Journal of Computational and Applied Mathematics 270 (2014) 63–77

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Heterogeneous architectures for computational intensive applications: A cost-effectiveness analysis

E. Danovaro *, A. Clematis, A. Galizia, G. Ripepi, A. Quarati, D. D'Agostino

Institute of Applied Mathematics and Information Technologies, National Research Council of Italy, Genoa, Italy

ARTICLE INFO

Article history:
Received 30 September 2013

Keywords:
Parallel programming
HPC
Heterogeneous architectures

ABSTRACT

Current workstations can offer really amazing raw computational power, in the order of TFlops on a single machine equipped with multiple CPUs and accelerators, which means less than half a dollar for a GFlop. Such result can only be achieved with a massive parallelism of the computational devices, but unfortunately not every application is able to fully exploit them. In this paper we analyze the performances of some widely used, computational intensive, applications, like FFT, convolution and *n*-body simulation, comparing the performance of a multi-core cluster node, with or without the contribution of GPUs. We aim to provide clear measure of the benefit of a heterogeneous architecture, in terms of time and cost, with a stress on the technology adopted at different levels of the software stack for the application parallelization.

1. Introduction

In 1965 the Intel co-founder Gordon E. Moore affirmed that, over the history of computing hardware, the number of transistors on integrated circuits doubles approximately every year, and the trend would continue for at least ten years [1]. This affirmation, recalibrated in "every two years" in 1975,[1] is still valid, but with a major change: every new generation of CPUs is more powerful than the previous one mostly because it provides more cores.

Multicore CPUs have been produced since 2005, when the dual-core Intel's Pentium D 800 was released, followed by the AMD's Athlon 64 X2. Presently, all the most important processor vendors have switched to multicore architectures, with great differences in terms of the homogeneity of these cores [2], as we will briefly describe in the following section. Moreover, as recognized in [3], "This shift toward increasing parallelism is not a triumphant stride forward based on breakthroughs in novel software and architectures for parallelism; instead, this plunge into parallelism is actually a retreat from even greater challenges that thwart efficient silicon implementation of traditional uniprocessor architectures". Each new generation of CPU double the cores, but each single core has more or less the same compute capabilities: "The dilemma is that a large percentage of mission critical enterprise applications will not automagically run faster on multicore servers. In fact many will actually run slower".[2]

Furthermore, while the traditional homogeneous HPC solutions offer a well-known architecture, and this makes easier to achieve performance figures very close to the theoretical peak, the efficient exploitation of the present more hierarchical

* Corresponding author.
E-mail addresses: danovaro@ge.imati.cnr.it (E. Danovaro), clematis@ge.imati.cnr.it (A. Clematis), galizia@ge.imati.cnr.it (A. Galizia), ripepi@ge.imati.cnr.it (G. Ripepi), quarati@ge.imati.cnr.it (A. Quarati), dagostino@ge.imati.cnr.it (D. D'Agostino).

[1] http://news.cnet.com/2100-1001-984051.html.

[2] Patrick Leonard, Rogue Wave Vice President for Product Development http://software.intel.com/en-us/blogs/2007/03/14/the-multi-core-dilemma-by-patrick-leonard.

http://dx.doi.org/10.1016/j.cam.2014.02.022
0377-0427/

and heterogeneous computing architectures requires an increased effort in the development of smart solution because, for example, the data movement among the memories of the different kinds of processors represents a new and major issue. This scenario was defined "a true computing jungle" [4].

This issue was recently discussed in a Workshop organized by the European Commission in preparation of HORIZON 2020: "Software development has not evolved as fast as hardware capability and network capacity.… Development and maintenance of software for advanced computing systems is becoming increasingly effort-intensive requiring dual expertise, both on the application side and on the system side".[3] The present frontiers is therefore the development of programming models and tools enabling developers to balance the competing goals of productivity and implementation efficiency: "Writing programs that scale with increasing numbers of cores should be as easy as writing programs for sequential computers" [5].

Many research projects aimed and aims to provide a solution to this problem. For example pattern based parallel programming models, in particular using skeletons, promise several benefits as a simplification of the code development and the easy portability and reusability of the solutions [6]. An example was the ASSIST programming environment [7,8], originally developed in the framework of the GRID.it Italian national project, while a complete survey is provided in [9].

Among the current projects it is worth citing ParaPhrase,[4] "aiming to provide developers with parallel patterns for the creation of re-mappable component based applications", the Hybrid for HPC (H4H) project,[5] "aiming at providing compute-intensive application developers with a highly efficient hybrid programming environment for heterogeneous computing clusters composed of a mix of classical processors and hardware accelerators", and the Open Network for High-Performance Computing on Complex Environments (ComplexHPC)[6] EU COST Action, "aiming to coordinate European groups working on the use of heterogeneous and hierarchical systems for HPC, foster HPC efforts to increase Europe competitiveness and train new generations of HPC scientists".

Also vendors have made significant efforts to assist developers in writing high performance applications. This also because such problems are not limited only to huge supercomputers: if we move toward a wider audience of scientists and IT developers, we can see that the same issues arise also with current workstations. In fact it is possible to achieve up to 10 TFlops on a single machine equipped with multiple CPUs and accelerators, with a cost of less than $5000.

In this paper we analyze some of these programming models and tools, both commercial and freely available, in terms of provided support and achievable performance with respect to some widely used, computational intensive algorithms, as for example the *N*-body, the convolution and Fourier transform. The focus is on providing a clear measure of the different efficiency figures with respect to the programming paradigm considered, the adopted tool and the achievable peak performance. We will see that it is nearly easy to achieve an order of magnitude speed improvements for selected compute kernels with a limited effort, but also that developers have to know the specific features of every compute capabilities to achieve much higher performance.

2. A low-cost heterogeneous computing environment

Current workstations can offer really amazing raw computational power, in the order of TFlops, on a single machine equipped with multiple CPUs and GPU devices. Such result can only be achieved with a massive parallelism of the computational devices. For example a workstation is usually equipped with 1–4 multicore processor and up to 4 accelerators connected via PCI express bus, as shown in Fig. 1.

2.1. Adopted computing environment

Workstation CPUs are produced by Intel and AMD. Intel XEONs are characterized by 6–8–10 cores per CPU, and the possibility to connect up to 8 CPUs on a single motherboard (up to 80 cores per board); each CPU usually has shared cache (up to 30 MB). AMD Opterons are composed by two dies, each die has up to 8 cores and 8 MB of cache. It is possible to install up to 4 Opterons (up to 64 cores on a single board). CPUs on the same board can communicate with a dedicated bus (QPI for Intel, HyperTransport for AMD), moreover each CPU integrates DDR3 memory controller and PCI express controller, granting good scalability and extremely high aggregated bandwidth. One drawback of such architecture is that accessing memory connected to another CPU requires multiple transmission steps (from memory to CPU, from CPU to CPU one or more times), and thus increased latency.

Table 1 list a few state-of-the-art CPUs from Intel and AMD, with number of cores, price, raw performance of a system with maximum number of CPUs installed, and bandwidth of communication buses to CPUs, Memory and PCI express. Please note that peak performances require the full exploitation of all the capabilities: all CPUs, all cores, and SIMD instruction in each core. In fact recent CPUs offer an extended set of instruction (i.e. SSE1 … SSE4, AVX) which are able to process up to 8 single precision or 4 double precision values. Adoption of such SIMD instruction is often referred as vectorization.

[3] http://cordis.europa.eu/fp7/ict/computing/documents/advanced_computing_ws_report.pdf.

[4] http://paraphrase-ict.eu/.

[5] www.h4h-itea2.org.

[6] http://complexhpc.org/.