




The survey on ARM processors for HPC

Daniel Yokoyama¹ · Bruno Schulze¹ · Fábio Borges¹  · Giacomo Mc Evoy²

Published online: 8 June 2019

© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

The ongoing effort to reach the exascale computing barrier has led to a myriad of research and publications in the topic of alternative energy-efficient architectures, such as ARM, for HPC systems. The staggering pace at which ARM architectures have evolved has increased the volume of publications on this topic even more. A complex subject as the race to exascale touches on several aspects such as floating-point performance, scalability issues in coupled workloads, net energy consumption and ratio of energy to performance. In this context, we see the opportunity to contribute to this subject by: (1) analyzing the state of the art to identify essential papers; (2) highlighting important developments of ARM architecture in support to HPC; (3) discussing both positive and negative trends observed regarding the use of ARM for HPC; and (4) listing key topics concerning the use of ARM for exascale computing, along with distinguished references for each one.

Keywords Exascale · ARM · High-performance computing · Energy efficiency · Heterogeneous computing

1 Introduction

For High-Performance Computing (HPC), the “performance” is crucial. Performance to solve larger problems faster is the driving force guiding its development. The exascale computing barrier (10^{18} floating-point operations per second (flops)) is

✉ Fábio Borges
borges@lncc.br

Daniel Yokoyama
dmassami@lncc.br

Bruno Schulze
schulze@lncc.br

Giacomo Mc Evoy
giacomo.valenzano@atos.net

¹ Laboratório Nacional de Computação Científica (LNCC), Petrópolis, Brazil

² R&D Atos/Bull (ARM Software Lab), Petrópolis, Brazil

a sought-out goal by the scientific HPC community. Unfortunately, due to the law of diminishing returns, a simple aggregation of computing nodes and cores of current technology is not enough. From a financial perspective alone, this option proves to be prohibitive in operating costs due to the growth of energy consumption [119]. In this context, alternative architectures with low power consumption, such as ARM and GPGPU, are being developed and studied to build HPC systems.

The number of scientific papers published dealing with the energy issue on HPC has increased dramatically in the last couple of years, highlighting the importance of the relationship between performance and power consumption in the development of HPC systems. However, the number of papers alone and a lack of structure in categorizing such papers could lead to missed opportunities.

The contributions of this survey are as follows:

- Compile, categorize and analyze a set of influential and distinguished papers about the subject of power efficiency in HPC systems in general, and in the use of ARM processors for HPC in particular. This work should aid in a better understanding of this subject and guide the research toward the exascale barrier.
- Provide a narrative that describes the evolution of server-class ARM architecture over time, contrasting the projections of past research with the actual results observed, discussing how the observed bottlenecks either have been tackled or persist.
- Discuss both positive and negative trends observed throughout the ongoing research and development of server-class ARM processors, particularly for ARMv8 and its prospects as one of the leading architectures for the future exascale HPC systems.

The remaining sections are: Sect. 2 presents a brief explanation of the topics addressed and some common terms used; Sect. 3 discusses related works; Sect. 4 contains a detailed review of distinguished papers that address the use of ARMv7 instruction set architecture (ISA) for HPC; Sect. 5 contains a detailed review of distinguished papers that focus on the use of ARMv8 ISA for HPC; Sect. 6 contains a detailed review of distinguished papers that address the use of ARM, co-processors and SIMD extensions for HPC; Sect. 7 highlights both negative and positive trends observed in this study; Sect. 8.1 categorizes important references for the advancement of the topic addressed; Sect. 9 summarizes the results of this survey.

2 Background

Since the 1960, Moore's law has been able to correctly predict the increase in the number of transistors in integrated circuits doubling roughly every 2 years [117]. This allowed chips manufacturers to improve the speed of processors and implement more instructions on hardware than in earlier generations. In particular, the x86 complex instruction set computer (CISC) architecture, heavily used for HPC systems, followed this law to improve performance. However, this increase in transistor numbers and reduced area also generated greater heat and power consumption. Heat is a problem

that could be dealt with by using even more power to dissipate it, worsening the power issue.

Reduced instruction set computer (RISC) architectures, such as ARM, were developed with a different approach. Because of power consumption constraints that those processors had to deal with, such as in mobile phones with limited power supply, they used to have fewer instructions, needing fewer transistors to implement its instruction set. As such, they have reduced heat dissipation and power consumption when compared to x86 processors.

As evidenced in 2003 by [128], the “Power Wall” would not allow this growth to continue indeterminately. A famous quote from Erik P. DeBenedictis from the Advanced Device Technologies department at Sandia National Laboratories, in Albuquerque, stated that “we could build an exascale computer today, but we might need a nuclear reactor to power it.”

In this scenario, low-power-consumption processors emerge as an alternative to x86 architectures, particularly when paired with specialized processors such as GPUs. The prospect of this alternative being able to deliver similar performance capabilities with lower power requirements, when compared to x86 processors, is fueling intense development and research.

2.1 TOP500 [115] and Green500 [38]

The TOP500 List is a ranking of the highest performance supercomputer. Since 1993, it has been publishing two lists annually, every June and November. It uses the High-Performance LINPACK (HPL) benchmark, a linear algebra problem, for ranking in terms of flops that a system can achieve. As of November 2018, the highest-ranking supercomputer is IBM’s Summit, at Oak Ridge National Laboratory, with 143.5 Petaflops.

The year of 2018 has been an important year for alternative architectures in large HPC systems. Summit is based on the IBM Power9 processor, a RISC processor that complies with the Power ISA specification by OpenPOWER foundation. A similar supercomputer (Sierra) occupies the second place. The third place goes to Sunway TaihuLight, which is based on the many-core 64-bit RISC Sunway architecture, and was ranked first for two years.

Since the growing environmental concern, in November 2007 the Green500 List has been published alongside the TOP500 List. It also uses HPL, but machines are ranked by the ratio of performance per wattage, or Flops/W. In November 2018, the first position belonged to Shoubu system B, a ZettaScaler-2.2 system with Xeon D-1571 16C operating at 1.3GHz and achieving 17.600 Megaflops/W. It is noteworthy to point out that this system is ranked 375 in the Top 500 list, while the Power9-based Summit and Sierra are ranked as third and sixth in the Green500 list, respectively.

3 Related work

The work presented in [41] tries to analyze the current shift in paradigm experience in HPC, that is, the increased focus on energy-efficient system. This paper used the study and comparison of current and past top ranking HPC systems (TOP500 and GREEN500) to verify disparities between the two lists. It also states that the trend of the TOP500 is toward increased maximum power consumption while at the same time improving the energy efficiency. The work analyzes different architectural designs, including heterogeneous systems with energy-efficient co-processors, and their impact in performance and energy efficiency, pointing out that many systems have significant differences between the theoretical and actual performance obtained with the HPL benchmark. Our work confirms the suggested trend of HPC systems aiming at higher energy efficiency as means of achieving more scalable performance. We also present a more up-to-date analysis of power-efficient heterogeneous systems, with a focus on ARM.

The work [58] is a survey on software-based methods to improve HPC energy efficiency. The work is a comprehensive study on the state of the art of the use of software to achieve improved overall system utilization focusing on energy efficiency. The authors mention ARM big.LITTLE strategy as means of addressing the issue of energy constraints as the number of cores per processor increases, and remark that heterogeneous computing is a trend that is going to continue in the future of energy-efficient supercomputing. Regarding the evolving architectural innovation, the authors highlight the need for the software layer to evolve alongside it, to facilitate the desired levels of energy efficiency. Our work complements this survey by providing a historical narrative of the evolution of heterogeneous ARM systems and discussing both positive and negative trends observed in the evolution of hardware and software of alternative HPC architectures.

The author in [125] recognizes the power wall as the main factor limiting the goal of HPC at the exascale level. Thus, to contribute to the current research, it studies trends and analyzes perspectives that factor in the power wall on the development of HPC systems. First, it concludes that in order to assess energy efficiency, not only the performance per watt should be used as a metric, as in the GREEN500, but the power-usage effectiveness (ratio between total facility energy and IT equipment energy) should also be reported. The article then highlights two new transistor technologies of interest in this field: the near-threshold voltage (NTV) and silicon photonics. The NVT approach aims to improve energy efficiency by lowering voltage to approach circuit threshold limits; it is however not ready for HPC due to increased gate delay. The silicon photonics aims to improve energy efficiency by using light propagating in optical fibers instead of electrical signals. The work also points to embedded chips as a research perspective to optimize energy efficiency, mentioning the ARM-based Mont-Blanc research project. Our current survey provides insight into the three latest Mont-Blanc projects, while exploring other initiatives based on ARM processors that aim at tackling the HPC power wall.

4 Articles based on ARMv7

The ARMv7 architecture, with its ARMv7-A microarchitecture, marked the beginning of the “application” profile for ARM architectures, which enable full-fledged operating systems for general purpose, user-oriented applications. Its main new features over its predecessor ARMv6 include out-of-order execution and the single instruction, multiple data (SIMD) engine called NEON, which only supported single-precision (SP) operations [20]. Its significant performance improvements started gathering the interest of the scientific community, starting off research on the possibility of HPC workloads on ARM.

Below follows a more in-depth analysis of some key papers regarding the ARMv7 architecture.

4.1 Power struggles: revisiting the RISC versus CISC debate on contemporary ARM and x86 architectures [21]

This paper studies the decades-long argument between RISC and CISC ISA; however, unlike the previous generations of processors focused on optimizing performance, nowadays manufacturers are equally concerned with energy efficiency. Even though ARM RISC processors currently reign in the world of smartphones and tablets, the authors note that there exist two growing trends: the growing interest for ARM in the server market, and in counterpart the interest for x86 in the mobile market. Thus, the article evaluates the impact of the instruction set architecture in the actual power consumption.

Although large gaps in performance were observed between the two architectures in the experiments executed by the study, the authors designed a methodology to normalize performance by accounting for different clock speeds among the tested processors. The experiments point to performance being architecture (RISC or CISC) agnostic.

It should be noted that although ARM is classified as RISC and x86 as CISC architectures, many of the defining aspects of such classification are recently being interchangeably used between the two ISA. For example, ARMv7 contains the THUMB2 extension for 16-bit instructions, while x86 ISA translates instructions into RISC like micro-ops.

4.2 Supercomputing with commodity CPUs: are mobile SoCs ready for HPC? [95]

This paper assesses the possibility of employing ARM-based mobile System on Chip (SoC) for HPC. It compares the current trend of migrating x86 supercomputer systems to ARM to similar earlier trends, such as vector and SIMD architectures being superseded by RISC ISA processors in the early 1990s, and then in turn replaced in the mid-2000s by x86 CISC processors. They argue that the x86, marketed for desktop computers, succeeded over RISC not because of performance (an order of magnitude slower) but because of price, because the RISC processors at the

time were about 30× more expensive. The world is living an equivalent transition nowadays. Low-power, mass produced processors are beginning to close the gap in performance while their cost is approximately 70× lower than x86 processors marketed for HPC systems.

To validate their claims, a series of experiments were executed with benchmarks on four different systems: NVIDIA Tegra 2 with dual core ARM Cortex-A9 clocked at 1GHz, NVIDIA Tegra 3 with quad-core ARM Cortex-A9 clocked at 1.3GHz, Samsung Exynos 5250 with dual core ARM Cortex-A15 clocked at 1.7 GHz and Intel Core i7-2760QM with quad-core Intel Sandy Bridge processor.

Although already competitive by a purely energy-efficient standpoint, the experiments highlighted the following limitations in adopting ARM-based mobile SoCs for HPC: lack of error-correcting code (ECC) capable memory, larger bandwidth I/O capabilities, better networking performance, memory addressing limitations (32 bits address) and unnecessary mobile-specific components.

4.3 Tibidabo: making the case for an ARM-based HPC system [97]

This article presented the first large-scale (256 cores) cluster build with ARM processors, the Tibidabo. The cluster is composed of 16 blades, each hosting eight nodes with the Nvidia Tegra2 SoC, a dual core Cortex-A9 running at 1GHz with 1GB DDR2 of memory.

To evaluate this cluster, they first analyzed a single node of the proposed system, comparing both the performance and energy to solution against Intel mobile core-i7 processors over a set of benchmarks. For the Dhrystone performance was 7.8× slower, for CINT2006 it was 9× slower, and for the CFP2006 performance was 9.4× slower. The energy to solution was shown in all cases to be slightly better for the ARMv7 node. Finally, the STREAM benchmark showed that the Cortex-A9 had a 50.6% copy and 27% add efficiency when compared to the 2666MB/s theoretical bandwidth, while Core-i7 had a 40.5% copy and 41% add efficiency when compared to the 17066MB/s theoretical bandwidth.

Because the Nvidia Tegra2 is marketed mobile, only about 35% of the die area of the SoC is used by HPC applications. The rest, even if not used, can contribute to power consumption due to leaks. Another aspect is that, even when overestimating, only 16% of the power is spent on computation components and the remaining 84% is overhead used to interconnect the computation components with other components in the system.

The cluster was then evaluated using weak and strong scalability tests. The former was performed with the HPL benchmark, showing that its 120 Megaflops/W on 96 nodes is competitive with AMD Opteron 6128 and Intel Xeon X5660-based clusters, but about 21× slower than Intel Xeon Phi (November 2012 Green500 #1). Regarding the latter tests, good strong scalability (decreasing time-to-solution) was observed with HYDRO, GROMACS and SPECfem3D benchmarks up to 96 nodes. On the other hand, the PEPC benchmark did not scale well after 32 nodes.

The article also includes projections extrapolated from results of HPL running in 192 cores of the Cortex-A15 SoC. Extrapolations include a 4× faster clock, and

up to 16 cores per SoC, taking into account a model for intra-node communication with the DIMEMAS simulator. Results showed that the increase in the number of cores could lead to a 1046 Megaflops/W rate when using Cortex-A15 with 16 cores per chip clocked at 2 GHz while making it competitive with the x86 Sandy Bridge in terms of energy efficiency. However, it should be noted that new unforeseen bottlenecks could appear such as cache and communication delays. Also, the authors stress that the 1 Gb/s bandwidth and 50μ latency of the Tibidabo's Ethernet interconnect were sufficient to support projections, but clock frequencies beyond 2 GHz would require more bandwidth, and larger clusters would require reduced latencies.

The article clearly sets some prerequisites for a successful ARM-based HPC system: the need of a HPC-ready solution (higher core density in the chips and stripped of unnecessary components), architecture optimized software such as MPI and algebra libraries, improved support to double-precision floating-point operations as specified in ARMv8 (instead of the ARMv7 used in this study) and the need of better networking solutions, such as InfiniBand.

4.4 Performance and energy efficiency analysis of HPC physics simulation applications in a cluster of ARM processors [20]

This article contributes by evaluating two physics simulation applications, Ondes3D and N-Body, in a small, 8-node ARM cluster, and comparing time-to-solution and energy to solution with a traditional x86 HPC system. The ARM cluster, named Yggdrasil, is built with the Cubietruck SoC, a dual core ARM Cortex-A7 processor board. The article evaluates real applications taking into account the number of processes, compilation flags and clock frequencies.

When dealing with different compilation flags, three scenarios were compared against the baseline of no optimization flags: (1) with O3; (2) with O3 and ARM-specific flags; (3) with O3, ARM-specific and NEON optimization flags. For Ondes3D, the time-to-solution and energy to solution decreased about 50% for all three optimizations when compared to baseline, with (3) yielding the best result marginally. For N-body, the time-to-solution decreased about 60%, while the energy to solution did so by [51–58%].

For clock frequencies, the article realized experiments with two profiles: performance (time-to-solution) and powersaving (energy to solution). The performance profile for Ondes3D presented a decrease in time-to-solution of about 20% when compared to powersaving, with similar energy-to-solution metrics. For N-Body, the time-to-solution presented comparable results, yielding slightly worse energy-to-solution experiment for performance over powersaving. Scalability tests showed that the most appropriate setting was four processes, with 50% less time-to-solution and same energy as with two processes. However, the trend was interrupted when moving to 16 processes, presumably due to communication overheads. As the number of processes increased, the speedup gets further away from a linear speedup for both applications, also the impact of compilation flags is greater with fewer nodes used.

When compared to a traditional x86 cluster, Yggdrasil presented higher time-to-solution (5.5 to $8.4 \times$ worse), albeit with lower energy-to-solution (2.9 to $4.5 \times$

better). The authors reach the conclusion that the execution of scientific applications is viable on low-power devices from an energy-to-solution standpoint, and point to the necessity of double-precision support, not available in Cubietruck.

4.5 The Mont-Blanc prototype: an alternative approach for HPC systems [96]

The Mont-Blanc project is a joint effort of the European scientific community and major HPC technology vendors to develop an exascale HPC system until 2020 with focus on energy efficiency. The Mont-Blanc prototype is an ARM HPC system developed in the first and second stages of the project. It consists of a system with 1080 nodes, 2160 Cortex-A15 CPU cores running at 1.7GHz and 4320 Mali T604 GPU cores at 533MHz. Each node uses 4GB of DDR3 main memory with 12.8GB/s bandwidth. The nodes are aggregated into groups of 15 composing a blade interconnected with 1 GbE.

The Mont-Blanc prototype system is theoretically capable of 107.7 Teraflops in Single Precision (SP). For double precision (DP), the system peaks at 30.3 Teraflops, of which 6.8 Gigaflops come from the CPU and 21.3 Gigaflops from the GPU of a single node.

On a core by core comparison with a traditional HPC system composed of Intel Xeon E5-2670 running at 2.6GHz (MareNostrum III), the Mont-Blanc prototype is 2.2× to 12.7× slower. On a node-by-node analysis, it is between 9× and 18× slower. The best performance was achieved with heterogeneous computation (CPU+GPU) with OmpSs [37]. The energy efficiency of Mont-Blanc prototype can be up to 40% better if compared to MareNostrum III when using heterogeneous computations.

Analyzing the entire system, the strong scalability showed that the Mont-Blanc prototype can scale up to 16 nodes to compensate for performance, some applications such as SMMP (molecular thermodynamics) may not scale further. When testing weak scaling, all applications achieve $\geq 60\%$ of ideal performance at the maximum problem size, with a majority scaling at $> 70\%$ and with some application scaling at $> 90\%$ efficiency. Compared to the traditional HPC system, Mont-Blanc was on average 3.5× slower and consumed only 9% less energy. However, those applications were not optimized for GPU and OmpSs (the best result for Mont-Blanc).

The authors point out the Mont-Blanc prototype needs to scale about 10× more to match the performance of x86. As such, the Ethernet-based interconnect produces significant overhead in the TCP/IP protocol stack. In addition to the lack of DP SIMD processing, the paper highlights the importance of having built-in energy profilers, and the software to support them.

4.6 Summary of works on ARMv7

Given that the main appeal of the ARMv7 is low power consumption, the majority of the research focused on energy efficiency. Additionally, many of the noteworthy works have explored the combination of ARMv7 cores with GPGPU for processing HPC workloads [11, 29, 44, 48, 95, 97, 99, 106, 113, 114]. These articles will be

further discussed in Sect. 6. Other articles focused on comparing the performance and energy consumption with the x86 architecture [7, 9, 21, 27, 28, 30, 49, 57, 72]. The consensus of these evaluations is that the ARMv7 was not ready for HPC workloads, not only due to their low flops count (at least one order of magnitude slower) but also because of several bottlenecks related to memory bandwidth and interconnect. This is somewhat expected considering that the existing hardware (e.g., Tegra K1) was designed for mobile usage instead of HPC. At the same time, there is general agreement that the energy to solution is already somewhat comparable to x86 architecture for small scales.

5 Articles based on ARMv8 ISA

The ARMv8 architecture provides significant improvements over its predecessor, the ARMv7. Not only it supports the new Aarch64 execution state with 64-bit registers and memory addressing, but also boosts the NEON SIMD with double-precision floating point and IEEE-754 arithmetic [108]. In addition to the server-class reference implementations offered by ARM (Cortex-A5x and Cortex-A7x families), several manufacturers are developing ARMv8 SoCs geared toward performance, most notably the processor families: AppliedMicro's X-Gene, NVIDIA's Tegra and Jetson, Qualcomm's Kryo, and Cavium ThunderX.

Below follows a more in-depth analysis of some key papers about the ARMv8 architecture.

5.1 Characterization and bottleneck analysis of a 64-bit ARMv8 platform [68]

This paper studies the X-Gene processor and compares it to other three x86 processors: the High-performance processors Xeon E5-2670v1 (Sandy Bridge) and Xeon E5-2667v3 (Haswell), and the low performance low consumption Atom C2758. Each AppliedMicro's X-Gene 1 processor has eight cores and was developed using 40nm lithography. The authors rely on a framework called Partial Least Squares (PLS) Path Modeling to analyze both performance and power consumption of these processors.

The results of executing over 400 workloads show that X-Gene is in average $2.3\times$ slower than Sandy Bridge, $3.4\times$ slower than Haswell and approximately 7% faster than Atom. Energy wise, Sandy Bridge consumes $1.2\times$ the energy of X-Gene. However, X-Gene consumes $1.3\times$ more energy than Haswell and $3.5\times$ the energy of Atom.

To understand those results, PLS Path Modeling is employed. In this model, latent variables are defined as points of architectural or micro-architectural interest such as cache, TLB and SIMD execution. The PLS algorithm is based on the calculation weights for the linear combination of input variables used to estimate each latent variable. The PLS allows to identify and measure the relationship between the result of the experiments and each latent variable. This study allowed identifying the memory hierarchy as the main architectural-factor-limiting performance.

Besides the results presented, the authors also noted that due to its licensing model ARM processor can achieve better performance due to an easier integration with accelerators. Greater maturity of ARMv8 software stack, such as compilers, can lead to improved performance. Future many-core ARMv8 implementations can achieve better results if memory bandwidth supports it and ARMv8 can further benefit from shrunken die printing (such as X-Gene2 that will use 28nm instead of the current 40nm). The results from this study suggest that for X-Gene to reach energy-delay product (EDP) parity with other platforms, it needs to improve performance by 1.4× or reduce energy consumption by 2×.

5.2 Quantifying energy use in dense shared memory HPC node [93]

This article features the AsianCat, one of the first many-core HPC systems built with the ThunderX processor. The ThunderX is Cavium's first server-class processor, designed with a 28nm process. The processor is designed as a dual-socket SoC with two ThunderX ARMv8 CPU, each with 48 cores (96 total). To allow for cache coherency in such a large number of cores, it implements the Cavium Coherence Protocol Interconnect (CCPI).

According to the article, designers for exascale will need to consider HPC systems under three aspects: performance, energy efficiency and programmability. The proposed system needs to be monitored and managed dynamically at system level. Each ThunderX is measured to be capable of 240Gflops in DP.

Both Splash-2 and PARSEC benchmarks suites were used to measure the system, specifically, their parallel execution stages. The benchmarks evaluated in this study are grouped in three sets: (1) benchmarks that once allocated reach peak power and stay stable at that level, (2) benchmarks for which power varies with periodic oscillation with efficient load balancing among threads; and (3) benchmarks with periodic oscillation and large load imbalances among threads. The fluctuations in power and the categorization lead to the possibility of energy savings through Dynamic Voltage Frequency and Scaling (DVFS).

The study also discovered that the CCPI did not influence the power level for the benchmark suites. Thus, the AsianCat system shows an energy-efficient L2 level cache coherent many-node system.

This work meticulously measured performance and power consumption of the ThunderX processor, even mapping PMU directly to avoid the perf or the OProfile overhead. The research on evaluating a many-core system and its cache interconnect highlight the viability of a future ARMv8 solution appropriate for HPC.

5.3 Advanced performance analysis of HPC workloads on Cavium ThunderX

The authors of this work [26] also evaluate the Cavium ThunderX processor. The evaluation relies on an environment for performance monitoring, developed by the Barcelona Supercomputing Center (BSC), and focuses on a Lattice Boltzmann HPC production code.

To obtain performance metrics on the ARMv8 cores, the Extrae instrumentation library and the Paraver trace analyzer were ported. These in turn rely on PAPI, which provides a standardized interface to hardware performance counters. The hardware used for the evaluation is a four-node cluster, each housing a two Cavium ThunderX SoCs with 48 cores each.

As an initial evaluation, the application was executed on a single processor, using 48 OpenMP threads and a single MPI process. Performance metrics revolved around two key portions of the application, the “propagate” and the “collide” functions. The propagate function reached its highest bandwidth of 12.6 GB/s (38% of STREAM available bandwidth) when using 16 threads, while the collide function achieved 73Gflops (38% of theoretical peak) when using 48 threads. These percentages are contrasted with 75% and 28% bandwidth for the propagate function when using Intel E5-2630-v3, and Intel Xeon Phi 7120X, respectively; also, with 36% and 30% of peak performance for the collide function for those x86 processors.

The authors then used their performance toolset to assess the nature of the bottlenecks that hindered performance of the propagate function in the ThunderX processor. Two main bottlenecks were identified, the first being the saturation of the L2 cache (which plateaus at 24 threads), which is shared among all 48 cores and is the last level cache; the second being a high TLB miss ratio for 32 and 48 threads. This analysis led them to choose a different data structure, resulting in a bandwidth of 20.5GB/s (52% of STREAM bandwidth), a 64% performance increase.

In addition to the promising results of executing HPC production code, this article provides a detailed description of how to identify performance bottlenecks on an ARMv8 processor. After porting existing tools, it is shown that it is possible to evaluate ARMv8 SoCs in a fashion similar to other multi-core processors such as the Xeon Phi. This adaptability may also translate to performance optimization such as switching data structures. While the authors did not present performance results for the whole cluster of four nodes, they recognize that the analysis is preliminary, and that a second release of the ThunderX processor is expected for HPC.

5.4 A performance analysis of the first generation of HPC-optimized Arm processors [82]

The authors in [82] discuss the benefits and short comings of using ARM-based nodes in HPC systems, when compared against modern x86 processors (Broadwell, Skylake). The evaluation spans a diverse range of mini-apps and scientific applications, as well as different compilers and performance libraries.

To achieve their goal, the Isambard cluster was built, a cabinet comprising of 42 blades of the XC50 “Scout” system. Each blade packs four nodes with the Cavium (Marvell) ThunderX2 processor, and each node includes two 32-core ThunderX2 processors running with a frequency of 2.1GHz and 2666MHz DDR4 memory channels, for a total of 10752 high-performance Armv8 cores.

The main conclusion of this effort is that the Cavium ThunderX2 processor can now be considered a viable alternative to x86 CPUs, especially when considering the cost of the hardware, measured as performance per dollar. A highlight is that compilers such

as GCC7 and GCC8 still lack support for scientific source codes, not only in matters of performance (such as the FFT library compiled with GCC), but also in compilation issues that prevent execution, particularly with GCC8. Between ARM HPC Compiler and the Cray Compiler (CCE), the former provided better build support, because CCE lacked support for some intrinsic and inline assembly.

It is important to stress that the results presented are all for a single node, without showing any scalability results. We believe this is a missed opportunity to provide more interesting comparisons against x86 HPC clusters. Also, while there are very important results for compilers, and the developers cared to detail which combinations of compilers and math libraries were available, the authors did not include information about which libraries were ultimately used to obtain the published results. Our revision of the source code and results suggests that the ARM Performance Libraries and the Cray Programming Environment (CPE) are able to significantly improve mini-app performance.

5.5 Summary of ARMv8

Since late 2017, there have been numerous scientific publications aiming at evaluating the performance of ARMv8 processors. There is of course a sustained interest in energy efficiency [14, 33, 45, 59, 68, 92, 93, 110, 111, 118]. Notwithstanding, the vast majority of power consumption metrics available for this survey belong to processors and SoCs geared toward mobile instead of HPC clusters. We argue that this is due to energy metrics being available through hardware-specific instrumentation devices, which are more easily installed in small or development nodes, e.g., for NVIDIA's boards. On the other hand, ARMv8 HPC clusters with power monitoring capabilities are still under development [82], for example the Dibona ARMv8 cluster built for the Mont-Blanc 3 project already enables energy metrics via a customized solution based on FPGAs [16].

Previous studies focused mainly on 32-bit ARM processors and, as those studies showed, due to architectural limitations such as 32-bit memory addressing, lack of ECC memory, slower interconnects and missing support for double-precision computation, that the 32-bit ARM processors lacked several crucial features needed for HPC viability. We can see that implementations such as AppliedMicro's X-Gene3 and Marvell's ThunderX2 processors remedy these limitations.

Regarding comparisons against the x86 architecture, we have compiled several works that evaluate the ARMv8 architecture: comparisons: [6, 15, 26, 39, 59, 63, 82, 92, 110, 118]. If we consider the state-of-the-art ThunderX2, the findings can be summarized as this processor having a performance significantly better as the Broadwell processor and somewhat inferior to a top-of-the-line Skylake processor. In Sect. 7, we elaborate on how this processor compares to existing x86 offers.

6 Articles that used ARM with co-processors

6.1 Computational and memory analysis of tegra SoCs [84]

This article evaluates how ARM SoCs with embedded GPUs can be used to leverage high performance with reduced energy consumption for some HPC workloads. Recent developments in the mobile industry required the execution of graphical demanding applications and games, leading to GPU accelerators manufacturers such as NVIDIA to develop SoCs with embedded GPUs, such as the Tegra K1 (Kepler architecture) and X1 (Maxwell architecture) SoCs. These High-Performance Embedded Computing (HPEC) boards have CUDA ready GPU units as well as energy-efficient ARM processors.

To study these SoCs, the article used the theoretical capabilities of the board alongside the computational density (CD) and external memory bandwidth (EMB) metrics as well as benchmark execution with the realizable utilization (RU) metric and HPC kernel execution. To compare theoretical results with measured benchmark, they used the previously developed CD-RU metric, a normalized ratio between computations benchmark and CD, and developed the EMB-RU, a normalized ratio between memory benchmark and EMB. Traditional HPC NVIDIA devices are also used for comparing the HPEC devices.

The GPU of the K1 SoC has CD of 182.40 SP giga operations per second (GOPS) and 7.60 DP GOPS and EMB of 14.93 GB/s with a Thermal Design Power (TDP) of 8 Watts, while X1 has 256 SP GOPS, 8 DP GOPS and 25.6 GB/s with a TDP of 10 Watts. Comparing those metrics with the traditional HPC Kepler GPU, the authors point to a performance gain of 11.76 \times and over 20 \times higher memory bandwidth with the Kepler K40, however noting that the latter has 15 Stream Multiprocessing (SMX) cores and consumes 29.38 \times more power. The Maxwell HPC equivalent (Titan X SC) shows an improved performance of 13.52 \times and 13.14 \times higher memory bandwidth than X1; however, it used 25 \times the power of the X1.

The computational intensity (CI) of each kernel executed displayed a very varied range. From the matrix transpose (MT), that does not use computation power, only memory reordering, to matrix multiply (MM) that vary from 25% to over 90% depending on the dataset size and FFT1D that has fixed 49% CI disregarding the dataset size and FFT2D that ranges between 45% and 55%.

For MM, CD-RU rises as the memory bound dominated by small datasets grows to enable for reuse of data with on-device caching. In addition, EMB-RU declines quickly due to the same reason. Both K1 and X1 reach saturation in the experiment, and K1 has lower CD-RU due to slower DDR3 memory. CD-RU and EMB-RU showed that both SoCs have similar results to the HPC ones and X1 and to a lower extent K1, with enough scaling, are feasible for HPC CPU-bound applications with SP. However, the results for DP show that those SoCs are not designed for DP computations.

The almost perfect 50% division of CPU and memory execution of FFT1D mixes the results of memory operations and computation for all devices (SoC and

traditional). The results also show that for small datasets, CD-RU and EMB-RU score is low for all devices, but that K1 and X1 could be effectively scaled into clusters and match performance of the HPC devices. Results with FFT2D are similar but with a positive CD-RU trend.

The MT kernel, that has CI of 0%, shows that due to their smaller memory and layout, K1 and X1 SoCs support a matrix of up to 16MB size. Both K1 and X1 and Titan X show a peak of approximately 80% RU. This workload helps identify the bottleneck of K1 as being the DDR3 memory speed (X1 has DDR4, while Titan X has GDDR5). The limited EMB points to poor scalability of SoCs GPUs with the MT kernel.

This article showed that the ARM processor with embedded GPU counterpart is a viable option for HPC, sometimes surpassing the traditional GPU approach. Weak points to watch out for in this computational model should be memory-bound kernels, that due to lower speeds cannot scale very well, as well as DP computations which are not well supported.

6.2 High-precision power modeling of the tegra K1 variable SMP processor architecture [114]

Motivated by the inefficiency in accuracy of rate-based power models (up to 30% for the CPU and 70% for the GPU and memory), this study aims to deliver a model for the Tegra K1 SoC that takes into account individual architectural discrepancies of the SoC, such as two distinct cores the Low Power (LP) and the High-Performance(HP), caches and main memory while also taking into account the application influence to deliver an extremely high-precision-power model, with almost 100% accuracy efficiency. The precision of the rate-based models is entirely dependent on the DVFS frequencies. Thus, generic models fail to take into account important aspects that influence the power model of the SoC, such as heterogeneous cost performance metrics (instructions, cycles, etc.) and power gating.

The work published on [114] improved on an earlier model developed for the GPU, CPU, and memory, published on [113], limited by hardware measurements specific for CPU. The new approach estimates average capacitive load per instruction on a per-process basis. While not useful as a generic model for the Tegra K1, this application specific model is a successful trade-off to achieve a precise model. These works also contributed by pointing out how the model can be applied to identify system-specific power usage for each application and how to use these data to improve the efficiency of the SoC.

7 ARM for HPC perspectives

Throughout this review, that encompassed the analysis of a large set of publications, white papers and public available information, some trends could be observed. A selected group of articles that distinguishes itself by its innovations and analysis were analyzed in depth.

This section is dedicated to analyzing major trends observed in the evolution of the ARM architectures, its feasibility as a venue for HPC and the evolution of the several hardware and software components that are being pieced together to provide a suitable execution environment for high-performance applications on ARM architectures.

7.1 Architectural evolution

The introduction of the energy-efficient ARMv7 processors was seen as an opportunity to execute scientific application in an alternative architecture [41, 96]. However, even though a necessary step toward HPC support, this architecture was found to be immature for scientific applications in key aspects:

- The SIMD vector unit only provides support for single-precision floating-point numbers. While double-precision floating-point operations are possible via the Vector Floating Point (VFP) unit [12], no vectorization is possible. Also, the implementations are limited by the architecture to support only IEEE 754-1985, and not the more updated IEEE 754-2008 standard.
- Projections made based on observed scalability of scientific applications on ARMv7 hardware already pointed out that the interconnect would become the bottleneck of several memory-intensive and network-intensive workloads, not only for ARMv8 but also for faster ARMv7 processors [97]. However, the SoCs manufactured at the time were geared toward mobile computing, so the available SoCs lack the necessary I/O interfaces to support high-bandwidth interconnect [95]. For example, the Tegra X1 board supports 5-lane PCIe 2.0 [89], while Mellanox Infiniband expects either PCIe 3.0 or 16-lane PCIe 2.0 [83]. As a result, no high-bandwidth interconnects formally support the ARMv7 operating system architecture.
- The 32-bit size registers present the important limitation of having limited addressable memory (theoretical maximum of 4GB), a concern that was also raised in some works [8, 95, 97]. Here, it is important to point out that ARM has taken steps to mitigate this limitation by extending ARMv7 to support 40-bit addressing with the Large Physical Address Extension [23].

These limitations haven been remedied with the advent of the ARMv8 architecture. Not only does the improved SIMD unit support double-precision floating-point numbers, thus effectively doubling performance [95], it is also fully compliant with IEEE 754-2008 standard [108].

The interconnect capabilities of ARMv8 processors have also improved significantly. Processors such as the ThunderX2 and the X-Gene3 have full support for the PCIe 3.0 ports, with 32-lane [40] and 42-lane [51] implementations, respectively. As such, both the Mellanox Infiniband [16] and Cray's Aries interconnect [82] have been successfully used in ARMv8 HPC clusters.

It is difficult to establish direct comparisons between recent x86 CISC architecture developments and the ARMv8 architecture. Careful analysis of the ISA [21]

suggests that one ISA is not inherently superior to another, and that metrics such as single-thread performance and power efficiency are ultimately dependent on implementation. Regarding comparisons between the state-of-the-art ThunderX2 processor and recent Intel processors, we see that performance results vary throughout different workloads, depending on how the applications can leverage the strengths of each processor. For example, the ThunderX2 has a significant higher core count and memory bandwidth, potentially improving results for memory-bound applications [82]. On the other hand, Intel processors are equipped with wider vectors (512-bit AVX versus 256-bit NEON), which can translate into peak floating-point advantage for compute-bound workloads, as well as more efficient L1 cache utilization.

In the context of scalability, the ThunderX2 processor seems to perform better under-saturating of the sockets: The STREAM benchmark showed best results when using a thread count of half the core count [82], already besting a Skylake processor in memory bandwidth efficiency, thanks to the higher number of channels (8 against 6). A similar finding was observed with the HPCG benchmark [16], showing that multi-node scalability was satisfactory (80% Rmax) for up to the tested 1024 cores.

The architectural refinement of the ARMv8 continues, and currently the first implementation of the ARMv8.2-A ISA corresponds to the A64FX processor, developed by Fujitsu using a 7nm process and with close collaboration with ARM [66]. At the time of writing, this processor has not yet been thoroughly evaluated, but initial performance reports [126] indicated over 90% efficiency (from theoretical peak) in the DGEMM benchmark, and over 80% efficiency in the STREAM Triad benchmark, using a 512-bit wide SIMD with SVE support.

7.2 Core software components

In 2010, an initiative to enhance OS support for ARM processors was established in the form of the engineering organization called Linaro [4]. It brings together industry and open-source community to work on free and open-source software such as the Linux kernel, the GNU Compiler Collection (GCC), and power management for the ARM architecture family.

Although the focus of Linaro has been mainly to enable a common software foundation for all modern Linux-based mobile devices running on ARM processors, their work has broadened to include software stacks and tools in networking, servers, and the Internet of Things (IoT). Relevant to this survey are efforts toward kernel functionalities such as input–output memory management unit (IOMMU), huge pages, DVFS, hardware performance counters, specific support for SoCs, among others. Additionally, Linaro continually improves the efficacy and efficiency of the GCC compiler, in terms of intrinsic support, auto-vectorization and instruction scheduling [2]. Unfortunately, the list of improvements produced by Linaro is not readily available as a changelog, and requires examination of each monthly release.

Another initiative that is cementing the HPC ecosystem for ARM is the OpenHPC project [104]. OpenHPC is a Linux Foundation Collaborative Project that aims to provide a reference collection of open-source HPC software components and best

practices, including provisioning tools, resource management, I/O clients, development tools, and a variety of scientific libraries.

OpenHPC began providing simple configuration recipes for HPC clusters, but ongoing efforts are focusing on providing automation for more advanced configuration and tuning to address scalability, power management, and high availability concerns [104]. OpenHPC fully supports ARMv8 architecture since its v1.2 release, packaging several key core components such as the OpenMPI library (v3.x), SLURM cluster manager (v18.x), Boost C++ libraries (v1.69), Nagios monitoring solution (v4.x), among others [5].

7.3 Libraries and applications

A new system architecture presents the challenge of not only compiling existing source codes, but also the code executing correctly and efficiently. This is of particular importance to scientific applications, which often have special requirements such as particular library dependencies.

Several key papers discussed in this survey have been developed in the context of the Mont-Blanc project, the European initiative to arrive at an exascale supercomputer by leveraging ARM architectures. The first phase of the project focused on providing initial ports for scientific libraries, developer tools and runtimes to the ARMv7 architecture [75].

A noteworthy port was the Automatically Tuned Linear Algebra Software (ATLAS) [122], containing routines from Basic Linear Algebra Subprograms (BLAS) and Linear Algebra Package (LAPACK). As for developer tools, we highlight the port of Scalasca, which supports the performance optimization of parallel programs by measuring and analyzing their runtime behavior, as well as Score-p measuring infrastructure. Regarding runtimes, the OmpSs programming model [35] was ported in the scope of the Mont-Blanc project (OpenMPI, MPI and OpenCL were ported by other groups). It extends OpenMP to support heterogenous architectures and thus directly enables ARM SoCs.

The second phase of the Mont-Blanc project saw the improvement in the software ports, broadening the supported functionality. While this phase focused on the development of the hardware prototype [96], several applications were evaluated on this platform, which also had to be ported to allow compilation and to leverage vectorization. Among these scientific applications, we highlight: i) BigDFT, a simulation software for electronic orbitals that depends on BLAS and LAPACK; ii) Quantum Espresso, a suite for modeling nanoscale materials that relies on several scientific libraries including FFTW.

The third phase of the project was characterized by the introduction of the ARMv8, for both hardware and software [16]. It was verified that porting the applications and libraries to ARMv7 considerably helped to execute scientific code in ARMv8, but some more fine tuning was needed. During this phase, ARM developed two important pieces of software for ARMv8: the ARM HPC Compiler and the ARM Performance Libraries. The ARM Performance Libraries is a set of high-performance mathematical libraries, and, along with the ARM HPC Compiler, has

been shown to consistently provide higher performance than GCC 7.1 for the HPCG benchmark [102].

Additionally, the ARM Allinea Studio [73] was released. It comprises MAP, a performance tool to profile an application using statistical sampling; and Performance Reports, a tool that summarizes resource and energy utilization. Also in the context of the Mont-Blanc project, the HPCToolkit performance suite [10] was ported to ARMv8, enabling low-overhead performance metrics based on statistical sampling and source code analysis. This effort required the port of several other libraries, as well as the enhancement of the Performance API (PAPI) to support the ThunderX2 processor.

To change the architecture on which HPC has developed in the last decade will demand not just the replacement of the processors, but a complete redesign of traditional libraries and tools used for HPC to achieve better performance on those new architectures. Not to mention that the applications will need to be ported and optimized for the new technologies.

7.4 The ARM scalable vector extension

The Scalable Vector Extension (SVE) is the evolution of the SIMD support on ARM processors [112] and builds upon the latest NEON extension to accommodate the needs of new markets that showed increased interest in the architecture. SVE aims to attend demands such as gather-load and scatter-store, per-lane predication, and longer vectors, in fact, giving most of the vector machines capabilities, albeit in a reduced scale, to ARM processors. This significant evolution clearly demonstrates ARM's interest for HPC.

One of the most interesting aspects of the extension is the fact that it does not define a size for the vector registers, allowing the manufacturer to choose between 128 bits to 2048 bits, with 128-bit wide increments. In accordance with the ARM licensing model, this allows to produce custom-made processors tailored for a specific market.

Another change from the previous ARM SIMD extension is the use of predication instead of branching. This allows for reduced number of instructions, consequently faster code, in the SIMD loops. Other SIMD extensions for different architectures that also employ predication such as the x86's AVX occupy registers for comparison. To avoid wasting a register for control, SVE included while instructions that make use of scalar counts and limit registers. The inclusion of those instructions and support register also helps auto-vectorization, which tends to align all the SIMD data to the size of the largest variable, thus resulting in wasted throughput when the size of the induction variable is greater than that of the data processed.

The SVE proposition opens the possibility of porting code between processors with different widths, without the need to recompile the code. To deal with this, the SVE introduces specific instructions such as *index* to initialize the variable, *inc* to increment based on current vector length and element size and *fadda* that allows for ordered floating-point reductions when the order must be preserved for correctness.

To assess SVE performance, a model was executed with a medium-sized, out-of-order processor (not related to any real one), using an experimental compiler with SVE support. Capabilities such as gather-load and scatter-store and per-lane predication enable vectorization of more complex structures and loops, and thus higher utilization of the SIMD, when compared to NEON. When using the same register width (128 bits), SVE achieves up to $3\times$ speedup, and up to $7\times$ speedup with a width of up to 512 bits.

SVE represents an answer to the AVX-512 SIMD instructions [3] present on x86 processors, such as the Xeon Phi 7200. The AVX-512 is an expansion of AVX-256, doubling the capacity of vector computations of the NEON 128-SIMD extension available in most ARMv8 implementations. SVE aims at extending (instead of replacing) NEON, and was developed specifically for vectorization of HPC scientific workloads, in particular for applications in Machine Learning [54]. The SVE not only supports the claim that SIMD is increasing in importance for performance, but also attests the development of ARM processors focused on intensive computing applications for HPC. The first implementation of an ARMv8-A processor with SVE support is the A64FX, by Fujitsu, described in Sect. 7.1.

7.5 ARM licensing program

Although the energy efficiency of ARM processors is regarded as the main factor for proposing its adoption in HPC, one important aspect that is frequently overlooked by studies supporting the adoption of ARM is its licensing program. ARM processors allow for manufacturers to license its architectures and modify it to better suit the computational needs that the processors are aiming to achieve. The custom-made processor needs only to pass through an architecture compatibility test, leaving plenty of room for specialization.

Examples of this flexibility are the NVIDIA's Denver processor and Qualcomm's Kryo 280. On the one hand, the Denver processor is an ARMv8-based processor optimized for using with GPU co-processors. This processor is available on the Jetson TX2 System-on-Module (SoM). This 7.5W average consumption module has 2 Denver, 4 Cortex-A57 and 256 Pascal Cuda cores. Due to its capabilities of hot plug cores (that is, turning cores on or off during runtime), it can power off some of those cores to allow for extremely high energy-efficient Single Instruction, Multiple Thread (SIMT) processing. On the other hand, the Kryo 280 is an ARMv8-based (supposedly based on the Cortex-A73) processor built with 10nm lithography aiming for high performance on mobile devices. This processor clocked up to 2.45GHz aims to deliver faster mobile experience. The Kryo 280 is available on the Snapdragon 835. The Snapdragon has an SoC with 8 Kryo 280 cores with 4 clocked at 1.9GHz for use in most applications to conserve battery lifetime and 4 clocked at 2.45GHz for intensive computing applications. While both processors are based on the same architecture, there are stark differences that attest to the licensing flexibility.

We argue that this freedom should be explored for HPC systems to deliver custom-made processors that are energy efficient to specific workload types while still maintaining good performance.

8 Summary of the scientific literature

In this section, we report a list of articles, white papers and other references with topics that address the current research to reach the HPC exascale barrier. We also extracted relevant performance metrics from these papers and present these metrics in graphs and a table.

8.1 Recommended articles

Table 1 lists more than 100 articles, which are mainly focused on the use of ARM processors to reduce the energy consumption of HPC infrastructure.

Table 1 presents some categories that are crucial to the use of ARM in HPC and improving computational capacity while keeping energy consumption low. The references listed in each category are works that are distinguished contributions for the topic. Most articles appear in more than one category, and it should also be noted that a work not appearing in another category does not equate to that work not

Table 1 Articles recommendations

| Category | Papers |
|-------------------------|---|
| ARMv7 | [7–9, 11, 19–21, 25, 27–30, 32, 33, 42–44, 48, 49, 52, 56, 57, 60–62, 64, 69–72, 76, 77, 85, 87, 88, 90–92, 94–101, 105–107, 113, 114, 120, 124, 127] |
| ARMv8 | [1, 6, 14–16, 22, 26, 33, 36, 39, 42, 52, 59, 60, 63, 67, 68, 74, 82, 86, 92, 93, 102, 110, 111, 118, 121, 123] |
| SIMD | [13, 20, 54, 77, 82, 112] |
| GPGPU | [11, 14, 19, 28, 29, 32, 33, 44, 48, 69, 74, 84, 87, 90, 96, 100, 106, 107, 111, 113, 114, 127] |
| Heterogeneous computing | [6, 18, 27–29, 31, 43, 49, 58, 74, 85, 99, 109, 120, 129] |
| HPC | [1, 6–9, 11, 14, 16, 18, 20, 26, 29–32, 39, 41–49, 57, 59, 64, 68, 72, 76–78, 82, 84, 85, 87, 90, 92, 93, 95–97, 99, 101, 102, 106, 107, 109, 110, 113, 114, 116, 118, 121, 125] |
| Big data | [50, 61–63, 70, 98, 127, 129] |
| Virtualization | [19, 60, 64, 91, 94, 120] |
| Mobile | [14, 30, 52, 95, 100, 106] |
| Energy efficiency | [7, 9, 11, 14, 15, 19–21, 25, 27, 29–31, 33, 36, 41, 44–46, 49, 52, 56–59, 61–63, 68–72, 74, 76, 77, 84, 85, 87, 88, 91–93, 95–97, 99–101, 105–107, 109–111, 113, 114, 116, 118, 120, 123, 125] |
| ARM/x86 comparison | [6–9, 14, 15, 20, 21, 26–31, 33, 39, 42, 49, 57, 59, 63, 67–74, 77, 82, 85, 87, 88, 90, 92, 95, 97, 99, 100, 106, 110, 111, 116, 118, 120, 127] |

contributing to the subject; it is only that we believe that its main contributions lie in other categories.

The ARMv7 and ARMv8 categories relate to articles that contribute to the research by analyzing those ISA in depth. The SIMD category deals with the use of SIMD instructions. The GPGPU category addresses the use (combined or not) of Graphics Processing Units to optimize execution time with reduced energy consumption. The Heterogeneous Computing category comprises works that use a variety of combined architectures (such as CPU, GPGPU, FPGA and ASIC) to contribute to the research. In the HPC category, there are works contributing with crucial aspects for large-scale computing such as scalability, shared and distributed-memory computing and intercommunication. The Big Data deals with the use of ARM in support of Big Data applications. Virtualization group works that study the hardware virtualization of ARM processors in particular for Cloud Computing. The Mobile category collects work that does not address the use of ARM for large-scale computing directly, but deals with subjects that can be of interest for the research on this topic. The Energy Efficiency category deals with work that contributes to the research of reducing the energy consumption of new architectures and methods. Finally, the ARM/x86 Comparison category deals with the work that present comparisons between ARM and x86 ISA, mainly highlighting differences in performance and energy efficiency.

8.2 Performance using the STREAM benchmark

To provide an overview of the performance and efficiency of ARM processors, and their relationship with the x86 architecture, we constructed Table 2. This table presents the measurements of the memory bandwidth obtained from executing the STREAM benchmark [81], as reported by other articles and whitepapers that have been referenced in the table. Reports spanning the years 2010 to 2018 are presented, comprising the ARMv7, ARMv8, x86_64 and MIC (Xeon Phi) architectures. The following methodology was used to build the table:

- We include results from the Triad vector kernel, which is consistently used in the literature. In some cases, the value was only available from graph, at other times the metric was expressed as a ratio or percentage of other metrics. In these cases, we either approximate or calculate the actual value, as indicated in the comment column.
- Table 2 includes the theoretical memory bandwidth of the processor. If available, this information is retrieved from the respective manufacturers. Otherwise, it is calculated using the reported memory speed, the number of memory channels and the memory bus width.
- We also show the Thermal Design Power (TDP) in Watts, as specified by the respective manufacturers. This represents the maximum allowed heat that is dissipated by a processor when working at full capacity. However, we were unable to find TDP values from some ARM processors. In these cases, we use the TDP of the board instead, as indicated in the comment column.

Table 2 Memory bandwidth per processor with STREAM Triad, according to diverse sources

| Date | Processor | Arch. | Th. BW | Triad BW | eff (%) | TDP | BW/TDP | Cite | Comments |
|---------|-----------------|--------|--------|----------|---------|------|--------|-------|------------------------------|
| Q3 2010 | Core i7-640M | X86_64 | 17.1 | 6.9 | 40.35 | 35 | 0.20 | [97] | Arrandale |
| Q1 2011 | Tegra 2 | ARMv7 | 2.66 | 1.2 | 45.11 | 20 | 0.06 | [95] | TDP from board |
| Q1 2011 | Core i7-2600 | X86_64 | 21 | 14.1 | 67.14 | 95 | 0.15 | [79] | Sandy bridge |
| Q4 2011 | Tegra 3 | ARMv7 | 5.86 | 1.3 | 22.18 | 20 | 0.07 | [95] | TDP from board |
| Q4 2011 | Core i7-2760QM | X86_64 | 25.6 | 10.3 | 40.23 | 45 | 0.23 | [95] | Sandy bridge |
| Q1 2012 | Xeon E5-2620 | X86_64 | 42.6 | 32.25 | 75.70 | 95 | 0.34 | [103] | Sandy bridge; BW halved |
| Q2 2012 | Core i3-3217U | X86_64 | 25.6 | 8 | 31.25 | 17 | 0.47 | [106] | Approx BW |
| Q2 2013 | X-Gene | ARMv8 | 59.7 | 19.41 | 32.51 | 45 | 0.43 | [42] | |
| Q4 2012 | Exynos 5250 | ARMv7 | 12.8 | 5.4 | 42.19 | 8 | 0.68 | [95] | TDP estimate only |
| Q4 2012 | Xeon Phi SE10P | MIC | 352 | 174.7 | 49.63 | 300 | 0.58 | [79] | Knight's corner |
| Q2 2013 | Exynos 5410 | ARMv7 | 12.8 | 4.5 | 35.16 | 4 | 1.13 | [30] | TDP estimate only |
| Q2 2013 | Xeon Phi 7120 | MIC | 352 | 177.9 | 50.54 | 300 | 0.59 | [79] | Knight's corner |
| Q3 2013 | Odroid XU | ARMv7 | 12.2 | 5.2 | 42.62 | 13 | 0.40 | [30] | 4xCortex-A15; TDP from board |
| Q3 2013 | Xeon E5-2650v2 | X86_64 | 59.7 | 38.55 | 64.57 | 95 | 0.41 | [103] | Ivy Bridge; BW halved |
| Q3 2013 | Xeon E5-2697v2 | X86_64 | 59.7 | 36.43 | 61.02 | 130 | 0.28 | [55] | Ivy Bridge; BW halved |
| Q3 2013 | Atom C2550 | X86_64 | 25.6 | 10.4 | 40.63 | 14 | 0.74 | [65] | Avoton |
| Q3 2013 | Atom C2758 | X86_64 | 25.6 | 12.9 | 50.39 | 20 | 0.65 | [65] | Avoton |
| Q1 2014 | Odroid XU3 Lite | ARMv7 | 14.8 | 4.5 | 30.41 | 12.4 | 0.36 | [106] | 4xCortex-A15; approx BW |
| Q2 2014 | Jetson TK1 | ARMv7 | 15 | 5.3 | 35.33 | 10.7 | 0.50 | [106] | TDP from board; approx BW |
| Q3 2014 | Xeon E5-2670v3 | X86_64 | 68 | 46.45 | 68.31 | 120 | 0.39 | [103] | Haswell; BW halved |
| Q4 2015 | Jetson TX1 | ARMv8 | 25.6 | 9.32 | 36.41 | 6 | 1.55 | [42] | Estimated TDP |
| Q1 2016 | Xeon E5-2695 | X86_64 | 76.8 | 48.925 | 63.70 | 120 | 0.41 | [55] | Broadwell; BW halved |
| Q2 2016 | Xeon E5-2699v4 | X86_64 | 76.8 | 65.5 | 85.29 | 145 | 0.45 | [82] | Broadwell; BW halved |
| Q2 2016 | Xeon Phi 7250 | MIC | 460.8 | 421 | 91.36 | 215 | 1.96 | [103] | Knight's landing; MCDRAM |

Table 2 (continued)

| Date | Processor | Arch. | Th. BW | Triad BW | eff (%) | TDP | BW/TDP | Cite | Comments |
|---------|--------------------|--------|--------|----------|---------|-----|--------|-------|-------------------------|
| Q4 2016 | ThunderX | ARMv8 | 153.6 | 39.6 | 25.78 | 120 | 0.33 | [26] | BW from ratio |
| Q2 2017 | X-Gene3 | ARMv8 | 136.5 | 67 | 49.08 | 125 | 0.54 | [34] | |
| Q3 2017 | Xeon Gold 6142 | X86_64 | 119.2 | 72.13 | 60.51 | 150 | 0.48 | [55] | Sky/lake; BW halved |
| Q3 2017 | Xeon Platinum 8160 | X86_64 | 153.6 | 85.945 | 55.95 | 150 | 0.57 | [16] | Sky/lake; BW halved |
| Q3 2017 | Xeon Gold 6138 | X86_64 | 128 | 100.2 | 78.28 | 125 | 0.80 | [82] | Sky/lake; BW from ratio |
| Q3 2017 | Xeon Platinum 8176 | X86_64 | 128 | 107.35 | 83.87 | 165 | 0.65 | [82] | Sky/lake; BW from ratio |
| Q3 2017 | Xeon Gold 6142 | X86_64 | 119.2 | 72.13 | 60.51 | 150 | 0.48 | [55] | Sky/lake; BW halved |
| Q2 2018 | ThunderX2 | ARMv8 | 170.7 | 97.61 | 57.18 | 180 | 0.54 | [16] | BW halved |
| Q4 2018 | A64FX | ARMv8 | 1024 | 830 | 81.05 | | | [126] | TDP undisclosed yet |

Bandwidths in GB/s, TDP in Watts

- For each processor, we report the bandwidth corresponding to a single processor. If the source specifies the metric of a dual-socket node, we halve the reported value. This was done to provide a more meaningful ratio with the TDP, which is related to a processor.
- The table includes the ratios between measured and theoretical bandwidths, as well as between the measured bandwidth and the TDP.

We justify the usage of the STREAM benchmark and the TDP as performance metrics, given the lack of extensive data usable for meaningful comparisons between different processors. Ideally, a benchmark such as LINPACK could also be presented, as well as the actual power consumption by either the compute node or the processor. However, the combination of these metrics is somewhat sparse in the literature, and what is available would not present a compelling trend in graphs. Given this scenario, we argue that the TDP can also be used as a standard energy footprint to enable comparisons between architectures and implementations.

The information contained in Table 2 is also presented in graph form. Figure 1 indicates that, before ARMv8 architecture, the memory bandwidth of ARMv7 lagged almost an order of magnitude behind Intel's Xeon and Core i7 architectures. However, ARMv7 nodes progressively reduced the gap, and is on par for the first generations of ARMv8 machines. But now, memory bandwidth is surpassing the top-of-the-line Xeon x86 processors. As discussed in Sect. 5, we attribute this to a higher count of memory controllers. The new A64FX is discussed in the next figure. Also important to take into account is the ability of the Intel Compiler to produce optimized code for AVX-512 processors. The Xeon Phi 7250 has maintained a very large lead; this can be explained by the use of a different type of memory, MCDRAM, which has a higher bandwidth but at a reduced size (DRAM yielded only 90GB/s instead [103]).

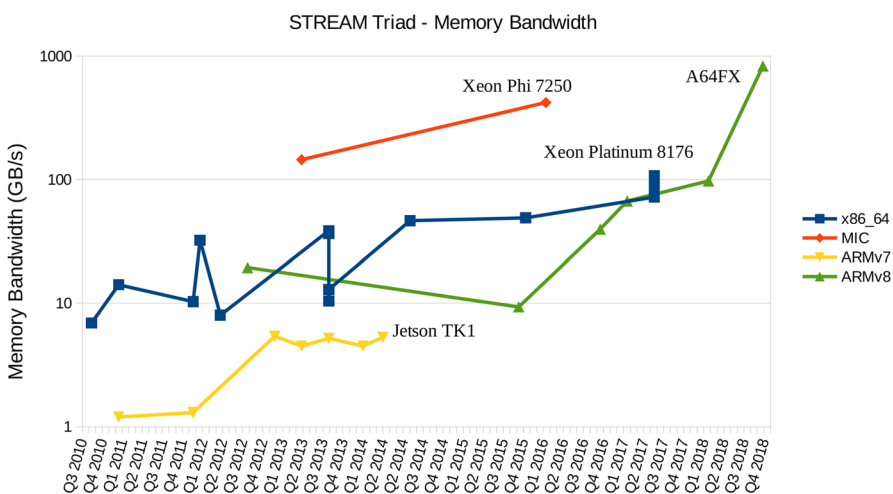


Fig. 1 Memory bandwidth for STREAM Triad by date of processor release, grouped by architecture

The bandwidth efficiency against the theoretical peak is presented in Fig. 2. The variance in values for x86_64 given a fixed date indicates that Intel released several processors at similar dates, each with different performance profiles, with Xeon Platinum yielding the highest performance for x86_64. From the figure trends, we infer that improvement in memory efficiency is difficult. Low efficiency for early ARMv8 processors may be attributed to insufficient last level cache, as memory saturation was reached before exhausting the cores with threads [16]. Recent significant improvements in ARMv8 in this regard may also be explained with better compilers, but this requires verification. The Xeon Phi 7250 once again holds the lead with its dedicated MCDRAM and optimized compilation process, but Fujitsu's A64FX now claims a similar high efficiency.

Finally, Fig. 3 shows the ratio between measured bandwidth and the TDP, as published by manufacturers. We can see much larger variations in the curves (Exynos 5410 is an outlier for ARMv7, and Jetson TX1 for ARMv8) which can be attributed to many things. We believe that these processors have been designed to be more efficient toward memory utilization than toward CPU processing. It is interesting to note that the Xeon Platinum was surpassed by a Xeon Gold here, hinting at this node being more energy efficient. However, aside from the workload-specific Xeon Phi and these outliers, the bandwidth per dissipated Watt appears somewhat flat over time. We argue that can mean that processors are becoming increasingly CPU-dominant, and that improvements in CPU efficiency are evolving faster than those for memory management, a view that has already been expressed [80]. We could not add the A64FX to this figure, as the TDP has not yet been disclosed.

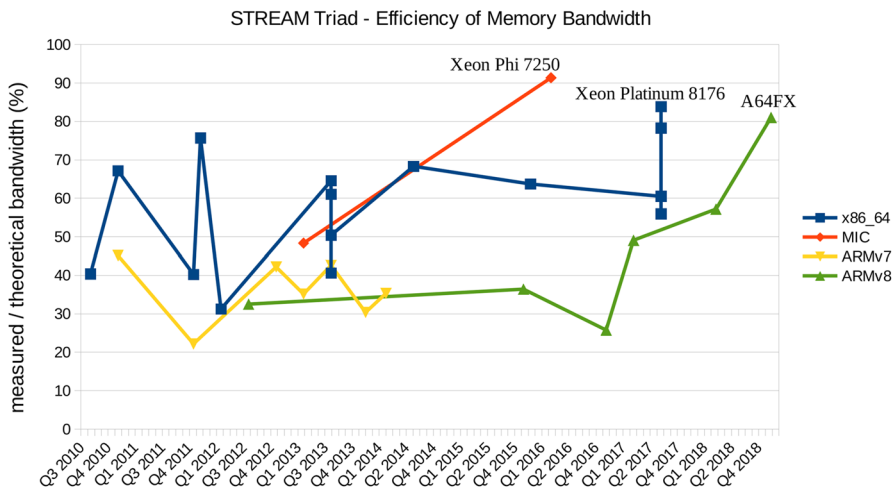


Fig. 2 Efficiency of memory bandwidth for STREAM Triad by date of processor release, grouped by architecture

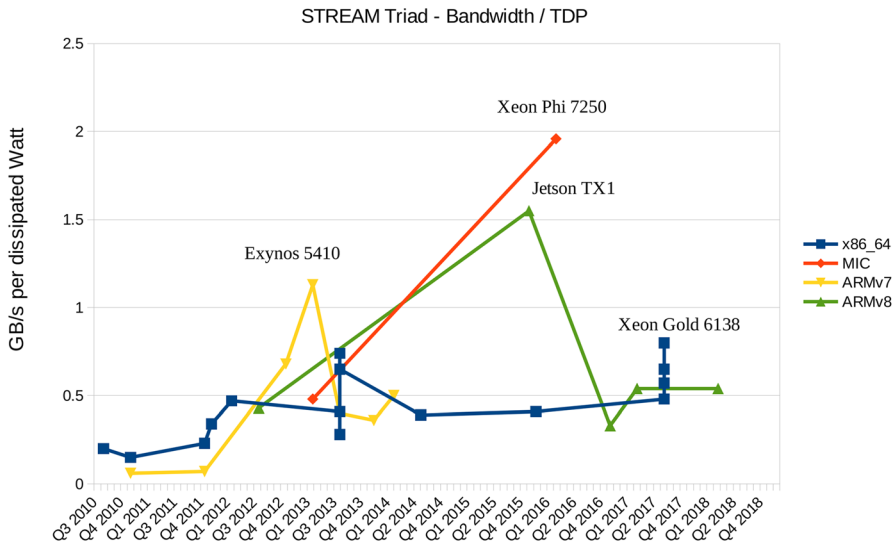


Fig. 3 Ratio between memory bandwidth for STREAM Triad and TDP reported by manufacturers

9 Conclusion

The need for energy efficiency to reach an Exaflop HPC system is driving intense research in the use of alternatives architectures and systems to replace the current, power hungry x86-based HPC. The number of proponents, as well as the volume of information and research about the viability of the ARM processor for HPC, is growing rapidly. In this context, we felt motivated to present a comprehensive study in the state of the art of the usage of ARM architecture for HPC. Our main contribution is to condense information on key aspects such as power efficiency, energy to solution, scalability, and support for specific architectural operations. Knowing that HPC is a moving target, we have strived to include the most recent developments regarding the ARMv8 architecture, including the latest A64FX processor by Fujitsu.

There have been many concerns and challenges for porting high-performance applications to ARM architectures, as an alternative to x86 in HPC systems. Architectural limitations such as lack of I/O ports for high-bandwidth interconnect, and lack of parallelization of double-precision floating points, compounded with the general poor state of the HPC software stack and need for porting scientific applications, all of these challenges required years of work. Here, we highlight the efforts of initiatives such as the Mont-Blanc project and Linaro, which have helped fill the software gap, at the same time driving necessary hardware changes into the ARMv8 architecture while maintaining an energy-efficient profile.

Besides experiments replacing x86 for ARM processors, the literature concerning this topic expanded into power models, scalability models, state of libraries and toolsets for ARM in support to scientific applications, use of co-processors and others. Although the use of the ARM architecture on HPC is recent, the literature is already quite extensive. Motivated by the volume of information available, this

review highlighted the main topics concerning the use of ARM, telling the story of the evolution of its architecture and development of the HPC ecosystem.

The use of co-processors on HPC has permitted major gains in performance and energy efficiency on recent HPC systems. The use of co-processors allows for specialized, computation-intensive applications to be offloaded to be processed on the co-processor. As seen in examples such as the Denver custom-made ARM processor for the NVIDIA TegraX2, the ARM architecture can be a better fit for those systems. From the collected results of memory bandwidth, we conclude that many-core CPUs such as the Xeon Phi can achieve very high efficiency, and that ARMv8 processors are following a more pronounced trend toward higher core count and efficiency compared to x86_64 processors. While the most recent server-class ARMv8 processors have been designed without GPUs, we see that there is a potential market for GPU-enabled SoCs that can achieve high performance with modest energy budget.

The introduction of the ARMv8 architecture has made ARM processors even more ubiquitous and beyond the realm of mobile computing. Already Amazon offers ARM-based compute instances for Cloud Computing, powered by Graviton processors that are modified Cortex-A72 versions [17]. Also in the server-side, the recent Kunpeng 920 processor by Huawei [53] is setting out to dominate the Big Data market with its 64 cores developed with 7nm process. ARM is also widening its offerings with a processor based on a new architecture, geared toward machine learning operations with low energy budget [24]. Compounded to the more aggressive lithography improvements in ARMv8 processors and cheaper processor prices, when compared to Intel, we can easily see ARM HPC becoming prevalent with its matured software ecosystem.

As the gap in single-threaded performance decreases, scalability concerns will become increasingly important. While the scientific community has already tested the scalability of ARMv8-powered clusters with a thousand cores, clusters with higher core counts are currently being built and evaluated, and so their scalability at hundreds of thousands of cores is still an open question.

Nonetheless, additional benefits of the ARM approach such as its licensing model, the improved capability to use it integrated together with co-processors and history of developing power-efficient processors are placing the ARM as a viable solution for the exascale power wall.

Acknowledgements The authors would like to acknowledge the National Laboratory for Scientific Computing Postgraduate program, CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) Ph.D. fellowship, the Atos Company through the cooperation project of Research, Development and Training of Human Resources in Computational Modeling and High-Performance Computing (conditioned to the receipt of resources by the Fundação de Apoio ao Desenvolvimento da Computação Científica-FACC) and CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) Grant Number 309873/2013-4.

References

1. HPC Case Study: CFD Applications on ARM. https://developer.arm.com/-/media/developer/products/software-tools/hpc/Documentation/UCAMB_Arm_CFDvFinal.pdf. Accessed 24 Apr 2019
2. Linaro releases. <https://releases.linaro.org/components/toolchain/gcc-linaro>. Accessed 5 May 2019

3. Viso geral sobre intel advanced vector extensions 512. <https://www.intel.com.br/content/www/br/pt/architecture-and-technology/avx-512-overview.html>. Accessed 22 Sept 2017
4. Linaro. <https://www.linaro.org/>. Accessed 5 May 2019
5. Openhpc releases (2019). <https://github.com/openhpc/ohpc/releases/>. Accessed 5 May 2019
6. Abdurachmanov D, Bockelman B, Elmer P, Eulisse G, Knight R, Muzaffar S (2015) Heterogeneous high throughput scientific computing with apm x-gene and intel xeon phi. *J Phys Conf Ser* 608(1):012033
7. Abdurachmanov D, Elmer P, Eulisse G, Knight R, Niemi T, Nurminen JK, Nyback F, Pestana G, Ou Z, Khan K (2015) Techniques and tools for measuring energy efficiency of scientific software applications. *J Phys Conf Ser* 608:012032
8. Abdurachmanov D, Elmer P, Eulisse G, Muzaffar S (2014) Initial explorations of arm processors for scientific computing. *J Phys Conf Ser* 523(1):012009
9. Addiego N (2017) Evaluation of the efficiency of an ARM-based beowulf cluster versus traditional desktop computing for high performance computing. Master's thesis, University of San Diego
10. Adhianto L, Banerjee S, Fagan M, Krentel M, Marin G, Mellor-Crummey J, Tallent NR (2010) Hpc toolkit: tools for performance analysis of optimized parallel programs. *Concurr Comput Pract Exp* 22(6):685–701
11. Allalen M, Brayford D, Tafani D, Weinberg V, Mohr B, Brömmel D, Halver R, Meinke J, Mohanty S (2015) The mont-blanc project: First phase successfully finished. arXiv preprint [arXiv :1508.05075](https://arxiv.org/abs/1508.05075)
12. ARM: ARM Architecture Reference Manual—ARMv7-A and ARMv7-R edition. https://static.docs.arm.com/ddi0406/c/DDI0406C_C_arm_architecture_reference_manual.pdf (2014). Accessed 24 Apr 2019
13. Armejach A, Caminal H, Cebrían JM, Langarita R, González-Alberquilla R, Adeniyi-Jones C, Valero M, Casas M, Moretó M (2019) Using arms scalable vector extension on stencil codes. *J Supercomput*. <https://doi.org/10.1007/s11227-019-02842-5>
14. Azimi R, Fox T, Gonzalez W, Reda S (2018) Scale-out vs scale-up: a study of arm-based socs on server-class workloads. *ACM Trans Model Perform Eval Comput Syst (TOMPECS)* 3(4):18
15. Azimi R, Zhan X, Reda S (2015) How good are low-power 64-bit socs for server-class workloads? In: 2015 IEEE International Symposium on Workload Characterization (IISWC), IEEE, pp 116–117
16. Banchelli F, Garcia M, Josep M, Mantovani F, Morillo J, Peiro K, Ramirez G, Teruel X, Mc Evoy G, Wanza J, Gracia J, Lumi A, Ganellari D, Schiffmann P (2019) Mb3 d6.9—performance analysis of applications and mini-applications and benchmarking on the project test platforms. Tech. rep. https://www.montblanc-project.eu/wp-content/uploads/2019/02/MB3_D6.9_Performance-analysis-of-applications-and-benchmarking-on-the-project-test-platforms.v1.0.pdf. Accessed 24 Apr 2019
17. Barr J (2018) EC2 Instances (A1) Powered by Arm-Based AWS Graviton Processors. <https://aws.amazon.com/blogs/aws/new-ec2-instances-a1-powered-by-arm-based-aws-graviton-processors>. Accessed 24 Apr 2019
18. Barrett BW, Brightwell R, Grant R, Hammond SD, Hemmert KS (2014) An evaluation of mpi message rate on hybrid-core processors. *Int J High Perform Comput Appl* 28(4):415–424. <https://doi.org/10.1177/1094342014552085>
19. Beserra D, Pinheiro MK, Souveyet C, Steffnel LA, Moreno ED (2017) Performance evaluation of os-level virtualization solutions for hpc purposes on soc-based systems. In: 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA), IEEE, pp 363–370
20. Bez JL, Bernart EE, Santos FF, Schnorr LM, Navaux POA (2016) Performance and energy efficiency analysis of HPC physics simulation applications in a cluster of arm processors. *Pract Exp Concurr Comput* 29:e4014
21. Blem E, Menon J, Sankaralingam K (2013) Power struggles: revisiting the risc vs. cisc debate on contemporary arm and x86 architectures. In: 2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA2013), IEEE, pp 1–12
22. Boggs D, Brown G, Tuck N, Venkatraman K (2015) Denver: Nvidia's first 64-bit arm processor. *IEEE Micro* 35(2):46–55
23. Brash D (2010) Extensions to the armv7-a architecture. In: 2010 IEEE Hot Chips 22 Symposium (HCS), IEEE, pp 1–21

24. Bratt I (2018) Arm's First Generation Machine Learning Processor. https://www.hotchips.org/hc30/2conf/2.07_ARM_ML_Processor_HC30_ARM_2018_08_17.pdf. Accessed 24 Apr 2019
25. Bull D, Das S, Shivashankar K, Dasika GS, Flautner K, Blaauw D (2011) A power-efficient 32 bit arm processor using timing-error detection and correction for transient-error tolerance and adaptation to pvt variation. *IEEE J Solid-State Circ* 46(1):18–31
26. Calore E, Mantovani F, Ruiz D (2018) Advanced performance analysis of hpc workloads on cavium thunderx. In: 2018 International Conference on High Performance Computing & Simulation (HPCS), IEEE, pp 375–382
27. Canuto M, Bosch R, Macías M, Guitart J (2016) A methodology for full-system power modeling in heterogeneous data centers. In: Proceedings of the 9th International Conference on Utility and Cloud Computing, ACM, pp 20–29
28. Castelló A, Duato J, Mayo R, Peña AJ, Quintana-Ortí ES, Roca V, Silla F (2014) On the use of remote gpus and low-power processors for the acceleration of scientific applications. In: The Fourth International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies (ENERGY), pp 57–62
29. Choi J, Dukhan M, Liu X, Vuduc R (2014) Algorithmic time, energy, and power on candidate hpc compute building blocks. In: 2014 IEEE 28th International Parallel and Distributed Processing Symposium, IEEE, pp 447–457
30. Cloutier MF, Paradis C, Weaver VM (2014) Design and analysis of a 32-bit embedded high-performance cluster optimized for energy and performance. In: Proceedings of the 1st International Workshop on Hardware-Software Co-Design for High Performance Computing, IEEE Press, pp 1–8
31. Cong J, Huang M, Wu D, Yu CH (2016) Heterogeneous datacenters: options and opportunities. In: Proceedings of the 53rd Annual Design Automation Conference, ACM, p 16
32. Corni E, Morganti L, Morigi MP, Brancaccio R, Bettuzzi M, Levi G, Peccenini E, Cesini D, Ferraro A (2016) X-ray computed tomography applied to objects of cultural heritage: Porting and testing the filtered back-projection reconstruction algorithm on low power systems-on-chip. In: 2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP), IEEE, pp 369–372
33. D'Agostino D, Quarati A, Clematis A, Morganti L, Corni E, Giansanti V, Cesini D, Merelli I (2019) Soc-based computing infrastructures for scientific applications and commercial services: performance and economic evaluations. *Future Gener Comput Syst* 96:11–22
34. De Gelas J (2017) AppliedMicro's X-Gene 3 SoC Begins Sampling: A Step in ARM's 2017 Server Ambitions. <https://www.anandtech.com/show/11189/appliedmicro-x-gene-3-soc-starts-sampling>. Accessed 24 Apr 2019
35. Duran A, Ayguadé E, Badia RM, Labarta J, Martinell L, Martorell X, Planas J (2011) Ompss: a proposal for programming heterogeneous multi-core architectures. *Parallel Process Lett* 21(02):173–193
36. Durand Y, Carpenter PM, Adami S, Bilas A, Dutoit D, Farcy A, Gaydadjiev G, Goodacre J, Katenen M, Marazakis M et al (2014) Euroserver: Energy efficient node for european micro-servers. In: 2014 17th Euromicro Conference on Digital System Design (DSD), IEEE, pp 206–213
37. Elangovan VK, Badia RM, Parra EA (2012) Ompss-openssl programming model for heterogeneous systems. In: International Workshop on Languages and Compilers for Parallel Computing, Springer, pp 96–111
38. Feng W, Scogland T. Green500. <https://www.top500.org/green500/>. Accessed 5 May 2019
39. Ferreron A, Jagtap R, Rusitoru R (2016) Identifying representative regions of parallel hpc applications: a cross-architectural evaluation. In: 2016 IEEE International Symposium on Workload Characterization (IISWC), IEEE, pp 1–2
40. Fialho L, Gracia J, Nigay A, Evoy M (2019) Mb3 d7.13—final report on enhancements to message passing. Tech. rep. https://www.montblanc-project.eu/wp-content/uploads/2019/02/MB3_D7.13_Final-report-on-enhancements-to-message-passing-2.pdf. Accessed 24 Apr 2019
41. Filiposka S, Mishev A, Juiz C (2016) Current prospects towards energy-efficient top hpc systems. *Comput Sci Inf Syst* 13(1):151–171
42. Fox T (2017) Revisiting the case of arm socs in high-performance computing clusters. Ph.D. thesis, School of Engineering, Brown University
43. Frid N, Ivošević D, Sruk V (2015) Heterogeneity impact on mp soc platforms performance. In: 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), IEEE, pp 1071–1076

44. Geveler M, Ribbrock D, Donner D, Ruelmann H, Höppke C, Schneider D, Tomaschewski D, Turek S (2016) The icarus white paper: a scalable, energy-efficient, solar-powered hpc center based on low power gpus. In: European Conference on Parallel Processing, Springer, pp 737–749
45. Gómez C, Martínez F, Armejach A, Moretó M, Mantovani F, Casas M (2019) Design space exploration of next-generation HPC machines. Barcelona Supercomputing Center. <http://hdl.handle.net/2117/131511>
46. Grant RE, Levenhagen M, Olivier SL, DeBonis D, Pedretti KT, Laros JH III (2016) Standardizing power monitoring and control at exascale. *Computer* 49(10):38–46
47. Grass T, Allandé C, Armejach A, Rico A, Ayguadé E, Labarta J, Valero M, Casas M, Moreto M (2016) Musa: a multi-level simulation approach for next-generation hpc machines. In: SC16: International Conference for High Performance Computing, Networking, Storage and Analysis, IEEE, pp 526–537
48. Grasso I, Radojkovic P, Rajovic N, Gelado I, Ramirez A (2014) Energy efficient hpc on embedded socs: Optimization techniques for mali gpu. In: 2014 IEEE 28th International on Parallel and Distributed Processing Symposium, IEEE, pp 123–132
49. Griessl R, Peykanu M, Hagemeyer J, Porrmann M, Krupop S, vor dem Berge M, Kiesel T, Christmann W (2014) A scalable server architecture for next-generation heterogeneous compute clusters. In: 2014 12th IEEE International Conference on Embedded and Ubiquitous Computing (EUC), IEEE, pp 146–153
50. Gu B, Kwak Y (2016) Map task allocation strategy in an arm-based hadoop cluster by using local storage as split cache. *Int J Adv Media Commun* 6(1):65–72
51. Gwennap L (2017) Performance Arms X-Gene 3 for Cloud. <https://www.linleygroup.com/uploads/x-gene-3-for-cloud.pdf>. Accessed 24 Apr 2019
52. Halpern M, Zhu Y, Reddi VJ (2016) Mobile cpu's rise to power: Quantifying the impact of generational mobile cpu design trends on performance, energy, and user satisfaction. In: 2016 IEEE International Symposium on High Performance Computer Architecture (HPCA), IEEE, pp 64–76
53. Huawei (2019) Huawei Unveils Industry's Highest-Performance ARM-based CPU. <https://www.huawei.com/en/press-events/news/2019/1/huawei-unveils-highest-performance-arm-based-cpu>. Accessed 24 Apr 2019
54. Iliescu DA, Petrogalli F. Arm scalable vector extension and application to machine learning. Tech. rep., ARM. <https://developer.arm.com/solutions/hpc/resources/hpc-white-papers/arm-scalable-vector-extensions-and-application-to-machine-learning>. Whitepaper Accessed 24 Apr 2019
55. Jackson A, Turner A, Weiland M, Johnson N, Perks O, Parsons M (2019) Evaluating the arm ecosystem for high performance computing. arXiv preprint [arXiv:1904.04250](https://arxiv.org/abs/1904.04250)
56. Jaquet D, Hasbani F, Flatresse P, Wilson R, Arnaud F, Cesana G, Di Gilio T, Lecocq C, Roy T, Chhabra A et al (2014) A 3 ghz dual core processor arm cortex tm-a9 in 28 nm utbb fd-soi cmos with ultra-wide voltage range and energy efficiency optimization. *IEEE J Solid-State Circ* 49(4):812–826
57. Jarus M, Varrette S, Oleksiak A, Bouvry P (2013) Performance evaluation and energy efficiency of high-density hpc platforms based on intel, amd and arm processors. In: European Conference on Energy Efficiency in Large Scale Distributed Systems, Springer, pp 182–200
58. Jin C, de Supinski BR, Abramson D, Poxon H, DeRose L, Dinh MN, Endrei M, Jessup ER (2016) A survey on software methods to improve the energy efficiency of parallel computing. *Int J High Perform Comput Appl* 31:1094342016665471
59. Jundt A, Cauble-Chantrenne A, Tiwari A, Peraza J, Laurenzano MA, Carrington L (2015) Compute bottlenecks on the new 64-bit arm. In: Proceedings of the 3rd International Workshop on Energy Efficient Supercomputing, ACM, p 6
60. Jung YW, Sok SW, Santoso GZ, Shin JS, Kim HY (2015) Prototype of light-weight hypervisor for arm server virtualization. In: Proceedings of the International Conference on Embedded Systems and Applications (ESA). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), p 36
61. Kaewkasi C, Srisuruk W (2014) Optimizing performance and power consumption for an arm-based big data cluster. In: TENCON 2014-2014 IEEE Region 10 Conference, IEEE, pp 1–6
62. Kaewkasi C, Srisuruk W (2014) A study of big data processing constraints on a low-power hadoop cluster. In: 2014 International Computer Science and Engineering Conference (ICSEC), IEEE, pp 267–272
63. Kalyanasundaram J, Simmhan Y (2017) Arm wrestling with big data: A study of arm64 and x64 servers for data intensive workloads. arXiv preprint [arXiv:1701.05996](https://arxiv.org/abs/1701.05996)

64. Keckskemeti G, Hajji W, Tso FP (2017) Modelling low power compute clusters for cloud simulation. In: 2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), IEEE, pp 39–45
65. Kennedy P. Intel Atom C2550 Benchmarks - 4 core Avoton tested. <https://www.servethehome.com/intel-atom-c2550-benchmarks-4-core-avoton-tested>. Accessed on 24 Apr 2019
66. Kodama Y, Odajima T, Asato A, Sato M (2019) Evaluation of the riken post-k processor simulator. arXiv preprint [arXiv:1904.06451](https://arxiv.org/abs/1904.06451)
67. Kumar D, Memon S, Thebo LA (2018) Design, implementation & performance analysis of low cost high performance computing (hpc) clusters. In: 2018 12th International Conference on Signal Processing and Communication Systems (ICSPCS), IEEE, pp 1–6
68. Laurenzano MA, Tiwari A, Cauble-Chantrenne A, Jundt A, Ward WA, Campbell R, Carrington L (2016) Characterization and bottleneck analysis of a 64-bit armv8 platform. In: 2016 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), pp 36–45. <https://doi.org/10.1109/ISPASS.2016.7482072>
69. Lee Y, Kim S (2015) Empirical characterization of power efficiency for large scale data processing. In: 2015 17th International Conference on Advanced Communication Technology (ICACT), IEEE, pp 787–790
70. Loghin D, Tudor BM, Zhang H, Ooi BC, Teo YM (2015) A performance study of big data on small nodes. *Proc VLDB Endow* 8(7):762–773
71. Lorenzon AF, Cera MC, Beck ACS (2015) On the influence of static power consumption in multicore embedded systems. In: 2015 IEEE International Symposium on Circuits and Systems (ISCAS), IEEE, pp 1374–1377
72. Lorenzon AF, Sartor AL, Cera MC, Beck ACS (2015) Optimized use of parallel programming interfaces in multithreaded embedded architectures. In: 2015 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), IEEE, pp 410–415
73. Luecke GR, Groth BM, Weeks NT, Kraeva M (2017) Comparing allinea's and intel's performance tools for hpc. In: Proceedings of the 25th High Performance Computing Symposium, HPC '17, pp. 3:1–3:12. Society for Computer Simulation International, San Diego, CA, USA. <http://dl.acm.org/citation.cfm?id=3108096.3108099>
74. Mantovani F, Calore E (2018) Performance and power analysis of hpc workloads on heterogeneous multi-node clusters. *J Low Power Electron Appl* 8(2):13
75. Mantovani F, Ruiz D, Vilarrubi O, Martorell X, Nieto D, Auweter A, Tafani D, Adeniyi-Jones C, Gloaguen H, Utrera G (2015) D5.11—Final report on porting and tuning of system software to ARM architecture. Tech. rep. <https://www.montblanc-project.eu/wp-content/uploads/2019/01/D7.17-Final-report-on-Arm-optimized-Fortran-compiler-and-math-libraries.pdf>. Accessed 24 Apr 2019
76. Mappuji A, Effendy N, Mustaghfirin M, Sondok F, Yuniar RP, Pangesti SP (2016) Study of raspberry pi 2 quad-core cortex-a7 cpu cluster as a mini supercomputer. In: 2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE), IEEE, pp 1–4
77. Maqbool J, Oh S, Fox GC (2015) Evaluating arm hpc clusters for scientific workloads. *Concurr Comput Pract Exp* 27(17):5390–5410
78. Maynard C, Selwood P (2016) Exascale computing research priorities for the met office forecasting research technical report no: 611
79. McCalpin J. STREAM Benchmark results. https://www.cs.virginia.edu/stream/by_date/Bandwidth.html. Accessed 24 Apr 2019
80. McCalpin JD (2016) Memory bandwidth and system balance in HPC systems. SC16 Invited Talk. <http://tiny.cc/ipvh7y>. Accessed 5 May 2019
81. McCalpin JD (1995) Memory bandwidth and machine balance in current high performance computers. *IEEE Comput Soc Tech Comm Comput Arch (TCCA) Newslett* 2:19–25
82. McIntoshSmith S, Price J, Deakin T, Poenaru A (2019) A performance analysis of the first generation of hpc-optimized arm processors. *Pract Exp Concurr Comput*. <https://doi.org/10.1002/cpe.5110>
83. Mellanox: InfiniBand Cards - Overview. http://www.mellanox.com/page/infiniband_cards_overview (2014). Accessed 24 Apr 2019
84. Milluzzi A, George A, Lam H (2016) Computational and memory analysis of tegra socs. In: 2016 IEEE High Performance Extreme Computing Conference (HPEC), pp 1–7. <https://doi.org/10.1109/HPEC.2016.7761602>

85. Moorthy P, Kapre N (2015) Zedwulf: Power-performance tradeoffs of a 32-node zynq soc cluster. In: 2015 IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), IEEE, pp 68–75
86. Morales F, Bismarck JL (2016) Evaluating gem5 and qemu virtual platforms for arm multicore architectures. Master's thesis, KTH, School of Information and Communication Technology (ICT)
87. Morganti L, Cesini D, Ferraro A (2016) Evaluating systems on chip through hpc bioinformatic and astrophysic applications. In: 2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP), IEEE, pp 541–544
88. Nikolskiy V, Stegailov V (2016) Floating-point performance of arm cores and their efficiency in classical molecular dynamics. *J Phys Conf Ser* 681(1):012049
89. NVIDIA: Tegra K1 Technical Reference Manual. http://l0ke.as.arizona.edu/~ckulesa/binaries/ONR/TegraK1_TRM_DP06905001.pdf (2014). Accessed 24 Apr 2019
90. Oyarzun G, Borrell R, Gorobets A, Mantovani F, Oliva A (2018) Efficient cfd code implementation for the arm-based mont-blanc architecture. *Future Gener Comput Syst* 79:786–796
91. Plugaru V, Varrette S, Bouvry P (2014) Performance analysis of cloud environments on top of energy-efficient platforms featuring low power processors. In: 2014 IEEE 6th International Conference on Cloud Computing Technology and Science (CloudCom), IEEE, pp 416–425
92. Pruitt DD, Freudenthal EA (2016) Preliminary investigation of mobile system features potentially relevant to hpc. In: Proceedings of the 4th International Workshop on Energy Efficient Supercomputing, IEEE Press, pp 54–60
93. Puzović M, Manne S, GalOn S, Ono M (2016) Quantifying energy use in dense shared memory hpc node. In: Proceedings of the 4th International Workshop on Energy Efficient Supercomputing, E2SC '16, pp 16–23. IEEE Press, Piscataway, NJ, USA. <https://doi.org/10.1109/E2SC.2016.7>
94. Raho M, Spyridakis A, Paolino M, Raho D (2015) Kvm, xen and docker: A performance analysis for arm based nfv and cloud computing. In: 2015 IEEE 3rd Workshop on Advances in IEEE Information, Electronic and Electrical Engineering (AIEEE), pp 1–8
95. Rajovic N, Carpenter PM, Gelado I, Puzovic N, Ramirez A, Valero M (2013) Supercomputing with commodity cpus: Are mobile socs ready for hpc? In: 2013 SC-International Conference for High Performance Computing, Networking, Storage and Analysis (SC), IEEE, pp 1–12
96. Rajovic N, Rico A, Mantovani F, Ruiz D, Vilarrubi JO, Gomez C, Backes L, Nieto D, Servat H, Martorell X, Labarta J, Ayguade E, Adeniyi-Jones C, Derradji S, Gloaguen H, Lanucara P, Sanna N, Mehaut JF, Pouget K, Videau B, Boyer E, Allalen M, Auweter A, Brayford D, Tafani D, Weinberg V, Brömmel D, Halver R, Meinke JH, Beivide R, Benito M, Vallejo E, Valero M, Ramirez A (2016) The mont-blanc prototype: An alternative approach for hpc systems. In: SC16: International Conference for High Performance Computing, Networking, Storage and Analysis, pp 444–455. <https://doi.org/10.1109/SC.2016.37>
97. Rajovic N, Rico A, Puzovic N, Adeniyi-Jones C, Ramirez A (2014) Tibidabo: making the case for an arm-based hpc system. *Future Gener Comput Syst* 36:322–334
98. Reeda R, Cox MA, Wrigley T, Mellado B (2015) A cpu benchmarking characterization of arm based processors. *Computer* 7(3):581–586
99. Rethinagiri SK, Palomar O, Moreno JA, Unsal O, Cristal A (2015) Trigeneous platforms for energy efficient computing of hpc applications. In: 2015 IEEE 22nd International Conference on High Performance Computing (HiPC), IEEE, pp 264–274
100. Ross JA, Richie DA, Park SJ, Shires DR, Pollock LL (2014) A case study of opencl on an android mobile gpu. In: 2014 IEEE High Performance Extreme Computing Conference (HPEC), pp 1–6
101. Rostrolla G, da Rosa Righi R, Rodrigues VF, Velho P, Padoin EL (2015) Greenhpc: a novel framework to measure energy consumption on hpc applications. In: Sustainable Internet and ICT for Sustainability (SustainIT), 2015, IEEE, pp 1–8
102. Ruiz D, Mantovani F, Casas M, Labarta J, Spiga F (2018) The HPCG benchmark: analysis, shared memory preliminary improvements and evaluation on an arm-based platform. Polytechnic University of Catalonia. <http://hdl.handle.net/2117/116642>. Accessed 5 May 2019
103. Rupp K. Knights Landing vs. Knights Corner, Haswell, Ivy Bridge, and Sandy Bridge: STREAM benchmark results. <https://www.karlsruhp.net/2016/07/knights-landing-vs-knights-corner-haswell-ivy-bridge-and-sandy-bridge-stream-benchmark-results>. Accessed 24 Apr 2019
104. Schulz KW, Baird CR, Brayford D, Georgiou Y, Kurtzer GM, Simmel D, Sterling T, Sundararajan N, Van Hensbergen E (2016) Cluster computing with openhpc. In: HPCSPRO16: Inaugural HPC systems professionals workshop. <http://hdl.handle.net/2022/21082>

105. Schürmans S, Onnebrink G, Leupers R, Ascheid G, Chen X (2016) Frequency-aware esl power estimation for arm cortex-a9 using a black box processor model. *ACM Trans Embed Comput Syst (TECS)* 16(1):26
106. Selinger A, Rupp K, Selberherr S (2016) Evaluation of mobile arm-based socs for high performance computing. In: *Society for Computer Simulation International Proceedings of the 24th High Performance Computing Symposium*, p 21
107. Sheen SK (2016) Astro-a low-cost, low-power cluster for cpu-gpu hybrid computing using the jetson TK1. Master's thesis, California Polytechnic State University. <https://doi.org/10.15368/these.s.2016.45>
108. Shore C. Porting to 64-bit arm. Tech. rep., ARM. <https://community.arm.com/developer/ip-products/processors/b/processors-ip-blog/posts/porting-to-arm-64-bit>. Whitepaper Accessed 24 Apr 2019
109. Silvano C, Agosta G, Bartolini A, Beccari AR, Benini L, Bispo J, Cmar R, Cardoso JM, Cavazzoni C, Martinović J et al (2016) Autotuning and adaptivity approach for energy efficient exascale hpc systems: the antarex approach. In: *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp 708–713. IEEE (2016)
110. Sirin U, Appuswamy R, Ailamaki A (2016) Oltp on a server-grade arm: power, throughput and latency comparison. In: *Proceedings of the 12th International Workshop on Data Management on New Hardware*, ACM, p 10
111. Stegailov V, Vecher V (2018) Efficiency analysis of intel, AMD and Nvidia 64-Bit hardware for memory-bound problems: a case study of Ab initio calculations with VASP, pp 81–90. https://doi.org/10.1007/978-3-319-78054-2_8
112. Stephens N, Biles S, Boettcher M, Eapen J, Eyole M, Gabrielli G, Horsnell M, Magklis G, Martinez A, Premillieu N et al (2017) The arm scalable vector extension. *IEEE Micro* 37(2):26–39
113. Stokke KR, Stensland HK, Griwodz C, Halvorsen P (2016) A high-precision, hybrid gpu, cpu and ram power model for generic multimedia workloads. In: *Proceedings of the 7th International Conference on Multimedia Systems, MMSys '16*, pp 14:1–14:12. ACM, New York, NY, USA. <https://doi.org/10.1145/2910017.2910591>
114. Stokke KR, Stensland HK, Halvorsen P, Griwodz C (2016) High-precision power modelling of the tegra k1 variable smp processor architecture. In: *2016 IEEE 10th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSOC)*, pp 193–200. <https://doi.org/10.1109/MCSOC.2016.28>
115. Strohmaier E, Dongarra J, Horst S, Meuer M, Meuer H (2019) Top 500 The List. <https://www.top500.org/>. Accessed 5 May 2019
116. Sundriyal V, Fought E, Sosonkina M, Windus TL (2016) Power profiling and evaluating the effect of frequency scaling on nwchem. In: *Society for Computer Simulation International Proceedings of the 24th High Performance Computing Symposium*, p 19
117. Thompson SE, Parthasarathy S (2006) Moore's law: the future of si microelectronics. *Mater today* 9(6):20–25
118. Tiwari A, Keipert K, Jundt A, Peraza J, Leang SS, Laurenzano M, Gordon MS, Carrington L (2015) Performance and energy efficiency analysis of 64-bit arm using games. In: *Proceedings of the 2nd International Workshop on Hardware-Software Co-Design for High Performance Computing*, ACM, p 8
119. Trader T (2014) The exascale revolution. <https://www.hpcwire.com/2014/10/23/exascale-revolution/>. Accessed 23 July 2016
120. Villebonnet V, Da Costa G, Lefevre L, Pierson JM, Stolf P (2014) Towards generalizing"" big little"" for energy proportional hpc and cloud infrastructures. In: *2014 IEEE Fourth International Conference on Big Data and Cloud Computing (BdCloud)*, IEEE, pp 703–710
121. Weloli JW, Bilavarn S, Derradji S, Belleudy C, Lesmanne S (2016) Efficiency modeling and analysis of 64-bit arm clusters for hpc. In: *2016 EuroMicro Conference on Digital System Design (DSD)*, pp 342–347. <https://doi.org/10.1109/DSD.2016.74>
122. Whaley RC, Petit A, Dongarra JJ (2001) Automated empirical optimizations of software and the atlas project. *Parallel Comput* 27(1–2):3–35
123. Whatmough PN, Das S, Hadjilambrou Z, Bull DM (2017) Power integrity analysis of a 28 nm dual-core arm cortex-a57 cluster using an all-digital power delivery monitor. *IEEE J Solid-State Circ* 52(6):1643–1654
124. Wrigleya G, Reed R, Mellado B (2015) Memory benchmarking characterisation of arm-based socs. *Computer* 7(3):607–617

125. Xie X (2016) Low-power technologies in high-performance computer: trends and perspectives. *Natl Sci Rev* 3(1):23–25
126. Yoshida T (2018) Fujitsu high performance cpu for the post-k computer. In: *Hot Chips 30 Symposium (HCS)*, Series Hot Chips, vol 18
127. Zhang J, You S, Gruenwald L (2015) Tiny gpu cluster for big spatial data: A preliminary performance evaluation. In: *2015 IEEE 35th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, IEEE, pp 142–147
128. Zhirnov VV, Cavin RK, Hutchby JA, Bourianoff GI (2003) Limits to binary logic switch scaling—a gedanken model. *Proc IEEE* 91(11):1934–1939
129. Zhu Y, Mattina M, Whatmough P (2018) Mobile machine learning hardware at arm: a systems-on-chip (soc) perspective. arXiv preprint [arXiv:1801.06274](https://arxiv.org/abs/1801.06274)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.