Journal of Computational Science 4 (2013) 439–443

Contents lists available at ScienceDirect

Journal of Computational Science

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

The low power architecture approach towards exascale computing

Nikola Rajovic [a,b,*], Lluis Vilanova [a,b], Carlos Villavieja [a,b], Nikola Puzovic [a], Alex Ramirez [a,b]

[a] Computer Sciences Department, Barcelona Supercomputing Center, Barcelona, Spain
[b] Departament d'Arquitectura de Computadors, Universitat Politècnica de Catalunya – BarcelonaTech, Barcelona, Spain

ARTICLE INFO

Article history:
Received 31 March 2012
Received in revised form
13 September 2012
Accepted 15 January 2013
Available online 1 February 2013

Keywords:
Exascale
Embedded
Mobile processors
Low power
Cortex-A9

ABSTRACT

Energy efficiency is a first-order concern when deploying any computer system. From battery-operated mobile devices, to data centers and supercomputers, energy consumption limits the performance that can be offered.

We are exploring an alternative to current supercomputers that builds on low power mobile processors. We present initial results from our prototype system based on ARM Cortex-A9, which achieves 120 MFLOPS/W, and discuss the possibilities to increase its energy efficiency.

1. Introduction

For a long time, the only metric that was used for assessing supercomputer performance was their speed. The Top500 list ranks supercomputers based on their performance when running the High-Performance LINPACK benchmark [1]. However, performance per watt is currently as important as raw computing performance: nowadays, system performance is limited by power consumption and power density. The Green500 list [2] ranks supercomputers based on their power efficiency. A quick look at this list shows that the most efficient systems today achieve around 2 GFLOPS/W, and that most of the top 50 power-efficient systems are built using heterogeneous CPU + GPU platforms. According to Ref. [2], among the most power-efficient supercomputers (Table 1) are either those based on processors designed with supercomputing in mind, or those based on general purpose CPUs with accelerators (Intel MICs or GPUs). Blue Gene/Q (Power A2) is an example of first type. Examples of the second type are the Intel Cluster (Intel Xeon E5-2670 and Intel Knights Corner), the Degima Cluster (Intel Core i5 and ATI Radeon GPU) and Bullx B505 (Intel Xeon E5649 and NVIDIA GPU).

Not only supercomputers but also servers and data-centers have power constrains. In recent years we have also seen a dramatic increase in the number, performance and power consumption in this domain. This market, which includes companies such as Google, Amazon and Facebook, is also concerned with power efficiency. Frachtenberg et al. [3] present an exhaustive description of how Facebook builds efficient servers for their data-centers, achieving a 38% reduction in power consumption by improving cooling and power distribution only.

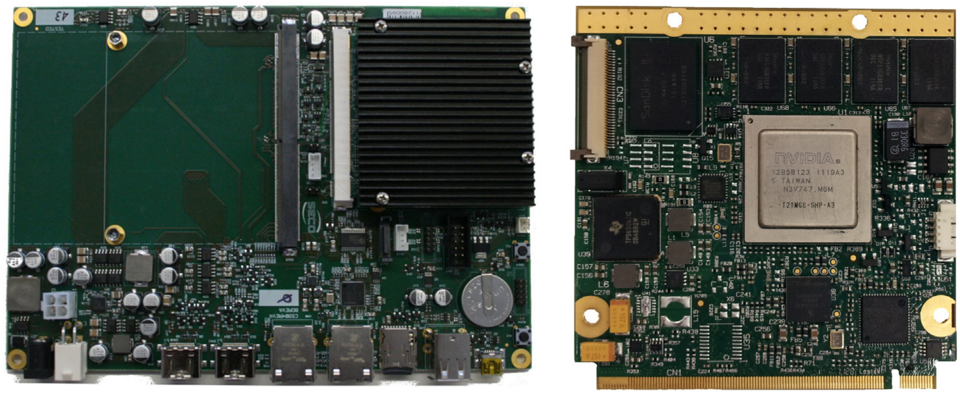
The performance of supercomputers has shown a constant exponential growth over time: according to the Top500 list of supercomputers [4], an improvement of 10× in performance is observed every 3.6 years. The Roadrunner Supercomputer achieved 1 PFLOPS (10^{15} floating point operations per second) in 2008 [5] on a power budget of 2.3 MW, and the current number one supercomputer,[1] Sequoia achieves 16 PFLOPS while consuming 7.9 MW.

Following this trend, exascale performance should be reached in 2018, but the required power for that will be up to 400 MW.[2] A realistic power budget for an exascale system is 20 MW [6], which requires an energy efficiency of 50 GFLOPS/W. As Ref. [6] suggests, we have to tackle a lot of issues towards achieving exascale – to improve on computing elements, memory technologies, networking, storage and cooling. Here we choose to deal with computing

* Corresponding author at: Computer Sciences Department, Barcelona Supercomputing Center, Jordi Girona 29, Nexus II Building, 08034 Barcelona, Spain. Tel.: +34 633138986.
E-mail addresses: nikola.rajovic@bsc.es (N. Rajovic), lluis.vilanova@bsc.es (L. Vilanova), cvillavi@ac.upc.edu (C. Villavieja), nikola.puzovic@bsc.es (N. Puzovic), alex.ramirez@bsc.es (A. Ramirez).

[1] As per June 2012 Top500 list.
[2] For comparison purposes, the total reported power of all the supercomputers as per June 2012 Top500 list is 336 MW [4].

1877-7503/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.jocs.2013.01.002



(a) Carrier Board (b) Daughter Board

Fig. 1. (a) Carrier board with daughter board in place and (b) daughter board without the heat sink.

Table 1
Power efficiency of several supercomputing systems.

Supercomputing system	Type	GFLOPS/W
Blue Gene/Q Cluster	Homogeneous	2.1
Intel Cluster	Intel MIC accelerator	1.38
Degima Cluster	ATI GPU accelerator	1.37
Bullx B505	NVIDIA GPU accelerator	1.27
iDataPlex DX360M4	Homogeneous	0.93

elements first by exploring an alternative microprocessor architecture for HPC.

A quick estimation, based on using 16 GFLOPS processors (like those in Sequioa and the Fujitsu K supercomputers), shows that a 1 EFLOPS system would require 62.5 millions of such processors. Based on current trends, if we observe that only 35–50% of the 20 MW allocated to the whole computer is actually spent on the CPUs, we can see that each of those processors has a power budget of only 0.15 W, including caches, network-on-chip, etc. Current high-end multicore architectures are one or two orders of magnitude away from that mark. The cores used in GPU accelerators are in the required range, but they lack general purpose computing capabilities. A third design alternative is to build a high performance system from low power components originally designed for mobile and/or embedded systems.

In this paper, we evaluate the feasibility of developing a high performance compute cluster based on the current leader in the mobile domain, the ARM Cortex-A9 processor [7]. First, we describe the architecture of our HPC cluster, built from Nvidia Tegra2 SoC[3] and a 1 Gb Ethernet interconnection network. To the best of our knowledge, this is the first large-scale HPC cluster built using ARM multicore processors.

Then, we compare the per-core performance of the Cortex-A9 with a contemporary power-optimized Intel Core i7,[4] and evaluate the scalability and performance per watt of our ARM cluster using the High-Performance LINPACK benchmark.

2. Prototype

2.1. Node

The prototype that we are building (named Tibidado[5]) consists of 256 nodes organized into 32 blades. Each blade has eight nodes and a shared power supply unit (PSU). The compute node is built around a SECO Q7-compliant carrier board (Fig. 1(a)) designed to host one microprocessor SoC and one low power MXM GPU. Each node also exposes two Ethernet NICs (network interface controllers) – 1 Gb for MPI communication and 100 Mb for a NFS (Network File System) which hosts Linux kernel and both system and user data. The node is designed to be used for embedded software development, not particularly tailored for HPC, and hence includes many features that are unnecessary in the HPC domain (e.g. multimedia expansions and related circuitry).

2.2. Microprocessor SoC

The compute power comes from an NVIDIA Tegra 2 SoC, which implements a dual-core ARM Cortex-A9 processor at 1 GHz. This SoC is mounted on the daughter board (Fig. 1(b)), which is connected to the carrier board via a Q7 connector. The use of a Q7-compliant daughter board eases future upgrades of the processor SoC. In addition to the SoC, the daughter board contains 1 GB of DDR2-667 RAM and a 1 Gb embedded Ethernet controller. The power consumption of the daughter board is approximately 4 W, and it provides 2 GFLOPS of peak double precision floating-point performance.

2.3. Interconnection network

Nodes are connected through 1 GbE network with a tree-like network topology. Each group of 32 nodes are connected on a first-level switch. We use the same switch model for all switching levels of the network (Cisco SF200-50 [8]). Each node is reachable within four hops in the network.

3. Initial results

3.1. Methodology

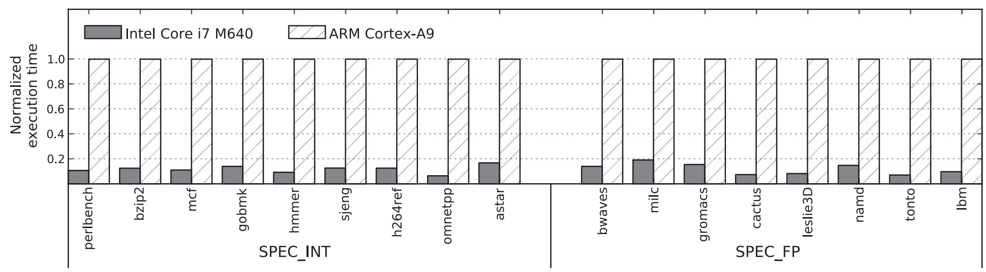
For single core comparison of both performance and energy, we use Dhrystone [9], STREAM [10] and SPEC CPU2006 [11] benchmark suites. Both platforms, Tibidabo node and a power optimized Intel Core i7 laptop, execute benchmarks with the same input set size in order to have comparable numbers. Both platforms run GNU/Linux OS and use the GCC 4.6 compiler. We measure power consumption at AC socket connection point for both platforms, and calculate energy-to-solution by integrating power samples.

We report initial performance and energy efficiency of a fraction of Tibidabo cluster (32 nodes) running High-Performance LINPACK

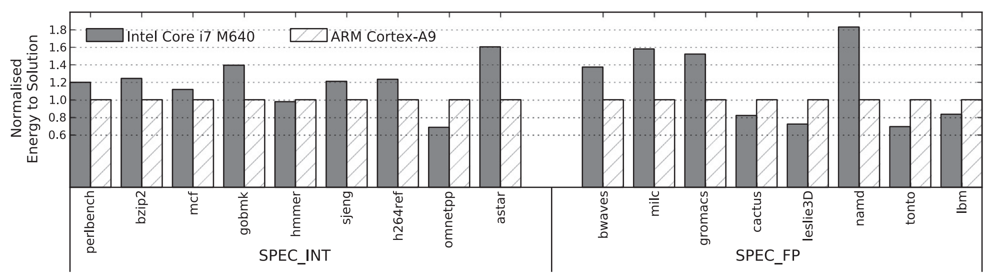
[3] NVIDIA Tegra2 implements dual-core power-optimized ARM Cortex-A9.
[4] Both the Nvidia Tegra2 and Intel Core i7 M640 were released on Q1 2010.
[5] Tibidabo is a mountain overlooking Barcelona.

Table 2
Dhrystone and STREAM: Intel Core i7 and ARM Cortex-A9 performance and energy-to-solution comparison.

Platform	Dhrystone			STREAM					
	perf (DMIPS)	*energy*		*perf (MB/s)*				*energy (avg.)*	
		abs (J)	*norm*	*copy*	*scale*	*add*	*triad*	*abs (J)*	*norm*
Intel Core i7	19,246	116.8	1.056	6912	6898	7005	6937	481.5	1.059
ARM Cortex-A9	2213	110.8	1.0	1377	1393	1032	787	454.8	1.0



(a) Performance comparison



(b) Energy to Solution comparison

Fig. 2. SPEC CPU2006 benchmark results: (a) comparison between the two platforms in terms of execution time and (b) comparison between the two platforms in terms of energy to solution. All results are normalized to ARM Cortex-A9.

benchmark [1]. This benchmark is used to rank supercomputers in the Top500 list while solving the biggest possible problem that can fit into system memory. We tested weak scaling and three different configurations for strong scaling (with different problem sizes). For the algebraic backend we use ATLAS 3.9.51 library [12]. Power samples are collected at blades' AC connection points and do not include network power consumption given that network is out of the scope of this paper.

3.2. Results

In terms of performance, on all the tested single-core benchmarks, the Intel Core i7 outperforms ARM Cortex-A9 core, as expected given the obvious design differences.

Table 2 shows the comparison between two platforms. In the case of Dhrystone, Core i7 performs better by a factor of nine, but ARM platform uses 5% less energy. Similarly, in the case of STREAM, Core i7 provides five times better performance but ARM platform uses 5% less energy to execute it.

In the case of SPEC suite (Fig. 2), Intel Core i7 core is significantly faster than ARM Cortex-A9 (up to 10 times), but at the same time, ARM platform uses less power resulting in 1.2 times smaller energy-to-solution (on average).

The performance of High-Performance LINPACK in weak scaling configuration scales linearly when the number of nodes is increased from 1 to 32 (see Fig. 3). For each node configuration of the benchmark, we chose the input size that provided maximum performance. Since the theoretical peak performance of a single node is 2 GFLOPS, and we achieve 1 GFLOPS per node in each configuration, the efficiency is 50%. We can attribute relatively small efficiency to the fact that algebra library is not particularly optimized for our platform but is relying on the compiler to do all optimizations. We expect that a hand-tuned version of algebraic backend should give an additional improvement in efficiency and thus in maximum achievable performance (and energy-efficiency at the end).

Strong scaling tests suggest that the communication overhead limits the scalability (increasing problem size gives better scalability). As a matter of fact, our network is a simplistic one, so we experience limitations in connectivity and congestion (as observed as timeouts in post-mortem trace analysis).

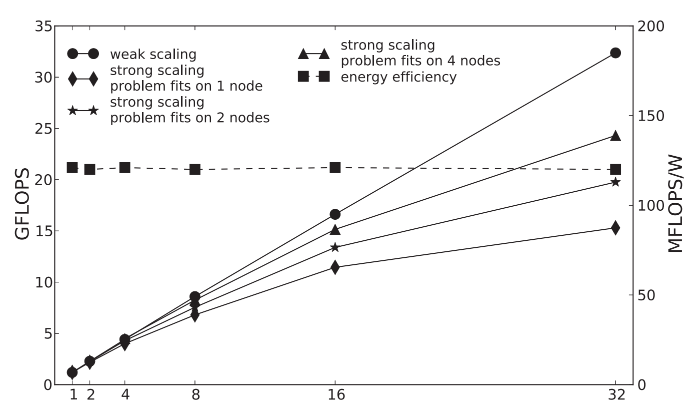


Fig. 3. Results for High-Performance LINPACK when changing number of nodes from 1 to 32. Left *y*-axis shows performance (GFLOPS), while right *y*-axis shows energy efficiency (MFLOPS/W).

4. Increasing the energy efficiency

So far, we have demonstrated the feasibility of building an ARM-based cluster for HPC system and deployed a full HPC software stack that allows for software development and tuning for the ARM-based multicores. On the other side, we have demonstrated that using low-power processor does not necessarily result in better energy efficiency. However, to better understand how to achieve an energy-efficient system, we need to see where the power is drawn, and to identify possible ways to increase the energy efficiency of the system.

When we analyze the power consumed by one node, only 6% of the total power is spent on the CPU cores [7], while the 1 GB DDR2-667 memory module and Ethernet controllers consume about 30% of the board power [13]. The remaining power, over 60% of the total, is spent on power supply inefficiencies, signaling logic, and other components which are necessary for an embedded system development kit, such as an integrated keyboard controller, HDMIs, USBs with related circuitry. Although necessary when used in the embedded domain, these components are not needed for an HPC system, and a first step towards the improved energy-efficiency could be the re-designing the boards to remove all unnecessary components.

The 6% of the total power consumed by the cores is much lower than the percentage that is consumed by high-end CPUs, which may consume up to 40% of the total power in a typical system [14]. This leaves room for improving energy efficiency by using the same techniques that are already seen in contemporary microprocessors: increasing the multicore density (the number of cores on a single chip) or adding advanced functional units, such as SIMD floating-point unit(s).

The first of the two possibilities is more expensive one in terms of additional power requirements given the design constrains, but it gives an opportunity for achieving balanced system integration by putting more power into computing compared to the other node components. Although by increasing multicore density we do increase the overall power, the power consumed by shared components (such as Ethernet ports, memory, etc.) does not scale linearly with the number of cores, and the overall energy-efficiency of the system increases. Implementation of SIMD floating-point unit comes at the cost of increasing power consumption but at the same time boosts floating-point performance which results in improved energy efficiency. In order to achieve the optimal result, a proper mix of these techniques is required, and design space exploration should answer how to mix them on an ARM architecture to get a more energy efficient system.

5. Conclusions

To the best of our knowledge, we are the first to deploy and evaluate a cluster for High Performance Computing built from commodity embedded components and ARM mobile processors. Unlike heterogeneous systems based on GPU accelerators, which require code porting and special-purpose programming models like CUDA or OpenCL, our system can be programmed using well known MPI + SMP programming models.

Our results show that the ARM Cortex-A9 in the Nvidia Tegra2 SoC is up to ten times slower than a mobile Intel i7 processor, but still achieves a competitive energy efficiency. Given that Tegra2 is among the first ARM multicore products to implement double-precision floating point functional units, we consider it very encouraging that such an early platform, built from off-the-shelf components achieves competitive energy efficiency results compared to multicore systems in the Green500 list.

If we account for upcoming mobile multicore products based on ARM Cortex-A15 CPUs, we expect better energy-efficiency and about the same power budget [15]. First implementations of the Cortex-A15 are built on 28 nm process (compared to 40 nm in Tegra2) which allows the same power requirements while improving the CPU floating-point performance, hence leading to the future ARM-based systems with increased energy efficiency.

Acknowledgements

This project and the research leading to these results is supported by Mont-Blanc project (European Community's Seventh Framework Programme [FP7/2007–2013] under grant agreement no. 288777) and PRACE Project (European Community funding under grants RI-261557 and RI-283493).

References

[1] J. Dongarra, P. Luszczek, A. Petitet, The LINPACK benchmark: past present and future, Concurrency and Computation: Practice and Experience 15 (9) (2003) 803–820.
[2] The Green500 List: Environmentally Responsible Supercomputing, http://www.green500.org/
[3] E. Frachtenberg, A. Heydari, H. Li, A. Michael, J. Na, A. Nisbet, P. Sarti, High-efficiency server design, in: Proceedings of the 2011 ACM/IEEE Conference on Supercomputing, IEEE Computer Society, 2011.
[4] Top500, TOP 500 Supercomputer Sites, http://www.top500.org
[5] K. Barker, et al., Entering the petaflop era: the architecture and performance of Roadrunner, in: in: Proceedings of the 2008 ACM/IEEE conference on Supercomputing, SC'08, IEEE Press, Piscataway, NJ, USA, 2008, pp. 1:1–1:11.
[6] K. Bergman, et al., Exascale computing study: technology challenges in achieving exascale systems, in: DARPA Technical Report, 2008.
[7] ARM Ltd., Cortex-A9 Processor, http://www.arm.com/products/processors/cortex-a/cortex-a9.php
[8] Cisco Systems Inc., Cisco 200 Series Switches Datasheet, http://www.cisco.com/en/US/prod/collateral/switches/ps5718/ps11229/data_sheet_c78-634369.pdf
[9] R. Weicker, Dhrystone: a synthetic systems programming benchmark, Communications of the ACM 27 (10) (1984) 1013–1030.
[10] J. McCalpin, A survey of memory bandwidth and machine balance in current high performance computers, IEEE TCCA Newsletter (1995) 19–25.
[11] J. Henning, SPEC CPU2006 benchmark descriptions, ACM SIGARCH Computer Architecture News 34 (4) (2006) 1–17.
[12] R. Whaley, J. Dongarra, Automatically tuned linear algebra software, in: Proceedings of the 1998 ACM/IEEE Conference on Supercomputing (CDROM), IEEE Computer Society, 1998, pp. 1–27.
[13] DDR2 SDRAM System-Power Calculator, http://www.micron.com/support/dram/power_calc.html
[14] A. Mahesri, V. Vardhan, Power consumption breakdown on a modern laptop, Power-Aware Computer Systems (2005) 165–180.
[15] Texas Instruments Inc., Going "beyond a faster horse" to transform mobile devices – Whitepaper, http://www.ti.com/lit/an/swpt048/swpt048.pdf, 2011.

Nikola Rajovic is a junior research fellow at Barcelona Supercomputing Center. He got his BSc (2008), MSc (2010) in electrical and computer engineering from School of Electrical Engineering, University of Belgrade, Serbia. From 2010, he is a PhD student at Universitat Polytecnica de Catalunya. His research interest is in the area of energy efficient multicore heterogeneous architectures and embedded systems.

Lluis Vilanova is a PhD student at the Barcelona Supercomputing Center and the computer architecture department at the Universitat Politecnica de Catalunya, where he also received his MSc Degree from. His interests cover computer architecture and operating systems.

Carlos Villavieja received the equivalent of the MSc degree in computer engineering from La Salle School of Engineering at Ramon Llull University in 2000 and his PhD in 2012 at UPC, Barcelona. Since 2012 he is a postdoc at the HPS group in the electrical and computer engineering at the University of Texas at Austin. His research interests are the interaction among the processor architecture and the runtime system/OS to improve applications performance, and the memory system in chip multiprocessors.

Nikola Puzovic is a senior researcher at Barcelona Supercomputing Center. He received his PhD degree (2009) in computer engineering from the University of Siena, Italy. His primary research interests are in the area of energy efficient high performance computing and in parallel programming models.

Alex Ramirez is an associate professor in the computer architecture department at the Universitat Politecnica de Catalunya, and leader of the computer architecture group at BSC. He has a BSc (1995), MSc (1997) and PhD (2002, awarded the UPC extraordinary award to the best PhD in computer science) in computer science from the Universitat Politecnica de Catalunya (UPC), Barcelona, Spain. He has been a summer student intern with Compaq's WRL in Palo Alto, CA for two consecutive years (1999–2000), and with Intel's Microprocessor Research Laboratory in Santa Clara (2001). His research interests include heterogeneous multicore architectures, hardware support for programming models, and simulation techniques. He has co-authored over 75 papers in international conferences and journals and supervised 3 PhD students.