Contents lists available at ScienceDirect

Future Generation Computer Systems

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

SoC-based computing infrastructures for scientific applications and commercial services: Performance and economic evaluations

Daniele D'Agostino [a], Alfonso Quarati [a], Andrea Clematis [a], Lucia Morganti [b], Elena Corni [b], Valentina Giansanti [c], Daniele Cesini [b], Ivan Merelli [c,*]

[a] *Institute for Applied Mathematics and Information Technologies "E. Magenes", National Research Council of Italy, Genoa, Italy*
[b] *CNAF Section - Italian Institute for Nuclear Physics, Bologna, Italy*
[c] *Institute for Biomedical Technologies, National Research Council of Italy, Segrate (Milan), Italy*

HIGHLIGHTS

- The article discusses the sustainability and effectiveness of adopting low-power processors in HPC and Cloud infrastructures.
- The analysis focuses on the achievable trade-off between time-to-solution and energy-to-solution together with economical aspects.
- The tests have been executed on the best state-of-the-art low-power System-on-Chip platforms.

ARTICLE INFO

Article history:
Received 14 May 2018
Received in revised form 29 November 2018
Accepted 15 January 2019
Available online 5 February 2019

Keywords:
Low power Systems-on-Chip
N-body benchmark
Deep learning
Next-Generation Sequencing
Performance and economic evaluations

ABSTRACT

Energy consumption represents one of the most relevant issues by now in operating computing infrastructures, from traditional High Performance Computing Centers to Cloud Data Centers. Low power System-on-Chip (SoC) architectures, originally developed in the context of mobile and embedded technologies, are becoming attractive also for scientific and industrial applications given their increasing computing performances, coupled with relatively low costs and power demands. In this paper, we investigate the performance of the most representative SoCs for a computational intensive N-body benchmark, a simple deep learning based application and a real-life application taken from the field of molecular biology. The goal is to assess the trade-off among time-to-solution, energy-to-solution and economical aspects for both scientific and commercial purposes they are able to achieve in comparison to traditional server-grade architectures adopted in present infrastructures.

1. Introduction

Nowadays the costs related to the running of applications, and consequently the provisioning of services, are more and more dominated by the electricity bill, therefore the adoption of energy-efficient solutions is crucial [1]. This is particular important for large Information Technology (IT) infrastructures, such as High Performance Computing (HPC) facilities or Cloud Data Centers, which demand energy efficient servers to keep their carbon footprint within acceptable limits. In fact, as reported in [2], data centers in the U.S. consumed in 2014 about 70 billion KWh, representing 1.8% of total U.S. electricity consumption. The perspective is to consume approximately 73 billion KWh in 2020. A similar issue holds true also in the HPC research field, which is facing major challenges on the path towards exascale computing. An exascale system can be defined as a supercomputer that can solve scientific problems 10x faster than the best petaflop systems[1] in a power envelope of 20–30 MW. By simply scaling current technologies to exascale, a supercomputer would consume 100 MW, i.e. 10x more than today. Thus, the need for new solutions arises.

Several techniques and best practices have been proposed to face this issue, e.g. [3]. In particular, on the one hand, high-end processors are quickly introducing more advanced power-saving and power-monitoring technologies [4–6]. On the other hand, a new class of low-power processors, often called Systems-on-Chip (SoCs), are gaining an increasing interest in many applicative fields [7,8]. Originally designed for the mobile market, SoCs are progressively reducing the performance gap with high-end processors, with the added value of keeping a competitive edge on costs, reducing their carbon footprint and preserving the environment [9].

* Corresponding author.
E-mail addresses: dagostino@ge.imati.cnr.it (D. D'Agostino), quarati@ge.imati.cnr.it (A. Quarati), clematis@ge.imati.cnr.it (A. Clematis), lucia.morganti@cnaf.infn.it (L. Morganti), elena.corni@cnaf.infn.it (E. Corni), valentina.giansanti@itb.cnr.it (V. Giansanti), daniele.cesini@cnaf.infn.it (D. Cesini), ivan.merelli@itb.cnr.it (I. Merelli).

[1] https://www.top500.org.

https://doi.org/10.1016/j.future.2019.01.024
0167-739X/© 2019 Elsevier B.V. All rights reserved.

The superior performance/consumption ratio of SoCs is driven by the growing demands for hand-held devices and energy-savvy boards in mobile and embedded industrial applications. An example can be represented by portable sequencing machines, such as the Oxford Nanopore Minion [10], where power consumption for on field data analysis represents a major issue. This device will lead to the direct analysis of genomes of humans, animals or plants in remote regions of the world, or to analyze the composition of the microbioma in air-filters, water or soil samples in a simple and portable way, possibly coupling Edge, Fog and Cloud computing [11].

Indeed, the primary design goal for embedded processors has been low power consumption because of their use in battery-powered devices. By contrast, the current power-hungry traditional servers were designed for high floating point performance. Moving away from their primordial mobile and embedded worlds, SoCs are conceivably becoming an interesting alternative architecture for running scientific, but also commercial-oriented, applications without sacrificing too much the performances and functionalities of traditional servers [12].

On the contrary, they presently have not received great attention for equipping computing infrastructures. Investigating the potential and performances of low power SoC architectures for scientific and non-scientific workloads in traditional HPC environments as well as in Cloud computing is the aim of the COmputing on SoC Architectures (COSA) project,[2] an ongoing initiative funded by the Italian Institute for Nuclear Physics (INFN). The COSA project is exploring limits and benefits of low power SoCs compared to the current mainstream high-end x86/CUDA server architectures [13].

Within this initiative, the goal of this work is to assess the achievable trade-off among time-to-solution, energy-to-solution and economical aspects of SoC-based infrastructures. This paper represents an extended version of [14], aiming at a more extensive evaluation of the performance of the heterogeneous SoC architectures that equip the COSA lab. In particular, we exploited most of the state-of-the-art SoC devices, and thus one of the original and key findings is a consistent and meaningful comparison among them.

At first we considered two benchmarks, represented by the widely used, computational intensive N-body algorithm and the use of a Deep Learning approach applied to a classification problem. In particular, Deep Learning is part of the broader family of machine learning methods and deals with algorithms inspired by the brain structure that can be used for several tasks, such as for example the classification of features and images. In this case, the focus was on the possibility of running classification approaches on SoC devices. Considering that we used very popular Deep Learning libraries for our tests, our results can be applied to a wide range of applications.

Even if benchmarking is a widespread method to measure and evaluate the performance of computer platforms, the most accurate answer to application-specific needs should take into account their actual implementation [15]. By analyzing the system indicators which are more stressed by the application under some typical and known conditions (e.g. input data size, degree of parallelism, average bandwidth consumption), it should be possible to figure out the behavior of the system when executing the application in similar future scenarios.

This is the reason why we also present an evaluation of the feasibility, performance and cost requirements for a small-medium enterprise (SME) offering a service based on a real-life application taken from the field of molecular biology, namely the Next-Generation Sequencing (NGS) analysis. We discuss the achievable trade-off figures among acquisition costs, energy costs and performance comparing SoCs and standards enterprise-oriented platforms.

The paper is organized as follows. Section 2 presents the related works and the COSA lab, i.e. the hardware, energy and economic specifications of the architectures employed in our tests. The performance achievable with the N-body algorithm are discussed in Section 3, while the bioinformatic and NGS applications are analyzed in Sections 4 and 5 respectively. Conclusions and future developments are outlined in Section 6.

2. Materials and methods

In this Section, after a brief overview of related works, we describe the platforms used in the tests.

2.1. Related works

The sustainability of IT infrastructures, and Cloud data centers in particular, is a major research topic. For example since 2010 Greenpeace has been calling to power data centers on renewable energy and monitors this process every year.[3] Power-related aspects of Cloud data-centers have been widely analyzed (e.g. [16, 17]) as well as the use of renewable energy [18] and the location of new data centers.[4]

In general, virtualization and consolidation of servers are effective methods to reduce energy usage [19]. Then, modern server designs have automatic power-off or power-saving capabilities built in, that can be exploited to save power when they are not running heavy loads [20]. Instead, the use of low-power SoCs has not been extensively investigated so far. Considering that the power consumption of the first petascale system was of about 2.4 MW, the HPC scientific community, started in 2009 the discussion on how to scale up to 1 Exaflop while staying in a power envelope of 20–30 MW [21]. So far, several research projects were funded to tackle the key issue of designing new hardware components, together with suitable programming environments.

The Mont-Blanc project[5] [22] has deployed several generations of HPC clusters based on ARM processors, developing also the corresponding eco-system of HPC tools targeted to this architecture [23]. The latest phase 3 system, named Dibona, is going to be commercialized in 2018.[6] Another project along this direction is the EUROSERVER[7] [24] coordinated by CEA, which aims at designing and prototyping technology, architecture, and system software for the next generation of data center "Micro-Servers", exploiting 64-bit ARM cores. Many research groups are on the same research line, exploring different hardware low-power platforms or software techniques: some are exploring Dynamic Voltage and Frequency Scaling (DVFS) techniques as a way to modulate power consumption of processor and memory, scaling the clock frequency of one or both sub-systems according to the execution of memory- or compute-bound application kernels [20]. More recently, also the Near Threshold Voltage (NTV) computing [25] technique has been employed for making the processor to work at even lower voltages. Other initiatives dealing with the creation of low-power based machines are the Spiking Neural Network Architecture (SpiNNaker) project[8] [26], proposed by the Advanced Processor Technologies Research group at the University of Manchester and the project ExaNeSt [27].

[2] http://cosa-project.it.

[3] http://www.clickclean.org/international/en/.

[4] http://www.vertatique.com/importance-location-green-data-centers.

[5] http://www.montblanc-project.eu/.

[6] http://montblanc-project.eu/prototypes.

[7] http://www.euroserver-project.eu/.

[8] http://apt.cs.manchester.ac.uk/projects/SpiNNaker/.

Table 1
Hardware specifications of the architecture of the COSA lab hosted at INFN-CNAF in Bologna plus the traditional reference architecture used for comparisons. The Bill Of Material — BOM corresponds to the money spent to acquire each single platform.

Platform	CPU cores	GPU cores	TDP (W)	RAM (GB)	BOM (€)
Pentium N3700	4	–	6	16	300
Pentium J4205	4	–	10	16	300
Jetson TK1	4	192	11	8	160
Jetson TX1	4	256	15	8	480
Avoton C2750	8	–	20	16	800
Xeon D1540	8	–	45	16	1100
Atom C3958	16	–	25	32	1050
Dual Xeon E5-2683	32	–	250	256	5000
NVIDIA Tesla K20	–	2496	225	5	600

2.2. Experimental setup

To validate our approach, we experimented different applications on the heterogeneous testbed summarized in Table 1. It is composed by two classes of machines: Xeon E5-2683v4 and NVIDIA Tesla K20 Graphic Card are server-grade platforms hosted in the server room of the CNAF production data center, while Pentium N3700, Pentium J4205, Jetson TK1, Jetson TX1, Avoton C2750, Xeon D1540 and Atom C3958, are mini-ITX boards hosted in the data center.

Despite the COSA lab is equipped with some ARM-based CPUs, in the tests we have considered mainly x86-based hardware because the porting of the software to these platforms was straightforward, having all the dependencies already compiled and available. The evaluation of the effort needed to exploit these architectures from the developers point of view will be investigated as a future work.

The differences between the platforms are evident looking at the columns reporting the thermal design power (TDP) and median Bill Of Material (BOM), i.e. the cost for acquiring each platform. Of course, the Xeon E5 platform is more expensive than the others, also because it represents an high-end server, i.e. it is one node of the HPC cluster equipped with multiple sockets, redundant power supplies, fans, disks, and huge RAM amounts.

This platform is our reference machine. In detail, it is based on a Dual Xeon E5-2683v4, a 64-bit 16 core x86 CPU introduced in early 2016. It operates at 2.1 GHz base frequency with a turbo boost of 3 GHz for a single active core and it has Hyper-Threading capabilities. It is manufactured on a 14 nm process and is based on the Broadwell microarchitecture. It supports up to 1.5 TB of ECC DDR4 RAM and it is equipped with 40 MB L3 cache. It is the only microprocessors in this work that can be installed in dual socket configuration.

For what concerns the GPUs, we consider as reference the K20 GPU accelerator card. Released in 2012, it is equipped with 2496 CUDA cores and it still represents a device with relevant performance at a low price, because the last generation of TESLA accelerators cost up to 10,000 €. In both cases, the price is related to the acquisition of the card itself, that has to be hosted in a server, e.g. the above one based on the Xeon E5.

The other platforms can be grouped in server-grade low power SoCs (Avoton C2750, Xeon D1540, Atom C3958) and in mobile-grade SoCs (Pentium N3700, Pentium J4205, Jetson TK1, Jetson TX1). In the following paragraphs, we briefly describe their main features.

Low-power mobile-grade.

- The Pentium N3700 is a very low power (i.e. 6 W TDP) 64-bit quad-core system on a chip released in early 2015. The N3700 is manufactured in 14 nm lithography and is based on the Airmont microarchitecture. It operates at 1.6 GHz with turbo mode of up to 2.4 GHz. It is equipped with a 2 MB L2 cache but no L3 cache. The N3700 SoC incorporates the HD Graphics GPU (16 execution units at 400 MHz). It supports 8GB of DDR3 low voltage RAM at 1600 MHz (no ECC).
- The Pentium J4205 is a quad-core 64-bit x86 SoC released in 2016. The processor is based on Goldmont microarchitecture and is manufactured on a 14 nm process. The CPU operates at 1.5 GHz (2.6 GHz burst frequency) and has a TDP of 10 W. The SoC incorporates the Intel's HD Graphics 505 GPU at 250 MHz (burst frequency of 800 MHz). It supports up to 8GB of DDR3 RAM (no ECC) and features a 2 MB L2 cache, but no L3.
- The Jetson TK1 board is based on the Tegra K1 NVIDIA SoC. It is a mobile processor (TDP of about 10 W) that features a CPU and a GPU based on the architecture similar to a modern desktop NVIDIA GPU. However, it still uses the low power draw of a mobile chip. Tegra K1 allows embedded devices to use the same CUDA code that runs on a desktop or on a server GPU. The CPU is an NVIDIA "4-Plus-1" 2.32 GHz ARM quad-core Cortex-A15 (plus an energy saving shadow core). The GPU is a NVIDIA Kepler GK20a GPU with 192 SM3.2 CUDA cores (up to 326 GFLOPS). The board is equipped with 2GB of RAM (no ECC).
- The Jetson TX1 is a 64 bit evolution of the previous board and it is based on an embedded system-on-module (SoM) with quad-core ARM Cortex-A57, 4GB LPDDR4 and with an integrated 256-core Maxwell GPU.

Low-power server-grade.

- The Avoton C2750 (now rebranded as Atom C2750) was released at the end of 2013 and provides a low power (20 W TDP) octa-core, 2.4GHz CPU. Produced with the 22 nm lithography technology, it is based on the Silvermont microarchitecture. As the C3958 (see below), it supports ECC memory, up to 64GB.
- The Xeon D1540 is a 64-bit octa-core x86-64 microserver SoC released in March 2015. It is based on the same Broadwell microarchitecture of the Xeon E5, it is manufactured in 14 nm process and can support up to 128GB of DDR4 ECC RAM. The D1540 features a 12 MB L3 cache and operates at 2 GHz with a turbo frequency of 2.6 GHz. Two 10 Gigabit-Ethernet connections are embedded into the SoC.
- Atom C3958 is a 64-bit multi core, low power (25 W TDP) x86 microserver system released in the third quarter of 2017. It is manufactured on a 14 nm process and it is based on the Goldmont microarchitecture. The base CPU clock is 2.0 GHz. The C3958 features 16 MB L2 cache and supports up to 256 GB of dual-channel DDR4-2400 ECC memory, with 10 Gigabit-Ethernet connections embedded into the SoC.

3. The N-body application

Many physical phenomena involve, or can be simulated with, particle systems, where each particle interacts with all other particles according to the laws of physics. Examples include the gravitational interaction among the stars in a galaxy, the Coulomb forces exerted by the atoms in a molecule for molecular dynamics applications and turbulent fluid flow studies. The challenge of efficiently carrying out the related calculations is generally known as the N-body problem, defined as follow: given an ensemble of N entities in space whose interaction is governed by a potential function, the N-body problem is to calculate the force on each body in the ensemble that results from its interaction with all other bodies.

Mathematically, given N bodies with mass m_i, initial position x_i and initial velocity v_i, the N-body problem can be formulated as

$$U(x_i) = \sum_{j=0}^{N-1} F(x_i, x_j)$$

where $U(x_i)$ is a physical quantity at x_i which can be obtained by summing the pairwise interactions with all the other particles of the system. In particular, the gravitational force exerted on the particle x_i by particle x_j is expressed as

$$F(x_i, x_j) = Gm_i m_j \frac{x_j - x_i}{\|x_j - x_i\|^3}$$

where G is the gravitational constant and m_i, m_j the particles masses.

The solution of the N-body problem proceeds over timesteps, each time computing the force on every body and thereby updating its position and other attributes. The All Pairs approach is the straightforward solution technique, which evaluates the interactions between all the pairs of the N bodies. It requires $O(N^2)$ operations at each timestep, which makes the algorithm very popular for demonstrating the speedups achievable with a computational system.

An in-depth evaluation of the performance achievable on classic HPC architectures using three implementations of the algorithm, i.e. a sequential implementation, its parallelization for single-node multi-core architectures and a parallelization for CUDA architectures has been presented in [28]. Here we adopted the same approach for evaluating the platforms of Table 1.

Listing 1: Main loop of the sequential algorithm for the solution of the N-body problem.

```
for (k=0; k<timesteps; k++) {    //FIRST loop
  swap(oldbodies,newbodies);
  for (i=0; i<N; i++) {    //SECOND loop
      tot_force_i[X] = tot_force_i[Y] = tot_force_i[Z] = 0.0;
      for (j=0; j<N; j++) {    //THIRD loop
 if (j==i) continue;
     //20 floating point operations
     r[X] = oldbodies[j].pos[X]    oldbodies[i].pos[X];
         // analogous for r[Y] and r[Z]
     distSqr = r[X]*r[X] + r[Y]*r[Y] + r[Z]*r[Z] + EPSILON2;
     distSixth = distSqr * distSqr * distSqr;
     invDistCube = 1.0/sqrtf(distSixth);
     s = oldbodies[j].mass * invDistCube;
     tot_force_i[X] += s * r[X];
         // analogous for Y and Z
      }
      //24 flops
      dv[X] = dt * tot_force_i[X] / oldbodies[i].mass;
      newbodies[i].pos[X] += dt * (oldbodies[i].vel[X] + dv[X]/2);
      newbodies[i].vel[X] = oldbodies[i].vel[X] +dt * dv[X];
      // analogous for Y and Z
  }
}
```

In Listing 1 the core of the All Pairs approach is shown. The main loop corresponds to the given number of timesteps; in the second loop at first the total force on each particle is computed as the result of the sum of all the gravitational attraction with all the other particles, obtained with the third loop, then each particle position and velocity is updated as result of this force.

In particular the number of floating point operations (FLOP) for each iteration (i.e. considering the second and third loops) is $(N*(N-1)*20)+(24*N)$, and the total amount of flops operation per execution is *timesteps* times this value.

3.1. Performance evaluation

We performed several tests considering a variable number of bodies and timesteps. We present here the best results, achieved with 100 timesteps and 10,000 bodies, corresponding to about 200 billion operations.

Table 2
Performance, in GFLOPS, of the N-body algorithm implementations.

	1	4	8	16	32	64	GPU
N3700	0.53	2.10					
J4205	1.29	4.67					
Avoton	0.58	2.33	4.64				
C3958	1.00	3.99	7.98	15.86			
Xeon D	2.46	4.68	9.26	17.42			
Xeon E5	2.66	9.42	16.88	32.14	64.36	102.97	
TK1	0.83	3.12					157.50
TX1	0.88	3.22					333.39
K20							1112.66

We implemented the parallel version for multi-core using the Open specifications for Multi-Processing (OpenMP) directives, while we used CUDA for the parallel version for GPU. We did not consider the use of multiple nodes for a single parallel run because of the poor performance of SoC in MPI applications. The performance using the gcc compiler with optimization flags O3 are shown in Table 2 and Fig. 1 in billion floating point operations per second (GFLOPS).

In the sequential case we can see that Xeon E5 and Xeon D provide the best performance. Results are nearly the same, as expected, because both are based on the Broadwell microarchitecture. However the greater amount of L2 and L3 caches in the Xeon E5 plus the greater number of available cores allow to get up to 103 GFLOPS, 6 times more than the parallel performance achievable with the Xeon D.

The actual best performance, however, are provided by the GPUs, as expected for an algorithm based on a regular problem like the N-body. All of them provide better performance than the Xeon E5, even the oldest TK1, and K20 is 11 times faster.

Moving to the other server-grade, low-power SoCs, we can see that C3958 presents results comparable with the Xeon D. While C3958 incorporates more processing cores than the Xeon D, the greater operating frequency plus the newer architecture of the latter result in better performance. Moreover the Xeon D features a richer instruction set (e.g. AVX and FMA3) that can be exploited to further improve the performance. The Intel Atom C3000 series succeeds the Intel Atom C2000 series, originally named Avoton, and this is reflected by the poorer performance of the Avoton SoC that, even if classified on the server-grade SoCs, presents results comparable with the mobile-grade J4205. It is worth to note, in fact, that both J4205 and C3958 are based on the Goldmont microarchitecture, with the main difference represented by the lower operating frequency and number of cores of the J4205. At last, the worst performance are provided by N3700. This is due to the fact that its architecture has been derived by the Silvermont one, thus it presents nearly the performance of the Avoton, but with 4 cores only.

For reference, considering the highest performance achievable by each architecture, we get the result in 95 seconds using N3700, in 43 sec. using both J4205 and Avoton, in 12 sec. using both C3958 and Xeon D, in less than 2 sec. using the Xeon E5, in about 1 sec. using the GPU of TK1 and less than 1 sec. using both the GPUs of TX1 and K20.

3.2. Energetic evaluation

A different scenario arises when we evaluate the power consumption for performing the computation. As stated in [14], the laboratory power measurement equipment consists of a high precision DC power supply, a high precision digital multimeter connected to a National Instruments data logging software, and a high precision AC power meter. To monitor the consumption of the SoCs, the DC current absorbed is measured by a Voltech PM300 Power Analyzer downstream of the power supply.

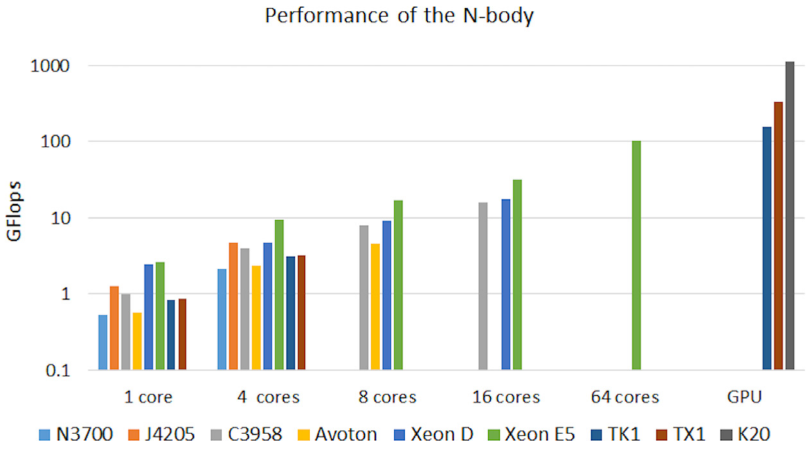


Fig. 1. Figure shows performance in GFLOPS of N-body simulation.

Table 3
Power consumption, in Joule, of the N-body simulation.

	1	4	8	16	64	GPU
N3700	3,754.83	1,089.55				
J4205	1,434.12	538.80				
Avoton	7,517.29	1,987.2	1,090.43			
C3958	6,316.13	1,674.84	899.90	515.12		
Xeon D	2,834.12	2,017.33	1,054.3	760.89		
Xeon E5	21,056.32	16 741.27	11 365.22	5872.12	659.6	
TK1	3,132.23	814.32				15.07
TX1	4,126.42	1,034.56				11.54
K20						43.56

Table 4
Summary of performance of a synthetic benchmark (HS06) figures for providing 400,000 HS06.

	Number of platforms	Acquisition cost Million €	Power (kW)	Daily energy (€)
N3700	14,277	2.27	100.0	268.8
J4205	12,117	3.47	121.3	326.0
Avoton	7,265	1.24	181.3	487.3
C3958	3,333	3.47	155.9	419.1
Xeon D	2,640	2.93	210.6	566.1
Dual Xeon E5	694	3.33	243.9	655.6

Results are shown in Table 3 and Fig. 2. Also in this case, the best results are provided by the GPUs, but TX1 requires one fourth of the energy needed by K20 for providing results in 0.6 s instead of 0.2.

C3958 provides the best result for the CPUs. In particular it uses 22% less energy of Xeon E5 for providing the results 6.5 times slower and it costs one fifth. It uses about 33% less energy in comparison to the Xeon D and in this case it requires just one second more to compute the result at about the same cost. Surprisingly, even J4205 presents better results in comparison to the Xeon D. This is justified by the feature of the Silvermont architecture, designed to be less power hungry in comparison to the Broadwell, but is however a key result to highlight the importance of considering the trade-off between time-to-solution (43 sec for J4205 vs. 12 using Xeon D) and energy-to-solution (539 Joule vs. 761 respectively). The Avoton, together with N3700, shows the worst performance, even if the latter costs about one third of the former.

It is important to note that the values actually measured do not always correspond to those declared by the manufacturers.

3.3. Economic evaluation

These results are important to evaluate the different scenarios arising if we would replace the existing equipments of a computing center with low-power clusters. In particular, we considered the characteristics of the CNAF data center[9] [29], which is equipped with about 30,000 heterogeneous cores providing a computational power of about 400 kHS06.

HS06[10] is a benchmark for measuring CPU performance in order to provide a consistent and comparable measure of existing computational equipments. It has been developed by the HEPiX Benchmarking Working Group using the industry standard SPEC CPU2006 benchmark suite.

Results are shown in Table 4 and Fig. 3. In this analysis we disregard GPUs, that clearly would represent the best choice, because they effectively support only application based on regular domains. We disregard also other hardware elements to be considered in the total cost of ownership of a computing center [30], mainly switches, cabling, and cooling systems.

We can see that only the solution based on the dual Xeon E5 requires less than 1000 platforms, while the others require from 4 to 20 times more elements. However, the cost for acquiring them is comparable or lower, about halved using Avoton. The cost saving is even more important considering an estimation of the daily energy cost in the worst case, i.e. when all the cores are busy for the whole day.

This value assumes an hourly energy price per KW of €0.112, determined by the average price supplied by Eustat in the second half of 2017.[11] As known, the energy market has strong spatial/temporal scale variations. Just to mention spatial fluctuations, we remind the 2015 price per KWh in three US states[12]: ¢7.4 (Washington), ¢15.42 (California) and ¢26.17 (Hawaii), with a US average price of ¢10.41 . In Europe for the same period, the price per KWh ranges from ¢7.8 (Bulgaria) to ¢16 (Italy), with EU-28 average of ¢11.9.

If we estimate the yearly bill, the saving due to the use of low-power SoC, in comparison to Xeon E5, can be of 33,000 € using Xeon D, of 120,000 € using J4205 and up to 141,000 € using N3700.

The question now becomes: do these raw numbers correspond to achievable performance figures? In the next Sections we discuss the performance of a different benchmark and then those of a real-life Bioinformatics application.

[9] https://www.cnaf.infn.it/en/core-computing.
[10] https://w3.hepix.org/benchmarking.html.
[11] http://ec.europa.eu/eurostat/statistics-explained/index.php/Energy.
[12] https://www.eia.gov/electricity/state/.

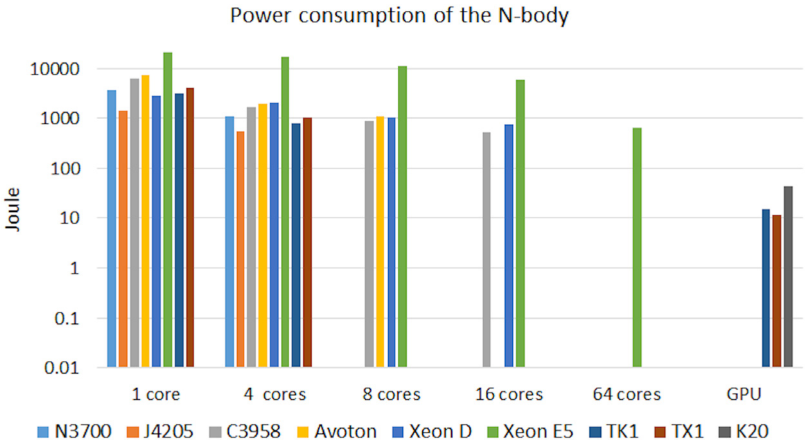


Fig. 2. Figure shows power consumption in Joule of N-body simulation.

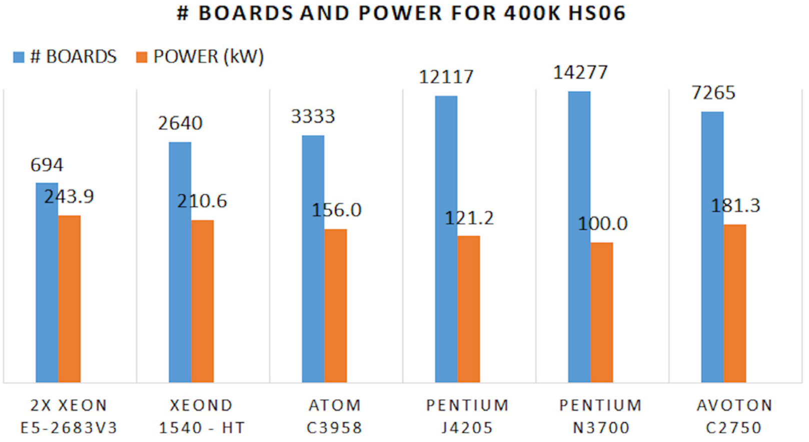


Fig. 3. Figure shows how many boards are needed to have a cluster of 400 kHS06 and their consumption.

4. The deep learning application

Deep Learning (DL) is a representation-learning method exploiting simple blocks that models the knowledge by transforming the information layer after layer using a hierarchical approach, from low-level to high-level features, making possible the learning of more abstract levels of representation [31].

In the last years, DL has attracted a lot of attention by tackling problems in different application domains and achieving results beyond expectations. For example, it has been applied in bioinformatics, game playing, imaging processing, object detection, robotics and drug discovery. One of the main reasons for the increasing use of DL algorithms is the need to implement approaches for the analysis of the large amount of heterogeneous data produced in every field, moving from statistical to model-based representations.

One of the main application of DL concerns classification problems, where these models are used to discriminate the data in a more accurate way compared to other techniques, eliminating all the variables that are not relevant for the study. In particular, DL can be used with a *supervised* or *unsupervised* approach. In the first case, the class of each data is known and the algorithm is run to determine the weights that can separate the classes in the best way. In the unsupervised learning, instead, the classification is performed observing the data and finding common and uncommon characteristics to divide the feature space. During the training, an error function is calculated and the weights are modified to reduce the error.

The number of weights to adjust is typical of each network and is also a proxy of the complexity of the problem. The most used procedure to evaluate the weights is the back-propagation of the stochastic gradient descent (SGD). Using this approach, the input examples are divided into small sets, the outputs and errors are evaluated for each set, the average gradient is calculated, and the weights modified accordingly. The procedure is repeated until the average of the objective function stops decreasing. The performance is then tested on an independent set, named the test set, to verify the ability of the trained model to classify unknown data.

Differences between models are due to the number of layers used to construct the network, types of layers considered, and the hyper-parameters used to refine the net. The most commonly used networks are the Convolutional Neural Network (CNN) and the Recurrent Neural Network (RNN), applied mainly on images and speech recognition, respectively. CNNs are composed by a sequence of convolution and pooling layers [32], while RNNs are characterized by feedback loops.

One of the main topics discussed up today is the possibility to run the training of DL models in a parallel fashion, in order to reduce the time needed to achieve the results. Many works have been published concerning these aspects: some summarized the state of the art related to multi-core and distributed setting testing the speedup in training a CNN using a single core CPU and GPU [33], whereas others highlighted parallelization as one of the main DL challenges, in order to get the resolution of which many research areas would benefit [34,35].

Some works discuss how DL can be applied to social network analysis and how the evolution of parallel techniques have increased their performance [36], while others propose approaches tested on specific networks: [37] is about intra-block parallel optimization to leverage data parallelism and blockwise mode-update filtering to stabilize learning process; [38] describes a large-scale distributed system of tens of thousands of CPU cores for training large deep neural networks; [39] proposed K-Brain, which performs deep learning and deep inference algorithms for image recognition.

4.1. Test case

Within this context we considered the use of DL applied to the bioinformatic problem of identifying the correct mRNA target for a specific miRNA. miRNAs are small non-coding RNA molecules that play an important role in a wide range of biological processes, since they interact with specific mRNAs affecting the expression of the corresponding proteins. As miRNAs dysfunction can lead to the development and progression of different kind of diseases, from cancer to cardiovascular dysfunction and neurological disorders, it is a priority to understand the complex methods behind the interactions between miRNAs and their targets.

However, the interaction mechanisms between miRNAs with their messenger RNA targets are poorly understood, because the experimental identification of miRNA-target interactions remains challenging due to their complexity and to a limited knowledge of rules governing these processes. This is the reason why many prediction algorithms have been developed to find possible miRNA-target interactions. Some of the most popular tools currently available and preferably used (TargetScan, miRanda, RNAhybrid) are based on different biological properties and can be highly inconsistent with each other. This lack of a large consensus in the predictions boosted the development of methods to reduce the false positive predictions, by integrating lists of miRNA-target genes predicted by other algorithms, relying on the hypothesis that predictions achieved by more than one tool have higher probability to be validated in the lab [40].

Many DL-based frameworks and tools have been proposed to solve this problem, and very interesting results have been achieved. In particular, deepTarget [41] is one of the most popular and relies on autoencoders and RNN to learn miRNA:mRNA interactions. Other examples are miRAW [42], which uses DL to analyze the mature miRNA transcript without making assumptions on which physical characteristics are the best suitable to impute targets, and DeepMirTar [43] that predicts human miRNA targets at the site level.

In our test case we designed a five-layer network using the Keras neural network library[13] and the TensorFlow library for numerical computation using data flow graphs.[14] All the layers are dense (fully-connected) ones: the first is characterized by 80 units from which the input data are processed, from the second to the fourth layer the units are 40, while the last layer, from which we receive the classification, has 2 units. The activation function is *relu* (rectified linear unit) for all the layers except the last, for which the *sigmoid* is used.

4.2. Performance and economic evaluations

We selected 572 experimentally validated miRNA:mRNA, considering both positive and negative interactions, to perform a supervised learning on the predictions achieved using TargetScan, miRanda and RNAhybrid. A total of 5 scores have been collected for each miRNA-target, considering the different results provided by each tool. The DL network was then trained using a 10-cross validation strategy to find the best weights to correctly separate the true and false interactions by mapping those scores on the right label.

Table 5
Summary of performance and energetic results for the miRNA target identification. T represents the neural network training part of the application, C the classification.

	Execution time		Power consumption	
	T (min)	C (s)	T (MJ)	C (J)
N3700	324	2.1	0.005	0.53
J4205	263	1.6	0.008	0.77
Avoton	271	2.8	0.026	4.44
C3958	219	1.3	0.056	5.56
Xeon D	110	2.2	0.082	27.2
Xeon E5	85	1.9	1.7	618
TK1	2025	12.02	13.7	1351
TX1	890	4.9	8.3	760
K20	98	8.5	1.5	2234

The code has been divided into two blocks, the neural network training and the classification, where the previously trained network is reloaded and used. We used Keras and Tensorflow with the default configuration, as they claim to be able to use automatically all the available CPUs/GPUs. Results are shown in Table 5 and Figs. 4 and 5.

As expected, SoC architectures in general do not represent a valid solution for the training of a DL network. While the Broadwell architectures (i.e. Xeon E5 and D) plus the K20 in fact accomplish this task in less than two hours, the other platforms require much more time.

In general, we measured a low usage of the computational capabilities, and this negatively affected in particular the results achieved using GPUs. According to several developers forum,[15] this seems a common issue, as the fact that the installation of the Keras and TensorFlow libraries is rather complicated on the Jetson boards,[16] while it is much easier on the K20 and actually straightforward on the x86 devices. We applied some common strategies to improve GPU performance, such as changing CUDA version,[17] implementing a manual tuning of the batch loading size,[18] or using a different loading procedure,[19] but without any considerable improvement. This aspect will be further investigated.

Despite this issue, the experimental results present a very interesting scenario. The performances of the Xeon D are the best considering both the execution time and the power consumption. While the training, in fact, requires about 23% more time that the Xeon E5, the power consumption is about twenty times lower. More important, this power consumption figure holds true also for the classification, but in this case the execution time is nearly the same.

Moreover, all the x86-based SoCs present the same execution time for the classification. Some devices, as the C3958, can exploit very fast storage solution based on SSD, therefore the I/O overheads are reduced in comparison to the Xeon E5 and, consequently, the execution time. This holds true also comparing the classification time of the Jetson TX1 and the K20. The TX1 board is equipped, in fact, with a 16 GB eMMC flash storage, resulting in very low latency

[13] https://keras.io.

[14] https://github.com/tensorflow/tensorflow.

[15] https://github.com/keras-team/keras/issues/249.

[16] http://cudamusing.blogspot.com/2015/11/building-tensorflow-for-jetson-tk1.html.

[17] https://github.com/tensorflow/tensorflow/issues/5995.

[18] https://stackoverflow.com/questions/42097115/keras-tensorflow-backend-slower-on-gpu-than-on-cpu-when-training-certain-netwo.

[19] https://stackoverflow.com/questions/44563418/low-gpu-usage-by-keras-tensorflow.

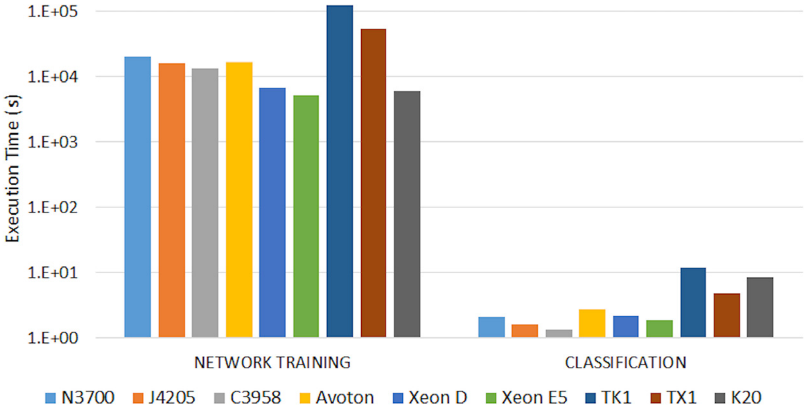


Fig. 4. The execution time for neural network training and classification. The scale is logarithmic.

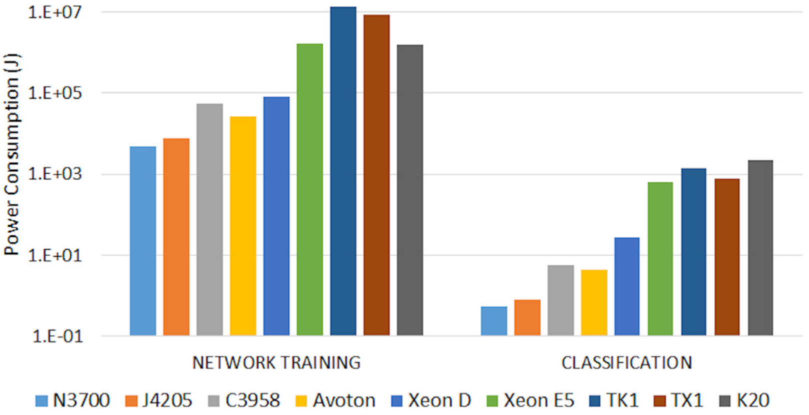


Fig. 5. The power consumption, in Joule, for neural network training and classification. The scale is logarithmic.

Table 6
Parameters of the applicative scenarios considered for the execution of the NGS application.

	Potential customers	Experiments per day	Amount of data for experiment	Note
Research labs	7	1	10–50 GB	Usually an experiment corresponds to many samples to analyze. There are also breaks periods
Hospitals	15	3	1–10 GB	The rate is constant
Farms	10	5	1–10 GB	The rate is constant
Food industry	5	10	500 MB–1 GB	The rate is constant and quite urgent

to move data in comparison to the SATA disk of the K20. Looking at the power consumptions, the N3700 is able to classify using 0.5 J in comparison to the 618 J required by the Xeon E5.

Therefore, we can conclude that the Xeon D represents a very good compromise when an application has to perform both the training and the classification, while cheaper, less powerful, but also more energy-saving devices are suitable candidates when only the classification is required. In particular, this is an interesting results for embedding DL applications in devices used on the field, as for example SoC-bases smart cameras for space monitoring [44] or IoT applications [45,46] and, more in general, when DL meets the edge computing[20] [47], because they provide greater and more general-purpose compute capabilities than the commonly used FPGAs at a reasonable price.

5. The NGS application

The increasing availability of molecular biology data resulting from improvements in experimental techniques represents an unprecedented opportunity for Bioinformatics and Computational Biology, but also a major challenge [48]. Due to the growing number of experiments involving genomic research, originating from the spreading of these techniques from research centers to hospitals and from farms to food industries, the amount and complexity of biological data is increasing very fast.

In particular, the high demand for low-cost sequencing has driven the development of high-throughput technologies that parallelize the sequencing process, producing thousands or millions of sequences concurrently [49]. Such huge and heterogeneous amount of digital information is fundamental for uncovering disease associated hidden patterns in data [50,51], allowing the creation of predictive models for real-life biomedical applications [52, 53].

The first step to accomplish for each analysis is the alignment of the reads achieved through sequencing to the reference genome. This is usually done on a single server using parallel applications

[20] https://towardsdatascience.com/deep-learning-on-the-edge-9181693f466c.

such as Bowtie [54], BWA [55] or STAR [56]. All these tools rely on the Burrows–Wheeler transform (also called block-sorting compression), which is useful to compress the reference database in order to make the search very fast.

These programs are very fast, basically CPU-bound, and scale linearly with the dimension of the input. Although each input sequence takes a different computational effort to align against the reference dataset, depending on the number of hits found and the general similarity with the reference genome (considering gaps and mismatches), these algorithms are usually implemented using smart multi-thread approaches that are able to balance the load among threads. On the other hand, network-based implementation of these aligners, generally relying on splitting the database, despite being tested by different groups, had little success, given that it is usually better to split the input sequences in chunks and compute them separately [57].

5.1. Test case

The experimental evaluations have been conducted considering an NGS application for performing clinical, zoo-technical, research and industrial analysis. Considering the health-care system, sequencing is moving from research, that needs few deep analyses in a specific amount of time, to hospitals, where patients are analyzed on a daily basis, although in this case the single experiments have a lower throughput of data.

On the other hand, zoo-technical analyses may aim at verifying the pedigree of animals or to prevent diseases, such as mastitis, while in food industry they may have the goal to certify food safety [58], both to be compliant with allergenic-free nutritions and for religious related diet constrains. Typically, these experiments are in large number, but with a lower throughput of reads, since they aim at the identification of peculiar and well-known patterns.

In detail, we considered that the input dataset for a clinical and zoo-technical analysis has a size of about 10 GB, while an in depth analysis – as those performed for research purposes – requires to process about 50 GB. Food industry analysis' datasets are assumed to be of 1 GB size. Table 6 summarizes the four applicative scenarios considered.

5.2. Performance evaluation

Although each sequence takes a different computational effort to align against the reference genome, the running times are normally distributed [59], resulting in an application that has almost linear scalability. In other words, this means that doubling the dimension of the dataset results in doubling the time required for the computation and so on and so forth.

In the tests we have considered only x86-based hardware because a full-featured GPU version of the software is not available. Table 7 shows the execution times, in minutes, required for performing the analysis of a 10 GB dataset. Analogously, an analysis for a dataset of 1 GB, tenfold smaller than the original one, has been performed on the considered platforms, achieving the same scalability figures, as expected. For sake of completeness, we repeated the analysis considering a dataset of 50 GB, five time bigger than the original one, achieving the same speed-up trends.

In contrast to the N-body algorithm, the NGS application is heavily data intensive, both considering RAM and storage. Therefore, we can see some interesting performance figures in comparison to Table 2. One example is the better scalability of the C3958, equipped with 16 real cores, in comparison to the use of Hyper-Threading in the Xeon D. However, the latter still presents better performance. The same situation holds true for the Xeon E5 in comparison to both of them, but it is able to provide results three times faster. Also the Avoton presents better results in comparison to J4205 thanks to its 8 cores. N3700 remains the slowest processor but the difference in comparison to J4205 is reduced to only 20%.

Table 7
Execution times, in minutes, for performing a clinical analysis on a 10 GB dataset.

Threads	Xeon E5	Xeon D	C3958	Avoton	J4205	N3700
1	438.4	406.0	673.9	877.0	752.2	897.6
4	121.9	112.0	156.8	223.3	204.4	247.1
8	65.7	59.2	88.1	119.1		
16	35.5	47.8	57.0			
32	20.6					
64	17.2					

Table 8
Summary of performance and economic figures of the considered platforms.

	Xeon E5	Xeon D	C3958	Avoton	J4205	N3700
Clinical (min)	17.2	47.8	57.0	119.1	204.4	247.1
Research (min)	86.0	239.1	285.4	595.3	1021.9	1235.5
Industry (min)	1.8	4.8	6.7	5.4	20.2	24.9
Platforms	2	5	6	12	22	24
Acquisition (€)	10,000	5500	6300	9600	6600	7200
Power (W)	702	395	276	300	220	168
Annual energy (€)	689	387	271	295	216	165

5.3. Economic evaluation

We collected all the data used for an economic evaluation in Table 8. In particular, we considered the scenario where a SME provides such analysis service on a regional basis. In Lombardia, where ITB has its own headquarter, it is possible to forecast requests from fifteen hospitals and ten farms, for a total amount of 95 clinical/zoo-technical analysis per day, while the demand of research analysis can be evaluated as an average of seven per day and of fifty daily arrivals for Industry (see Table 6).

At first, we evaluated the number of platforms the SME have to acquire to assure all the analysis are performed on the due time. On the basis of the best performance achieved, a single Xeon E5 platform can accomplish up to 85 clinical/zoo-technical or 17 research analysis every 24 h, therefore there is the need to buy two platforms. The worst case is represented by the Intel N3700, where a single platform can process only 6 clinical/zoo-technical or 1 research analyses per day. The time required for the analysis of food industries is very limited except for J4205 and again N3700, that can accomplish respectively only 72 and 48 of these tasks. It is worth to note that, in every case, each platform behaves independently for each analysis and it is not considered as part of a cluster.

We can therefore conclude that the best choice is represented by the acquisition of 5 platforms equipped with Xeon D processors, because such infrastructure is the cheapest one and it is able to provide clinical results in less than one hour.

Finally, in Table 8 we report the power needed by the platforms along with the yearly cost, as discussed in Section 3.3 . We observe that notwithstanding consumption noticeably varies amongst the architectures, due to the different computation times, the energy cost is rather negligible because of the small number of servers needed to operate the service. Nevertheless, starting from these energetic considerations, it is possible to conclude that low-power architectures still represent a suitable solution for SMEs willing to provide services like the one discussed because of the trade-off between performance and costs. As said before, in the considered scenario the acquisition of 5 Xeon D platforms is the cheapest solution that allows to support the considered workload, and it is able to satisfy the expectation of customers in terms of delivering results on time.

We would like to stress that the analysis presented here has the purpose to answer the question whether low-power SoC architectures are feasible also in a commercial environments where SMEs offer their services, as stated in the Introduction. In general, the dual Xeon E5 system represents, in fact, a platform suitable for other types of workloads that need HPC-oriented capabilities. Moreover, higher cost servers are typically equipped with enterprise-level components as opposed to low-end system, which require service in the long run. However, for the purpose of our analysis, we have shown that low-power architectures can represent an equivalent or even better choice in comparison to traditional enterprise system and, in general, this is an aspect that can be taken in consideration in brokering cloud services [60,61].

6. Conclusions and future works

This paper presents an analysis of performance and economic aspects related to the adoption of low-power System-on-Chips in computational infrastructures. Starting from two benchmarks, i.e. the widely used N-body algorithm and a deep learning based application, we discussed the achievable performance of the state-of-the-art low-power SoC architectures in comparison to the traditional server-grade CPUs and GPUs equipping present computing infrastructures. Then we moved the focus on assessing if the interesting raw energetic and computational figures holds true also in a real-life applicative scenario. SoC represent, in fact, an interesting alternative to run a number of scientific and commercial applications. In the paper we analyzed a scenario where a SME is willing to offer a service based on a real life application taken from the field of molecular biology using a SoC-based infrastructure.

It is to note that comparing high-end commercial/HPC servers with motherboards based on low-power SoC taken from the mobile and embedded world can be considered unfair, but nevertheless the results presented assess that also for time-consuming applications, like NGS data analysis, the use of low-power architectures represents a feasible choice in terms of trade-off among time-to-solution, energy-to-solution and economical aspects.

Future developments of this work are twofold. From one side we will evaluate other use cases coming from applicative domains with different requirements. In particular, we are considering the use of such low-power architectures in combination with the Edge computing paradigm for IoT applications and in pharmaceutical industries for the development of new drugs through in-silico simulations of proteins–chemicals interactions. Furthermore, we plan to extend the analysis for the adoption of SoC-based clusters in computing infrastructures by including a wider range of the elements in the total cost of ownership, such as network interconnections, the cooling and the cost for developing applications able to exploit such a greater number of low-power computational nodes. This aspect partially overlaps with the research efforts of the parallel computing community paving the way for the exascale [62].

Acknowledgments

This work has been supported by the Italian Ministry of Education and Research (MIUR) through the Flagship (PB05) InterOmics, the EC-FP7 innovation project MIMOMICS, Italy (no. 305280), and the EC-FP7 strep project REPARA, Italy (no. 609666), and it was partly funded by the Scientific Commission 5 of the INFN, Italy through the COSA project.

References

[1] A. Winston, G. Favaloro, T. Healy, Energy strategy for the C-suite, Harv. Bus. Rev. (2017) 138–146.

[2] A. Shehabi, S.J. Smith, N. Horner, I. Azevedo, R. Brown, J. Koomey, E. Masanet, D. Sartor, M. Herrlin, W. Lintner, United States Data Center Energy Usage Report, Lawrence Berkeley National Laboratory, Berkeley, California, 2016, LBNL-1005775.

[3] Best Practices Guide for Energy-Efficient Data Center Design. National Renewable Energy Laboratory (NREL), a national laboratory of the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, 2011.

[4] D. Hackenberg, T. Ilsche, R. Schone, D. Molka, M. Schmidt, W. Nagel, Power measurement techniques on standard compute nodes: a quantitative comparison, in: 2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), 2013, pp. 194–204.

[5] D. Horak, L. Riha, R. Sojka, J. Kruzik, M. Beseda, Energy consumption optimization of the Total-FETI solver and BLAS routines by changing the CPU frequency, in: Proceedings of the 14th International Conference on High Performance Computing and Simulation, HPCS 2016, 2016, pp. 1031–1032.

[6] S. Catalán, J.R. Herrero, E.S. Quintana-Orti, R. Rodriguez-Sanchez, Energy balance between voltage-frequency scaling and resilience for linear algebra routines on low-power multicore architectures, Parallel Comput. (2017).

[7] M. Daneshtalab, N. Bagherzadeh, H. Sarbazi-Azad, Special issue on on-chip parallel and network-based systems, J. Comput. 97 (6) (2015) 539–541, (Computing-Springer).

[8] L. Morganti, D. Cesini, A. Ferraro, Evaluating systems on chip through HPC bioinformatics and astrophysics applications, in: Proceedings of the 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP), 2016, pp. 541–544.

[9] N. Rajovic, P. Carpenter, I. Gelado, N. Puzovic, A. Ramirez, M. Valero, Supercomputing with commodity CPUs: Are mobile socs ready for HPC? in: Proceedings of SC13: International Conference for High Performance Computing, Networking, Storage and Analysis, 2013.

[10] M. Jain, H.E. Olsen, B. Paten, et al., The Oxford Nanopore MinION: delivery of nanopore sequencing to the genomics community, Genome Biol. 17 (1) (2016) 239.

[11] D. D'Agostino, L. Morganti, E. Corni, D. Cesini, I. Merelli, Combining edge and cloud computing for low-power, cost-effective metagenomics analysis, Future Gener. Comput. Syst. 90 (2019) 79–85.

[12] E. Calore, S.F. Schifano, R. Tripiccione, Energy-performance tradeoffs for HPC applications on low power processors, in: S. Hunold, et al. (Eds.), Euro-Par 2015: Parallel Processing Workshops. Euro-Par 2015, in: Lecture Notes in Computer Science, vol. 9523, 2015, pp. 737–748.

[13] D. Cesini, E. Corni, A. Falabella, A. Ferraro, L. Morganti, E. Calore, S.F. Schifano, M. Michelotto, R. Alfieri, R. De Pietri, T. Boccali, A. Biagioni, F. Lo Cicero, A. Lonardo, M. Martinelli, P.S. Paolucci, E. Pastorelli, P. Vicini, Power efficient computing: the experience of the COSA project, Sci. Program. 2017 (2017) 7206595.

[14] D. D'Agostino, D. Cesini, E. Corni, A. Ferraro, L. Morganti, A. Quarati, I. Merelli, Performance and economic evaluations in adopting low power architectures: A real case analysis, in: International Conference on the Economics of Grids, Clouds, Systems, and Services, Springer, Cham, 2017, pp. 177–189.

[15] A. Clematis, A. Corana, D. D'Agostino, A. Galizia, A. Quarati, Job-resource matchmaking on Grid through two-level benchmarking, Future Gener. Comput. Syst. 26 (8) (2010) 1165–1179.

[16] P. Ruiu, A. Bianco, C. Fiandrino, P. Giaccone, D. Kliazovich, Power comparison of cloud data center architectures, in: Communications (ICC), 2016 IEEE International Conference on, IEEE, 2016, pp. 1–6.

[17] L.D. Radu, Green cloud computing: A literature survey, Symmetry 9 (12) (2017) 295.

[18] I. Goiri, W. Katsak, K. Le, T.D. Nguyen, R. Bianchini, Parasol and greenswitch: Managing datacenters powered by renewable energy, ACM SIGARCH Comput. Archit. News 41 (1) (2013) 51–64, ACM.

[19] Al-Dulaimy, et al., Power management in virtualized data centers: state of the art, J. Cloud Comput. Adv. Syst. Appl. 5 (2016) 6.

[20] D. Horak, L. Riha, R. Sojka, J. Kruzik, M. Beseda, Energy consumption optimization of the Total-FETI solver and BLAS routines by changing the CPU frequency, in: Proceedings of the 14th International Conference on High Performance Computing and Simulation, HPCS 2016, 2016, pp. 1031–1032.

[21] A. Geist, R. Lucas, Major computer science challenges at exascale, Int. J. High Perform. Comput. Appl. 23 (4) (2009) 427–436.

[22] The Mont-Blanc prototype: An alternative approach for HPC systems, in: SC16: International Conference for High Performance Computing, Networking, Storage and Analysis, 2016, pp. 444–455.

[23] G. Oyarzun, R. Borrell, A. Gorobets, F. Mantovani, A. Oliva, Efficient CFD code implementation for the ARM-based Mont-Blanc architecture, Future Gener. Comput. Syst. 79 (2018) 786–796.

[24] M. Marazakis, et al., EUROSERVER: share-anything scale-out microserver design, in: 2016 Design, Automation & Test in Europe Conference & Exhibition, DATE 2016, Dresden, Germany, March (2016) 14-18, 2016, pp. 678–683.

[25] S. Catalan, et al., Energy balance between voltage-frequency scaling and resilience for linear algebra routines on low-power multicore architectures, Parallel Comput. (2017) http://dx.doi.org/10.1016/j.parco.2017.05.004.

[26] S. Furber, S. Temple, Neural systems engineering, J. R. Soc. Interface ((4) 13) (2007) 193–206, http://dx.doi.org/10.1098/rsif.2006.0177.

[27] M. Katevenis, et al., The ExaNeSt project: Interconnects, storage and packaging for exascale systems, in: Digital System Design (DSD), 2016 Euromicro Conference on, 2016, pp. 60–67, http://dx.doi.org/10.1109/DSD.2016.106.

[28] E. Danovaro, A. Clematis, A. Galizia, G. Ripepi, A. Quarati, D. D'Agostino, Heterogeneous architectures for computational intensive applications: A cost-effectiveness analysis, J. Comput. Appl. Math. 270 (2014) 63–77.

[29] S. Bovina, D. Michelotto, The evolution of monitoring system: the INFN- CNAF case study, J. Phys.: Conf. Ser. 898 (2017) 092029.

[30] S. Wienke, H. Iliev, D. an Mey, M.S. Muller, Modeling the productivity of HPC systems on a computing center scale, in: J. Kunkel, T. Ludwig (Eds.), in: ISC 2015, Lecture Notes in Computer Science, vol. 9137, Springer International Publishing, 2015, pp. 358–375.

[31] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436.

[32] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[33] V. Hegde, S. Usmani, Parallel and Distributed Deep Learning. Tech. report, Stanford University, 2016, 2016.

[34] M.M. Najafabadi, F. Villanustre, T.M. Khoshgoftaar, N. Seliya, R. Wald, E. Muharemagic, Deep learning applications and challenges in big data analytics, J. Big Data 2 (1) (2015) 1.

[35] X.W. Chen, X. Lin, Big data deep learning: challenges and perspectives, IEEE Access 2 (2014) 514–525.

[36] G. Cybenko, Parallel computing for machine learning in social network analysis, in: Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2017 IEEE International, IEEE, 2017, pp. 1464–1471.

[37] K. Chen, Q. Huo, Scalable training of deep learning machines by incremental block training with intra-block parallel optimization and blockwise model-update filtering, in: Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on, IEEE, 2016, pp. 5880–5884.

[38] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A.Y. …Ng, Large scale distributed deep networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1223–1231.

[39] H.J. Yoo, S. Park, K. Bong, D. Shin, J. Lee, S. Choi, A 1.93 tops/w scalable deep learning/inference processor with tetra-parallel mimd architecture for big data applications, in: IEEE International Solid-State Circuits Conference, IEEE, 2015, pp. 80–81.

[40] P. Alexiou, M. Maragkakis, G.L. Papadopoulos, M. Reczko, A.G. Hatzigeorgiou, Lost in translation: an assessment and perspective for computational microRNA target identification, Bioinformatics 25 (23) (2009) 3049–3055.

[41] B. Lee, J. Baek, S. Park, S. Yoon, DeepTarget: end-to-end learning framework for microRNA target prediction using deep recurrent neural networks, in: Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, ACM, 2016, pp. 434–442.

[42] A. Pla, X. Zhong, S. Rayner, MiRAW: A deep learning-based approach to predict microRNA targets by analyzing whole microRNA transcripts, PLoS Comput. Biol. 14 (7) (2018) e1006185.

[43] M. Wen, P. Cong, Z. Zhang, H. Lu, T. Li, DeepMirTar: a deep-learning approach for predicting human miRNA targets, Bioinformatics 1 (2018) 7.

[44] I. Haider, B. Rinner, Private space monitoring with SoC-based smart cameras, in: Mobile Ad Hoc and Sensor Systems (MASS), 2017 IEEE 14th International Conference on, IEEE, 2017, pp. 19–27.

[45] I. Haider, B. Rinner, Trustworthy and Privacy-Aware Sensing for Internet of Things. arXiv preprint arXiv:1808.08549, 2018.

[46] H. Li, K. Ota, M. Dong, Learning IoT in edge: deep learning for the internet of things with edge computing, IEEE Network 32 (1) (2018) 96–101.

[47] Y. Huang, X. Ma, X. Fan, J. Liu, W. Gong, When deep learning meets edge computing, in: 2017 IEEE 25th International Conference on Network Protocols (ICNP), IEEE, 2017, pp. 1–2.

[48] J.C. Fuller, P. Khoueiry, H. Dinkel, et al., Biggest challenges in bioinformatics, EMBO Rep. 14 (4) (2013) 302–304.

[49] G.M. Church, Genomes for all, Sci. Am. 294 (1) (2006) 46–54.

[50] I. Merelli, A. Calabria, P. Cozzi, et al., SNPranker 2.0: a gene-centric data mining tool for diseases associated SNP prioritization in GWAS, BMC Bioinformatics 14 (1) (2013) S9.

[51] I. Merelli, P. Cozzi, D. D'Agostino, A. Clematis, L. Milanesi, Image-based surface matching algorithm oriented to structural biology, IEEE/ACM Trans. Comput. Biol. Bioinform. 8 (4) (2011) 1004–1016.

[52] F. Chiappori, I. Merelli, L. Milanesi, A. Marabotti, Static and dynamic interactions between GALK enzyme and known inhibitors: Guidelines to design new drugs for galactosemic patients, European J. Med. Chem. 63 (2013) 423–434.

[53] F. Chiappori, P. D'Ursi, I. Merelli, L. Milanesi, E. Rovida, In silico saturation mutagenesis and docking screening for the analysis of protein-ligand interaction: the Endothelial Protein C Receptor case study, BMC Bioinformatics 10 (12) (2009) S3.

[54] B. Langmead, C. Trapnell, M. Pop, S.L. Salzberg, Ultrafast and memory-efficient alignment of short DNA sequences to the human genome, Genome Biol. 10 (3) (2009) R25.

[55] H. Li, R. Durbin, Fast and accurate short read alignment with Burrows–Wheeler transform, Bioinformatics 25 (14) (2009) 1754–1760.

[56] A. Dobin, C.A. Davis, F. Schlesinger, J. Drenkow, C. Zaleski, S. Jha, P. Batut, M. Chaisson, T.R. Gingeras, STAR: ultrafast universal RNA-seq aligner, Bioinformatics 29 (1) (2013) 15–21.

[57] H. Lin, X. Ma, W. Feng, N.F. Samatova, Coordinating computation and I/O in massively parallel sequence search, IEEE Trans. Parallel Distrib. Syst. 22 (4) (2011) 529–543.

[58] Applications of Whole Genome Sequencing in food safety management, www.fao.org/3/a-i5619e.pdf.

[59] C. Misale, G. Ferrero, M. Torquati, M. Aldinucci, Sequence alignment tools: one parallel pattern to rule them all? Biomed. Res. Int. (2014).

[60] A. Quarati, A. Clematis, D. D'Agostino, Delivering cloud services with qos requirements: Business opportunities, architectural solutions and energy-saving aspects, Future Gener. Comput. Syst. 55 (2016) 403–427.

[61] D. D'Agostino, A. Clematis, A. Quarati, D. Cesini, F. Chiappori, L. Milanesi, I. Merelli, Cloud infrastructures for in silico drug discovery: economic and practical aspects, Biomed. Res. Int. (2013).

[62] A. Rigo, C. Pinto, K. Pouget, D. Raho, D. Dutoit, P.Y. Martinez, V. .Bartsch, Paving the way towards a highly energy-efficient and highly integrated compute node for the Exascale revolution: the ExaNoDe approach, in: Digital System Design (DSD), 2017 Euromicro Conference on, IEEE, 2017, pp. 486–493.

Daniele D'Agostino, Ph.D., is a researcher at the Institute of Applied Mathematics and Information Technologies of National Research Council. His research activities concern the design of science gateways in different research fields, the resource allocation in Grid/Cloud environments and the development of parallel software. He co-organized the 22th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, several special issues on ISI journals and co-authored more than 90 scientific papers, published in journals, book chapters and conference proceedings.

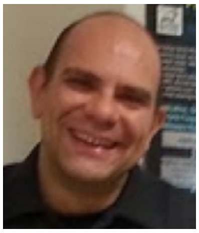
Alfonso Quarati has been a Researcher at the Institute for Applied Mathematics and Information Technology (IMATI) of the Italian National Research Council (CNR) in Genoa since 1997. His current research interests cover models and tools for grid and cloud computing, QoS assessment and performance evaluation methodologies, green computing, mashup and web-services design, collaborative working technologies. He has co-authored more than 70 scientific papers, published in journals, book chapters, technical reports and conference proceedings.

Andrea Clematis is a Research Director at the Institute of Applied Mathematics and Information Technologies of Italian CNR in Genova (IMATI-CNR). His research interests are in the area of high performance, parallel and distributed computing, with emphasis on programming environments and applications of parallel computing. He co-authored more than 120 technical papers, published in journals, book chapters and conference proceedings. Dr. Clematis acted as project leader for IMATI in about 20 cooperative research projects founded by different institutions including European Commission, Nato, and British Council. He is the Chair of the Euromicro Technical Committee on Parallel and Distributed Processing, and the subject area editor for Parallel and Distributed Processing of the Journal of Systems Architecture (Elsevier).

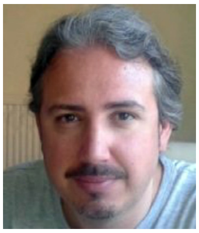
Lucia Morganti, Ph.D. in Astrophysics from the Ludwig Maximilians University of Munich, works at CNAF, the INFN National Center dedicated to research and development on IT technologies, located in Bologna, Italy. She is a member of both the Storage and the User support unit of the data center, and she is involved in the H2020 ExaNeSt project for exascale-level supercomputers, and in the INFN-COSA (Computing On SoC Architecture) project for power-efficient scientific computing.

Elena Corni, master degree in Medical Physics at University of Bologna, works at INFN-CNAF in Bologna, Italy. She is a member of the User Support unit of the data center, and she collaborates in the INFN-COSA (Computing On SoC Architecture) project.

Valentina Giansanti is working as a research fellow at the Institute for Biomedical Technologies (ITB) of the Italian National Research Council (CNR). She earned a master degree in Biomedical Engineering at the University of Rome La Sapienza in 2017. She was a visiting research at Cambridge University in 2018 and she focused her work on the application of deep learning to Genomics and Proteomics data.



Daniele Cesini is working as a Researcher in Technology at the Italian Institute for Nuclear Physics (INFN). He is currently a member of the Data Handling group at INFN-CNAF. He is the coordinator of the User Support Team of the INFN Tier1. Since 2004, he acquired experience working within national and international initiatives dealing with distributed and parallel computing. He focused his research in the field of efficient tasks-scheduling in distributed environments for mixed High Performance/High Throughput Computing workflows. He is expert in the application porting to different computing platforms: distributed architectures, low power processors and HPC hybrid systems.



Ivan Merelli is staff scientist at the Institute for Biomedical Technologies (ITB) of the Italian National Research Council (CNR). He earned a Master's Degree in Biomedical Engineering from the Politecnico di Milano in 2004 and a Ph.D. in Computer Science from the University of Milano-Bicocca in 2009. He was visiting scientist at Harvard University in 2014 and at Cambridge University in 2015. He co-authored more than fifty papers published in international peer-reviewed journals and more than fifty contributions in international Conference Proceedings. His research activities concern statistical analysis, software development and data integration in the field of *Genomics* and *Proteomics*, in particular for the management of *Big Data* by using *High-Performance Computing* facilities.