Supercomputing with Commodity CPUs: Are Mobile SoCs Ready for HPC?

Nikola Rajovic[§‡], Paul M. Carpenter[§], Isaac Gelado[§], Nikola Puzovic[§],
Alex Ramirez[§‡], Mateo Valero[§‡]

[§]Barcelona Supercomputing Center
C/ Jordi Girona 29
08034 Barcelona, Spain

[‡]Universitat Politècnica de Catalunya
C/ Jordi Girona 1-3
08034 Barcelona, Spain

{first.last}@bsc.es

ABSTRACT

In the late 1990s, powerful economic forces led to the adoption of commodity desktop processors in high-performance computing. This transformation has been so effective that the June 2013 TOP500 list is still dominated by x86.

In 2013, the largest commodity market in computing is not PCs or servers, but mobile computing, comprising smartphones and tablets, most of which are built with ARM-based SoCs. This leads to the suggestion that once mobile SoCs deliver sufficient performance, mobile SoCs can help reduce the cost of HPC.

This paper addresses this question in detail. We analyze the trend in mobile SoC performance, comparing it with the similar trend in the 1990s. We also present our experience evaluating performance and efficiency of mobile SoCs, deploying a cluster and evaluating the network and scalability of production applications. In summary, we give a first answer as to whether mobile SoCs are ready for HPC.

1. INTRODUCTION

During the early 1990s, the supercomputing landscape was dominated by special-purpose vector and SIMD architectures. Vendors such as Cray (vector, 41%), MasPar (SIMD,[1] 11%), and Convex/HP (vector, 5%[2]) designed and built their own HPC computer architectures for maximum performance on HPC applications. During the mid to late 1990s, microprocessors used in the workstations of the day, like DEC Alpha, SPARC and MIPS, began to take over high-performance computing. About ten years later, these RISC CPUs were, in turn, displaced by the x86 CISC architecture used in commodity PCs. Figure 1 shows how the number of systems, of each of these types, has evolved since the first publication of the TOP500 list in 1993 [41].

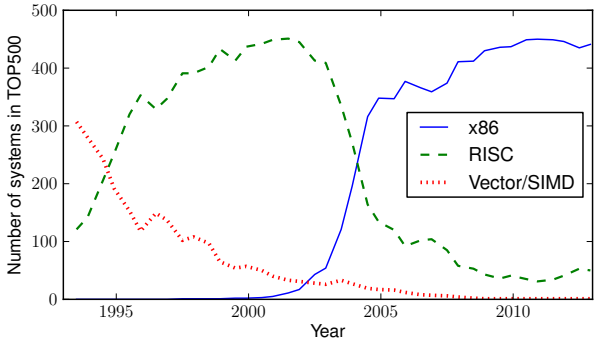


Figure 1: TOP500: Special-purpose HPC replaced by RISC microprocessors, in turn displaced by x86

Building an HPC chip is very expensive in terms of research, design, verification, and creation of photomask. This cost needs to be amortized over the maximum number of units to minimize their final price. This is the reason for the trend in Figure 1. The highest-volume commodity market, which was until the mid-2000s the desktop market, tends to drive lower-volume higher-performance markets such as servers and HPC.

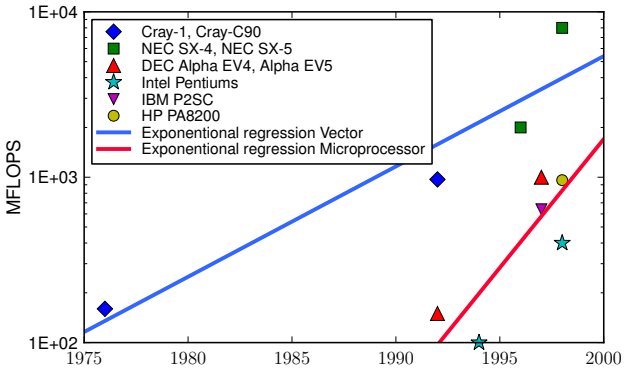
The above argument requires, of course, that lower-end commodity parts are able to attain a sufficient level of performance, connectivity and reliability. To shed some light on the timing of transitions in the HPC world, we look at the levels of CPU performance during the move from vector to commodity microprocessors. Figure 2(a) shows the peak floating point performance of HPC-class vector processors from Cray and NEC, compared with floating-point-capable commodity microprocessors. The chart shows that commodity microprocessors, targeted at personal computers, workstations, and servers were around ten times slower, for floating-point math, than vector processors, in the period 1990 to 2000 as the transition in HPC from vector to microprocessors gathered pace.

The lower per-processor performance meant that an application had to exploit ten parallel microprocessors to achieve the performance of a single vector CPU, and this required new programming techniques, including message-passing pro-

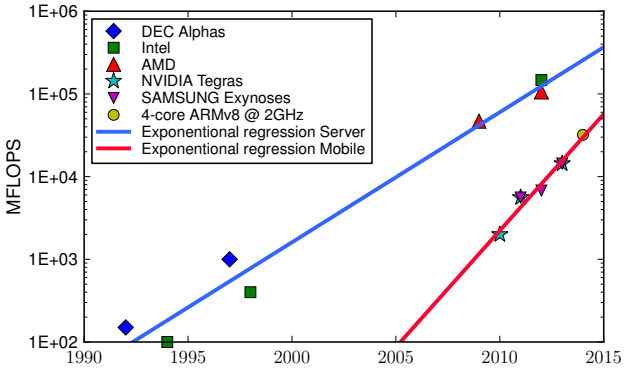
[1]SIMD: Single-Instruction Multiple Data
[2]Figures are vendor system share in June 1993 TOP500 [41].

SC '13, November 17 - 21 2013, Denver, CO, USA

http://dx.doi.org/10.1145/2503210.2503281



(a) Vector vs. commodity



(b) Commodity vs. mobile

Figure 2: Evolution of peak double-precision floating point performance over the years

gramming models such as MPI. Commodity components, however, did eventually replace special-purpose HPC parts, simply because they were 30 times cheaper. Even though a system may have required ten times as many microprocessors, it was still cheaper overall.

As a consequence, a new class of parallel computers built on commodity microprocessors and distributed memories, gained momentum. In 1997, the Intel ASCI Red supercomputer [27] became the first system to achieve 1 TFLOPS performance in the High Performance Linpack (HPL) benchmark by exploiting 7,246 parallel Intel Pentium Pro processors [41]. Most of today's HPC systems in the TOP500 are still built on the same principle: exploit a massive number of microprocessors, based on the same technology used for commodity PCs.[3] Vector processors are almost extinct, although their technology is now present in most HPC processors in the form of widening SIMD extensions, and in HPC compute accelerators—GPUs and Intel Xeon Phi.

Nowadays we observe a similar situation: low-power microprocessors targeted at mobile devices, such as smartphones and tablets, integrate enough transistors to include an on-chip floating-point unit capable of running typical HPC applications. Figure 2(b) shows the peak floating point performance of current HPC microprocessors from Intel and AMD, compared with new floating-point capable mobile processors, which are integrated into mobile SoCs[4] from NVIDIA and Samsung. The chart shows that mobile SoCs are not faster than their HPC counterparts. In fact, they are still ten times slower, but the trend shows that the gap is quickly being closed: the recently introduced ARMv8 Instruction Set Architecture (ISA), not only makes double-precision floating point (FP-64) a compulsory feature, but it also introduces it into the SIMD instruction set. That means that ARMv8 processors, using the same micro-architecture as the ARMv7 Cortex-A15, would have double the FP-64 performance at the same frequency. Furthermore, mobile SoCs are approximately 70 times cheaper[5] than their HPC counterparts, matching the trend that was observed in the past.

Given the trend discussed above, it is reasonable to consider whether the same market forces that replaced vectors with RISC microprocessors, and RISC processors with x86 processors, will replace x86 processors with mobile phone processors. That makes it relevant to study the implications of this trend before it actually happens.

We are not arguing about the superior energy efficiency of mobile processors, or about fundamental energy efficiency advantages of RISC vs CISC instruction sets. We agree that energy efficiency and performance are two axes in a design space, and that currently HPC and mobile processors have different targets, but this is subject to change at any time.

We do argue that the higher volume of the mobile market makes it easier for its vendors to amortize the costs of a new design, enables multiple vendors to survive and compete, and leads to faster product evolution, lower prices, more features, and higher performance.

If mobile processors add the required HPC features, and displace the current server processors, it will likely be due to economic reasons, rather than fundamental technology differences or superior implementation skills.

The contributions of this paper are:

- To find out whether mobile SoCs are ready for HPC, or close to being so, we evaluate the performance and energy efficiency of three mobile SoCs on their developer boards. We compare the results with a current HPC processor.

- We describe our experiences in designing and deploying HPC cluster prototypes using mobile SoCs. We encountered several problems, big and small, along the

[3]These systems represent ∼80% of the total systems in the June 2013 TOP500 list.

[4]A System-on-Chip (SoC) integrates multiple subsystems into a single chip, including general-purpose multi-core processors, caches, memory controllers, DSPs, VLIW cores, graphic processors, special purpose accelerators, I/O interfaces and analog circuitry.

[5]We compare the official tray list price of an Intel Xeon E5-2670 [22] with the leaked volume price of NVIDIA Tegra 3 [19]: $1552 vs. $21. A fairer comparison; i.e. of the same price type, would be between the recommended list price for the Xeon with an Intel Atom S1260 [20]: $1552 vs. $64 which gives the ratio of ∼24. The latter is, however, not a mobile processor but a low-power server solution from Intel, and it serves only as a comparison reference.

way, and we discuss them here for the benefit of other researchers.

- We evaluate the interconnection networks, in terms of latency and effective bandwidth. This is especially important because one potential weakness of mobile SoCs is the interconnect. We also give scalability results for ARM multicore cluster on production applications.
- We *discuss* the status of the HPC software stack on ARM processors.
- We discuss the features needed to enable mobile SoCs for HPC, for example the lack of ECC protection in memory, on-chip network controllers, or the lack of server I/O interfaces.

The rest of the paper is structured as follows. In Section 2 we compare with related work. In Section 3 we evaluate the performance, energy efficiency and memory bandwidth of a single mobile SoC. In Section 4 we investigate the performance, energy efficiency and scalability of our ARM multicore cluster. In Section 5 we discuss the maturity of the ARM software stack, and in Section 6 we discuss the problems encountered in building our Tibidabo cluster, alongside the more general limitations of mobile SoCs. Finally, in Section 7, we conclude the paper.

2. RELATED WORK

ASCI Red was a landmark supercomputing machine deployed in 1997, it was the first supercomputer to break the TFLOPS barrier [27], and it remained at number one on the TOP500 list for three years [41]. Rather than powerful vector processors, it integrated 7,246 Pentium Pro processors,[6] the first x86 devices to support integrated FP-64. From this time, as shown in Figure 1, x86 systems expanded their share of the TOP500 to their current dominance in HPC.

GreenDestiny [44] represents an early attempt to use low-power processors in HPC, using the Transmeta TM5600 processor. Although the proposal had good energy efficiency and compute density, a large-scale HPC system was never produced. MegaProto Systems [29] was another approach in this direction, based on later versions of Transmeta's processors, namely TM5800 and TM8820. These systems also achieved good energy efficiency for 2005, achieving 100 MFLOPS/W on a 512-processor system.[7] These are examples of how technical qualities may not be enough to compensate for the strengths of the commodity market.

The BlueGene family of supercomputers, introduced in 2004, and now in its third generation [1, 2, 18], introduced a new approach for energy-efficient HPC. In these machines, compute power comes from embedded cores integrated on an ASIC, together with architecture-specific interconnect fabrics. The first such system, BlueGene/L, is based on a dual-core PowerPC 440 processor at 750 MHz, extended with a dual FP-64 functional unit, for a peak floating point performance of 5.6 GFLOPS. It's successor, the BlueGene/P system, integrates a 4-core PowerPC 450 processor at 850 MHz, giving an increased peak floating point performance per compute chip of 13.6 GFLOPS. BlueGene/Q further increases the multi-core density to 16 cores per chip,[8] doubles the width of the functional unit, and increases the frequency to 1.6 GHz, delivering a peak floating point performance of 204.8 GFLOPS, in a power budget of about 55 W[16].

Indeed BlueGene represents the introduction of embedded energy-efficient processors in HPC, but it still integrates them in an HPC specific SoC, which suffers from the same low volume limitations as other HPC products. We envision those same embedded processors, integrated into a high volume mobile SoC entering HPC in the future.

FAWN was a proposal to use Intel Atom processors in clusters [43]. The cluster was built and tested with a range of workloads, but evaluation did not include a suite of true HPC workloads. The authors did a thorough study to determine the type of workloads where Intel Atom can offer a competitive energy-efficiency compared to an Intel Core i7 processor. A follow-up of this work found that a homogeneous cluster of low-power Intel Atom processors is not suited to complex database workloads [25]. The authors propose future research in heterogeneous clusters using low-power nodes combined with conventional ones.

Low-power ARM commodity processor IP is already being integrated into SoCs targeting the server market, in the form of micro-servers. Calxeda's EnergyCore ECX-1000 [7] is an ARM-based SoC targeted at data centres, with four ARM Cortex-A9 cores, five 10 GbE links, and SATA. Applied Micro's X-Gene [3] is a server-class SoC with eight ARMv8 (64-bit) cores, four 10 GbE links.

Several companies have developed micro-servers based on Intel Atom or ARM processors. AMD SeaMicro SM10000-64 is a 10U chassis with 256 dual-core Intel Atom N570 processors. Quanta Computer S900-X31A is a 3U microserver with up to 48 Intel Atom S1200 "Centerton" at 10W per node. HP's Project Moonshot is working on micro-servers in partnership with Intel, Calxeda and Texas Instruments. The first available product is the HP Moonshot 1500, which uses Intel Atom S1200. Dell is working on ARM-based servers with SoCs from Marvell, Calxeda and Applied Micro. The first product, Dell Copper is a 3U chassis with 48 Marvell quad-core ARMv7 ARM server nodes.

Unless these ARM server products achieve a large enough market share, they may follow the same path as GreenDestiny and MegaProto, and be overtaken by the larger volume commodity products, which are built on the same ARM processor IP.

Blem et al. [6] recently examined whether there were any inherent differences in performance or energy efficiency between the ARM and x86 ISAs. They emphasized the RISC (ARM) vs. CISC (x86) debate, and found that although current ARM and x86 processors are indeed optimised for different metrics, the choice of ISA had an insignificant effect. We do not contradict this finding. We argue that

[6]Later the number of processors was increased to 9,632 and processors were upgraded to Pentium II processors

[7]It would have ranked between 45 and 70 in the first edition of the Green500 list (November 2007)

[8]The BlueGene/Q compute chip has 18 cores, of which 16 are for computation, one is for running operating system services and one is a spare to increase the fabrication yield.

whichever architecture dominates the mobile industry will, provided the level of performance demanded in the mobile space is sufficient, eventually come to dominate the rest of the computing industry.

Li et al. [26] advocate the use of highly-integrated SoC architectures in the server space. They performed a design space exploration, at multiple technology nodes, for potential on-die integration of various I/O controllers, including PCIe, NICs and SATA. They predicted that, for future data centers at the 16nm technology node, an SoC architecture would bring a significant reduction in capital investment and operational costs. This paper is also advocating the use of SoCs, but for future HPC systems.

3. MOBILE SOC EVALUATION

In this section we examine the performance, energy efficiency, and memory bandwidth of a single system-on-chip (SoC). We chose developer boards with three different SoCs: NVIDIA Tegra 2 and Tegra 3, and Samsung Exynos 5250[9]. These SoCs cover two successive ARM processor microarchitectures: Tegra 2 and Tegra 3 have Cortex-A9 cores, capable of one Fused Multiply-Add (FMA) operation every two cycles, and the Samsung Exynos has Cortex-A15, with a fully-pipelined FMA. The Cortex-A15 also improves on the Cortex-A9 in terms of microarchitecture, including higher number of outstanding cache misses, longer out-of-order pipeline, and improved branch predictor [42]. The characteristics of the different SoCs and their developer kits are shown in Table 1.

In addition, we include one laptop platform, which contains the same Intel Sandy Bridge microarchitecture used in current state-of-the-art Xeon server processors.[10] We chose the laptop as a platform for comparison since the laptop integrates a set of features similar to those of mobile developer kits. In order to achieve a fair comparison in energy efficiency between the developer boards and the Intel Core i7 we boot the laptop directly into the Linux terminal, and we switch off the screen in order to reduce the non-essential power consumption. We give a quantitative measure of the difference in performance between mobile SoCs and high-performance x86 cores, which is driven by the different design points.

Mobile SoC providers are beginning to integrate compute-capable GPUs into their products. Tegra 2 and Tegra 3 have Ultra Low Power (ULP) GeForce GPUs [31], which support 1080p video with OpenGL ES 2.0. These current GPUs cannot be used for computation, but the GPU in the next product in the Tegra series, Tegra 5 ("Logan"), will support CUDA. The Samsung Exynos 5250 has a four-core Mali-T604 GPU, which supports OpenCL [23]. The SoCs used in this evaluation either have a non-programmable GPU (Tegra 2 and Tegra 3) or they do not have an optimized driver (Exynos 5). We therefore do not include the integrated GPU in the evaluation.

[9]Commercial name of this SoC is Samsung Exynos 5 Dual

[10]We used an Intel Core i7. A server-class Xeon also integrates QPI links and PCIe Gen3, but these are not relevant in this section.

3.1 Performance and energy efficiency

We measure performance and energy efficiency using our micro-kernels [34], which are listed in Table 2. In future, we plan to test multiple architectures, some of which include compute accelerators, and in such cases the effort of porting real world applications, with thousands of lines of code, is too high. In order to evaluate all these platforms, we designed this suite of micro-kernels to stress different architectural features and to cover a wide range of algorithms employed in HPC applications.

In our experiments, the problem size for the kernels is the same for all platforms, so that each platform has the same amount of work to perform in one iteration. We set the number of iterations so that the total execution time is similar for all platforms, and that the benchmark runs for long enough to get an accurate energy consumption figure.

To measure the power consumption of the platforms and energy to solution we use the Yokogawa WT230 [46] power meter. The meter is connected to act as a bridge between the power socket and the device, and it measures the power of the entire platform, including the power supply. The power meter has a sampling frequency of 10 Hz and 0.1% precision. Both power and performance are measured only for the parallel region of the application, excluding the initialization and finalization phases.[11]

3.1.1 Single-core evaluation

Figure 3 shows the single CPU performance and energy efficiency for each SoC as we vary the CPU frequency. The results are averaged across all benchmarks, and the baseline is the Tegra 2 platform running at its maximum frequency of 1GHz: for performance results we show the speedup with respect to the baseline, and for energy efficiency we normalize the results against the baseline. We can see that the performance improves linearly as the frequency is increased.

Tegra 3 brings a 9% improvement in execution time over Tegra 2 when they both run at the same frequency of 1GHz. Although the ARM Cortex-A9 core is the same in both cases, Tegra 3 has an improved memory controller which brings a performance increase in memory-intensive micro-kernels. The Arndale platform at 1 GHz shows a 30% improvement in performance over Tegra 2, and 22% over Tegra 3, due to the improved ARM Cortex-A15 microarchitecture. Compared with the Intel Core i7 CPU, the Arndale platform is just two times slower.

Averaged across all benchmarks, the Tegra 2 platform at 1GHz consumes 23.93 Joules to complete the work in one iteration. At the same frequency, Tegra 3 consumes 19.62J, giving an improvement of 19%, and Arndale consumes 16.95J, a 30% improvement. The Intel platform, meanwhile, consumes 28.57J, which is higher than all ARM-based platforms.

When we run the CPUs at their highest frequencies, instead of all at 1GHz, the Tegra 3 platform is 1.36 times faster than the Tegra 2 platform, and it requires 1.4 times less energy. The Exynos 5 SoC from the Arndale platform brings addi-

[11]It was not possible to give a fair comparison of the benchmarks including initialization and finalization, since the developer kits use NFS whereas the laptop uses its hard drive.

SoC name	NVIDIA Tegra 2	NVIDIA Tegra 3	Samsung Exynos 5250	Intel Core i7-2760QM
CPU				
Architecture	Cortex-A9	Cortex-A9	Cortex-A15	SandyBridge
Max. frequency (GHz)	1.0	1.3	1.7	2.4
Number of cores	2	4	2	4
Number of threads	2	4	2	8
FP-64 GFLOPS	2.0	5.2	6.8	76.8
GPU	Integrated (graphics only)	Integrated (graphics only)	Integrated Mali-T604 (OpenCL)	Intel HD Graphics 3000
Cache				
L1 (I/D)	32K/32K private	32K/32K private	32K/32K private	32K/32K private
L2	1M shared	1M shared	1M shared	256K private
L3	-	-	-	6M shared
Memory controller				
Number of channels	1	1	2	2
Width (bits)	32	32	32	64
Max. frequency (MHz)	333	750	800	800
Peak bandwidth (GB/s)	2.6	5.86	12.8	25.6
Developer kit				
Name	SECO Q7 module + carrier	SECO CARMA	Arndale 5	Dell Latitude E6420
DRAM size and type	1 GB DDR2-667	2 GB DDR3L-1600	2 GB DDR3L-1600	8 GB DDR3-1133
Ethernet interfaces	1 Gb, 100 Mb	1 Gb	100 Mb	1 Gb

Table 1: Platforms under evaluation

Kernel tag	**Full name**	**Properties**
vecop	Vector operation	Common operation in regular numerical codes
dmmm	Dense matrix-matrix multiplication	Data reuse and compute performance
3dstc	3D volume stencil computation	Strided memory accesses (7-point 3D stencil)
2dcon	2D convolution	Spatial locality
fft	One-dimensional Fast Fourier Transform	Peak floating-point, variable-stride accesses
red	Reduction operation	Varying levels of parallelism (scalar sum)
hist	Histogram calculation	Histogram with local privatisation, requires reduction stage
msort	Generic merge sort	Barrier operations
nbody	N-body calculation	Irregular memory accesses
amcd	Markov Chain Monte Carlo method	Embarrassingly parallel: peak compute performance
spvm	Sparce Vector-Matrix Multiplication	Load imbalance

Table 2: Micro-kernels used for platform evaluation

tional improvements in performance: it is 2.3 times faster than Tegra 2 and 1.7 times faster than Tegra 3. The Intel core at its maximum frequency is 3 times faster than the Arndale platform.

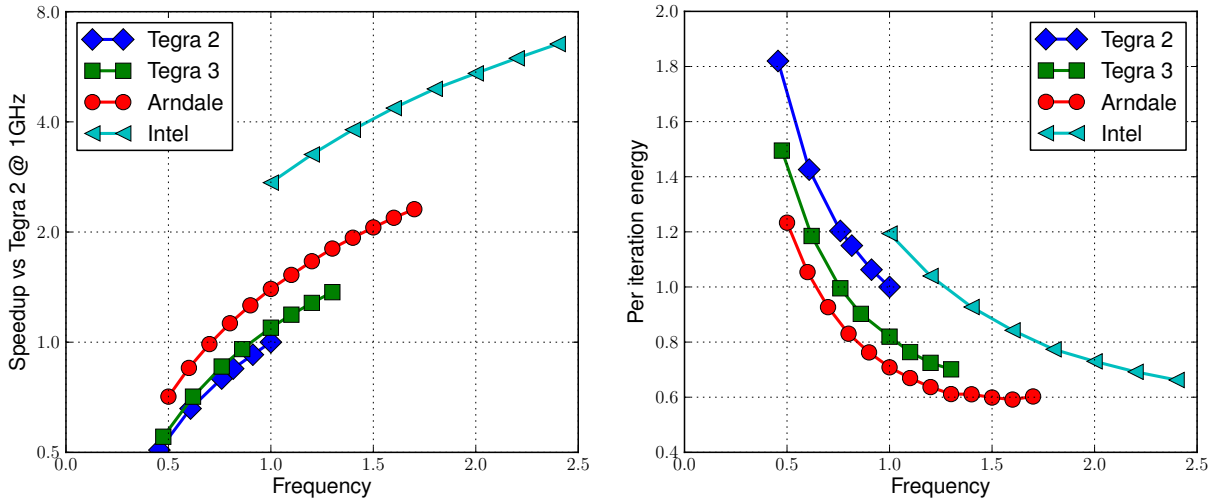
We can see that the change in generation of the ARM cores has closed the gap in performance with respect to their Intel counterparts. From the situation when Tegra 2 was 6.5 times slower we have arrived to the position where Exynos 5 is just 3 times slower than the Sandy Bridge core.

3.1.2 Multi-core evaluation

Figure 4 shows the results for the OpenMP version of the micro-benchmarks. The micro-benchmarks always use all the cores that are available in the platform (two in Tegra 2, four in Tegra 3, two in Arndale and four in Sandy Bridge). In all cases, multithreaded execution has brought improvements, both in performance and in energy efficiency, with respect to the serial version of the micro-benchmarks. In case of Tegra 2 and Tegra 3 platforms, the OpenMP version uses 1.7 times less energy per iteration. Arndale shows better improvement (2.25 times), while the Intel platform reduces energy to solution 2.5 times.

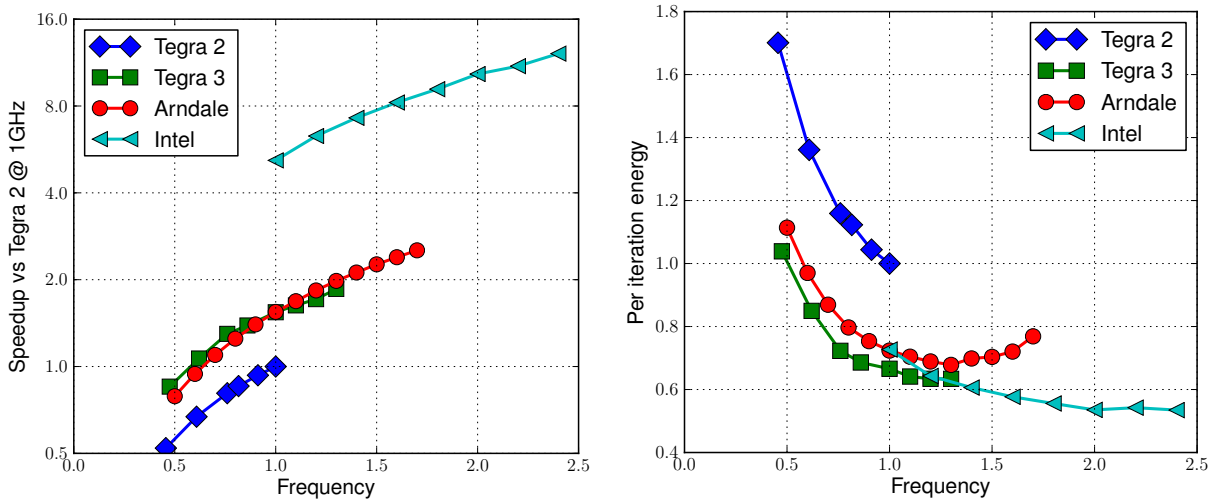
Our energy efficiency results show that for all platforms the SoC is not the main power sink in the system. When we increase the frequency of the CPU, its power consumption increases (at least) linearly with the frequency, but we see that the overall energy efficiency improves. This leads to conclusion that the majority of the power is used by other components.

Based on these results, we can conclude that if the trend in performance increase is continued, future mobile processors will continue to exhibit significant performance and energy efficiency improvements. The performance will increase with quad-core Cortex-A15 products running at higher frequencies (for example, Tegra 4 or Exynos 5 Octa will have four Cortex-A15 cores). The introduction of the ARMv8 ISA will bring a further increase in performance, due to the introduction of double-precision floating point support in the NEON vector unit [15]. This would bring double the performance, while keeping the power of the core itself at almost the same level.



(a) Performance (b) Energy

Figure 3: Single-core performance and energy efficiency results of micro-benchmarks with frequency sweep. Baseline is the Tegra 2 platform running at 1GHz.



(a) Performance (b) Energy

Figure 4: Multi-core performance and energy efficiency results of micro-benchmarks with frequency sweep. Baseline is the Tegra 2 platform running on 1GHz.

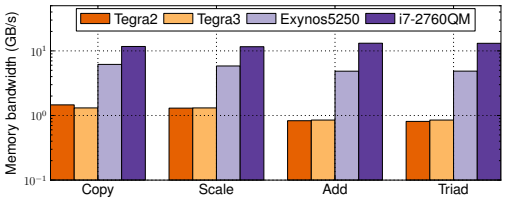
3.2 Memory bandwidth

Figure 5 shows the memory bandwidth for each platform, measured using the STREAM benchmark [28].

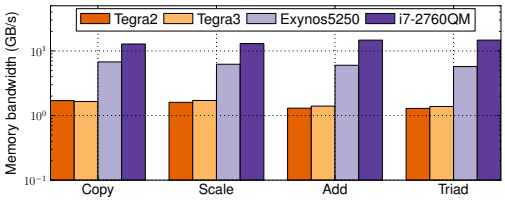
Our results show a significant improvement in memory bandwidth, of about 4.5 times, between the Tegra platforms (ARM Cortex-A9) and the Samsung Exynos 5250 (ARM Cortex-A15). This appears to be mostly due to the better Cortex-A15 microarchitecture which also improves the number of outstanding memory requests [42], and due to an additional channel in memory controller. Compared with the peak memory bandwidth, the multicore results imply an efficiency of 62% (Tegra 2), 27% (Tegra 3), 52% (Exynos 5250), and 57% (Intel Core i7-2760QM).

4. PARALLEL SCALABILITY

In this section we investigate the performance, energy efficiency and scalability of an ARM multicore cluster on production applications. Our single-node performance evaluation in Section 3.1 shows that the Tegra 2 is almost eight times slower than the Intel Core i7, both at their maximum operating frequencies, so the applications must be able to exploit a minimum of eight times as many parallel processors in order to achieve competitive time-to-solution. Fortunately, as described in the previous section, newer ARM SoCs are narrowing the gap. In this section, we also evaluate in detail the interconnection networks available on Tegra 2 and Exynos 5 platforms, in terms of latency and effective bandwidth.



(a) single core



(b) MPSoC

Figure 5: Memory bandwidth

Tibidabo [33] is the first large-scale cluster to be deployed using multi-core ARM-based SoCs. Tibidabo has 192 nodes, each with an Nvidia Tegra 2 SoC on a SECO Q7 module. Each node has a 1 GbE NIC, connected to the Tegra 2 SoC over PCIe. The cluster has a hierarchical 1 GbE network built with 48-port 1 GbE switches, giving a bisection bandwidth of 8 Gb/s and a maximum latency of three hops.

The scalability tests were performed on Tibidabo using the MPI applications listed in Table 3. High-Performance Linpack (HPL) [9] solves a random dense linear system of equations in double precision, and is widely known as the single benchmark used in the TOP500 list. PEPC [8] computes long-range Coulomb forces for a set of charged particles; HYDRO is a 2D Eulerian code for hydrodynamics based on the RAMSES code [40]; GROMACS [5] is a versatile package for molecular dynamics simulations; and SPECFEM3D_GLOBE [24] simulates continental and regional scale seismic wave propagation. Following common practice, we perform a weak scalability test for HPL and a strong scalability test for the rest.[12] All applications were compiled and executed out-of-the-box, without any tuning of the source code.

The results of the scalability tests are shown in Figure 6. Some of the strong scaling benchmarks cannot run on a single node, because they need too much memory. In this case, we calculated the speed-up assuming linear scaling on the smallest number of nodes that could execute the benchmark. For example, PEPC with the reference input set requires at least 24 nodes; so the results were plotted assuming that on 24 nodes the speed-up was 24.

SPECFEM3D shows good strong scaling, using an input set that fits in the memory of a single node. HYDRO, however, starts losing linear strong scalability trend after 16 nodes. GROMACS was executed using an input that fits in the memory of two nodes; its scalability improves as the input size is increased. PEPC also shows relatively poor strong scalability partly because the input set that we can fit on our cluster is too small [8].

[12]Under strong scaling, the total problem size is held constant as the number of nodes is varied. Under weak scaling, the problem size *per node* is held constant.

Application	Description
HPL	High-Performance LINPACK
PEPC	Tree code for N-body problem
HYDRO	2D Eulerian code for hydrodynamics
GROMACS	Molecular dynamics
SPECFEM3D	3D seismic wave propagation (spectral element method)

Table 3: Applications for scalability evaluation

In our previous study [35] we presented strong scaling results of HPL on 32 nodes of the Tibidabo cluster, with input sets that fit in the memory of one to four nodes. We observed that a change in the input set size affects the scalability—the bigger the input set the better the scalability. We looked further into the problem and discovered timeouts in post-mortem application trace analysis. These results, together with the poor strong scalability of other applications, motivate the further evaluation of the interconnect in Subsection 4.1.

Tibidabo shows good weak scaling on HPL, achieving a total 97 GFLOPS on 96 nodes and an efficiency of 51%. In order to determine the MFLOPS/W metric, which is used for ranking systems in the Green500 list [38], we also measured the system's power consumption while executing HPL, giving an energy efficiency of 120 MFLOPS/W. This is competitive with AMD Opteron 6174 and Intel Xeon E5660-based clusters, but is nineteen times lower than the most efficient homogeneous non-accelerated systems (BlueGene/Q), and almost 27 times lower than the number one GPU-accelerated system in the June 2013 Green500 list (Eurotech Eurora cluster, Intel Xeon E5-2687W and NVIDIA K20 GPU). The reasons for the low energy efficiency include the use of developer kits,[13] low multi-core density, the lack of a compute-capable GPU, no vendor-provided (tuned) linear algebra library, and no optimization of the MPI communication stack.

In our previous study [13], we compared energy-efficiency and performance of Tibidabo and an Intel Nehalem-based cluster. This was done using three different classes of numerical solvers for partial differential equations (including SPECFEM3D). All applications used in this study showed linear scalability. In addition, we found that while Tibidabo had a 4 times increase in simulation time, it achieved up to 3 times lower energy-to-solution.

4.1 Interconnection network

In order to improve the latency and bandwidth of the ARM clusters, we wanted to know whether a large overhead was being introduced by the TCP/IP software stack. We therefore compared TCP/IP with a direct communication Ethernet protocol called Open-MX [14]. Open-MX is a high-performance implementation of the Myrinet Express message-passing stack that works over ordinary Ethernet networks.

[13]Many of the components on the Q7 board are not required for HPC, and would be removed in a production system to reduce power consumption; e.g. multimedia adaptors and related circuitry, USB, HDMI, SATA, and RTC.

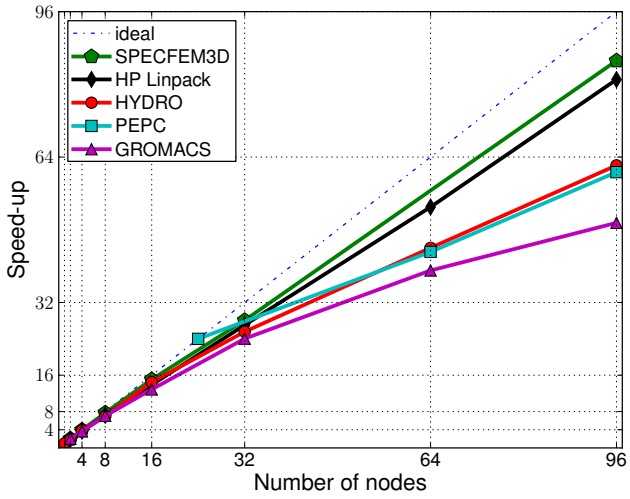


Figure 6: Scalability of HPC applications on Tibidabo.

Platform	FP64 Bytes/FLOPS		
	1GbE	10GbE	40Gb InfiniBand
Tegra 2	0.06	0.63	2.50
Tegra 3	0.02	0.24	0.96
Exynos 5250	0.02	0.18	0.74
Intel Sandy Bridge	0.00	0.02	0.07

Table 4: Network bytes/FLOPS ratios (FP64, excluding GPU)

It integrates seamlessly with the OpenMPI message passing library, which was used for these tests.

Open-MX bypasses the heavyweight TCP/IP stack and reduces the number of memory copies as much as possible, both of which reduce latency, CPU load, and cache pollution. For large messages, over 32kB, it uses rendezvous and memory pinning to achieve zero copy on the sender side and single copy on the receiver side. All actual communication management is implemented in the user-space library, with a kernel driver that takes care of initialization, memory registration, and passing and receiving raw Ethernet messages.

The latency and bandwidth results were measured using the ping-pong test from the Intel MPI Benchmark suite [21]. The ping-pong test measures the time and bandwidth to exchange one message between two MPI processes. Figure 7 shows the results, for (a) two SECO Q7 boards with Tegra 2 at 1GHz, (b) two Arndale boards with Exynos 5 at 1.0GHz, and (c) two Arndale boards at 1.4GHz. In all cases we used 1GbE links; on SECO boards the network controller is connected via PCI Express and on Arndale it is connected via a USB 3.0 port.

From Figure 7(a), it can be seen that the latency of Tegra 2 with TCP/IP is around 100μs, which is large compared to today's top HPC systems. When Open-MX is used, the latency drops to 65μs. Arndale running at 1GHz shows a higher latency (Figure 7(b)), on the order of 125μs with TCP/IP and 93μs when Open-MX is used. When the frequency is increased to 1.4GHz (Figure 7(b)), latencies are reduced by 10%.

Although the Exynos 5 gives better performance than Tegra 2, all network communication has to pass through the USB software stack and this yields higher latency, both with MPI and Open-MX. When the frequency of the Exynos 5 SoC is increased, the latency decreases, which indicates that a large part of the overhead is caused by software, rather than the network controller and the network itself. Hence, it is crucial to reduce this overhead either by using more agile software solutions, such as Open-MX, or by introducing hardware support to accelerate the network protocol. Some SoCs, such as Texas Instrument's KeyStone II [39] already implement protocol accelerators.

A recent study found that, for a CPU with performance similar to the Intel Sandy Bridge, a total communications latency of 100μs translates to a 90% higher execution time compared to the baseline case with zero latency [36] (geometric mean across nine benchmarks, with between 64 and 256 nodes).[14] A total latency of 65μs translates to a 60% higher execution time. If we assume the single core performance difference between the Arndale and Intel Core i7 platforms from Figure 3(a), the first order estimate is that latency would penalize execution time with approximately 50% and 40% for the aforementioned latencies respectively.

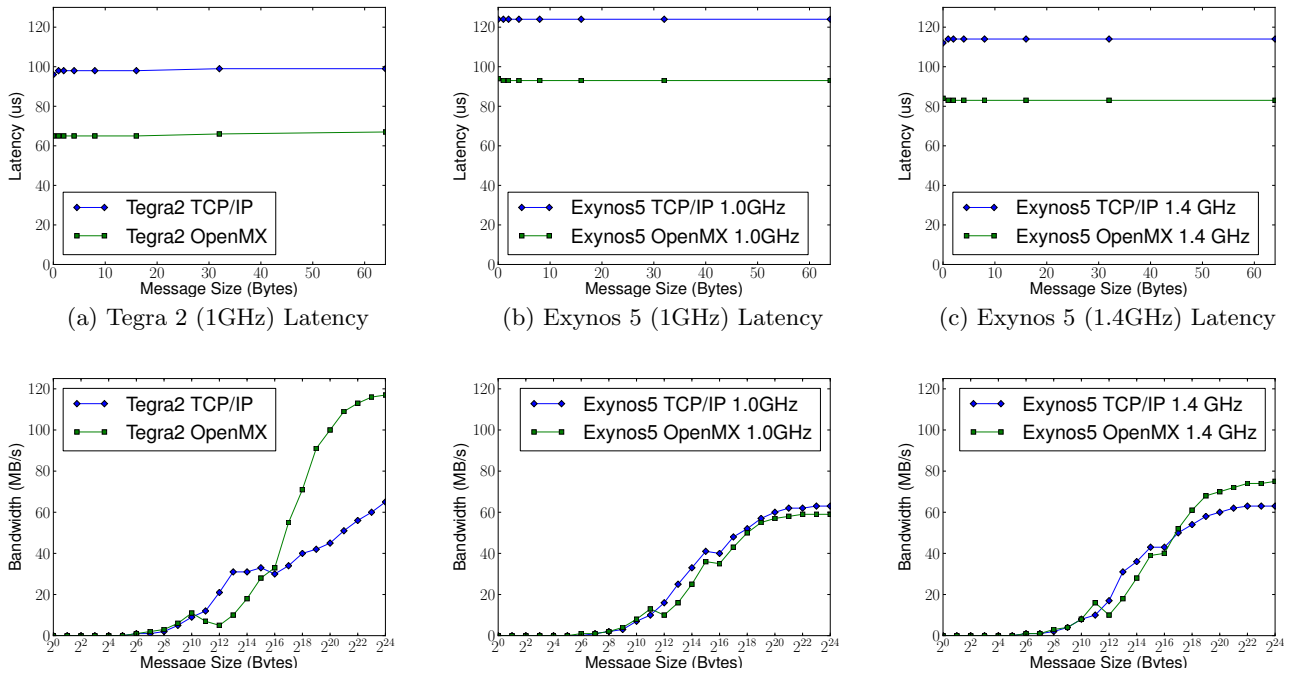
Figure 7 shows the effective bandwidth achieved as a function of the message size. The maximum bandwidth that can be achieved on the 1GbE link is 125 MB/s, and with MPI over TCP/IP none of the platforms is achieving it. In this case Tegra 2 can achieve 65 MB/s, and Exynos 5 can achieve 63 MB/s – utilizing less than 60% of the available bandwidth. When Open-MX is employed, the situation improves significantly for Tegra 2 that is now capable of reaching 117 MB/s – 93% of the theoretical maximum bandwidth of the link. Due to the overheads in the USB software stack, Exynos 5 shows smaller bandwidth than Tegra 2, but with an improvement over TCP/IP: 69 MB/s running at 1GHz and 75 MB/s running at 1.4GHz.

Table 4 shows the ratio of network bandwidth (in bytes per second) divided by peak double-precision performance (from Table 1). The figures include all CPU cores but exclude the GPU. This is because the only platform with a compute-capable GPU is the Samsung Exynos 5250, but ARM has not disclosed the Mali-T604's peak FP64 performance. A 1GbE network interface for a Tegra 3 or Exynos 5250 has a bytes/FLOPS ratio close to that of a dual-socket Intel Sandy Bridge.

5. SOFTWARE STACK

The software stack available on our ARM clusters is the same as would be found on a normal HPC cluster. The important components are shown in Figure 8. The OmpSs compiler is based on the Mercurium source-to-source compiler [4]. The GNU Compiler Collection has full support for ARM, in gcc, gfortran and g++. Runtime libraries include MPICH2, OpenMPI, OpenMX, CUDA [30], Nanos++, and

[14]The benchmarks were CPMD, gadget, GROMACS, HPL, MILC, NAMD, PEPC, Quantum, and WRF.



(a) Tegra 2 (1GHz) Latency (b) Exynos 5 (1GHz) Latency (c) Exynos 5 (1.4GHz) Latency

(d) Tegra 2 (1GHz) Bandwidth (e) Exynos 5 (1GHz) Bandwidth (f) Exynos 5 (1.4GHz) Bandwidth

Figure 7: Interconnect measurements

libGOMP. Developer tools include the Paraver [32] tracing tool and Allinea DDT debugger. Each node was also configured to run a SLURM client for job scheduling across the cluster nodes.

The operating system was a custom deployment of a standard Debian/*armhf* distribution. We built custom Linux kernels for each platform from the sources provided by NVIDIA for the Tegra platforms, and from Linaro for the Arndale board. All Linux kernels were tuned for HPC by selecting the non-preemptive scheduler, setting the default DVFS policy to *performance*. Furthermore, we stripped out all the kernel support for multimedia and interactive devices, and enabled the profiling support to access hardware performance counters and support to mount the node root filesystem by NFS.

We have natively compiled the ATLAS [45], FFTW [12], and HDF5 [11] scientific libraries. These libraries were compiled using the combination of flags that better suited each platform. The compilation of ATLAS required modifications to the source code to support the interface for processor identification provided by the ARM Linux kernel. Moreover, ATLAS required us to fix the processor frequency to its maximum to ensure that the auto-tuning steps of this library produced reliable results.

A modified version of CUDA 4.2 provided by NVIDIA is also installed in the nodes based on the CARMA kit. This installation also required the compilation of the NVIDIA kernel driver. Currently, the CUDA runtime is only provided for the *armel* ABI, so we have also deployed a Debian/*armel* filesystem that can be used to run CUDA applications, at the cost of a lower CPU performance. The installed CUDA runtime is still experimental and the performance of CUDA application is far from optimal, so this filesystem is only used for experimentation.

We have also deployed an experimental filesystem and a old version of the Linux kernel provided by ARM for the cluster based on Arndale boards. This configuration allows applications to run OpenCL code on the Mali GPU included in this platform. This experimental OpenCL driver and runtime is still on early stages of development, so it suffers from stability and performance issues. Moreover, the old version of the Linux kernel does not currently implement the thermal support for the Exynos 5 Dual MPSoC, so the chip cannot be clocked to frequencies higher than 1GHz.

NVIDIA and ARM are currently working on improving these experimental CUDA and OpenCL support. We expect support for using compute accelerators in our clusters to be available in the short-term.

6. CHALLENGES IN ARM HPC

In this section, we describe our experience and lessons learnt in designing and deploying the *Tibidabo* prototype HPC system. We encountered several problems, both big and small, and we discuss the most important ones for the benefit of other researchers. We also discuss the general limitations of current mobile SoCs that must be addressed before they can be suitable for large-scale HPC systems.

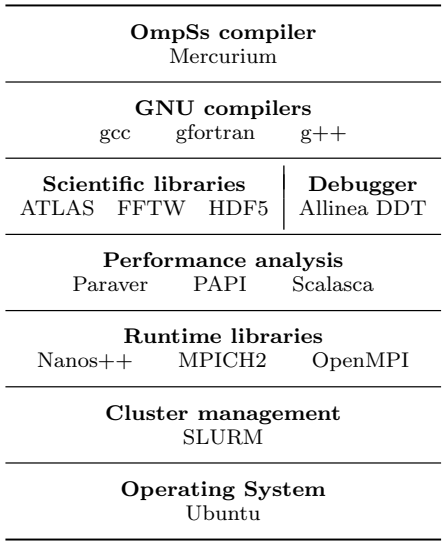


Figure 8: Software stack deployed on ARM-based clusters

6.1 Hardware

First, developer kits for mobile SoCs are not designed for continued high-performance operation. There is no cooling infrastructure (heatsink or fan), and after continued use at the maximum frequency, both the SoC and power supply circuitry overheat, causing the board to become unstable. Packaging and cooling have to be planned in advance with real TDPs and power consumption profiles provided from the vendors.

Second, the integrated PCIe in Tegra 2 and Tegra 3 was unstable. Sometimes the PCIe interface failed to initialize during boot, and sometimes it stopped responding when used under heavy workloads. The consequence was that the node crashed, and post-mortem analysis was not possible. Stability of I/O interfaces and their OS drivers should become a priority from the SoC vendors.

6.2 Software

First, hardware floating point (VFP) is still an optional feature in ARMv7. For backwards compatibility, most Linux distributions still use *soft-float* calling conventions. User programs must be compiled with the *softfp* ABI, which uses hardware instructions inside functions, but function calls still pass floating-point values through the integer registers, reducing performance accordingly. We had to build our own *hardfp* Linux images, which required new kernels to be provided by the SoC vendors.

Second, the low 100Mbit Ethernet bandwidth was not enough to support the NFS traffic in the I/O phases of the applications, resulting in timeouts, performance degradation and even application crashes. This required application changes to serialize the parallel I/O in the applications, and in some cases this limited the maximum number of nodes that application could utilize.

6.3 Mobile SoC limitations

Although performance and energy efficiency results suggest that mobile SoCs are ready for HPC, certain limitations must be addressed before a production system is viable.

First, the memory controller does not support ECC protection in the DRAM. A Google study in 2009 found that, within a year, 4% to 20% of all DIMMs encounter a correctable error [37]. There was considerable variation between platforms, but these figures suggest that a 1,500 node system, with 2 DIMMs per node, has a 30% error probability on any given day. However, ECC protection is not required for PCs and mobile devices.

Second, the lack of high bandwidth I/O interfaces in mobile SoCs prevents the use of high-bandwidth interconnects such as 10-40 GbE, or QDR-FDR Infiniband. The available interfaces are USB 3.0, which is 5Gb/s, and MIPI and SATA3, which are both 6Gb/s. Given the lower per-node performance, the balance between I/O and GFLOPS is still adequate, but will fall behind as soon as compute performance increases.

Third, the lack of integrated network interfaces and/or PCIe, and the lack of hardware support for interconnect protocols, leads to high network latencies and high CPU overheads. Poor network performance will not be critical for applications that communicate infrequently, or that use large (bandwidth-bound) messages. But it will severely impact applications with frequent and short messages. These overheads can be alleviated to some extent using latency-hiding programming techniques and runtimes [10].

Fourth, ARMv7 is a 32-bit architecture, which limits the addressable memory per process to 4GB. The ARM Cortex-A15 provides a 40-bit physical address space, via the Large Physical Address Extensions (LPAE), which potentially supports 1TB of physical memory per node; each process is still limited to 4GB. Current mobile SoCs, however, still integrate 32-bit buses and memory controllers, so they are unable to take advantage of this feature.

All these limitations are design decisions, related to features that are not required for smartphones or tablets, rather than real technical limitations. For example, the Calxeda EnergyCore, and the TI Keystone II, integrate ECC-capable memory controllers, the EnergyCore and X-Gene also integrate multiple 10 Gb/s Ethernet interfaces. The KeyStone II even has a protocol off-load engine. Finally, 64-bit ARMv8 ISA already provides a 42-bit address space, and the requirement for tablets and smartphones to grow beyond 4GB of memory will progressively lead to its adoption.

As soon as the inclusion of such features creates revenues from the server and HPC markets to compensate for their extra cost, these design decisions may change, and the above features may appear in the next generation of mobile SoCs.

A final point is that mobile SoCs will probably always contain special-purpose processors and hardware that are not necessary for HPC. Examples include audio processor, video encode and decode processor, display controllers, HDMI, and even integrated baseband processors. It is crucial that these subsystems can be powered off when they are not being used, in order to avoid unnecessary power consumption. We have already argued that mobile SoCs enjoy a cost advantage, and this is still true, even though these features increase chip area. Mobile SoCs may evolve into multi-use solutions with different operating modes, including HPC sup-

port through the features listed above: NICs, ECC, PCIe and multi-socket support via interfaces like QPI. The relevant blocks, mobile, server or HPC, would be powered up depending on the mode of operation.

7. CONCLUSIONS

We have shown that mobile processors have promising qualities that make them candidates for HPC in the near future. Applications that scale to a larger number of parallel nodes may benefit from a lower cost platform with competitive performance and energy efficiency. These application requirements are similar to those of BlueGene systems. Since mobile processors are driven by commodity component business dynamics, they have more aggressive roadmaps, and faster innovation, and are driven by a rapidly growing market that is demanding more and more performance.

Although performance and energy consumption suggest that mobile SoCs are becoming ready for HPC, the limitations described in Section 6.3 must be addressed before a production system is viable: lack of ECC protection, slow interconnect, 32-bit address space, and low-grade thermal package. All these limitations are design decisions, since the missing features are needed for HPC and servers but not smartphones and tablets. We encourage the providers of mobile SoCs to consider adding the necessary support to their products.

Another advantage is that embedded processors, and mobile processors in particular, are likely to become early adopters of 3D memory packaging. This approach is being considered for next-generation HPC systems, including the future NVIDIA Volta [17] GPU. In the mobile domain, the higher cost of 3D packaging of the CPU and memory can be amortized across a large market, while in HPC, because of the lower volume, a greater cost will be passed on to the end user.

It is possible that a new class of server SoC will be successful, built on the same basic technologies as mobile SoCs, and implementing the missing features. Alternatively, mobile vendors may decide to include the minimal set of features required to target the server and HPC markets. Either way, the cost of supercomputing may be about to fall because of the descendants of today's mobile SoCs.

8. ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their comments. We would also like to thank Gabor Dodza from ARM for his help with benchmarking the interconnect. This project and the research leading to these results was supported by the Mont-Blanc project (European Community's Seventh Framework Programme [FP7/2007-2013] under grant agreement n^o 288777), the PRACE project (European Community funding under grants RI-261557 and RI-283493), the Spanish Ministry of Science and Technology through *Computacion de Altas Prestaciones (CICYT) VI* (TIN2012-34557), and the Spanish Government through *Programa Severo Ochoa* (SEV-2011-0067).

9. REFERENCES

[1] N. R. Adiga, G. Almási, G. S. Almasi, Y. Aridor, R. Barik, D. Beece, R. Bellofatto, G. Bhanot, et al. An overview of the BlueGene/L supercomputer. In *ACM/IEEE 2002 Conference on Supercomputing*. IEEE Computer Society, 2002.

[2] S. Alam, R. Barrett, M. Bast, M. R. Fahey, J. Kuehn, C. McCurdy, J. Rogers, P. Roth, R. Sankaran, J. S. Vetter, P. Worley, and W. Yu. Early evaluation of IBM BlueGene/P. In *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, SC '08, pages 23:1–23:12, Piscataway, NJ, USA, 2008. IEEE Press.

[3] Applied Micro. APM "X-Gene" Launch Press Briefing. `http://www.apm.com/global/x-gene/docs/X-GeneOverview.pdf`, 2012.

[4] J. Balart, A. Duran, M. Gonzàlez, X. Martorell, E. Ayguadé, and J. Labarta. Nanos Mercurium: a research compiler for OpenMP. In *Proceedings of the European Workshop on OpenMP*, volume 8, 2004.

[5] H. Berendsen, D. van der Spoel, and R. van Drunen. Gromacs: A message-passing parallel molecular dynamics implementation. *Computer Physics Communications*, 91(1):43–56, 1995.

[6] E. Blem, J. Menon, and K. Sankaralingam. Power Struggles: Revisiting the RISC vs. CISC Debate on Contemporary ARM and x86 Architectures. In *19th IEEE International Symposium on High Performance Computer Architecture (HPCA 2013)*, 2013.

[7] Calxeda. Calxeda EnergyCore ECX-1000 Series. `http://www.calxeda.com/wp-content/uploads/2012/06/ECX1000-Product-Brief-612.pdf`, 2012.

[8] DEISA project. DEISA 2; Distributed European Infrastructure for Supercomputing Applications; Maintenance of the DEISA Benchmark Suite in the Second Year. Technical report.

[9] J. Dongarra, P. Luszczek, and A. Petitet. The LINPACK Benchmark: past, present and future. *Concurrency and Computation: Practice and Experience*, 15(9):803–820, 2003.

[10] A. Duran, E. Ayguade, R. M. Badia, J. Labarta, L. Martinell, X. Martorell, and J. Planas. OmpSs: a proposal for programming heterogeneous multi-core architectures. *Parallel Processing Letters*, 21(02):173–193, 2011.

[11] M. Folk, A. Cheng, and K. Yates. HDF5: A file format and I/O library for high performance computing applications. In *Proceedings of Supercomputing*, volume 99, 1999.

[12] M. Frigo and S. G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. Special issue on "Program Generation, Optimization, and Platform Adaptation".

[13] D. Göddeke, D. Komatitsch, M. Geveler, D. Ribbrock, N. Rajovic, N. Puzovic, and A. Ramirez. Energy efficiency vs. performance of the numerical solution of PDEs: an application study on a low-power ARM-based cluster. *Journal of Computational Physics*, 237:132–150, Mar. 2013.

[14] B. Goglin. High-Performance Message Passing over generic Ethernet Hardware with Open-MX. *Elsevier Journal of Parallel Computing (PARCO)*, 37(2):85–100, Feb. 2011.

[15] R. Grisenthwaite. ARMv8 Technology Preview. `http://www.arm.com/files/downloads/ARMv8_Architecture.pdf`, 2011.

[16] R. Haring, M. Ohmacht, T. Fox, M. Gschwind, D. Satterfield, K. Sugavanam, P. Coteus,

P. Heidelberger, M. Blumrich, R. Wisniewski, a. gara, G. Chiu, P. Boyle, N. Chist, and C. Kim. The IBM Blue Gene/Q Compute Chip. *Micro, IEEE*, 32(2):48 –60, march-april 2012.

[17] J.-H. Huang. GTC 2013 Keynote, March 2013.

[18] IBM Systems and Technology. IBM System Blue Gene/Q Data Sheet, November 2011.

[19] IHS iSuppli News Flash. Low-End Google Nexus 7 Carries $152 BOM, Teardown Reveals. `http://www.isuppli.com/Teardowns/News/pages/Low-End-Google-Nexus-7-Carries-\$157-BOM-Teardown-Reveals.aspx`.

[20] Intel. Intel ATOM S1260. `http://ark.intel.com/products/71267/Intel-Atom-Processor-S1260-1MB-Cache-2_00-GHz`.

[21] Intel. Intel MPI Benchmarks 3.2.4. `http://software.intel.com/en-us/articles/intel-mpi-benchmarks`.

[22] Intel. Intel Xeon Processor E5-2670. `http://ark.intel.com/products/64595/Intel-Xeon-Processor-E5-2670-20M-Cache-2_60-GHz-8_00-GTs-Intel-QPI`.

[23] Khronos OpenCL Working Group. The OpenCL Specification, version 1.0.29. `http://khronos.org/registry/cl/specs/opencl-1.0.29.pdf`, 2008.

[24] D. Komatitsch and J. Tromp. Introduction to the spectral element method for three-dimensional seismic wave propagation. *Geophysical Journal International*, 139(3):806–822, 1999.

[25] W. Lang, J. Patel, and S. Shankar. Wimpy node clusters: What about non-wimpy workloads? In *Proceedings of the Sixth International Workshop on Data Management on New Hardware*, pages 47–55. ACM, 2010.

[26] S. Li, K. Lim, P. Faraboschi, J. Chang, P. Ranganathan, and N. P. Jouppi. System-level integrated server architectures for scale-out datacenters. In *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 260–271. ACM, 2011.

[27] T. Mattson and G. Henry. An Overview of the Intel TFLOPS Supercomputer. *Intel Technology Journal*, 2(1), 1998.

[28] J. D. McCalpin. Memory bandwidth and machine balance in current high performance computers. *IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter*, pages 19–25, Dec. 1995.

[29] H. Nakashima, H. Nakamura, M. Sato, T. Boku, S. Matsuoka, D. Takahashi, and Y. Hotta. Megaproto: 1 TFlops/10kW rack is feasible even with only commodity technology. In *Proceedings of the ACM/IEEE SC 2005 Conference on Supercomputing*. IEEE, 2005.

[30] NVIDIA. CUDA Programming Guide 2.2, 2009.

[31] NVIDIA. Bringing High-End Graphics to Handheld Devices, 2011.

[32] V. Pillet, J. Labarta, T. Cortes, and S. Girona. Paraver: A tool to visualize and analyze parallel code. *WoTUG-18*, pages 17–31, 1995.

[33] N. Rajovic, A. Rico, N. Puzovic, C. Adeniyi-Jones, and A. Ramirez. Tibidabo: Making the case for an ARM-based HPC system. *Future Generation Computer Systems*, 2013. DOI: http://dx.doi.org/10.1016/j.future.2013.07.013.

[34] N. Rajovic, A. Rico, J. Vipond, I. Gelado, N. Puzovic, and A. Ramirez. Experiences With Mobile Processors for Energy Efficient HPC. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 464–468, 2013.

[35] N. Rajovic, L. Vilanova, C. Villavieja, N. Puzovic, and A. Ramirez. The low power architecture approach towards exascale computing. *Journal of Computational Science*, 2013. DOI: http://dx.doi.org/10.1016/j.jocs.2013.01.002.

[36] K. P. Saravanan, P. M. Carpenter, and A. Ramirez. Power/Performance evaluation of Energy Efficient Ethernet (EEE) for High Performance Computing. In *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, Austin, TX, April 2013.

[37] B. Schroeder, E. Pinheiro, and W.-D. Weber. DRAM errors in the wild: a large-scale field study. In *Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems*, pages 193–204. ACM, 2009.

[38] B. Subramaniam and W.-c. Feng. The Green Index: A Metric for Evaluating System-Wide Energy Efficiency in HPC Systems. In *8th IEEE Workshop on High-Performance, Power-Aware Computing (HPPAC)*, Shanghai, China, May 2012.

[39] Texas Instruments. AM5K2E04/02 Multicore ARM KeyStone II System-on-Chip (SoC) Data Manual, November 2012.

[40] R. Teyssier. Cosmological hydrodynamics with adaptive mesh refinement. *Astronomy and Astrophysics*, 385(1):337–364, 2002.

[41] TOP500. Top500®supercomputer cites. `http://www.top500.org/`.

[42] J. Turley. Cortex-A15 "Eagle" flies the coop. *Microprocessor Report*, 24(11):1–11, November 2010.

[43] V. Vasudevan, D. Andersen, M. Kaminsky, L. Tan, J. Franklin, and I. Moraru. Energy-efficient cluster computing with fawn: Workloads and implications. In *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, pages 195–204. ACM, 2010.

[44] M. Warren, E. Weigle, and W. Feng. High-density computing: A 240-processor beowulf in one cubic meter. In *Supercomputing, ACM/IEEE 2002 Conference*, pages 61–61. IEEE, 2002.

[45] R. Whaley and J. Dongarra. Automatically tuned linear algebra software. In *Proceedings of the 1998 ACM/IEEE conference on Supercomputing (CDROM)*, pages 1–27. IEEE Computer Society, 1998.

[46] Yokogawa. WT210/WT230 Digital Power Meters. `http://tmi.yokogawa.com/products/digital-power-analyzers/digital-power-analyzers/wt210wt230-digital-power-meters/`.