Challenges and limitations of High Performance Computing

Basic terms, benchmarking, workloads, interconnections, dark performance, slow learning and failed supercomputers

János Végh

©János Végh (Vegh.Janos@gmail.com)

V0.1.2 @2020.01.00 (https://github.com/jvegh/MultEdu/V0.5.9)

Printed in the World, using recycled electrons

Abstract

This booklet is the take-away form . It can be accompanied with the slide show, available from the conference side.

This booklet is the take-away form of the tutorial held at ISC High Performance Computing event,June 21-25, 2020, Frankfurt, for the participants. Compared to the slide show of the tutorial that was told during the lectures, the present booklet has additional material (in more concise and compact, ordered form). The booklet and the slide show are generated from the same source. The figures and some of the text are the same in both appearance. Compared to the lecture told at the tutorial, the audience, the questions, the atmosphere, etc. however may cause smaller differences, like another wording or omitting/adding some details. Recall also your memories with browsing the slide show, and read this material in parallel with that.

First edition: February 2019

V0.1.2 @2020.01.00 (https://github.com/jvegh/MultEdu V0.5.9)
10 09 08 07 06 05 04 03 02 01 19 18 17 16 15 14 13

Contents

Chapter

1 Overview of the tutorial B

1.1 Intro to the parallelized sequential computing

After that the dynamic growing of the single-processor performance has practically stalled about two decades ago [1], the only way to achieve the required high computing performance remained parallelizing the work of a very large number of sequentially working single processors. However, as was very early predicted [2] and decades later experimentally confirmed [3], the scaling of the parallelized computing is not linear. Even, as it was predicted, "*there comes a point when using more processors . . . actually increases the execution time rather than reducing it*" [3]. The parallelization operation has its own rules of game and has its inherent performance limitations [4, 5]. The present commonly used computing paradigm (and its technical implementation) also limits the performance of supercomputers [6].

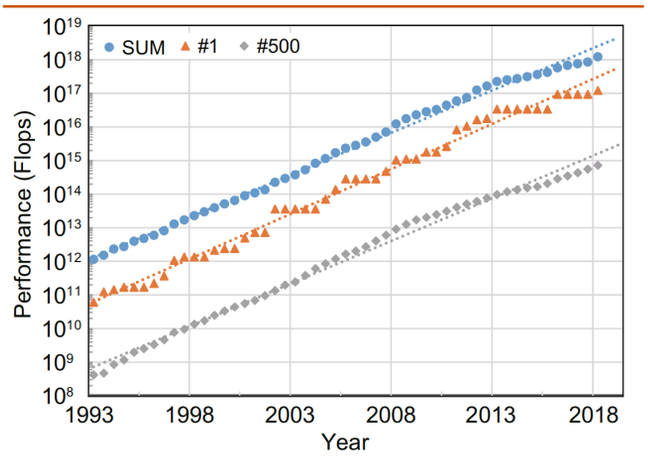

Figure 1.1 : The performances at the beginning of 2018 [7]

The expectations against supercomputers are excessive. Although even the Eflops payload performance has not yet been achieved, already the implementation of the Zflops supercomputers are planned [7, 8]. It looks like that in the feasibility studies an analysis whether some inherent performance bound exists remained out of sight either in USA [9, 10] or in EU [11] or in Japan [12] or in China [7]; although serious counter-arguments are also listed [13]. The confusion is growing: some "must work" world-class supercomputers (like Gyoukou, Aurora, SpiNNaker) are failed. In addition to the previously existing "*two different efficiencies of supercomputers*" [14] further efficiency/performance value appeared[1] (and several more can easily be derived).

Ignorance in this fields is dangerous: "*In December 2017, PEZY President Motoaki Saito, and PEZY employee, Daisuke Suzuki, were arrested on a charges of fraud – that is – padding expenses*"[2]. They were neither beginners nor outsiders: "*In 2015, computers using PEZY processors occupied the top 3 slots on the Green 500 supercomputer list*". In Japan, the company PEZY[3] expected infinitely large parallelized computing performance. Accordingly they assumed (and announced in advance) that Japan will have the #1 supercomputer, with about 0.13 Eflops. Finally, Gyoukou was nominated with 0.019 Eflops and conquered slot #4. However, only 2.4M cores (out of the 19.8M cores available) were measured.

Actually, there was no fraud. They were simply not aware of that a supercomputer performance limit existed, and they attempted to exceed it. "padding expenses" here means that it was assumed that some of the delivered processors were not "real" processors. Actually, those processors contributed to the "dark performance" only, because of the limitations discussed in this tutorial.

The same happened with the Aurora'18 in the US. (The Intel-Cray Aurora supercomputer which was planned for 2018 has been shifted to 2021, scaling up its performance from 180 petaFLOPS to 1 exaFLOPS[4]). Initially it was communicated that "*Aurora was retargeted*"[15], just weeks before its announced startup time, and that "*DOE Witholds Details of First Exascale Supercomputer, Even as it Solicits Researchers to Apply for Early Access*" [16]. Intel learned the lesson[5]: "*the company would be replacing the next-gen Phi (Knights Hill) with "a new platform*

[1] https://blogs.nvidia.com/blog/2019/06/17/hpc-ai-performance-record-summit/

[2] https://en.wikipedia.org/wiki/PEZY_Computing

[3] BTW: The name **PEZY** is an acronym derived from the Greek derived metric prefixs **p**eta-, **e**xa-, **z**etta-, and **y**otta

[4] https://fuse.wikichip.org/news/478/intel-axes-knights-hill-plans-a-new-microarchitecture-for-exascale/

[5] https://itpeernetwork.intel.com/unleashing-high-performance-computing/

Figure 1.2 : The news story of the PEZY fraud

and new microarchitecture specifically designed for exascale"." Because the single thread optimized processor cannot be optimized for many-processor environment. For today it was quietly admitted that "Aurora failed".

The same is happening today with the "mystic China supercomputers "[6] expected to deliver 0.2 Eflops payload performance. This is expected also be the fate of the planned EU supercomputers expected to deliver 0.13-0.18 Eflops: they are positioned in the "death zone", see Fig. 1.3.

The exa-scale race is going on [17, 18, 19], without seeing the rules of the game clear. This is the target of this tutorial.

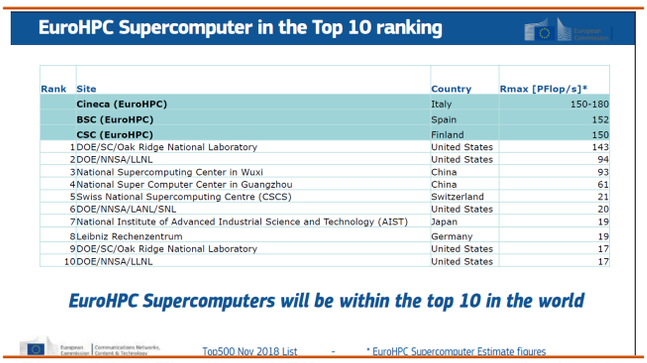

Figure 1.3 : The planned EU supercomputers are in the "death zone"

- Lesson B1 recalls the general limitations affecting supercomputers based on parallelized sequential processing, as concluded from the model of parallelization. Some numerical values of the limiting parameters of the technical implementations are presented.
- Lesson b2 studies the history of the supercomputing using the database TOP500 through calculating the "effective parallelization", and provides evidence that Amdahl's Law directs the history of parallelized sequential computing. It makes clear that the resulting parallel performance has stalled.
- Lesson B3 scrutinizes benchmarking: what bechmarks measure, why computers have different efficiencies, how the workflow affects supercomputer performance. The effect of different technical implementations (including GPGPU acceleration, half precision, OpenCAPI bus and interconnection quality) will also be demonstrated.
- In lesson B4 the parallel with the modern science is completed: the "quantal nature of time" and "communicational collapse" will be intrduced and a surprising parallel between measuring quantum states and measuring supercomputer performances will be drawn. *Given that the major contributor to the non-parallelizable portion of the task is the computation/communication itself, the further technical enhancement, without changing the principle of computation is "mission impossible"*: only increase the "dark performance" of the computing systems with extreme size.

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

Chapter

2 B3: Benchmarking performance of supercomputing

The most obvious field to apply our model to is supercomputing. Here the number of processors is extremely high and – as will be demonstrated – all contributions to $(1-\alpha_{eff})$ have been greatly reduced during their very well documented history spanning a quarter of century [20]. Our model is flexible enough to describe those hardware (HW)/software (SW) architectures and also indirectly prove the validity of the principles used in the model.

2.1 The effect of workflow on the performance

It is also known since decades that "*the inherent communication-to-computation ratio in a parallel application is one of the important determinants of its performance on any architecture. The higher the ratio, the less likely is a machine to provide effective performance on that application.*"[3] This observation is demonstrated in Fig. 2.1.

The left column of the figure displays different common communication intensities (different workflow types). The bottom subfigure shows the common case of Artificial Intelligence, where some intermediate layers exchange information with each other and the rest of the "neurons". Notice that here the communication intensity is proportional with m^2, the square of the number of "neurons", in the hidden layer.

The middle and bottom subfigures in the left column depict the communication intensity of the two popular supercomputer benchmark programs High Performance Linpack (HPL) and High Performance Conjugate Gradients (HPCG) in the same style. Here the initiating and terminating node is a single core and the "hidden layer" comprises all the rest of the cores. The communication intensity of HPL is proportional with N (the total number of processing units) and that of HPCG with $h \cdot N$, because h iterations take place. Let us notice that when a supercomputer is utilized for calculation in AI mode, the AI mode means performance proportional with the number of neurons in the hidden layer. If the supercomputer having 1M core is running in HPL configuration, and the AI mode system runs on x:1k:1k:y cores, they will have the same performance [21]. (The absolute times must not be compared, but their ratio can.)

The right hand column requires more attention. The right-hand scale and the blue line refer to the payload performance. The left hand scale refers to the α contributions of different kinds. For visibility, only the looping (operating system (OS)) contribution and the SW (calculation+communication) contributions are shown. The communication intensity is the lowest for the HPL case and the highest is for the Artificial Intelligence (AI) case. Correspondingly, the payload performance is the best for HPL and worst for HPL.

The reason is the different amounts of α contributions. Between subfigures A and B, the amount (and so the contribution) of the calculation is increased, mainly because of the need of iteration. Between subfigures B and C, the communication intensity is increased, leading to orders of magnitude decrease in the efficiency and also the inflexion point moves towards much lower nominal performance values.

2.2 The effect of interconnection on the performance

In a somewhat simplified view, the resulting performance can be calculated using the contributions to α as

$$P(N,\alpha) = \frac{N \cdot P_{single}}{N \cdot (1 - \alpha_{Net} - \alpha_{Compute} - \alpha_{Others}) + \approx 1} \tag{2.1}$$

That is, two of the contributions are handled with emphasis. The theory easily provides values for the contributions for the interconnection and calculation separately. Fortunately, the public database TOP500 [20] also provides data measured under conditions greatly similar to the 'net' contribution. Of course, the measured data contain the contribution of all components. However, as will be shown below, in the total contribution those mentioned contributions dominate, so the measured α can be directly compared with the calculated α, although here only qualitative agreement can be expected.

Both the quality of the interconnection and the nominal performance are a parametric function of the time, so one can assume on the theory side that (in a limited time span) the interconnection contribution was changing as shown in Fig. 2.2A. The other major contribution is assumed to be the calculation[1] itself. The benchmark

[1] This time also accessing data ("communicating" is included)

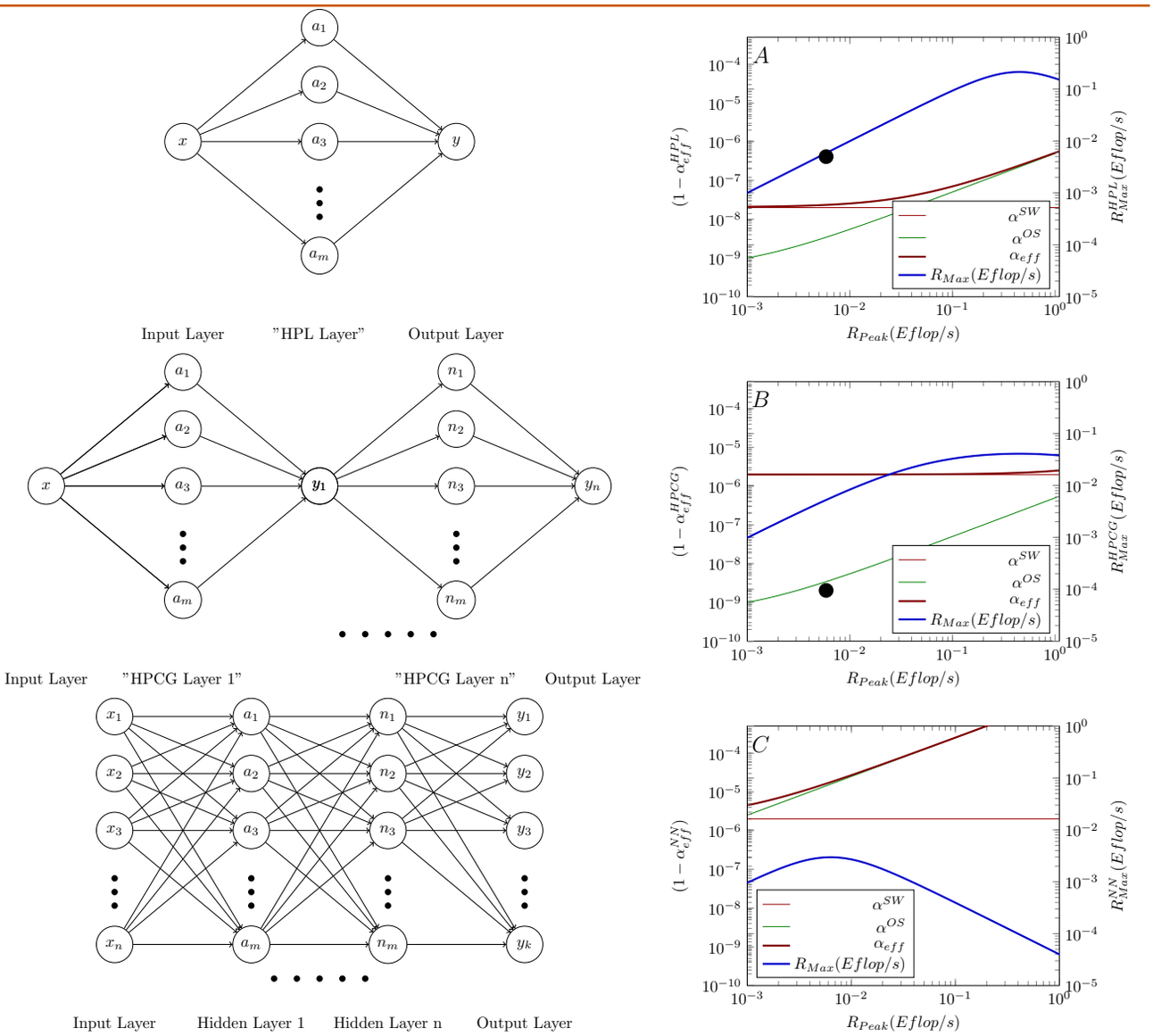

Figure 2.1 : The dependence of the payload performance of the different contrinutions and on the workflow type.

calculation contributions for HPL and HPCG are very different, so the sum of the respective component plus the interconnection component are also very different. Given that at the beginning of the considered time span the contribution from the HPCG calculation and that of the interconnection are in the same order of magnitute, the sum only changes marginally, i.e. the measured performance changes only marginally.

The case with the HPL calculation is drastically different. Since in this case the contribution from the interconnection is very much larger than that from the calculation, the sum of these two contributions changes sensitively as the speed of the interconnection improves. As soon as the contribution from the interconnection decreases to a value comparable with that of the calculation, the decrease of the sum slows down considerably, and the further improvement of the interconnection causes only marginal decrease in the value of the resulting α (and so only a marginal increase in the payload performance).

The measured data enable to draw the same conclusion, but one must consider that here multiple parameters may have been changed. The tendency, however, is surprisingly clear. Fig. 2.2.B is actually a 2.5D diagram: the size of the marks is proportional with the time passed since the beginning of the considered time period. A decade ago, the speed of interconnection gave the major contribution to α_{total}; enhancing it drastically in the past few years, increased the efficacy. At the same time, because of the stalled single-processor performance, the other technology components only changed marginally. The calculation contribution to α from benchmark HPL remained constant in function of the time, so the quick improvement of the interconnection technology resulted in a quick decrease of α_{total}, and the relative weights of α_{Net} and $\alpha_{Compute}$ reversed. *The decrease in value of*

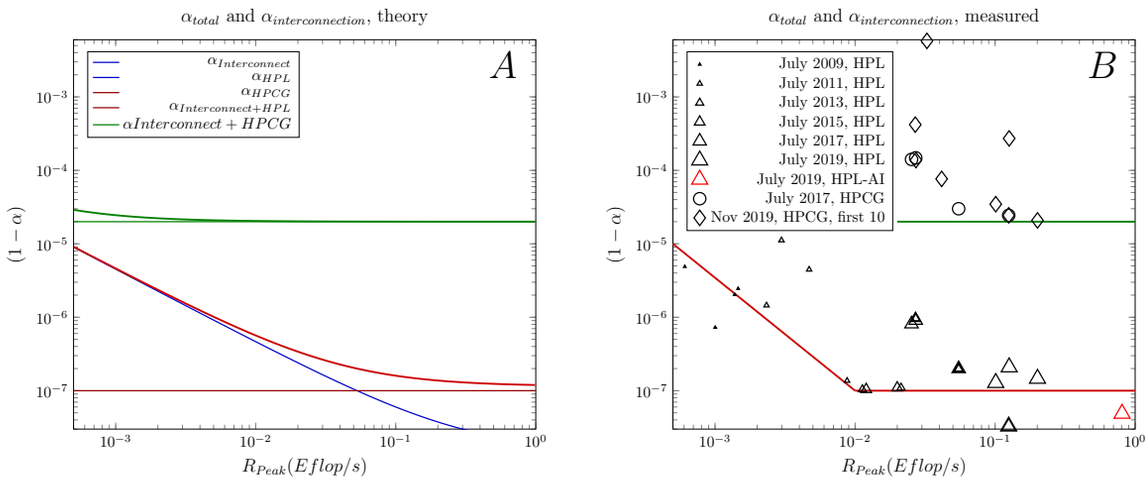

Figure 2.2 : The effect of changing the dominating contribution. The left subfigure shows the theoretical estimation, the right subfigure the corresponding measured data, as derived from the public database TOP500 [20].

$(1-\alpha)$ *can be considered as the result of the decreased contribution from the interconnection.*

However, the total α contribution decreased considerably *only* until α_{Net} reached the order of magnitude of $\alpha_{Compute}$. This occurred in the first 4-5 years of the time span shown in Fig. 2.2.B: the sloping line is due to the enhancement of the interconnection. Then, they changed their role, and the constant contribution due to the calculation started to dominate, i.e. the total α contribution decreased only marginally. As soon as the computing contribution took over the dominating role, the total α_{total} did not decrease any more: all measured data remain above that value of α. Correspondingly, the payload performance (due to the enhanced interconnection) improved only marginally (and due to factors other than the interconnection).

At this point, as a consequence of that the dominating contributor changed, it was noticed that the efficacy of the benchmark HPL and that of the real-life applications started to differ by up to two orders of magnitude. At that point was introduced the new benchmark program HPCG, since "*HPCG is designed to exercise computational and data access patterns that more closely match a different and broad set of important applications*" [22]. Since the major contributor is computing, the different benchmarks contribute differently and since that time "*supercomputers have two different efficiencies*" [14]. Yes, if the dominating α contribution (from the benchmark calculation) is different, then the same computer shows different efficiencies in function of the calculation it runs.

This enhancement of the interconnection has two important consequences. First, that the HPL benchmarks at the beginning of the period measured mostly the contribution of the interconnection, after that they measure mostly the contribution due to the HPL algorithm; see also below. Second, since that time, that the interconnection provides less contribution, than the calculation of the benchmark, enhancing the interconnection contributes only to the *dark performance*, rather than to the *payload performance*.

2.3 The effect of operand length on the performance

Reducing the communication really makes sense, however. The so called *HPL-AI* benchmark uses Mixed Precision[2] [23] rather than Double Precision calculations. This enabled to achieve apparently nearly 3 times better performance gain, that (as correctly stated in the announcement) "*Achieving a 445 petaflops mixed-precision result on HPL (equivalent to our 148.6 petaflops DP result)*", i.e. the peak DP performance did not change.

Unfortunately, this achievement has not much to do with AI: it utilizes the data representation commonly used in AI, but *the achievement comes from accessing less data in memory and using quicker operations on the shorter data rather than reducing the communication intensity.* For AI applications, the limitations remain the same as described above; except that when using Mixed Precision, the efficiency will be better by a factor of 2-3.

Similarly, exchanging data directly between the processing units [24] (without using the global memory) also enhances $(1-\alpha)$ (and payload performance) [25], but

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

it represents a (slightly) different computing paradigm. Only the two mentioned measured data fall below the limiting line of $(1-\alpha)$ in Fig. 2.2.B.

A warning sign is that two of the first ten supercomputers did not provide their HPCG performance and other two used only a small portion of their cores in the HPCG benchmarking. As predicted: "*scaling thus put larger machines at an inherent disadvantage*" [3]. The reason is Eq. (**??**): using all of their cores the achievable performance is not higher (or maybe even lower), only the power consumption is higher. The cloud-like supercomputers have definitely a disadvantage in the HPCG competition: the Ethernet-like operation results in relatively high $(1-\alpha)$ values.

It is expected that when using half precision (FP16), four times less data are transferred and manipulated by the system (for Summit, the measured power consumption data [23] underpin the statement), so it is expected that

$$\alpha_{HPL}^{FP64} = 4 * \alpha_{HPL}^{FP16}$$

However, the performance is only 3 times higher[3] than in the case of using 64-bit (FP64) operands. Given that the measured payload performance directly reflects the sum of all contributions, one can assume that the contributions of the two calculations plus the rest of the contributions define the α values we can conclude from the measurements for supercomputer Summit:

$$1 - \alpha_{HPL}^{FP0} - \alpha_{HPL}^{FP64} = 1.465 * 10^{-7}$$

$$1 - \alpha_{HPL}^{FP0} - \alpha_{HPL}^{FP16} = 0.488 * 10^{-7}$$

where α_{HPL}^{FP0} is the contribution of all parts independent from the floating manipulation. This quantity is a "zero bitlength floating operation" contribution: the supercomputer makes the stuff needed to perform the HPL benchmark, but the actual FP operations are not performed[4]. From this,

$$\alpha_{HPL}^{FP0} = 0.19 * 10^{-7}$$

$$\alpha_{HPL}^{FP16} = 0.33 * 10^{-7}$$

$$\alpha_{HPL}^{FP64} = 1.3 * 10^{-7}$$

$$\alpha_{HPCG}^{FP64} = 208 * 10^{-7}$$

These data directly underpin that the technology is (almost) perfect: the contribution from the benchmark calculation α_{HPCG}^{FP64} is orders of magnitude larger than the contribution from all the rests. Recalling that the benchmark program imitates the behavior (as defined by the resulting α) of real-life programs, one can see that *the contribution from the non-computational actors is about thousand times smaller than the contribution of the computation+communication itself.*

The ironic remark that *'Perhaps supercomputers should just be required to have written in small letters at the bottom on their shiny cabinets: "Object manipulations in this supercomputer run slower than they appear."* [14]' is becoming increasingly relevant.

The imposant numbers about performance of the individual components (including single-processor performance and speed of interconnection) are becoming less relevant when going to the extremes. Given that the largest α contribution today takes its origin in the calculation the supercomputer runs, even the best possible benchmark HPL dominates the performance measurement. Enhancing the other contributions, such as interconnection, result only in marginal enhancement of the performance, i.e. the overwhelming majority of the expenses increases the "dark performance" only. Because of this, *there are as many performance values as many measurement methods, and actually the benchmarks measure how much mathematics/communication the benchmark program does, rather than the supercomputer architecture (provided that all components deliver the technically achievable best parameters).*

As it can be concluded from that the many-processor performance *has* a maximum, depending on the effective parallelization, and that the different workflows result in different effective parallelization, "*Two Different Top500 Supercomputing Benchmarks Show Two Different Top Supercomputers*" [14].

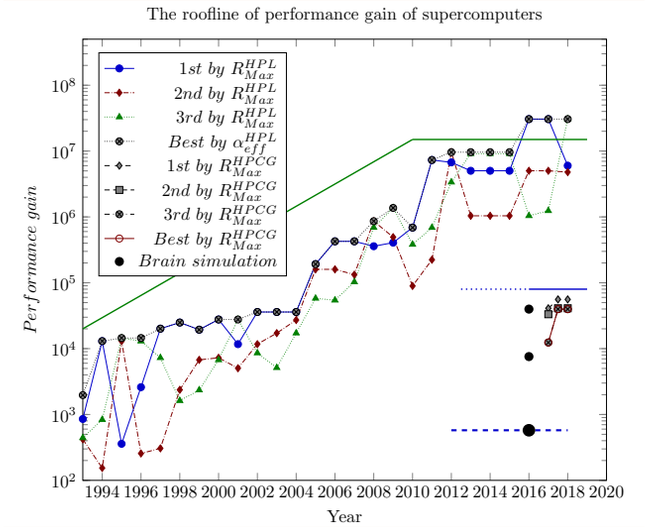

Figure 2.3 : The "rooflines" of supercomputer performance for three different workflows.

As discussed above and the theoretical discussion is illustrated in Fig. 2.1, the different workflows really contribute differently and they result in different performance gain rooflines [26] (this expresses the resulting performance without the single processor performance). The measured values are shown in Fig. 2.3 for the commonly used benchmark programs HPL and HPCG. The third roofline level is concluded from the brain simulation measurement [27], so it is subject of uncertainty. The roofline values shall be compared to the theoretically derived dependencies demonstrated in Fig. 2.1. The fact that the theoretical diagram lines consider pure and consequently calculated performance values, while

[3] https://blogs.nvidia.com/blog/2019/06/17/hpc-ai-performance-record-summit/

[4] The role of α_{HPL}^{FP0} is akin to execution time of the "empty loop" in programming.

the measured values may contain "foreign" contributions (such as the contribution of the interconnection discussed above) must be kept in mind.

2.4 The performance of brain simulation and AI

Today we live in the age of artificial intelligence and machine learning; from small startups to HW or SW giants everyone wants to build machine intelligence chips, applications, etc. The task, however, is hard: not only because of the size of the problem: the technology one can utilize (and the paradigm it is based upon) strongly degrades the chances to succeed efficiently. The general principles are of course well known, and the AI systems work more or less as expected for simple tasks, but on large systems they show up miserably small efficacy (extremely high learning times) [21].

There are, of course, very enhanced solutions, but their technical implementation is "top secret". Because of this, in this section the "experimental data" refer to the published data of brain simulation [27, 28], where also enough implementation details are known. Although the two cases are quantitatively different, the common feature of them is that as we approach the extremes with the size of the computing units, the nonlinearity of the scaling becomes more and more obvious.

2.4.1 Supercomputer efficiency in terms of AI

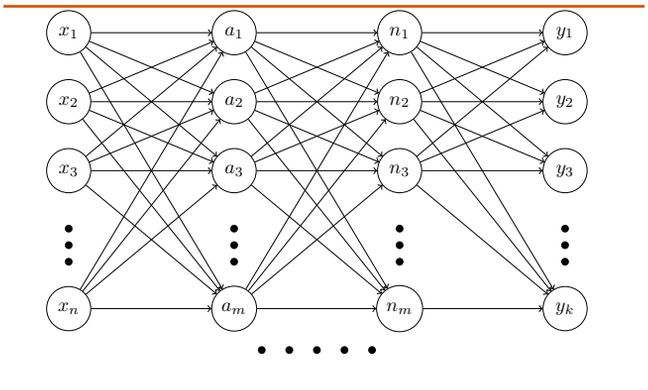

Figure 2.4 : The communication density of the AI workload

Recall the communication density of the AI workflow, shown in Fig. 2.4 (formerly shown as subfigure of Fig. 2.1). The life begins in several input channels (rather than one as in the HPL and HPCG cases) that would be advantageous. However, the values must be communicated to *all* nodes in the top hidden layer: the more input nodes and the more nodes in the hidden layer(s), the many *times* more communication is required for the operation. The same happens also when the first hidden layer communicates data to the second one; except that here *the square of the number of the nodes* is to be used as a weight factor of communication.

Initially the n input nodes issue messages, each one m messages (queuing#1) to the nodes in the first hidden layer, i.e. altogether nm messages. If a commonly used shared bus is utilized to transfer the messages, these nm messages must be queued (queuing#2). Also, every single node in the hidden layer receives (and processes) m input messages (queuing#3). Between the hidden layers the same is repeated (maybe several times) with mm messages, and finally km messages are sent to the output nodes. In all cases queuing 3 times.

To make a fair comparison with benchmarks HPL and $HPCG$, let us assume one input and one output node. In this case the AI execution time is $O(h \times m^2)$, provided that h hidden layers are implemented. (Here it was assumed that the messaging mechanism between layers is independent from each other. It is not so if they share a global bus. [5])

For a numerical example: let us assume that in the supercomputers 1M cores are used, and in the AI network 1K nodes are present in the hidden layers, and only one input and output nodes are used. In that case all execution times are $O(1M)$ (again, the amount of calculation is strongly different, so the scaling can be compared, but not the execution times). This communication intensity explains why in Fig. 3 the $HPCG$ "roofline" falls hundreds of times lower than that of the HPL: the increased communication need strongly decreases the achievable performance gain.

Notice that the number *calculation* operations increases with m, while the number of *communication* operations with m^2. In other words: the more nodes in the hidden layers, the higher is the communication intensity (communication/calculation ratio) and because of this, the lower is the efficiency of the system. Recall, that since the AI nodes perform simple calculations compared to the functionality of the supercomputer benchmarks, the communication/calculation ratio is much higher, making the efficacy even worse.

The massively "bursty" nature of the data (the different nodes of the layer want to use the communication at the same moment) also makes the case harder. The commonly used global bus is overloaded with messages. The possibility for wired point-to-point communication is obviously limited; but deploying them at least for the inter-layer communication buses can help a lot.

The communication circuits receive the task to send the data to N other nodes. The calculation and communication are *ab ovo* sequential, and the communication channel can only transfer one data value at a time. What is worse, bus arbitration, addressing, latency, etc. prolong the transfer time (and in his way decreases efficacy of the system).

2.4.2 Accelerating supercomputer using GPGPU

[5] *"The idea of using the popular shared bus to implement the communication medium is no longer acceptable, mainly due to its high contention."* [29]

Bibliography

[1] S. H. Fuller and L. I. Millett, Eds., *The Future of Computing Performance: Game Over or Next Level?* National Academies Press, Washington, 2011.

[2] 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.

[3] J. P. Singh, J. L. Hennessy, and A. Gupta, "Scaling parallel programs for multiprocessors: Methodology and examples," *Computer*, vol. 26, no. 7, pp. 42–50, Jul. 1993.

[4] J. Végh, J. Vásárhelyi, and D. Drótos, "The performance wall of large parallel computing systems," in *Lecture Notes in Networks and Systems 68*. Springer, 2019, pp. 224–237.

[5] J. Végh, "The performance wall of parallelized sequential computing: the roofline of supercomputer performance gain," *Parallel Computing*, vol. in review, p. http://arxiv.org/abs/1908.02280, 2019.

[6] J. Végh and A. Tisan, "The need for modern computing paradigm: analogies with classic versus modern physics," in *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2019, p. In print. [Online]. Available: http://arxiv.org/abs/1908.02651

[7] Liao, Xiang-ke and Lu, Kai and Yang, Canqun and Li, Jin-wen and Yuan, Yuan and Lai, Ming-che and Huang, Li-bo and Lu, Pingjing and Fang, Jian-bin and Ren, Jing and Shen, Jie, "Moving from exascale to zettascale computing: challenges and techniques," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 10, p. 1236–1244, Oct 2018. [Online]. Available: https://doi.org/10.1631/FITEE.1800494

[8] M. Feldman, "Exascale Is Not Your Grandfather's HPC," https://www.nextplatform.com/2019/10/22/exascale-is-not-your-grandfathers-hpc/, 2019.

[9] US Government NSA and DOE, "A Report from the NSA-DOE Technical Meeting on High Performance Computing," https://www.nitrd.gov/nitrdgroups/images/b/b4/NSA_DOE_HPC_TechMeetingReport.pdf, December 2016.

[10] R. F. Service, "Design for U.S. exascale computer takes shape," *Science*, vol. 359, pp. 617–618, 2018.

[11] European Commission, "Implementation of the Action Plan for the European High-Performance Computing strategy," http://ec.europa.eu/newsroom/dae/document.cfm?doc_id=15269, 2016.

[12] Extremtech, " Japan Tests Silicon for Exascale Computing in 2021." https://www.extremetech.com/computing/272558-japan-tests-silicon-for-exascale-computing-in-2021, 2018.

[13] H. Simon, "Why we need Exascale and why we won't get there by 2020," in *Exascale Radioastronomy Meeting*, ser. AASCTS2, 2014. [Online]. Available: https://www.researchgate.net/publication/261879110_Why_we_need_Exascale_and_why_we_won't_get_there_by_2020

[14] 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.

[15] 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-sys 2017.

[16] ——, "DOE Witholds Details of First Exascale Supercomputer, Even as it Solicits Researchers to Apply for Early Access," https://www.top500.org/news/doe-witholds-details-of-first-exascale-supercomputer-even-as-it-so 2018.

[17] US DOE, "The Opportunities and Challenges of Exascale Computing," https://science.energy.gov/~/media/ascr/ascac/pdf/reports/Exascale_subcommittee_report.pdf, 2010.

[18] J. Dongarra, "The Global Race for Exascale High Performance Computing," http://ec.europa.eu/newsroom/document.cfm?doc_id=45647, 2017.

[19] Top500.org, "The race to exascale," https://sciencenode.org/feature/the-race-to-exascale.php, 2018.

[20] TOP500.org, "The top 500 supercomputers," https://www.top500.org/, 2019.

[21] J. Végh, *How deep the machine learning can be*, ser. A Closer Look at Convolutional Neural Networks. Nova, 2020, p. In press.

[22] HPCG Benchmark, "Hpcg benchmark," http://www.hpcg-benchmark.org/, 2016.

[23] A. Haidar, S. Tomov, J. Dongarra, and 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*, ser. SC '18. IEEE Press, 2018, pp. 47:1–47:11.

[24] F. Zheng, H.-L. Li, H. Lv, F. Guo, X.-H. Xu, and X.-H. Xie, "Cooperative computing techniques for a deeply fused and heterogeneous many-core processor architecture," *Journal of Computer Science and Technology*, vol. 30, no. 1, pp. 145–162, Jan 2015.

[25] Y. Ao, C. Yang, F. Liu, W. Yin, L. Jiang, and Q. Sun, "Performance Optimization of the HPCG Benchmark on the Sunway TaihuLight Supercomputer," *ACM Trans. Archit. Code Optim.*, vol. 15, no. 1, pp. 11:1–11:20, Mar. 2018.

[26] S. Williams, A. Waterman, and D. Patterson, "Roofline: An insightful visual performance model for multicore architectures," *Commun. ACM*, vol. 52, no. 4, pp. 65–76, Apr. 2009.

[27] S. J. van Albada, A. G. Rowley, J. Senk, M. Hopkins, M. Schmidt, A. B. Stokes, D. R. Lester, M. Diesmann, and 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*, vol. 12, p. 291, 2018.

[28] S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, and A. D. Brown, "Overview of the SpiNNaker System Architecture," *IEEE Transactions on Computers*, vol. 62, no. 12, pp. 2454–2467, 2013.

[29] L. de Macedo Mourelle, N. Nedjah, and F. G. Pessanha, *Reconfigurable and Adaptive Computing: Theory and Applications*. CRC press, 2016, ch. 5: Interprocess Communication via Crossbar for Shared Memory Systems-on-chip.

List of Figures

The stalling of the single-processor performance about two decades ago, accompanied with the explosion-like growing demand on computing, put under pressure all fields where some kind of performance increase can be expected. Those fields include researching different materials, developing more or less different computing principles, developing interconnections with higher speed, developing and combining different kinds of accelerators, introducing reconfiguration both in computing and in the interconnection, etc. During this, achieving higher numbers describing the new developments is a natural goal, both from engineering and marketing point of view. However, as all engineering solutions at least approached their limitations, enabled by laws of nature, the segregated optimizations started to block each other.
The present booklet attempts to put the base terms in order. It starts from the very beginning, and brushes up principles known already decades ago. Some issues have been pointed out decades ago and could be validly neglected up to now, but the respectable technical development caused the old issues to return in a technically different form. The booklet

‘Challenges and limitations of High Performance Computing ’

wants to provide a systematic review of the basic ideas, merits, procedures, conventions, etc. It provides a good starting point for understanding the performance limitations, the understanding some mystic phenomena. Through providing a solid understanding of the principles of the parallelized sequential processing, the careful reader can attain a solid background for understanding the operation and performance limitations of the infinite variety of such systems. The principles are carefully underpinned by publicly available data.