2020 International Conference on Computational Science and Computational Intelligence (CSCI)

Do we know the operating principles of our computers better than those of our brain?

János Végh
Kalimános BT
Debrecen, Hungary
Vegh.Janos@gmail.com ORCID: 0000-0002-3247-7810

Ádám J. Berki
University of Medicine, Pharmacy,
Sciences and Technology of Targu Mures, Romania
berki.adam@yahoo.com ORCID: 0000-0001-7099-167X

Abstract—**The increasing interest in understanding the behavior of biological neural networks, and the expanding utilization of artificial neural networks in different fields and scales, both require a thorough understanding of how technological computing works. However, von Neumann in his classic "First Draft" warned that it would be *unsound* to use his suggested paradigm to model neural operation, furthermore that using "too fast" processors *vitiates his paradigm*, which was intended *only* to describe (the timing relations of) vacuum tubes. Thus, it is worth scrutinizing how the present technology solutions can be used to mimic biology. Some electronic components bear a surprising resemblance to some biological structures. However, combining them using different principles can result in systems with inferior efficacy. The paper discusses how the conventional computing principles, components, and thinking about computing limit mimicking biological systems.**

I. INTRODUCTION

For today, the definition of 'neuromorphic computing' started to diverge: from "very large scale integration (VLSI) with analog components that mimicked biological neural systems" changed to "brain-inspired computing for machine intelligence" [1], or "a realistic solution whose architecture and circuit components resemble to their biological counterparts" [2], or "implementations that are based on biologically-inspired or artificial neural networks in or using non-von Neumann architectures" [3]. How much resemblance to biology is required depends on the authors. Some initial resemblance indeed exists, and even some straightforward systems can demonstrate functionality in some aspects similar to that of the nervous system.

"*Successfully addressing these challenges [of neuromorphic computing] will lead to a new class of computers and systems architectures*" [4] was hoped. However, as noticed by the judges of the Gordon Bell Prize, "*surprisingly, [among the winners,] there have been no brain-inspired massively parallel specialized computers*" [5]. Despite the vast need and investments, the concentrated and coordinated efforts, just because of mimicking the biological systems with computing inadequately.

On one side, "*the quest to build an electronic computer based on the operational principles of biological brains has attracted attention over many years*" [6]. On the other side, more and more details come to light about the brain's computational operations.

As that the operating principles of the large computer systems tend to deviate not only from the operating principles of the brain but also from those of a single processor, it is worth reopening the discussion on a decade-old question "*Do computer engineers have something to contribute. . . to the understanding of brain and mind?*" [6]. Maybe, and they surely have something to contribute to the understanding of computing itself. *There is no doubt that the brain does computing, the key question is how?*

As we point out in section II, the terminology makes it hard to discuss the terms. Section III presents that, in general, large-scale computing systems have enormously high energy consumption and low computing efficiency. Section IV discusses the primary reasons for the issues and failures. The necessity of fair imitation of biological objects' temporal behavior in computing systems is discussed in section V. Neuromorphic computing is a particular type of workload in forming the computational efficiency of computing systems, as section VI discusses it. Section VII draws parallel with classic versus modern science and classic versus modern computing. Section VIII provides examples, why a neuromorphic system is not a simple sum of its components.

II. TERMINOLOGY OF NEUROMORPHIC COMPUTING

In his "First Draft" [7], von Neumann mentions the need to develop the ***structure***, giving consideration to both their design and architecture issues. Given that von Neumann discussed computing operations in parallel with neural operations, his model is *biology-inspired*, but because of its timing relations, it is not *biology-mimicking*. After carefully discussing the timing relations of vacuum tubes in section 6.3 of his draft, he made some reasonable omissions and provided an *abstraction* (this is known as "*classic computing paradigm*") having a clear-cut range of validity. He warned that a technology using "too fast" processors *vitiates* that paradigm, and that it would be anyhow *unsound* to apply that neglection to describe operations in the nervous system, for a discussion see [8]). "This is . . . the first substantial work . . . that clearly *separated logic design from implementation. . . .* The computer defined in the 'First Draft' *was never built, and its architecture and design seem now to be forgotten.*" [9].

The comprehensive review [3] analyzed the complete amount of three decades of related publications, based on fre-

978-1-7281-7624-6/20/$31.00 ©2020 IEEE
DOI 10.1109/CSCI51800.2020.00120

quently used keywords. Given that von Neumann constructed a brain-inspired model and defined no architecture, it is hard to interpret even the taxonomy that "*Neuromorphic computing has come to refer to a variety of brain-inspired computers, devices, and models that contrast the pervasive von Neumann computer architecture*." [3] If it is the architecture that his followers implemented von Neumann's *brain-inspired model and abstraction* meant for the "vacuum tube interpretation" only, *what is its contrast, the definition of "neuromorphic computing"?* Because of the "all-in-one" handling and the lack of clear definitions, the area seems to be divergent; improper methods or proper methods outside their range of validity are used.

Von Neumann's model distinguished the [payload] processing time and the [non-payload, but needed] transport time. Any operation, whether neuromorphic or conventional or analog, must comprise these two components. Von Neumann provided a perfect *model*, valid for any kind of computing, including neuromorphic one. Notice that *these components mutually block each other*: an operation must not start until its operands are delivered to the input of the processing unit, and its *results cannot be delivered until their processing finished. This latter point must be carefully checked, especially when using accelerators, recurrent relations of computed feedback.*

Importantly, the transfer time depends both on the physical distance of the respective elements in the computing chain and on the method chosen to transfer the data [10]; it can only be neglected after a meticulous analysis of the corresponding timing relations. The need for also using the time in neural spikes was early noticed: "Activity is assumed to consist of neuronal ensembles – spikes clustered in space *and* in time" [11] (emphasis in original). However, if the packets are sent through a bus with congestion, the information is lost (or distorted) in most cases, especially in large systems. Here we attempt to focus on one vital aspect of biology-mimicking technical computing systems: *how truly they can represent the biological time*.

III. ISSUES WITH LARGE SCALE COMPUTING

Given that we can perform a "staggering amount of (10^{16}) synaptic activity per second" [2], assuming one hundred machine instructions per synaptic activity, we arrive in the exaOps region. It is the needed *payload* performance, rather than *nominal performance* (orders of magnitude between), for neural operation. However, the worst limiting factor is not a large number of operations. In the bio-inspired models, up to billions of entities are organized into specific assemblies. They cooperate via communication, which increases exponentially with increasing complexity/number. (The communication here means sending data and sending/receiving signals, including synchronization.) *The major issue is that technology cannot mimic the "private buses" of biology.*

To get nearer to the biological brain's computationally and energetically efficient operation, we must mimic a complete set of biological features. Only a small portion of the neurons are working simultaneously in solving the actual task; there is a massive number of very simple ('very thin') processors rather than a 'fat' processor; only a portion of the functionality and connection are pre-wired, the rest is mobile; there is an inherent redundancy, replacing a faulty neuron may be possible via systematic training.

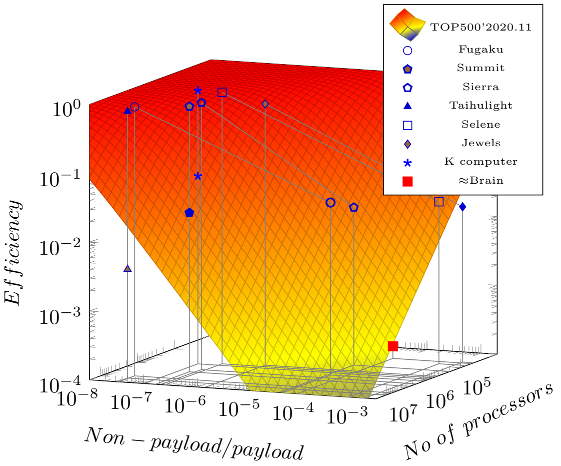

Fig. 1. The 2-parameter efficiency surface (in function of parallelization efficiency measured by benchmark HPL and number of processing elements) as concluded from Amdahl's Law (see [12]), in first order approximation. Some sample efficiency values for selected supercomputers are shown, measured with benchmarks HPL and HPCG, respectively. Also, the estimated efficacy of brain simulation using conventional computing is displayed.

The vast computing systems can cope with their tasks with growing difficulty. The examples include demonstrative failures already known (such as the supercomputers Gyoukou and Aurora'18, or the brain simulator SpiNNaker) and many more may follow: such as Aurora'21 [13], the China mystic supercomputers[1] and the EU planned supercomputers[2]. Also, the present world champion (as of 2020 July) *Fugaku* stalled [14] at some 40% of its planned capacity, and in a half year could increase only marginally. As displayed in Fig. 1, the efficiency of computers assembled from parallelized sequential processors depends *both* on their parallelization efficiency and *number of processors*. It was predicted: "*scaling thus put larger machines at an inherent disadvantage*", since "*this decay in performance is not a fault of the architecture, but is dictated by the limited parallelism*" [15]. As discussed in detail in **[12], [16]**, the efficiency of the parallelization of the large systems is essentially defined by the workload they run. The *artificial intelligence, … it's the most disruptive workload from an I/O pattern perspective*[3]. For orientation, see **[17]** and the estimated efficiency of simulating the brain in Fig. 1.

The performance scaling is strongly nonlinear **[16]**. When targeting neuromorphic features such as "deep learning train-

---

[1] https://www.scmp.com/tech/policy/article/3015997/china-has-decided-not-fan-flames-super-computing-rivalry-amid-us

[2] https://ec.europa.eu/newsroom/dae/document.cfm? doc_id =60156

[3] https://www.nextplatform.com/2019/10/30/cray-revamps-clusterstor-for-the-exascale-era/

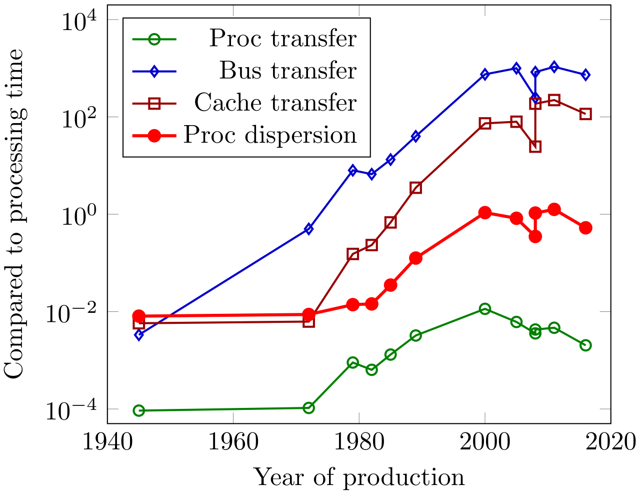

Fig. 2. The history of some relative temporal characteristics of processors, in function of their year of production. Notice how cramming more transistors in a processor changed disadvantageusly their temporal characterisctics.

ing", the issues start to manifest at just a couple of dozens of processors [18]**[19]**.

IV. LIMITATIONS DUE TO TECHNICAL IMPLEMENTATION

Biology uses purely event-driven (asynchronous) computing, while modern electronics uses clock-driven (synchronous) systems. The changing technology and the ever-growing need for more computing also increased the physical size, both of the processors and the systems targeting high processor performance. Synchronizing the operation of elements of a computing chain that have logical dependence, furthermore that because the finite physical size leads to "time skew" in the same signal's arrival times, it introduces a dispersion to the synchronization signal. *The dispersion of synchronizing the computing operations* vastly increases the cycle time, decreases the utilization of all computing units, and enormously increases the power consumption of computing [20], [21], and it is one of the primary reasons for the inefficiency [22] of the Application Specific Integrated Circuit (ASIC) circuits.

Fig. 2 shows how the merit "dispersion" has changed during the years due to demands against computing and the implementation technology. The definition of dispersion, and its detailed discussion is given in **[8]**. The red diagram line "Proc dispersion" (i.e., the dispersion inside the processor) has grown to a value about a hundred times higher than the one at which von Neumann justified his "procedure". The dispersion diagram line, alone, *vitiates* applying the classic paradigm to our processors.[4] From the other diagram lines, one can see that the technical implementation that the data must be delivered between the technology blocks "memory" and "processor", and make the value of "system dispersion" orders of magnitude higher. This feature is mistakenly attributed to the consequence of the "von Neumann architecture" as the "von Neumann bottleneck". A case where the classic paradigm is really "unsound" given that it neglected the transfer time.

How the workload turns this "technical implementation" (stemming from the Single Processor Approach (SPA)) into a real bottleneck is illustrated with the case when neural-like communication shall be served. The inset in Fig. 3 shows a simple neuromorphic use case: one input neuron and one output neuron are communicating through a hidden layer, comprising only two neurons. Fig. 3.A mostly shows the biological implementation: all neurons are directly wired to their partners, i.e., a system of "parallel buses" (the axons) exists. Notice that the operating time also comprises two non-payload times: the data input and data output, which coincide with the non-payload time of the other communication party. The diagram displays the logical and temporal dependencies of the neuronal functionality.

The payload operation ("the computing") can only start after the data is delivered (by the, from this point of view, non-payload functionality: input-side communication), and the output communication can only begin when the computing finished. Important that: i/ *the communication time is an integral part of the total execution time*, and ii/ *the ability to communicate is a native functionality* of the system. In such a parallel implementation, *the performance of the system*, measured as the resulting total time (processing + transmitting), *scales linearly with increasing both the non-payload communication speed and the payload processing speed.*

The present technical approaches assume a similar linearity of the performance dependence of the computing systems as "*Gustafson's formulation [24] gives an illusion that as if N [the number of the processors] can increase indefinitely*" [25]. *Gustafson's 'linear scaling' neglects the communication entirely* (which is not the case, especially in neuromorphic computing). The interplay of the improving parallelization and the general hardware (HW) development covered for decades that *the scaling was used far outside of its range of validity* **[16]**.

Fig. 3.B shows a *technical implementation of a high-speed shared bus* for communication. To the grid's right, the activity that loads the bus at the given time is shown. A double arrow illustrates the communication bandwidth, the length of which is proportional to the number of packages the bus can deliver in a given time unit. The high-speed bus is only very slightly loaded. We assume that the input neuron can send its information in a single message to the hidden layer furthermore that the processing by the neurons in the hidden layer both starts and ends at the same time. However, *the neurons must compete for accessing the bus*, and only one of them can send its message immediately, the other(s) must wait until the bus gets released. The neurons at a few picoseconds distance from each other must communicate through the shared bus, in the several nsec range[5]. As the timing analysis in **[26]** pointed out, the resulting transfer time

---

[4] Notice that here we speak about single (although typically multicore) processors. The wafer-scale systems, the multi-wafer scale systems, deserve special discussion, and we notice that supercomputers' efficiency is also a special form of dispersion.

[5] Notice here again the dispersion

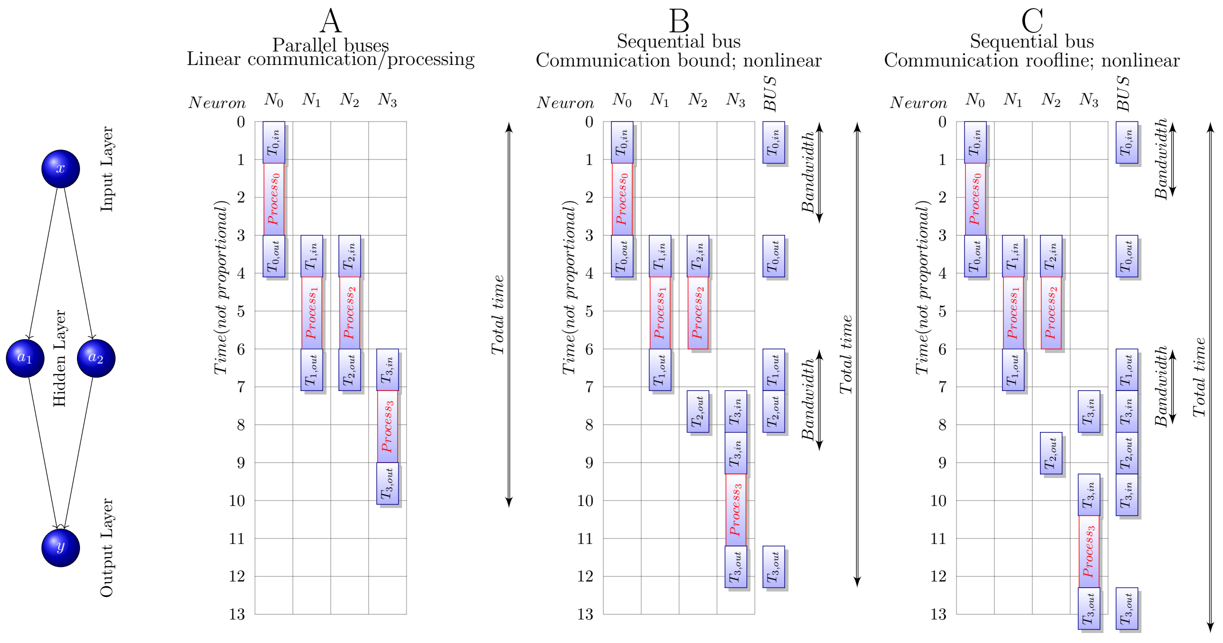

Fig. 3. Implementing neuronal communication in different technical approaches. A: the parallel bus; B and C: the shared serial bus, before and after reaching the communication ”roofline” [23]

depends linearly on the number of ”neurons” in the system, and the systems spend most of their time waiting for the arbiter. The ”high-speed bus” has marginal importance in this case: it is only slightly utilized. This effect is so strong, that in vast systems a ”communicational collapse” [27], follows, when the ”roofline” [23] approached.

The output neuron can only receive the message when the first neuron completed it. Furthermore, the output neuron must first acquire the second message from the bus, and the processing can only begin after having both input arguments. *This constraint results in sequential bus delays both during non-payload processing in the hidden layer and the payload processing in the output neuron*. At this point, one can stick to synchronous computing, which increases dispersion to an intolerable level. It leads, however, to the need of mitigating the dispersion using methods such as ”*spikes … are dropped if the receiving process is busy over several delivery cycles*” [28]. The other option is to proceed without synchronization (i.e., mixing the synchronous and asynchronous operating modes). The HW gates always have an input signal, and when started, they perform the operation they were designed for. If the correct input signal reached its input, then it works as we expect. Otherwise, the output is undefined, as its input is.

Adding one more neuron to the layer introduces one more delay, which explains why ”*shallow networks with many neurons per layer … scale worse than deep networks with less neurons*” [18]: the system sends them at different times in the different layers (and even they may have independent buses between the layers), although the shared bus persists in limiting the communication.

The neuromorphic systems that need much more communication, making their non-payload to payload ratio very wrong. The linear dependence at low nominal performance values explains why the initial successes of *any new technology, material or method* in the field, using the classic computing model, can be misleading: in simple cases (at ”toy” level [29]), the classic paradigm performs tolerably well thanks to that compared to biological neural networks, current neuron/dendrite models are simple, the networks small, and learning models appear to be rather basic.

V. THE IMPORTANCE OF IMITATING THE TIMELY BEHAVIOR

In both biological and electronic systems, both the distance between the entities of the network, and the signal propagation speed is finite. Because of this, in the physically large-sized systems the ’idle time’ of the processors defines the final performance a parallelized sequential system can achieve. In the conventional computing systems, the ’data dependence’ limits the available parallelism: we must compute the data before using it as an argument for another computation. Thanks to the ’weak scaling’ [24], *this ’communication time’ is neglected*.

In neuromorphic computing, however, as discussed in connection with Fig. 3, the transfer time is a vital part of information processing. A biological brain must deploy a

”speed accelerator” to ensure that the control signals arrive at the target destination before the arrival of the controlled messages, despite that the former derived from a distant part of the brain [30]. *This aspect is so vital in biology that the brain deploys many cells with the associated energy investment to keep the communication speed higher for the control signal.*

To arrive at a less limiting architecture, additional ideas shall be borrowed from the biological architectures. Although the ”private buses” cannot be reproduced technologically, introducing hierarchical buses and organized traffic, using computer network-like logical addressing and the ”small world” nature of communication, more close imitation can be reached [22]. The approach uses the following key novel ideas: 1) implementing directly-wired connections between physically neighboring cells; 2) creating a particular hierarchical bus system; 3) placing a special communication unit, the (Inter-Core Communication Block (ICCB), Fig. 4B, purple) between the computer cores mimicking neurons (Fig. 4B, green); 4) creating a specialized ’fat core’ neuron (Fig. 4B) with the extra abilities to access the local and far memories (Fig. 4 M) and to forward messages via the gateway (Fig. 4 G) to other similar ’fat core’ neurons (similar gateways can be implemented for the inter-processor communication, and higher organizational levels). This organization enables both easy sharing of locally important state variables, keeps local traffic away from the bus(es), and reduces wiring inside the chip. The ICCBs can closely mimick the correct parallel behavior of biology. The resemblance between Fig. 4A and Fig. 7 in reference [30] underlines the importance of making a clear distinction between handling ’near’ and ’far’ signals. Furthermore, it underlines as well as the necessity of their simultaneous utilization. The conduction time in biological systems must be separately maintained in biology-mimicking computing systems. Making time-stamps and relying on the computer network delivery principles is not sufficient for maintaining correct relative timing. *The timely behavior is a vital feature of the biology-mimicking systems and can not be replaced with the synchronization principles of computing*.

One possible way is to put a ”time grid” on the processes simulating biology. This requirement results in the neurons continuing their calculation periodically from some concerted state. Such a synchronization method introduces a ”biological clock period” that is a million-fold longer than the processor’s clock period. Although this effect drastically reduces the achievable computing temporal performance **[17]**, the synchronization principle is so common that also the special-purpose neuromorphic chips [32], [33] use it as a built-in feature. In their case, the speed of neuronal functionality is hundreds of times higher than that of the competing solutions, and the communication principles are slightly different (i.e., the non-payload/payload ratio is vastly different), the performance-limiting effect of the ”quantal nature of computing time” persists when used in extensive systems.

VI. THE ROLE OF THE WORKLOAD ON THE COMPUTING EFFICIENCY

As was very early predicted [34] and decades later experimentally confirmed [15], the scaling of the parallelized computing is not linear. Even, ”*there comes a point when using more processors . . . actually increases the execution time rather than reducing it*” [15]. Paper **[12]** discusses first/second-order approaches to explain the issue. The first-order approach explains the experienced saturation, and the second-order the predicted decrease. As **[12]** discusses, the different workloads, mainly due to their different communication-to-computation ratio, work with different efficiency on the same computer system [35]. The neuromorphic operation on conventional architectures shows the same issues **[19], [16]**.

VII. LIMITATIONS DUE TO THE CLASSIC COMPUTING PARADIGM

The careful analysis discovers a remarkable parallel between the proposed ’modern computing’ **[36]** versus the classic computing and the modern science versus the classic science. The parallel can help accept that *what one can not experience in every-day computing can be true when using computing under extreme conditions*. Using another computing theory is a must, especially when targeting neuromorphic computing. *In the frames of ”classic computing”*, as was bitterly admitted [28], *”any studies on processes like plasticity, learning, and development exhibited over hours and days of biological time are outside our reach”*.

VIII. A SYSTEM IS NOT A SIMPLE SUM OF ITS COMPONENTS

One must not conclude from a feature of a component to a similar feature of the system: the non-linearity discussed above is especially valid for the large-scale computing systems mimicking neuromorphic operation. We mention two prominent examples here. One can assume that if the time of the operation of a neuron can be shortened, the performance of the whole system gets proportionally better. Two distinct options are to use shorter operands (move less data and to perform less bit manipulations) and to mimic the operation of the neuron using quick analog signal processing instead of slow digital calculation.

The so-called *HPL-AI* benchmark used *Mixed Precision*[6] [37] rather than Double Precision operands in benchmarking their supercomputer. *The name suggests as if in solving AI tasks, the supercomputer can show that peak efficiency.*

We expect that when using half-precision (FP16) rather than double precision (FP64) operands in the calculations, four times less data are transferred and manipulated by the system. The measured power consumption data underpin our

---

[6] Both names are used rather inconsequentially. On one side, the test itself has not much to do with Artificial Intelligence (AI), just uses the operand length common in AI tasks; the benchmark HPL, similarly to AI, is a workload type. On the other side, the Mixed Precision is Half Precision: it is natural for multiplication twice as long operands are used temporarily. It is a different question that the operations are contracted.

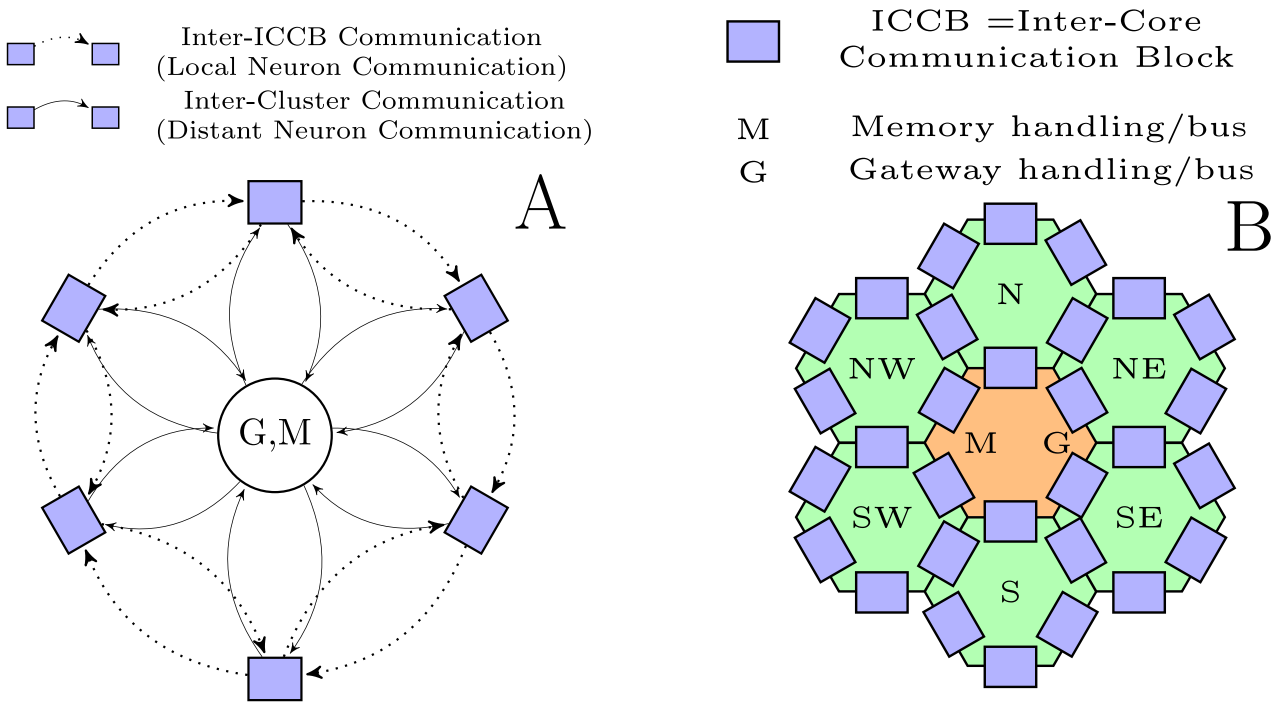

Fig. 4. Subfigure A (compare to Fig.7 in [30]) shows a proposal [31] of how to reduce the limiting effect of the SPA, via mimicking the communication between local neurons using direct-wired inter-core communication and the communication between the farther neurons via using the inter-cluster communication bus, in the cluster head. Subfigure B suggests a possible implementation of the principle: the Inter-Core Communication Blocks represent a ”local bus” (directly wired, with no contention), while the neurons can communicate with each other through the ’G’ gateway and the ’M’ (local and global) memory.

expectations. However, the computing performance is only three times higher than in the case of using 64-bit (FP64) operands. The non-linearity has its effect even in this simple case. In the benchmark, the housekeeping activity also takes time **[16]**.

Another plausible assumption is that if we use quick analog signal processing to replace the slow digital calculation, as proposed in [38], [39] or using different materials/effects [29], the system gets proportionally quicker. Adding analog components to a digital processor, however, has its price. Given that the digital processor cannot handle resources outside of its world, one must call the operating system (OS) for help. The required context switching takes time in the order of executing 10^4 instructions [40], [41], which dramatically increases the total execution time and makes the non-payload to payload ratio much worse. Similarly, *it is not reasonable to decrease processing speed if the corresponding transfer speed cannot be reduced* **[26]**. Although these cases seem to be very different, they share at least the common feature. They change not only one parameter: *they also change the non-payload to payload ratio* that defines the efficiency. The analysis of their temporal behavior (including their connection to the computing system) limits the utility of any new material/ effect/ technology; the detailed discussion see in **[26]**.

IX. SUMMARY

The authors have identified some critical bottlenecks in current computational systems/neuronal networks rendering the conventional computing architectures unadaptable to large (and even medium) sized neuromorphic computing. Built with the segregated processor (SPA, wording from Amdahl [34]), the current systems lack autonomous communication of processors and have an inefficient method of imitating biological systems; mainly their temporal behavior.

REFERENCES

[1] K. Roy, A. Jaiswal, P. Panda, Towards spike-based machine intelligence with neuromorphic computing., Nature 575 (2019) 607–617. `doi:https://doi.org/10.1038/s41586-019-1677-2`.

[2] M. R. Ahmed, B. K. Sujatha, A review on methods, issues and challenges in neuromorphic engineering, in: 2015 International Conference on Communications and Signal Processing (ICCSP), 2015, pp. 0899–0903.

[3] C. D. S. et al, A Survey of Neuromorphic Computing and Neural Networks in Hardware`arXiv:1705.06963`.

[4] US DOE Office of Science, Report of a Roundtable Convened to Consider Neuromorphic Computing Basic Research Needs, https://science.osti.gov/-/media/ascr/pdf/programdocuments/docs/Neuromorphic-Computing-Report_FNLBLP.pdf (2015).

[5] G. Bell, D. H. Bailey, J. Dongarra, A. H. Karp, K. Walsh, A look back on 30 years of the Gordon Bell Prize, The International Journal of High Performance Computing Applications 31 (6) (2017) 469–484. URL https://doi.org/10.1177/1094342017738610

[6] Steve Furber and Steve Temple, Neural systems engineering, J. R. Soc. Interface 4 (2007) 193–206. `doi:10.1098/rsif.2006.0177`.

[7] J. von Neumann, First Draft of a Report on the EDVAC, https://web.archive.org/web/20130314123032/http://qss.stanford.edu/~godfrey/vonNeumann/vnedvac.pdf (1945).

[8] J. Végh, von Neumann's missing "Second Draft": what it should contain, in: The 2020 International Conference on Computational Science and Computational Intelligence; CSCI'20: December 16-18, 2020, Las Vegas, USA, paper CSCI2019, IEEE, 2020, p. paper CSCI2019.
URL https://arxiv.org/abs/2011.04727

[9] M. D. Godfrey, D. F. Hendry, The Computer as von Neumann Planned It, IEEE Annals of the History of Computing 15 (1) (1993) 11–21.

[10] J. Végh and Ádám J. Berki, On the spatiotemporal behavior in biology-mimicking computing systems, Brain Informatics (2020) in review.
URL https://arxiv.org/abs/2009.08841

[11] K. Boahen, Point-to-Point Connectivity Between Neuromorphic Chips UsingAddress Events, IEEE Trans. on Circuits and Systems Part 47 (2000) 416–434.

[12] J. Végh, Finally, how many efficiencies the supercomputers have?, The Journal of Supercomputing 76 (12) (2020) 9430–9455.
URL http://link.springer.com/article/10.1007/s11227-020-03210-4

[13] Top500.org, Retooled Aurora Supercomputer Will Be America's First Exascale System, https://www.top500.org/news/retooled-aurora-supercomputer-will-be-americas-first-exascale-system/ (2017).

[14] J. Dongarra, Report on the Fujitsu Fugaku System, Tech. Rep. Tech Report ICL-UT-20-06, University of Tennessee Department of Electrical Engineering and Computer Science (June 2016).

[15] J. P. Singh, J. L. Hennessy, A. Gupta, Scaling parallel programs for multiprocessors: Methodology and examples, Computer 26 (7) (1993) 42–50. doi:10.1109/MC.1993.274941.

[16] J. Végh, Which scaling rule applies to Artificial Neural Networks, in: Computational Science and Computational Intelligence (CSCE) The 22nd Int'l Conf on Artificial Intelligence (ICAI'20), IEEE, 2020, pp. Accepted ICA2246, in print; in review in Neurocomputing. arXiv:2005.08942.
URL http://arxiv.org/abs/2005.08942

[17] J. Végh, How Amdahl's Law limits performance of large artificial neural networks, Brain Informatics 6 (2019) 1–11.
URL https://braininformatics.springeropen.com/articles/10.1186/s40708-019-0097-2/metrics

[18] J. Keuper, F.-J. Preundt, Distributed Training of Deep Neural Networks: Theoretical and Practical Limits of Parallel Scalability, in: 2nd Workshop on Machine Learning in HPC Environments (MLHPC), IEEE, 2016, pp. 1469–1476. doi:10.1109/MLHPC.2016.006.
URL https://www.researchgate.net/publication/308457837

[19] J. Végh, How deep machine learning can be, A Closer Look at Convolutional Neural Networks, Nova, In press, 2020, pp. 141–169.
URL https://arxiv.org/abs/2005.00872

[20] R. Waser (Ed.), Advanced Electronics Materials and Novel Devices, Nanoelectronics and Information Technology, Wiley, 2012.

[21] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, et al., Dark Silicon and the End of Multicore Scaling, IEEE Micro 32 (3) (2012) 122–134.

[22] R. Hameed, W. Qadeer, M. Wachs, O. Azizi, A. Solomatnikov, B. C. Lee, S. Richardson, C. Kozyrakis, M. Horowitz, Understanding sources of inefficiency in general-purpose chips, in: Proceedings of the 37th Annual International Symposium on Computer Architecture, ISCA '10, ACM, New York, NY, USA, 2010, pp. 37–47. doi:10.1145/1815961.1815968.
URL http://doi.acm.org/10.1145/1815961.1815968

[23] S. Williams, A. Waterman, D. Patterson, Roofline: An insightful visual performance model for multicore architectures, Commun. ACM 52 (4) (2009) 65–76.

[24] J. L. Gustafson, Reevaluating Amdahl's Law, Commun. ACM 31 (5) (1988) 532–533. doi:10.1145/42411.42415.

[25] Y. Shi, Reevaluating Amdahl's Law and Gustafson's Law, https://www.researchgate.net/publication/228367369_Reevaluating_Amdahl's_law_and_Gustafson's_law (1996).

[26] J. Végh, Introducing Temporal Behavior to Computing Science, in: 2020 CSCE, Fundamentals of Computing Science, IEEE, 2020, pp. Accepted FCS2930, in print. arXiv:2006.01128.
URL https://arxiv.org/abs/2006.01128

[27] S. Moradi, R. Manohar, The impact of on-chip communication on memory technologies for neuromorphic systems, Journal of Physics D: Applied Physics 52 (1) (2018) 014003.

[28] S. J. van Albada, A. G. Rowley, J. Senk, M. Hopkins, M. Schmidt, A. B. Stokes, D. R. Lester, M. Diesmann, S. B. Furber, Performance Comparison of the Digital Neuromorphic Hardware SpiNNaker and the Neural Network Simulation Software NEST for a Full-Scale Cortical Microcircuit Model, Frontiers in Neuroscience 12 (2018) 291.

[29] D. Markovic, A. Mizrahi, D. Querlioz, J. Grollier, Physics for neuromorphic computing, Nature Reviews Physics 2 (2020) 499–510. doi:https://www.nature.com/articles/s42254-020-0208-2.pdf.

[30] György Buzsáki and Xiao-Jing Wang, Mechanisms of Gamma Oscillations, Annual Reviews of Neurosciences 3 (4) (2012) 19:1–19:29. doi:10.1146/annurev-neuro-062111-150444.

[31] J. Végh, How to extend the Single-Processor Paradigm to the Explicitly Many-Processor Approach, in: 2020 CSCE, Fundamentals of Computing Science, IEEE, 2020, pp. Accepted FCS2243, in print. arXiv:2006.00532.
URL https://arxiv.org/abs/2006.00532

[32] J. Sawada et al, TrueNorth Ecosystem for Brain-Inspired Computing: Scalable Systems, Software, and Applications, in: SC '16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2016, pp. 130–141.

[33] M. Davies, et al, Loihi: A Neuromorphic Manycore Processor with On-Chip Learning, IEEE Micro 38 (2018) 82–99.

[34] G. M. Amdahl, Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities, in: AFIPS Conference Proceedings, Vol. 30, 1967, pp. 483–485. doi:10.1145/1465482.1465560.

[35] IEEE Spectrum, Two Different Top500 Supercomputing Benchmarks Show Two Different Top Supercomputers, https://spectrum.ieee.org/tech-talk/computing/hardware/two-different-top500-supercomputing-benchmarks-show\-two-different-top-supercomputers (2017).

[36] J. Végh, A. Tisan, The need for modern computing paradigm: Science applied to computing, in: International Conference on Computational Science and Computational Intelligence CSCI The 25th Int'l Conf on Parallel and Distributed Processing Techniques and Applications, IEEE, 2019, pp. 1523–1532. doi:10.1109/CSCI49370.2019.00283.
URL http://arxiv.org/abs/1908.02651

[37] A. Haidar, S. Tomov, J. Dongarra, N. J. Higham, Harnessing GPU Tensor Cores for Fast FP16 Arithmetic to Speed Up Mixed-precision Iterative Refinement Solvers, in: Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis, SC '18, IEEE Press, 2018, pp. 47:1–47:11.

[38] E. Chicca, G. Indiveri, A recipe for creating ideal hybrid memristive-CMOS neuromorphic processing systems, Applied Physics Letters 116 (12) (2020) 120501. arXiv:https://doi.org/10.1063/1.5142089, doi:10.1063/1.5142089.
URL https://doi.org/10.1063/1.5142089

[39] Building brain-inspired computing, Nature Communications 10 (12) (2019) 4838.
URL https://doi.org/10.1038/s41467-019-12521-x

[40] F. M. David, J. C. Carlyle, R. H. Campbell, Context Switch Overheads for Linux on ARM Platforms, in: Proceedings of the 2007 Workshop on Experimental Computer Science, ExpCS '07, ACM, New York, NY, USA, 2007. doi:10.1145/1281700.1281703.
URL http://doi.acm.org/10.1145/1281700.1281703

[41] D. Tsafrir, The context-switch overhead inflicted by hardware interrupts (and the enigma of do-nothing loops), in: Proceedings of the 2007 Workshop on Experimental Computer Science, ExpCS '07, ACM, New York, NY, USA, 2007, pp. 3–3.