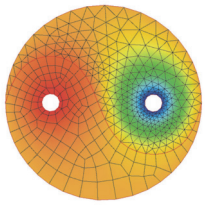
Publishing House
AKAPIT

COMPUTER METHODS IN MATERIALS SCIENCE

Informatyka w Technologii Materiałów

Vol. 15, 2015, No. 3



ON APPLICATION OF HETEROGENEOUS HARDWARE ARCHITECTURES BASED ON ADVANCED RISC MACHINES TO FEM CALCULATIONS

KRZYSZTOF BZOWSKI*, ŁUKASZ RAUCH

AGH – University of Science and Technology, Mickiewicza 30, 30-059 Kraków, Poland
**Corresponding author: kbzowski@agh.edu.pl*

Abstract

Advanced RISC Machine (ARM) hardware architectures are nowadays one of the most popular solutions among processors widely present in mobile and embedded systems. Due to relatively low power consumption and high multithreaded capabilities they can be found in more than 75% of 32-bits devices (Frenzel Jr, 2010). Modern ARM processors also contain integrated high efficiency graphics units like Mali T6xx which made them particularly useful for growing market of mobile devices. Mali processors support OpenCL standard which made them valuable for wide range of scientific computing, where processing power is as much important as power consumption. Presented paper contains proof of concept of Finite Element Method (FEM) software capable to compute transient heat transfer analysis and implemented for ARM architecture. Exemplary implementation using OpenCL was prepared. Efficiency data as well as comparison between modern GPGPU, accelerators and ARM devices are included in the paper.

Key words: heterogeneous computing, GPGPU, ARM, finite element method

1. INTRODUCTION

Heterogeneous computing is currently one of the most investigated field of high performance computing (Bientinesi et al., 2015). Problems of modern science often require huge computing power, which is related to the costs as well as energy requirements (Rodero & Parashar, 2012; Rauch, 2013). The main goal of heterogeneous computing is combining different architectures to obtain better overall performance and solve the task faster. The second objective applied to heterogeneous architecture is minimization of power consumption by careful distribution of computational tasks to suitable computational units. GPGPU computing is already well known and popular among computational scientist (Rauch et al., 2012). High amount of multiprocessors present in modern GPU devices allow to introduce new level of parallelism. Libraries such as NVIDIA CUDA provides comprehensive, but limited only to that manufacturer, computational solution. Intel Xeon Phi is relatively new family of computing devices in the world of heterogeneous hardware, which offers alternative architecture. These processors can be easily adapted to existing application due to native supports of x86 binaries and easy to use existing parallel libraries like MPI (Kruze & Banas, 2014).

Recently, the trends observed in green computing force manufacturers to create more environmental friendly computational solutions. There are specific environments with very limited power supplies, where well known conventional solutions cannot be used. ARM ltd. introduced architecture with overarching objective focused on power efficiency. Such a condition was possible due to significantly simpler architecture (Reduced Instruction Set Computing) compared to modern x86 processors, and the possibility of extending processors with dedicated con-

 ISSN 1641-8581

trollers, e.g. graphics co-processors. Currently, this type of the processors play a key role in the world of modern mobile devices, but also some attempts were already made to use them to build energy efficient HPC clusters (Rajovic et al., 2014). Some adaptation for general purpose scientific calculations using ARM processors was already done in (Abdurachmanov et al., 2014b) with very promising results.

Coexistence of many incompatible programming frameworks (CUDA, OpenMP, RenderScript) and language extensions (Intel Cilk Plus, C++ Accelerated Massive Parallelism) causes that it was necessary to create a common standard for heterogeneous computing, which would allow to merge different types of devices into one coherent system and use them together. Due to cooperation of significant manufactures like Intel, NVidia and AMD, it was possible to create an open standard for heterogeneous computing called OpenCL[1], which allows to create software working on different platforms – hardware as well as software. OpenCL was created as open source framework, which offers similar performance in many tasks to existing solution and it is not limited to particular vendor. Additional code modifications can still be required due to differences between hardware architectures to obtain a desirable performance. The major problem in performance of HPC is choice of the most efficient data structures for specific problem and target computing device. Well known schemes from conventional approaches are, in the most cases, not suitable to direct usage. Because of this, additional data translations are often needed. The main goal of this publication is to evaluate the possibility of using ARM in comparison to conventional CPUs and GPUs in scientific computing, particular for simulation of the materials behavior.

2. ARM-BASED COMPUTING

Global interest in ARM-based servers is growing despite of the lack of the significantly higher performance in comparison to conventional servers processors. Compared to x86 processors, huge technological leap of ARM processors is observed in the last few years. In many cases, the data processing efficiency of ARM processors has increased 10 or even 20 times over the last five years. One of the first ARM servers was announced in 2011, i.e. Calxeda EnergyCore ECX-1000, featuring four 32-bit ARMv7 cores. The latest great breakthrough was AppliedMicro X-Gene server – first ARMv8 64-bit server for general purpose (Abdurachmanov et al., 2014a). It was the matter of time for ARM products to start playing important role also in HPC area. Thus, the main goal of the Tibidabo project (Rajovic et al., 2014) was to create an energy efficient HPC cluster built using ARM processors. To achieve this goal NVIDIA Tegra2 System on Chip (SoC) architecture was used, with a dual-core ARM Cortex-A9 running at 1 GHz. Benchmarks, on a single computing node as well as by using a whole cluster, were performed using Dhrystone, SPEC CPU2006, STREAM and other scientific software. Scalability, performance as well as a power consumption were analyzed in details. However, relatively low energy efficiency was obtained. As a conclusions, authors believed that to achieve satisfactory performance, comparable to conventional x86 processors it is necessary do increase a compute density (by using processors with much more cores on chip) and use higher-end ARM multicore chips like the ARMv8, which will support double precision SIMD operations. Moreover, introducing support for standard industrial programming models such as CUDA or OpenCL in graphics cores can further improve energy efficiency and performance. Maintaining high memory bandwidth-to-flops ratio by using modern memory controllers was also recommended.

3. BENCHMARK – DESCRIPTION AND IMPLEMENTATION DETAILS

The main goal of presented paper is verification that modern mobile ARM GPU such as Mali-T628 can be used for scientific computing. Two dimensional transient thermal problem was chosen as a benchmark. Problem described by partial differential equation (1) was solved using FEM implemented using OpenCL.

$$k\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}\right)=\rho c_p \frac{\partial T}{\partial t} \tag{1}$$

where T – temperature, t – time, k – thermal conductivity coefficient, ρ – density, c_p – heat capacity. The weak formulation of equation (1) was introduced. Transient problem was solved using the theta-scheme (2) with coefficient equal 1 that discretizes the equation in time.

$$\{T\}=\theta\{T_1\}+(1-\theta)\{T_0\} \tag{2}$$

[1] https://www.khronos.org/opencl/

Finally system of linear equation was obtained as:

$$\left(H+\frac{C}{\Delta t}\right)T=\frac{C}{\Delta t}T_0-P$$
$$H=\int k\left[\frac{\partial N}{\partial x}\right]^T\left[\frac{\partial N}{\partial y}\right] \quad (3)$$
$$C=\int c_p\rho[N]^T[N]$$
$$P=\int \alpha[N]^T[N]$$

where: α – heat transfer coefficient, T_0 – nodal temperatures values from previous time step, T – nodal temperature values from current time step, N – interpolation (shape) functions, Δt – time step.

Process definition with geometry and meshes was created using Abaqus software. Implementation was divided into two parts for host and computational device. Host code was prepared using C++. ViennaCL (Rupp et al., 2010) was applied as a computing device manager. Kernels were written using OpenCL C99 language. Gaussian quadrature, aggregation of local and global systems of equations and Dirichlet boundary conditions were implemented as an OpenCL kernels to avoid additional data transfer between host and computational device. The nature of the solutions based on FEM caused that they fit very well to Single Instruction Multiple Data (SIMD) computational scheme, where the same instructions are being executed for different data.

Table 1. *Benchmarked hardware information*

Device name	Theoretical performance [GFLOPS]	Memory bandwidth [GBytes/sec]	Maximum power consumption [W]	Number of cores / computational units
Intel E5-2620	120	42.6	95	12
NVidia Tesla M2090	665	177	225	512
Intel Xeon Phi 7110P	1220	352	300	61
ARM Mali-T628	109	12.8	8	8

The main problem is gathering partial data together (obtained from integration) into global system of equations. In the most cases, gathering coefficient matrix for system of equations required only information about connections between nodes in finite elements mesh. However, the parallel algorithm presented in this paper required additional data such as inverse connectivity matrix, which provides information about elements attached to each node. The main purpose of that data structure was avoiding the problem of parallel addition of different cells, which can lead to race condition situation. Thus, proposed solution requires additional memory in order to avoid data transfer or execution of single-threaded algorithm. In presented approach each thread iterates on different rows of global coefficient matrix and adds values from single elements matrices. These matrices are flattened matrix of local heat capacity matrices, which were obtained using numerical integration. Coefficient matrix of system of equation is stored in compressed matrix using Sliced ELLPACK (Kreutzer et al., 2014) algorithm, which was designed for SIMD calculation, especially using GPGPU. Final system of equation was solved using ViennaCL Conjugate Gradient (CG) solver. Additional code optimization in OpenCL kernels as well as memory management proposed by Grasso et al. (2014) was also applied.

4. RESULTS

Four different types of devices was tested: modern multicore CPU – Intel E5-2620, professional computational GPU – NVidia Tesla M2090, coprocessor – Intel Xeon Phi 7110P, and ARM GPU – Mali-T628. Theoretical performance and power consumption of each device are shown in table 1.

Presented theoretical computing power of each device is unachievable in practical applications. Also in case of FEM simulations Floating Point Operations Per Second (FLOPS) is not suitable indicator due to memory bandwidth limitation. Therefore, all conditional source codes were significantly reduced in exchange for additional data stored in memory, which also prevents from reliable determination of the performance using FLOPS. One of the trustworthy method of performance determination, in the case of this particular algorithm, is measurement of the efficiency with different sizes of input data. The benchmark set included mesh grids between 340 and 750 000 nodes made for two dimensional square with side length equal 0.1m. The measurement methodology included one iteration of the FEM simulation. Typical material properties of low carbon steel was used. Dirichlet boundary conditions (constant temperature) was attached on the bottom

part of geometry. Results obtained for different devices are shown in figure 1.

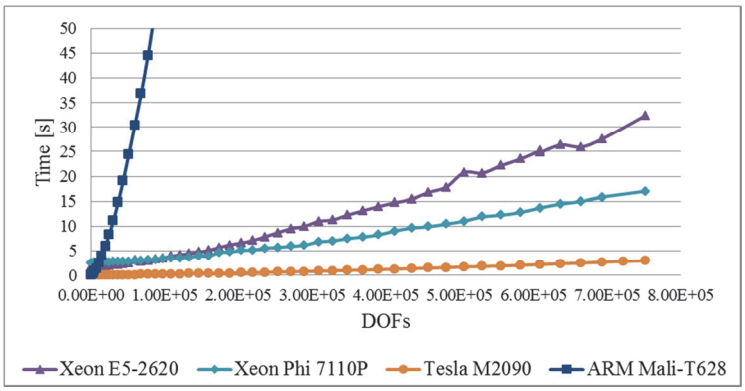


Fig. 1. *Time measurements of FEM simulations obtained for different computing devices as a function of Degrees of Freedom (DOFs).*

Table 2. *Measured power consumption of computing devices*

Device	Average power consumption [W] only calculations / whole time
Intel E5-2620	71
NVidia Tesla M2090	171 / 82
Intel Xeon Phi 7110P	137 / 111
ARM Mali-T628	2.15

consumption of the other devices was done with the dedicated monitoring software utilities. Obtained results were gathered in table 2.

The characteristics presented for each device in table 2 show average power consumption, however this measurement is not reliable when we look deeper at the measurements of the same factor in time. Figure 2 shows very irregular power consumption in case of GPGPU, which charges a lot of power during quite short calculations, but remains at the level of 80W when the device stays idle during data upload. In case of Xeon Phi the situation is similar, but the differences between power consumption during calculations and in the idle time are not so sharp.

It is necessary to mention, that all unconventional computational devices required host with one or more standard CPUs as a controller and executor of IO operation. This means that the total demand for electric power always includes both host and computing devices power requirements. The power consumption and computational efficiency of the applied devices cannot be analyzed separately. Therefore, we tried to analyze the behavior of hardware infrastructure composed of many computing

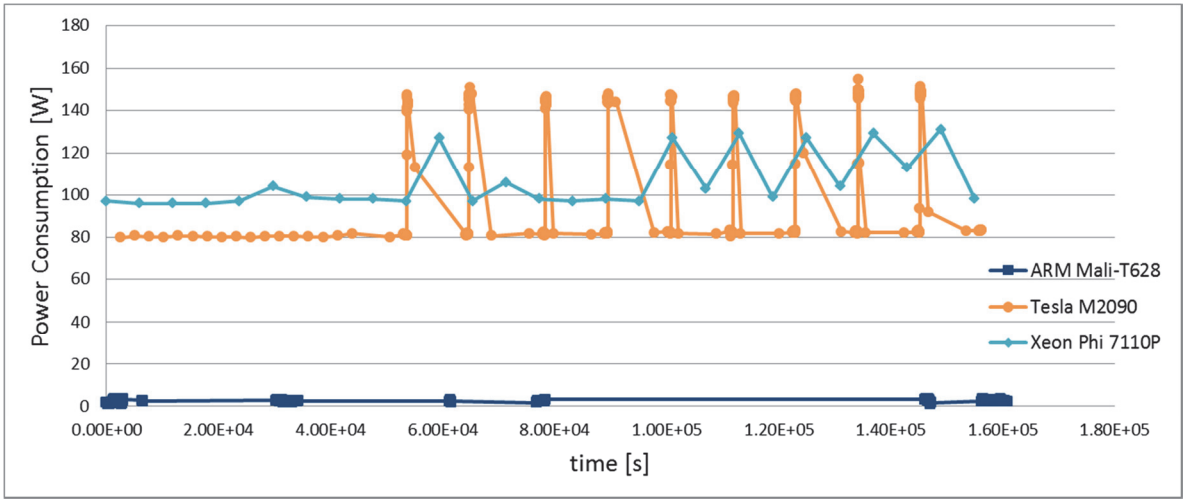


Fig. 2. *Example of measurements of power consumption in a function of time for three selected devices with heterogeneous hardware architecture.*

Figure 1 clearly shows superiority of Tesla M2090 over conventional CPU and Xeon Phi. Speedup obtained for ARM processor is significantly worse. This is due to much lower computational power as well as lower memory bandwidth in comparison to other devices. In addition to performance tests, power consumptions tests were also performed. Intel CPU power consumption was measured using electrical meter. Measuring the energy nodes, especially in the case of ARM processors, which are characterized by weaker capabilities than the rest of the analyzed hardware. The power efficiency was adopted as a measure of computational efficiency in a function of the number of DOFs. For simplicity of the further analysis we assumed that hardware has negligible communication overhead and the distribution of computing tasks is perfectly balanced. Due to these assumptions, approximate

number of computing nodes in each hardware configuration was calculated to cope with problems which exceeds device capabilities. Afterwards, the number of devices was multiplied by the time required for calculations and measured power consumption. The results of these calculations are presented in figure 3.

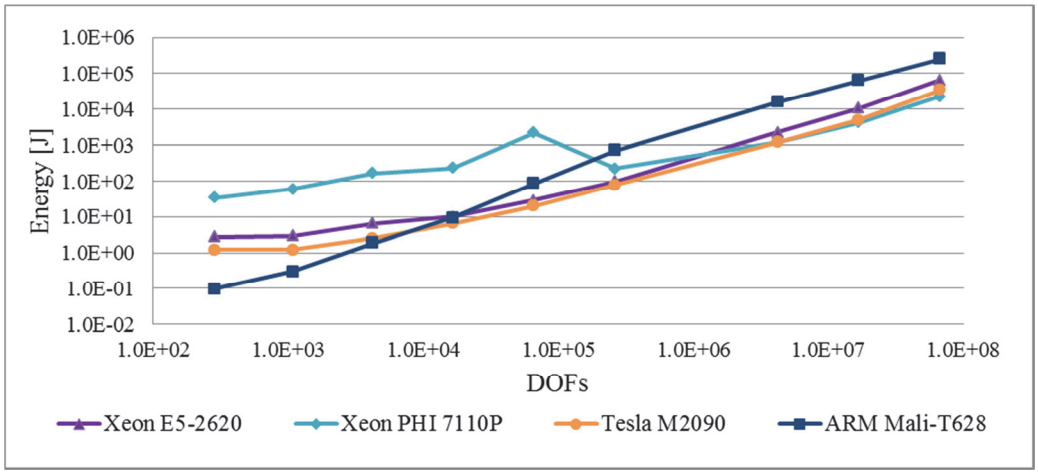


Fig. 3. *Power efficiency of computing devices within the context of the size of the system of equations*

It can be seen that for problems of a lower complexity, total energy required to solve such system of equations is also lower. In case of energy consumption, based on obtained data, explicit DOF threshold for each device can be distinguished in which the choice of ARM processors is less power consumption option e.g. if DOF is lower than 6481 in the case of the analyzed problem it would be better to use ARM Mali-T628 instead of Tesla M2090, otherwise GPGPU is characterized by better efficiency. Detailed information on DOF threshold values are presented in table 3.

Table 3. *Threshold DOFs below which the computation devices are less energy efficient then ARM processors.*

	Tesla M2090	Xeon E5-2620	Xeon PHI 7110P
Mali-T628	6481	17129	226052

5. CONCLUSION

Implementation of typical FEM solution of heat transfer model in OpenCL was presented. Benchmarks was performed on standard CPU as well as on non-conventional computing devices (GPU, co-processors) and ARM GPU device. Current development stage of modern ARM architectures does not allow to use them efficiently in HPC environment even in multi devices configurations, while in practice most of the computing problems are characterized by large number of DOFs. It should, however, be considered to use ARM processors as a controllers on hosts for other, more energy-requiring devices. Obtained results shows insufficient computing power of Mali-T628 processor in the case of typical FEM calculations. Authors believe, that it is still possible to find suitable area of application for ARM devices in heterogeneous computing, but further investigation is required. The future of the ARM HPC belongs to accelerators such as PEZY-SC, which currently (January 2015) occupied top place in GREEN 500 list.

ACKNOWLEDGEMENTS

Financial support of the NCN, project no. 2011/01/D/ST6/02023, is acknowledged.

REFERENCES

Abdurachmanov, D., Bockelman, B., Elmer, P., Eulisse, G., Knight, R., Muzaffar, S., 2014a, Heterogeneous High Throughput Scientific Computing with APM X-Gene and Intel Xeon Phi, *ArXiv e-prints*, available online at http://arxiv.org/pdf/1410.3441, accessed: 12.10.2015.

Abdurachmanov, D., Elmer, P., Eulisse, G., Muzaffar, S., 2014b, Initial explorations of ARM processors for scientific computing, *Journal of Physics Conference Series*, 523, 1-6.

Bientinesi, P., Herrero, J.R., Quintana-Ortí, E.S., Strzodka, R., 2015, Parallel computing on graphics processing units and heterogeneous platforms, *Concurrency and Computation: Practice and Experience*, 27, 1525-1527.

Frenzel Jr, L.E., 2010, Chapter 6 - How Microcomputers Work: The Brains of Every Electronic Product Today, *Electronics Explained*, ed. Frenzel, L.E., Newnes, Boston, 123-145.

Grasso, I., Radojkovic, P., Rajovic, N., Gelado, I., Ramirez, A., 2014, Energy Efficient HPC on Embedded SoCs: Optimization Techniques for Mali GPU, Proc. Conf. *Parallel and Distributed Processing Symposium, 2014 IEEE 28th International*, 123-132.

Kreutzer, M., Hager, G., Wellein, G., Fehske, H., Bishop, A.R., 2014, A unified sparse matrix data format for efficient general sparse matrix-vector multiply on modern processors with wide SIMD units, *ArXiv e-prints*, available online at http://arxiv.org/pdf/1307.6209v2.pdf, accessed: 12.10.2015.

Kruze, F., Banas, K., 2014, Finite Element Numerical Integration on Xeon Phi coprocessor, eds. Ganzha, M., Maciaszek, L., Paprzycki, M., Proc. Conf. *Federated Conference on Computer Science and Information Systems*, Lodz, 603-612.

Rajovic, N., Rico, A., Puzovic, N., Adeniyi-Jones, C., Ramirez, A., 2014, Tibidabo: Making the case for an ARM-based HPC system, *Future Generation Computer Systems*, 36, 322-334.

Rauch, L., 2013, Heterogeneous Hardware Implementation of Molecular Static Method for Modelling of Interatomic Behaviour, *Procedia Computer Science*, 18, 1057-1067.

Rauch, L., Bzowski, K., Rodzaj, A., 2012, OpenCL Implementation of Cellular Automata Finite Element (CAFE) Method, *Parallel Processing and Applied Mathematics*, eds. Wyrzykowski, R., Dongarra, J., Karczewski, K., Waśniewski, J., Springer Berlin Heidelberg, 381-390.

Rodero, I., Parashar, M., 2012, Energy Efficiency in HPC Systems, *Energy-Efficient Distributed Computing Systems*, John Wiley & Sons, Inc., 81-108.

Rupp, K., Rudolf, F., Weinbub, J., 2010, ViennaCL - A High Level Linear Algebra Library for GPUs and Multi-Core CPUs, eds. Mehofer, E., Schordan, M., Quinlan, D., Di Martino, B., Proc. Conf. *International Workshop on GPUs and Scientific Applications (GPUScA 2010)*, 51-56.

ZASTOSOWANIE HETEROGENICZNYCH ARCHITEKTUR SPRZĘTOWYCH BAZUJĄCYCH NA ARCHITEKTURZE ADVANCED RISC MACHINES DO OBLICZEŃ MES

Streszczenie

Architektura Advanced RISC Machine (ARM) jest obecnie jedną z najbardziej popularnych rozwiązań wśród procesorów mobilnych i systemów wbudowanych. W związku ze znacznie mniejszym zużyciem energii elektrycznej i wysoką wielowątkowością znalazły one zastosowanie w ponad 75% obecnie stosowanych systemów 32-bitowych (Frenzel Jr, 2010). Nowoczesne procesory ARM zawierają często zintegrowane jednostki graficzne wysokiej wydajności, takie jak Mali T6xx, co sprawia że stały się one szczególnie użyteczne dla dynamicznie rozwijającego się rynku urządzeń mobilnych. Procesory z rodziny Mali T6xx wspierają standard OpenCL, co powoduje, że mogą one również zostać wykorzystane w szerokiej gamie obliczeń naukowych, w których moc obliczeniowa jest tak samo istotna jak oszczędność energii. W artykule przedstawiono koncepcję oprogramowania wykorzystującego metodę elementów skończonych do obliczeń niestacjonarnych przepływów ciepła z wykorzystaniem architektury obliczeniowej ARM. Przedstawiono przykładową implementację z wykorzystaniem technologii OpenCL, jak również wykonano testy porównawcze z nowoczesnymi architekturami GPGPU oraz analizy energetyczne.

Received: December 11, 2014
Received in a revised form: February 2, 2015
Accepted: May 13, 2015