Real-time cortical simulations: energy and interconnect scaling on distributed systems

Francesco Simula*, Elena Pastorelli†, Pier Stanislao Paolucci*, Michele Martinelli*, Alessandro Lonardo*,
Andrea Biagioni*, Cristiano Capone*, Fabrizio Capuani*, Paolo Cretaro*, Giulia De Bonis*,
Francesca Lo Cicero*, Luca Pontisso*, Piero Vicini* and Roberto Ammendola‡
*INFN Sezione di Roma
Rome, Italy
{first-name.family-name}@roma1.infn.it
†INFN Sezione di Roma and PhD Program in Behavioural Neuroscience, "Sapienza" University of Rome
Rome, Italy
elena.pastorelli@roma1.infn.it
‡INFN Sezione di Tor Vergata and Electronic Engineering Dept., University of Roma "Tor Vergata"
Rome, Italy
roberto.ammendola@roma2.infn.it

***Abstract*—We profile the impact of computation and inter-processor communication on the energy consumption and on the scaling of cortical simulations approaching the real-time regime on distributed computing platforms. Also, the speed and energy consumption of processor architectures typical of standard HPC and embedded platforms are compared. We demonstrate the importance of the design of low-latency interconnect for speed and energy consumption. The cost of cortical simulations is quantified using the Joule per synaptic event metric on both architectures. Reaching efficient real-time on large scale cortical simulations is of increasing relevance for both future bio-inspired artificial intelligence applications and for understanding the cognitive functions of the brain, a scientific quest that will require to embed large scale simulations into highly complex virtual or real worlds. This work stands at the crossroads between the WaveScalES experiment in the Human Brain Project (HBP), which includes the objective of large scale thalamo-cortical simulations of brain states and their transitions, and the ExaNeSt and EuroExa projects, that investigate the design of an ARM-based, low-power High Performance Computing (HPC) architecture with a dedicated interconnect scalable to million of cores; simulation of deep sleep Slow Wave Activity (SWA) and Asynchronous aWake (AW) regimes expressed by thalamo-cortical models are among their benchmarks.**

***Index Terms*—neural network; real-time; energy-to-solution; interconnect; scaling; distributed computing;**

I. Introduction

In modern HPC and embedded systems, the most constraining limits to scaling are those related to power draw and dissipation. In HPC the electricity bill is the main contributor to the total cost of running an application so that energy-efficiency is becoming a fundamental requirement for large scale platforms, and power is a critical design figure for any embedded system. In this context, the feasibility of a computing system must not only pass through performance assessment of processors but also their performance-per-watt ratio. Several scientific communities are exploring non-traditional many-core processors architectures looking for a better tradeoff between time-to-solution and energy-to-solution. Some architectures of this kind are the Graphics Processing Unit (GPU) or those like the MPSoC that come from the embedded world, where ARM-based System-on-Chip designs dominate the market of low-power and battery-powered devices such as tablets and smartphones.

A number of research projects are active in trying to design an actual HPC platform along this direction. The Mont-Blanc project [1], [2], coordinated by the Barcelona Supercomputing Center, has deployed two generations of HPC clusters based on ARM processors, developing also the corresponding ecosystem of HPC tools targeted to this architecture. Another example is the EU-FP7 EuroServer [3] project, coordinated by CEA, which aims to design and prototype technology, architecture, and systems software for the next generation of datacenter "microservers", exploiting 64-bit ARM cores.

Unraveling how the brain works is a formidable scientific and HPC undertaking. The human brain includes about 10^{15} synapses and 10^{11} neurons activated at a mean rate of several Hz; as a digital simulation, it is a significant coding challenge and has very exacting requirements for an adequate computing architecture, even at the highest abstraction level.

Fast simulation of spiking neural network models plays a dual role: (i) it contributes to the solution of a scientific grand challenge — *i.e.* the comprehension of brain activity — and, (ii) by including it into embedded systems, it can enhance applications such as autonomous navigation, surveillance and robotics, requiring real-time performances. Moreover, real-time simulation of neural networks will be essential for understanding the mechanisms underlying the cognitive functions of the brain. Indeed, brain simulations should be embedded in complex environments, *e.g.* robotic platforms interacting with the world in real-time, which makes requirements on power consumption so much tighter. Therefore, cortical simulations assume a driving role in shaping the architecture of either specialized and general-purpose multi-core/many-core systems to come, standing at the crossroads between embedded

2377-5750/19/$31.00 ©2019 IEEE
DOI 10.1109/PDP.2019.00049

and HPC. See, for example [4], describing the TrueNorth low-power specialized hardware architecture dedicated to embedded applications, and [5] discussing the power consumption of the SpiNNaker hardware architecture, based on embedded multi-cores, dedicated to brain simulation. Worthy of mention are also [6], [7] as examples of approaches based on standard HPC platforms and general-purpose simulators.

The WaveScalES experiment in the Human Brain Project (HBP) has the goal of matching experimental measures with simulations of Slow Wave Activity (SWA) during deep sleep and anaesthesia, the transition to other brain states, and the interplay between cortical waves and memories with a focus in developing dedicated, parallel/distributed technologies able to overcome some of the limits faced by current attempts at brain simulation. On a different line of research, the ExaNeSt [8] and EuroExa projects investigates the design of an ARM-based, low-power High Performance Computing (HPC) architectures with dedicated interconnects scalable to million of cores; thalamo-cortical simulations are among their motivating benchmarks. At the joint between these projects stands the Distributed and Plastic Spiking Neural Network (DPSNN) simulation engine, developed by the APE Parallel/Distributed Computing Laboratory at INFN; its C/C++ code is written according to the MPI multi-process paradigm and is designed to be easily portable to exotic architectures and to stress either the available networking or computing resources. In this paper, the simulator is used to compare the scaling in time and energy consumption of an Intel-based HPC cluster — equipped with high-performance InfiniBand connectivity in addition to ordinary Ethernet — with that of different ARM-based platforms, taken as representatives of the class of new, low-power HPC systems like those pursued by ExaNeSt and EuroExa.

In previous works, we demonstrated the scalability of our simulation engine up to 1K processes [9] when applied to realistic long range synaptic connectivity [10] and described its internal architecture [11]. Within HBP the simulator has been applied to the study of Slow Waves Activity [12], [13], [14] in large scale cortical fields (up to 14 billion synapses), a different HPC challenge in which real-time is not required (Figure 1).

Originally, this simulation engine was developed as a mini-application benchmark in the framework of the EURETILE FP7 project [15].

The code organization and its compactness endow our application with a high degree of tunability and with it the chance of testing different areas of the executing platforms; by varying the number of neurons per core in the simulated neural net, the analysis can be moved from the performances of the platform interconnect — with relatively few neurons, each one projecting thousands of synapses — to the computing and memory resources — with more neurons per core. Full biological realism of a cortical tissue would require a number of synapses per neuron in the range between 5000 and 10000. Plus, the representation of large scale cortical systems needs the projection of long range intra-areal sparse connectivity described either by distance and layer dependent probability rules or by explicit lists of connections. Inter-areal connectivity is instead derived from the description of the sparse long-range connectome. Both kind of synaptic adjacency matrices depend on the spatial location of source and target neurons. If a large scale neural network is distributed on a grid of processes using a spatial mapping (i.e. a set of neighbouring neurons and incoming synapses is assigned to each process), the transport of spiking messages carried by the sparse synaptic adjacency matrix does not typically require an all-to-all interconnection between processes. Indeed, we demonstrated the advantages of such a reduction of the adjacency matrix between processes for the scaling of simulations of large networks with biologically plausible intra-areal long-range connections in [9]. However, the execution time for such large scale systems (Figure 1) is still one or two order of magnitude slower than the real time domain we focus on in this paper. If smaller number of neurons are considered, as necessary to reduce the network size to compatibility with real time execution, the sparsity of the synaptic adjacency matrix would also be reduced and the simulation would typically require all-to-all interprocess communications. As we will see, the communication of spikes among neurons is dominated by latency for network sizes in the explored range on contemporary HPC and embedded platforms. Therefore, we adopted in this paper a simple synaptic adjacency matrix: an homogeneous connection probability that simplifies the analysis of the scaling behaviour. Finally, we reduced the number of synapses per neuron to 1125, this way further stressing intercommunication latency (moderate size of payloads) and enabling the simulation of networks with a few more neurons, with a potentially higher representational power, but still needing the support of all-to-all inter-process communication.

This paper addresses the measure of power consumption and energy-to-solution for real-time cortical simulation and profiles the relative scaling of computation, communication

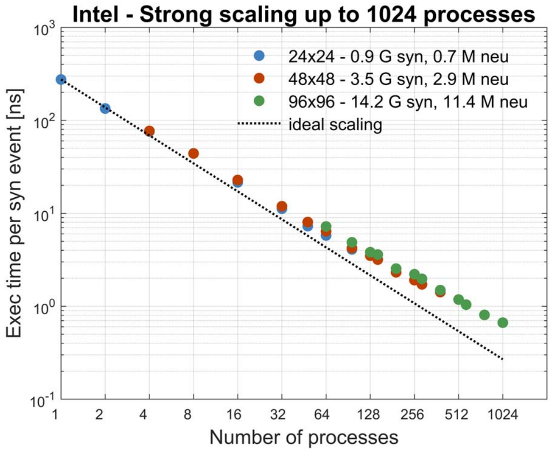

Fig. 1. Strong scaling up to 1024 processes of large neural networks on an IB-equipped Intel-based cluster.

and synchronization. Specifically, we perform a number of neural simulations to compare the performances of ARM- and Intel-based multi-core platforms, with further focus on the possible impact of the usage of off-the-shelf vs. custom networking components.

II. Mini-application benchmarking tool

Evaluation of HPC hardware is a key element especially in the first stages of a project — *i.e.* definition of specification and design — and during the development and implementation. Key components impacting performance should be identified in the early stages of the development, but full applications are too complex to be run on simulators and hardware prototypes. In usual practice, hardware is tested with very simple kernels and benchmarking tools which often reveal their inadequacy as soon as they are compared with real applications running on the final platform, showing a huge performance gap.

In the last years, a new category of compact, self-contained proxies for real applications called *mini-apps* have appeared. Although a full application is usually composed by a huge amount of code, the overall behaviour is driven by a relatively small subset of it. Mini-apps are composed by these core operations providing a tool to study different subjects: (i) analysis of the computing device — *i.e.* the node of the system. (ii) evaluation of scaling capabilities, configuring the mini-apps to run on different numbers of nodes, and (iii) study of the memory usage and the effective throughput towards the memory.

This effort is led by the Mantevo project [16], that provides application performance proxies since 2009. Furthermore, the main research computing centers provide sets of mini-applications, adopted when procuring the systems, as in the case of the NERSC-8/Trinity Benchmarks [17], used to assess the performance of the Cray XC30 architecture, or the *Fiber Miniapp Suite* [18], developed by RIKEN Advanced Institute for Computational Science (RIKEN AICS) and the Tokyo Institute of Technology.

In this work, we used DPSNN as a mini-application benchmarking tool to simulate networks of point-like spiking neurons of size compatible with reaching the real-time target. The network is composed of 80% Leaky Integrate-and-Fire neurons with Spike Frequency Adaptation (SFA), representing cortical pyramidal excitatory neurons with fatigue and 20% inhibitory neurons. SFA is switched off for inhibitory neurons. This network is a down-scaling of a grid of cortical columns [9] with realistic long range inter-columnar synaptic connectivity [10]. This network is able to enter both an asynchronous awake-like regime and a deep-sleep-like slow wave activity, by tuning the values of SFA and stimulation. Within the Wavescales experiment, a similar model with SFA is extended to study the interactions between Slow Waves Activity, memory association and synaptic homeostasis in a thalamo-cortical model applied to the classification of MNIST handwritten digits [19]. In this paper, synapses inject instantaneous post-synaptic currents while synaptic plasticity is disabled. The simulator implements a mixed event-driven (synaptic and neural dynamics) and time-driven (exchange of spiking messages) integration scheme. As discussed in the previous section, the number of synapses projected by each neuron is kept constant with an average value of 1125 synapses per neuron, the synaptic adjacency matrix is homogenously sparse, and neurons are evenly distributed among processes.

Each neuron receives also the stimulus of 400 “external” synapses, each one delivering a Poissonian spike train at a rate of about 3 Hz. After an initial transient, the neural network enters an asynchronous irregular firing regime at a mean rate of about 3.2 Hz in all simulations used for the scaling measures of this paper.

Inter-process communication is necessary to deliver spikes to target neurons residing on a process different from the one hosting the source neuron. Spikes are delivered using the AER representation (spiking neuron ID, emission time) [20]; in our case 12 byte per spike are required. The exchange of spikes is implemented in the set-up of this paper by means of synchronous MPI collectives. In a process, all spikes produced by neurons and targeted to neurons belonging to another are packed into a single message and delivered. The total number of messages required for all-to-all communication increases with the square of the number of processes on which the simulation is run. This throttles the application into different regimes, allowing to stress and test several elements of the execution platform.

Here is a rundown of the application tasks that the simulator performs and that allow to gauge the components of the architecture under test:

- **Computation**: event-driven integration of all neural dynamics and synaptic current injection events, occurring in a single network synchronization time step (set to 1 ms). This includes a component dominated by memory access to: 1- time delay queues of axonal spikes, 2- lists of neuro-synaptic connections, 3- lists of synapses.
- **Communication**: transmission along the interconnect system of the axonal spikes to the subset of processes where target neurons exist (in the specific set-up of this paper, all processes).
- **Synchronization**: synchronization barrier inserted to simplify the weighting of computation and communication components.

Fluctuations in computation load or communication congestion cause idling cores and diminished parallelization. The relative weight of computation increases with the number of incoming synapses per process. On the other side, a higher number of processes results in higher relative communication costs.

III. Scaling towards real-time

In this domain, being “real-time”, under a “soft” assumption, means a work point for the application such that the total wall-clock time for running it is not greater than the total simulated time, a condition necessary, but not sufficient, for robotics applications and embedding HPC simulations into virtual or real world environments, that would impose more stringent “hard” constraint to be satisfied at the scale of each step, lasting at most a few tens of ms each. The aim of this work is to identify the obstacles that impede reaching the real-time target for large neural networks. We performed a set of strong scaling tests on neural networks of increasing size executed on both Intel and ARM-based distributed platforms. For all network, we simulated 10 s of neural activity. The first testbed we used, representative of standard HPC systems, is made of Intel Xeon E5-2630 v2 processors (clocked at 2.60GHz) communicating over a ConnectX-class InfiniBand interconnect. For what concern the case of ARM-based distributed systems, representative of the embedded system world, we used two different testbeds, respectively based on Trenz and Jetson boards, detailed later in this section.

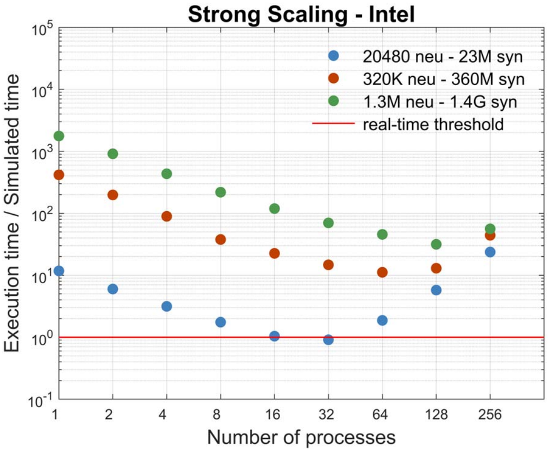

Fig. 2. Strong scaling of different problem sizes on an IB-equipped Intel-based platform. The red line is the threshold to be reached for soft real-time execution.

Figure 2 shows the runtimes for three neural network sizes. They should all be able to run in real-time if the scaling valid for larger configurations applied (see Figure 1). Indeed, the 20480 neurons configuration reaches real-time (9.15 seconds to simulates 10 seconds of activity). The network with 20480 neurons reached its maximum speed when distributed on 32 processes (Figure 2). Communication and synchronization are the main obstacles against scaling (see Figure 3 and Table I). For the 20480 neuron configuration they block a further acceleration over 32 processes and start impeding the scaling toward real-time of larger neural networks after a threshold corresponding to a larger number of processes for the configurations with 320K neurons (16× the 20480 network) and 1280K neurons (64×).

In our simulation, the network communicates spikes every simulated millisecond, the payload for each spike is 12 byte and the average firing rate is about 3 Hz. As a consequence, when the number of cores increases, the network produces a very large number of small message packets. Therefore, this test highlights a “latency” limitation of the interconnect. In general, commercial off-the-shelf interconnects offer adequate throughput when moving large amounts of data but typically trudge when the communication is latency-dominated. This issue with communication — manifesting here with a number of computing cores which is, by today’s standards, not large — is similar to that encountered by the parallel cortical simulator C2 [21] — targeting a scale in excess of that of the cat cortex — on the Dawn Blue Gene/P supercomputer at LLNL, with 147456 CPUs and 144 TB of main memory. The capability to replicate the behaviour of a supercomputer with a mini-app running on a limited number of 1U servers hints at an interesting performance improvement at both larger scale and smaller real-time configuration if identified obstacles to scaling were removed.

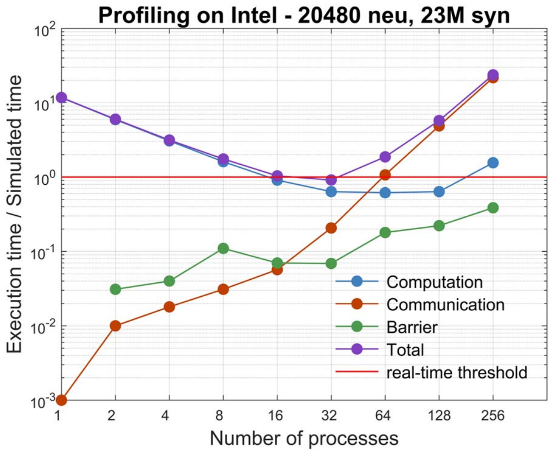

Fig. 3. DPSNN analysis of the Intel-based platform.

TABLE I
PROFILING OF EXECUTION COMPONENTS FOR DIFFERENT NETWORK SIZES.

Neurons	20480N			320KN		1280KN	
Synapses	2.30E+07			3.60E+08		1.44E+09	
Procs	4	32	256	4	256	4	256
Wall-clock (s)	31.5	**9.15**	237	893	441	4341	561
Computation	**97.6%**	69.7%	6.6%	**98.1%**	21.7%	**99.4%**	50.0%
Communicat.	0.6%	22.7%	**91.7%**	0.1%	**79.9%**	0.1%	**48.1%**
Barrier	1.3%	7.5%	1.6%	1.8%	1.1%	0.5%	1.9%

Similar results are obtained performing the same test on two ARM-based platforms; one is the ARM-based prototype of the ExaNeSt project [22] and the other is a commercial development board by NVIDIA equipped with an ARM SoC (Jetson TX1).

The ExaNeSt prototype is composed by four nodes, each node consisting of a TEBF0808 Trenz board equipped with a Trenz TE0808 UltraSOM+ module. The Trenz UltraSOM+ consists of a Xilinx Zynq UltraScale+ xczu9eg-ffvc900-1-e-es1 MPSoC and 2 Gbytes of DDR4 memory. The Zynq UltraScale+ MPSoC incorporates both a processing system composed by quad-core ARM Cortex-A53 and the programmable logic — left unused in this test. All four nodes are connected together through a 1 Gbps

Ethernet-based network. Given that the available cores are limited to 16, the scaling was pushed further up by using the "heterogeneous" mode of MPI, which allows launching an application as a single MPI instance that simultaneously uses distinct executables for different architectures; in this way, the simulation of the neural network is split between partitions of processes executing on ARM and Intel cores. A similar partitioning approach has been used also for the platform based on Jetson boards. Intel cores are about ten times faster than the ARMs on the Trenz boards and about 5 times faster than those on the Jetson (see execution times for 1, 2 and 4 processes in Figures 3, 5 and 6 in a scaling regime dominated by computation). Therefore, the Intel "bath" of processes executed on the Intel partition does not slow down the execution of the ARM Trenz and Jetson boards embedded in it.

The scaling of the system including the Trenz boards up to 64 processes and the profiling of the computation, communication and synchronization components are reported in Figures 4 and 5.

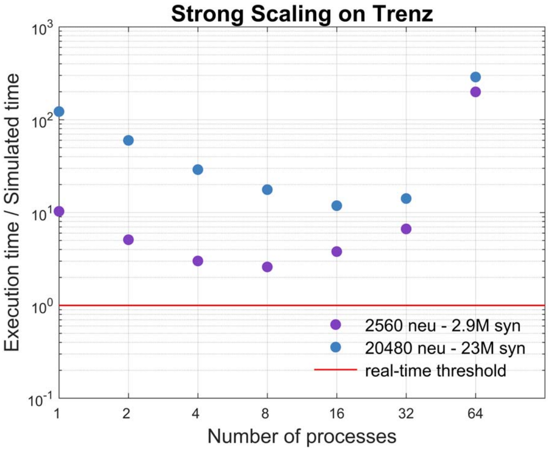

Fig. 4. Strong scaling of a grid simulated on the Trenz platform equipped with GbE interconnect.

The very same test was performed on two NVIDIA Jetson TX1 boards connected by an Ethernet 1 Gbit/s switch to emulate a dual-socket node, each equipped with four ARM Cortex-A57@2 GHz cores plus four ARM Cortex-A53@1.5 GHz cores in 20 nm CMOS technology in big.LITTLE configuration; results are in Figure 6.

IV. Energy-to-Solution analysis

We estimate and compare the instantaneous power, total energy consumption, execution time and energetic cost per synaptic event of a spiking neural network simulator distributed on MPI processes running the DPSNN simulation engine on both the low-power and standard HPC platforms.

The measures were performed with AC/DC current readings by a high-precision GW Instek GDM-8351 digital multimeter connected via USB to a PC: for the SoCs the DC current was sampled downstream the power supply — rated as 19V DC output — as long as only one board was used; for two SoC boards and for the Intel servers the AC current was sampled between the power strip feeding the systems' plugs and the mains outlet — rated as 220V AC output. Such difference should not affect significantly the results, given the closeness to one of the cos φ factor of the server power supply.

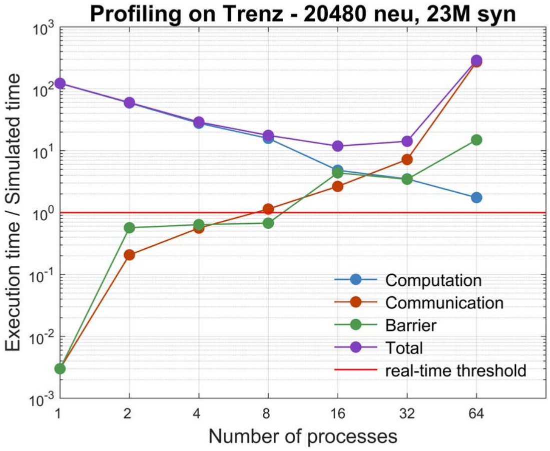

Fig. 5. DPSNN analysis of the Trenz platform.

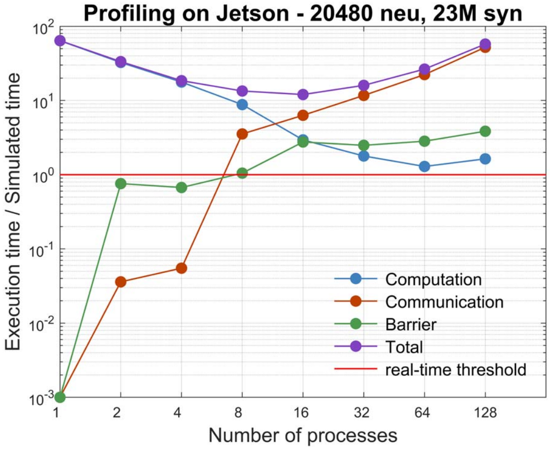

Fig. 6. DPSNN analysis of the NVIDIA SoC platform.

The traditional computing system — *i.e.* "server platform" — is based on SuperMicro X8DTG-D 1U dual-socket servers equipped with a mix of Xeon computing cores in the 32 nm CMOS technology of the Westmere-family, *i.e.* exa-core X5660@2.8 GHz and quad-core E5620@2.4 GHz. This "server platform" is juxtaposed to a typical "embedded platform", which is composed by the two Jetson boards.

The "embedded platform" has 4 GB — 1 GB per core — of LPDDR4 memory, with a memory bandwidth declared as 25.6 GB/s; the "server platform" has a varying amount of DDR3 memory (operating at 1333 MHz) per node — amounting to 1.5÷4 GB per core — and a max declared bandwidth of 32 GB/s.

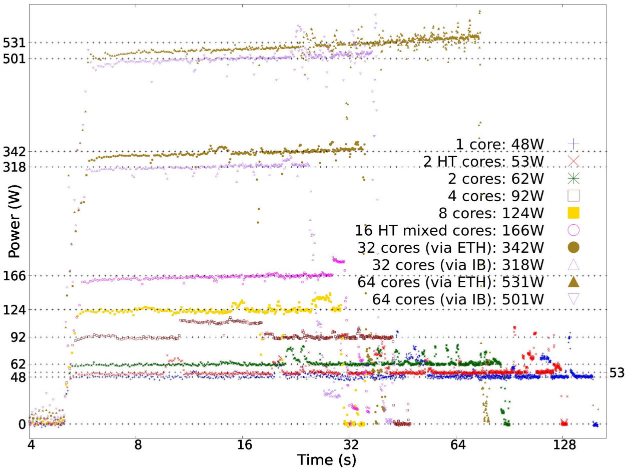

Fig. 7. Scaling of the total power consumption on x86: the same total simulation workload is executed on a number of cores doubling from 1 to 64. Temporal axis in logarithmic scale.

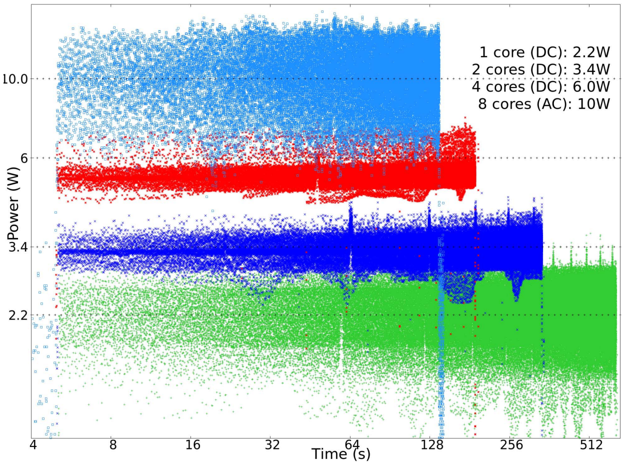

Fig. 8. Power consumption on ARM. Temporal axis in logarithmic scale.

Power and energy consumption were obtained simulating 10 s of activity of a network including ∼20480 neurons. The results of a strong scaling test are reported in Figure 7 for the "server platform" and in Figure 8 for the "embedded" one. In both plots, the legend reports the number of processes employed. Elapsed time is on the X-axis and the power draw is on the Y-axis, the meter reading subtracted from a baseline that is inferred by inspecting the plateau at application start, where 5 s of artificial pause was inserted in the application. Immediately after, a steep knee signals the real start of the simulation and the final drop marks its end. Note that, for the "embedded platform", the plot is split in two ranges: measures between one and four cores are performed on a single board; while two boards are used for eight cores. For both measures we used only one multimeter; attaching the probe at the output of a single power supply - which is DC - was the approach used for a single board. For two boards, we put the multimeter upstream the two power supplies, that implies an AC measure. The transformers' draw causes a significantly higher baseline while the readings are clearly noisier and more spread out. These baselines stand at 564 W for the "server platform" — from Figure 7 — and 49.2 W for the "embedded platform" — upper range of Figure 8.

Energy-to-solution and execution times for the Intel based platform, computed using data from Figure 7, are summarized in Table II, while those from Figure 8 (ARM platform), are in Table III.

TABLE II
DPSNN time, power and energy to solution on x86.

x86 cores	Time (s)	Power (W)	Energy to solution (J)
1	150.9	48	7243.2
2 HT	121.8	53	6455.4
2	80.7	62	5003.4
4	37.4	92	3440.8
8	25.3	124	3137.2
16	26.1	166	4332.6
32 plus ETH	30.0	342	10260.0
32 plus IB	19.7	318	6264.6
64 plus ETH	69.3	531	36798.3
64 plus IB	32.1	501	16082.1

TABLE III
DPSNN time, power and energy to solution on ARM.

ARM cores	Time (s)	Power (W)	Energy to solution (J)
1	636.8	2.2	1273.6
2	334.1	3.4	1135.9
4	185.0	6.0	1110.0
8	133.8	10	1338.0

A peculiar corner case is relative to the second row of Table II, where one physical core was used as two Hyper-Threaded (HT) cores to host two MPI processes; the scaling is clearly not as good as using two real, physical cores (third row), but a small gain is attained nonetheless using what is fundamentally a single core, which, at least for DPSNN, does not completely rule out using HyperThreading as often advised for general HPC applications. The system hosted on a single cluster node can use only up to 16 cores; as can be seen, the minimum time-to-solution is reached with 32 cores, *i.e.* 2 nodes, and only when choosing a low-latency transport for the inter-node communication such as InfiniBand — "IB" in Table II and in Figure 7 — as opposed to Ethernet — marked as "ETH". Moreover, InfiniBand has another significant difference compared with Ethernet; it draws measurably less power when in operation (∼30 W), as the two branches of the 32-core and the 64-core cases show.

Another interesting result is that the absolute minimum for energy-to-solution at 8 cores requires not even using remote communication, which is comprehensible given the relatively small size of the problem being simulated.

V. Conclusion

The computational cost of neural simulations is approximately proportional to the number of synaptic events. The total number of synaptic events is the product of the number of neurons, the number of synapses per neuron, the average firing rate and the total simulation time. The power efficiency can therefore be estimated with a J per synaptic event metric by dividing the total energy-to-solution by the total number of synaptic events. As a reference, using this metric, the energetic cost of Compass [23] — an optimized simulator for the architecture of the TrueNorth ASIC-based platform [4] —, run on an Intel Core i7 CPU 950@3.07 GHz (45 nm CMOS process) with 4 cores and 8 threads, is 5.7 μJ/synaptic event, also in that case excluding the base-line power consumption. Table IV summarizes the power consumption of DPSNN executed on ARM and Intel against that of the Compass/TrueNorth simulator.

TABLE IV
COMPARISON OF ENERGETIC EFFICIENCIES.

DPSNN simulator		**Compass/TrueNorth sim.**
ARM	**Intel**	**Intel**
1.1 (μJ / syn event)	3.4 (μJ / syn event)	5.7 (μJ / syn event)

The ARM architecture on Jetson requires about 3× less energy than Intel, but is about 5× slower (see ARM 4-core row in Table III 1110 J and 185 s vs 3440 J and 37.4 s, 4 Intel cores in Table II). Moreover, the way lower baseline for ARM makes it an interesting candidate for clusters that can be populated much more densely than what is actually possible with Intel.

The profiling of the computation and communication components reported in Figure 3, Figure 5 and Figure 6 demonstrated the critical impact of interconnect on the scaling, limiting the size of the network that can be simulated in real-time. The last two rows of Table II prove the burden of interconnect design on the energy-to-solution. Packets carrying spikes at each simulation step are small, as quantified in Section III. The observed effect on scaling is therefore latency-related, not due to lack of bandwidth.

In conclusion, the design of low-latency, energy-efficient interconnects supporting collective communications is of primary importance to enable a time- and energy-efficient exchange of neural spikes; this is expected to not only make cortical simulations possible at a larger scale but also push their use in embedded systems where it is often precluded by tight real-time constraints and limited power budget.

Availability of code and data

The source code of the DPSNN engine and the data that support the findings of this study are openly available in GitHub at https://github.com/APE-group/201812RealTimeCortSim. The DPSNN code also corresponds to the internal svn release 1163 of the APE group repository.

Acknowledgment

This work has received funding from the European Unions Horizon 2020 Framework Programme for Research and Innovation under Specific Grant Agreements No. 785907 (Human Brain Project SGA2) and No. 720270 (HBP SGA1), Grant Agreement No. 671553 (ExaNeSt) and Grant Agreement No. 754337 (EuroEXA).

References

[1] N. Rajovic *et al.*, "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, Nov 2016.

[2] "The montblanc project." Accessed: 27/Sep/2017.

[3] M. Marazakis, J. Goodacre, D. Fuin, P. Carpenter, J. Thomson, E. Matus, A. Bruno, P. Stenstrom, J. Martin, Y. Durand, and I. Dor, "Euroserver: Share-anything scale-out micro-server design," in *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 678–683, March 2016.

[4] P. A. Merolla *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014.

[5] E. Stromatias, F. Galluppi, C. Patterson, and S. Furber, "Power analysis of large-scale, real-time neural networks on spinnaker," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, Aug 2013.

[6] M.-O. Gewaltig and M. Diesmann, "Nest (neural simulation tool)," *Scholarpedia*, vol. 2, no. 4, p. 1430, 2007.

[7] D. S. Modha, R. Ananthanarayanan, S. K. Esser, A. Ndirango, A. J. Sherbondy, and R. Singh, "Cognitive computing," *Commun. ACM*, vol. 54, pp. 62–71, Aug. 2011.

[8] M. Katevenis *et al.*, "The ExaNeSt Project: Interconnects, Storage, and Packaging for Exascale Systems," in *2016 Euromicro Conference on Digital System Design (DSD)*, pp. 60–67, Aug 2016.

[9] E. Pastorelli, P. S. Paolucci, F. Simula, A. Biagioni, F. Capuani, P. Cretaro, G. D. Bonis, F. L. Cicero, A. Lonardo, M. Martinelli, L. Pontisso, P. Vicini, and R. Ammendola, "Gaussian and exponential lateral connectivity on distributed spiking neural network simulation," in *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, pp. 658–665, March 2018.

[10] P. Schnepel, A. Kumar, M. Zohar, A. Aertsen, and C. Boucsein, "Physiology and impact of horizontal connections in rat neocortex," *Cerebral Cortex*, vol. 25, no. 10, pp. 3818–3835, 2015.

[11] P. S. Paolucci, R. Ammendola, A. Biagioni, O. Frezza, F. Lo Cicero, A. Lonardo, E. Pastorelli, F. Simula, L. Tosoratto, and P. Vicini, "Distributed simulation of polychronous and plastic spiking neural networks: strong and weak scaling of a representative mini-application benchmark executed on a small-scale commodity cluster," *arXiv:1310.8478*, Oct. 2013. http://arxiv.org/abs/1310.8478.

[12] C. Capone, B. Rebollo, A. Muoz, X. Illa, P. Del Giudice, M. V. Sanchez-Vives, and M. Mattia, "Slow waves in cortical slices: How spontaneous activity is shaped by laminar structure," *Cerebral Cortex*, pp. 1–17, 2017.

[13] M. Ruiz-Mejias, L. Ciria-Suarez, M. Mattia, and M. V. Sanchez-Vives, "Slow and fast rhythms generated in the cerebral cortex of the anesthetized mouse," *Journal of Neurophysiology*, vol. 106, no. 6, pp. 2910–2921, 2011.

[14] A. Stroh, H. Adelsberger, A. Groh, C. Rühlmann, S. Fischer, A. Schierloh, K. Deisseroth, and A. Konnerth, "Making waves: Initiation and propagation of corticothalamic ca2+ waves in vivo," *Neuron*, vol. 77, no. 6, pp. 1136 – 1150, 2013.

[15] P. S. Paolucci *et al.*, "Dynamic many-process applications on many-tile embedded systems and HPC clusters: The EURETILE programming environment and execution platforms," *Journal of Systems Architecture*, vol. 69, pp. 29–53, 2016.

[16] M. A. Heroux, D. W. Doerfler, P. S. Crozier, J. M. Willenbring, H. C. Edwards, A. Williams, M. Rajan, E. R. Keiter, H. K. Thornquist, and R. W. Numrich, "Improving Performance via Mini-applications," Tech. Rep. SAND2009-5574, Sandia National Laboratories, 2009.

[17] M. J. Cordery, B. Austin, H. J. Wassermann, C. S. Daley, N. J. Wright, S. D. Hammond, and D. Doerfler, *Analysis of Cray XC30 Performance Using Trinity-NERSC-8 Benchmarks and Comparison with Cray XE6 and IBM BG/Q*, pp. 52–72. Cham: Springer International Publishing, 2014.

[18] "Fiber miniapp suite." http://fiber-miniapp.github.io/. Accessed: 10/Oct/2017.

[19] C. Capone, E. Pastorelli, B. Golosio, and P. S. Paolucci, “Sleep-like slow oscillations improve image classification through synaptic homeostasis and memory association in a thalamo-cortical model,” *arXiv:1810.10498*, 2018.

[20] J. Lazzaro, J. Wawrzynek, M. Mahowald, M. Sivilotti, and D. Gillespie, “Silicon auditory processors as computer peripherals,” *IEEE Transactions on Neural Networks*, vol. 4, pp. 523–528, May 1993.

[21] R. Ananthanarayanan, S. K. Esser, H. D. Simon, and D. S. Modha, “The cat is out of the bag: cortical simulations with 10^9 neurons, 10^{13} synapses,” in *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, pp. 1–12, Nov 2009.

[22] M. Katevenis, R. Ammendola, A. Biagioni, P. Cretaro, O. Frezza, F. L. Cicero, A. Lonardo, M. Martinelli, P. S. Paolucci, E. Pastorelli, F. Simula, P. Vicini, G. Taffoni, J. A. Pascual, J. Navaridas, M. Lujn, J. Goodacre, B. Lietzow, A. Mouzakitis, N. Chrysos, M. Marazakis, P. Gorlani, S. Cozzini, G. P. Brandino, P. Koutsourakis, J. van Ruth, Y. Zhang, and M. Kersten, “Next generation of exascale-class systems: Exanest project and the status of its interconnect and storage development,” *Microprocessors and Microsystems*, vol. 61, pp. 58 – 71, 2018.

[23] R. Preissl *et al.*, “Compass: A scalable simulator for an architecture for cognitive computing,” in *SC ’12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, pp. 1–11, Nov 2012.