy efficiency is a significant challenge. Usually the power consumption of a processor itself is under consideration. It seems to be more reasonable to consider a power consumption of an entire system (a server, a minicomputer, a smartphone). In this work we compare the energy consumption of several servers under conditions on the full CPU load by the Lennard-Jones liquid MD test.

Figure 2 displays power consumption changes during sequential executions of 4 independent MD test calculations. Results show that the ODROID-C1 power consumption at the full load can be estimated as 2.4 W (subtracting the power consumed by microHDMI and USB ports).

According to the data of Abdurachmanov et al. [17] it is possible to determine the power consumption at the full load of the ODROID XU+E minicomputer based on four Cortex-A15 cores (and probably overestimated due to the presence of idle Cortex-A7 cores on the same CPU chip). At the same time there are the data on the energy consumption of the servers based on Intel Xeon processors with the Ivy Bridge and Haswell architectures [15,16]. It allows us to estimate the amount of energy required to calculate 1 MD step for 1 atom for the MD test considered (table 2). The calculation times are estimated using a universal dependence (figure 1) of the peak performance without consideration of FMA operations (128 GFlops for E5-2650 v2, 240 GFlops for E5-2690 v2 and 249.6 GFlops for E5-2690 v3). The calculation time for ODROID-C1 with the use of all 4 cores is 7 μ s (in correspondence to figure 1). Table 2 shows an estimate for a system with a quad-core Cortex-A15 processor as well.

We see that the calculations of MD tests on systems with ARM Cortex-A processors require approximately 5 times less energy than the calculations on the latest models of Intel processors.

6. Conclusions

We have analyzed the results of the RGBenchMM test (similar to Linpack) for a variety of systems based on ARM Cortex-A CPUs including smartphones and the minicomputer ODROID-C1. Floating point peak performance of Cortex-A5 cores in ODROID-C1 has been estimated.

We have compiled the LAMMPS for ODROID-C1 and performed classical MD benchmarks. We have used the metric "calculation time vs. peak performance". By comparison with other CPUs we have shown the efficient use of floating point hardware capabilities of the Cortex-A architecture in combination with the GCC 4.9 compiler. The Cortex-A5 architecture provides 2 times faster calculations with LAMMPS than others CPU architectures at the same level of peak performance and without the need for manual tuning of the code.

The power consumption of ODROID-C1 has been measured. ARM Cortex-A cores have been found to be 3-5 times more energy efficient for the MD benchmark considered than Intel Xeon Ivy Bridge and Haswell cores.

Acknowledgments

The research (of Stegailov) was supported by the Basic Research Program at the National Research University HSE, Russia in 2015.

References

- Stegailov V V and Norman G E 2014 Program Systems: Theory and Applications 5 111 152 ISSN 2079-3316 URL http://psta.psiras.ru/read/psta2014_1_111-152.pdf
- [2] An interview with Steve Furber URL http://cacm.acm.org/magazines/2011/5/107684-an-interview-with-steve-furber/fulltext
- [3] Mitra G, Johnston B, Rendell A, McCreath E and Zhou J 2013 Use of simd vector operations to accelerate application code performance on low-powered arm and intel platforms *Parallel and Distributed Processing* Symposium Workshops PhD Forum (IPDPSW), 2013 IEEE 27th International pp 1107–1116
- [4] Stegailov V V, Orekhov N D and Smirnov G S 2015 Hpc hardware efficiency for quantum and classical molecular dynamics *Parallel Computing Technologies (Lecture Notes in Computer Science* vol 9251) ed Malyshkin V (Springer International Publishing) pp 469-473 ISBN 978-3-319-21908-0 URL http://dx.doi.org/10.1007/978-3-319-21909-7_45
- [5] Norman G E and Stegailov V V 2013 Mathematical Models and Computer Simulations 5 305-333 ISSN 2070-0482 URL http://dx.doi.org/10.1134/S2070048213040108
- [6] Podryga V O and Polyakov S V 2015 Mathematical Models and Computer Simulations 7 456-466 ISSN 2070-0482 URL http://dx.doi.org/10.1134/S2070048215050105
- [7] 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, Niethammer C, Bode A and Bungartz H J 2013 591 TFLOPS multitrillion particles simulation on SuperMUC Supercomputing (Lecture Notes in Computer Science vol 7905) ed Kunkel J, Ludwig T and Meuer H (Springer Berlin Heidelberg) pp 1–12 ISBN 978-3-642-38749-4
- [8] Piana S, Klepeis J L and Shaw D E 2014 Current Opinion in Structural Biology 24 98 105 ISSN 0959-440X
 [9] Curnow H J and Wichmann B A 1976 The Computer Journal 19 43 49
- [10] LAMMPS web site URL http://lammps.sandia.org
- [11] Swedroid Nordens största Android-community URL http://www.swedroid.se
- [12] Garg R Exploring the floating point performance of modern arm processors
- $\label{eq:URL} {\tt http://www.anandtech.com/show/6971/exploring-the-floating-point-performance-of-modern-arm-processors}$
- [13] Garg R Rgbenchmm android apps on google play
- URL https://play.google.com/store/apps/details?id=org.codedivine.rgbench&hl=en [14] Garg R Prelim analysis of RGBenchMM
- URL http://codedivine.org/2012/09/25/prelim-analysis-rgbenchmm/
- [15] Supermicro Ivy bridge based DCO superserver power & cost savings
- URL http://www.supermicro.com/white_paper/white_paper_Ivy-Bridge-Power.pdf [16] Intel Xeon E5-2600 v3 review: Haswell-EP redefines fast
- ${\rm URL}\ {\tt http://www.tomshardware.com/reviews/intel-xeon-e5-2600-v3-haswell-ep, 3932-9.{\tt html}}$
- [17] Abdurachmanov D, Arya K, Bendavid J, Boccali T, Cooperman G, Dotti A, Elmer P, Eulisse G, Giacomini F, Jones C D, Manzali M and Muzaffar S 2014 Journal of Physics: Conference Series 513 052008 URL http://stacks.iop.org/1742-6596/513/i=5/a=052008