Statistical considerations on limitations of supercomputers

János Végh

Received: date / Accepted: date

Abstract Supercomputer building is a many scene, many authors game, comprising a lot of different technologies, manufacturers and ideas. Checking data available in the public database in a systematic way, some general tendencies and limitations can be concluded, both for the past and the future. The feasibility of building exa-scale computers as well as their limitations and utilization are also discussed. The statistical considerations provide a strong support for the conclusions.

Keywords Supercomputer · efficiency · Limits

1 Introduction

For now, supercomputing has a quarter of century history and a well-documented and verified database [5] on their architectural and performance data. The huge variety of solutions and ideas does not enlighten drawing conclusions and especially making forecasts for the future of supercomputing.

In section 2 Amdahl’s law is reconsidered, interpreting it for the modern computing architectures, with keeping an eye on measurability. Choosing the right merit [10] of their characteristics and utilizing a large number of reliable measured data [5], clear conclusions are drawn in section 3. After validating the method, some predictions are made through extrapolating the tendencies for the near future in section 4.

Project no. 125547 has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the K funding scheme.

J. Végh
University of Miskolc
Tel.: +36-46-565-111/1753
E-mail: J.Vegh@uni-miskolc.hu
2 Supercomputers and Amdahl’s law

Amdahl’s law [1] on the joint performance of parallelly working systems is a very basic law of computing which seems to be nearly forgotten in the field of supercomputing. As taught in introductory courses on parallel processing, some fraction of the computing job cannot be parallelized (i.e. cannot be distributed among the parallelly working units), and this fraction limits the achievable resulting computing performance.

Although Amdahl only wanted to draw the attention to that the so called Single-Processor Approach introduces some serious limitations on computing performance (especially when large number of processors is utilized in the system of parallelly working processors), his successors formulated his idea (commonly known as Amdahl’s law) differently. A common misconception is to assume that Amdahl’s law is valid for software only and that parallelizable fraction \( \alpha \) contains something like ratio of numbers of the corresponding instructions to the respective total number.

Amdahl’s law is much more general, and is actually used on many different fields [4]. If Amdahl’s law is interpreted correctly: for the time needed for some activity rather than for some fraction of the code, it should describe also performance and limits of operation of supercomputers. Even, supercomputers are an excellent playground to check validity of Amdahl’s law in the case of extremely large number of processors.

2.1 Terms in Amdahl’s law

First the notations used in [10] are introduced and a summary of the ideas explained and illustrated in details there is given. If \( \alpha \) stands for the time fraction of activity that can be outsourced to several parallelly working processing units, all the rests, \((1 - \alpha)\) fraction, independently of their origin, fall into the category of non-parallelizable activity and (as discussed by Amdahl) appear as if they were sequential-only activity. If the parallelizable fraction is distributed among \( k \) processing units, the speedup \( S \) which can be achieved is

\[
S^{-1} = (1 - \alpha) + \alpha/k
\]

The speedup multiplied with the \( P \) absolute performance of one processor, the (apparent) resulting performance is given as

\[
P_{\text{Max}} = P \frac{1}{(1 - \alpha)}
\]

This is a theoretical upper limit for the performance (also of a supercomputer) which can only be achieved in idealistic case, as discussed in [10]. This usually cannot be computed in advance, because \( \alpha \) is not known in advance. However, on a ”black box” supercomputer one can measure \( R_{\text{Max}} \) and it is also known that \( R_{\text{Peak}} = kP \). Since
\[
S = \frac{(1 - \alpha) + \alpha}{(1 - \alpha) + \alpha/k} = \frac{k}{k(1 - \alpha) + \alpha}
\]

and the efficiency
\[
E = \frac{S}{k} = \frac{1}{k(1 - \alpha) + \alpha} = \frac{R_{Max}}{R_{Peak}}
\]

the measured payload performance provides information also on the "effective parallelism". That is, only a fraction of nominal performance can be utilized as payload performance, the rest remains a kind of "dark performance". One can easily express the "effective parallelization" \( \alpha_{eff} \) from the measured efficiency as

\[
\alpha_{eff} = \frac{k \cdot S - 1}{k - 1} \cdot \frac{1}{S}
\]

or equivalently

\[
\alpha_{eff} = \frac{Ek - 1}{E(k - 1)}
\]

Using measured performance values published for supercomputers [5], \( \alpha_{eff} \) values for the supercomputer configurations can be calculated, see Fig 2. Notice that for a given configuration \( \alpha_{eff} \) depends on \( k \).

2.2 A simple model for supercomputing

To understand the meaning of the values derived in this way, a simple model shown in Fig. 1 should be derived. Although the model is empirical rather than technical, with slightly extending it and giving technical meaning to its terms, it can easily be converted to a technical model. Also note that here no communication is assumed between the parallelly working units, but the model can be trivially extended to the case when the parallelly working processors communicate (explicitly or implicitly, like sharing some resource). The model assumes that several components contribute to the total execution time, as simple sum of either some components or the largest of some components.

The access time is usually small: whether the time is measured on the parallelized system or outside of it, one must compensate for its contribution (in the case of supercomputers, it is usually negligible). The contribution of the executed program \( \alpha_{SW} \) depends heavily on the nature of the program.

The contributions due to OS and HW are tightly connected, so it is not easy to separate them without making dedicated measurements; at this level their joint contribution will be handled as \( \alpha_{HW+OS} \). Within that contributions there are some parts which may become critical, like the looping delay \( T_x \) due to utilizing extremely large number of processors or the propagation delay \( PD_{xx} \) due to having large physical size of the supercomputer; they will be mentioned separately, and in the technical model they shall be handled
specifically. The time scale shown in the figure serves only for illustration, the actual contributions will strongly vary with the actual conditions.

From the figure the meaning of $\alpha_{eff}$ can be easily identified as Payload/Total. Also, the reasons of "dark performance" can be identified: the ready-to-fire processing units are simply idle. The common mistake of handling the access time improperly can falsify the conclusions, although in the case of long measurement times this effect can be neglected.

3 Performance and architecture checks

The available, rigorously validated database [5] enables to draw reliable conclusions, although the variety of sources of components, different technologies and ideas as well as the interplay of different factors cause a considerable scatter and requires extremely careful analysis.

3.1 Supercomputer timeline

As a quick test, Equ. (6) can be applied to data from [5], see Fig. 2. As shown, supercomputer history is about development of effective parallelism,
and Amdahl’s law formulated by Equ. (6) is actually what Moore’s law is for the size of electronic components. (The effect of Moore’s law is eliminated when calculating $\frac{R_{\text{max}}}{R_{\text{peak}}}$. ) To understand the behavior of the trend line, just recall Equ. (4): to increase the absolute performance, more processors shall be included, and to provide reasonable efficiency, the value of $(1 - \alpha)$ must be properly reduced.

3.2 Single-processor performance

As suggested by Equ. (2), the trivial way to increase the absolute performance of a supercomputer is to increase the single-processor performance of its processors. Since the single processor performance has reached its limitations, some kind of accelerators are frequently used for this goal. Fig. 3 shows how utilizing accelerators influences ranking of supercomputers.

As the left side of the figure depicts, GPU accelerated processors really increase performance of processors by a factor of 2-2.5, however this increased performance is about 40 times lower than the nominal performance of the GPU accelerator. The right side of the figure, however, discovers, that the
Figure 3 shows the correlation of performance of processors using accelerators and effective parallelism with ranking. In 2017, the performance of accelerated processors is nearly an order of magnitude worse than that of the non-accelerated processors, i.e., the resulting efficiency is worse than in the case of utilizing unaccelerated processors; this can be a definite disadvantage when GPUs are used in systems with extremely large numbers of processors. This makes at least questionable whether it is worth utilizing GPUs in supercomputers.

As the left figure shows, neither type of processor shows correlation between ranking of supercomputer and type of acceleration. Essentially the same is confirmed by the right side of the figure: the effective parallelization raises with the ranking position, and the slope is the same for any kind of acceleration.

3.3 Number of processors

Since the resulting performance depends both on the number of processors and effective parallelization, both quantities are correlated in Figure 4. As expected, in TOP50 the higher the ranking position is, the higher is the required number of processors.
of processors in the configuration, and as outlined above, the more processors, the lower \((1 - \alpha_{\text{eff}})\) is required (provided that the same efficiency is targeted).

In TOP10, the slope of the regression line sharply changes in the left figure, showing the strong competition for the better ranking position. On the right figure, TOP10 data points provide the same slope as TOP50 data points, demonstrating that to produce a reasonable efficiency, the increasing number of cores must be accompanied with a proper decrease in value of \((1 - \alpha_{\text{eff}})\), as expected from Equ. (4), furthermore, that to achieve a good ranking a good value of \((1 - \alpha_{\text{eff}})\) must be provided.

3.4 Architectural solution

As shown in Fig. 5, with time the composition of the type of the architectural solutions as well as the value of parallelization efficiency have considerably changed. However, in neither time the architectural solution caused significant difference compared to the other one; the slope is the same for both solutions, in both years. At the same time, \((1 - \alpha_{\text{eff}})\) has improved independently and considerably.

3.5 Benchmarking

According to the model, the SW (including benchmark programs) also contributes to the measured \((1 - \alpha_{\text{eff}})\), and its contribution is different for the different programs. Fortunately, since the beginnings the same benchmark program HPL is used to qualify supercomputers. Fortunately, HPL contributes only a low amount of overhead activity, so it can be used as the best estimator for describing the HW+OS environment of a supercomputer. Unfortunately, most real-life applications have much higher SW contribution, so recently benchmark HPCG has been suggested to imitate their behavior. Fig. 6 shows
how \((1 - \alpha_{eff})\) correlates with number of processing units, for the two mentioned benchmark programs. The behavior is quite similar on the left and right figures, but the value differs by about two orders of magnitude. Because of this, it can be safely stated that HPCG measures the behavior of the program on the architecture rather than the architecture \((\alpha_{HW+OS})\) itself. Notice also, how the relative \(\alpha_{eff}\) measured values change between the two benchmarks.

3.6 Ranking

For ranking, different merits can be used. One possible approach is to measure \(R_{Max}\), using benchmarks either HPL or HPCG. Of course, these two measurements lead to different rankings. Another possible approach is to rank by \(\alpha_{eff}\), measured with either of the two benchmarks. Fig. 7 compares how these two measurements correlate with each other. Data points on the left figure show no correlation, strongly supporting the statement that HPL measures the architecture, HPCG measures the SW contribution, and so they are not correlated at all. In contrast, the two \((1 - \alpha_{eff})\) values strongly correlate, al-
Statistical considerations of supercomputers

3.7 Efficiency

Although $\frac{R_{\text{Max}}}{R_{\text{Peak}}}$ measured with benchmark HPL is an important feature of the HW+OS assembly, it is a reliable merit only when $\alpha_{eef}^{\text{SW}}$ is less than $\alpha_{eef}^{\text{HW+OS}}$. As long as HPL is used to rank supercomputers, architects keep efficiency around 0.73 (see the left side of Fig. 8); although in the case of Taihulight [3] (because of the extremely large number of processors) it is only possible through using special HW units MPE. In the case of real-life programs, however, $\alpha_{eef}^{\text{SW}}$ is about two orders of magnitude higher than $\alpha_{eef}^{\text{HW+OS}}$ (see Fig. 6), so in that case the efficiency steeply decreases as the number of processors increases, see the right side of Fig. 8. Notice that the measured efficiency of Taihulight changes drastically: utilizing MPEs decreases $\alpha_{eef}^{\text{HW+OS}}$ which is considerable in the case of benchmark HPL, but in the case of HPCG $\alpha_{eef}^{\text{SW}}$ dominates, so the effect of MPEs are negligible.

It is important to notice, that $\alpha_{eef}$ sensitively changes with number of processors (see Fig. 6), while efficiency does not.

4 Future of supercomputers

The race for achieving Eflop/s performance is continuing. From the presently existing implementations some conclusions can be already drawn. From Fig. 8 one can conclude for the near future an optimistic single processor performance $P$ of value 50 Gflop/s. From Equ. (2) one shall conclude that for achieving 1 Eflop/s payload performance $(1 - \alpha)$ of value $5 \times 10^{-8}$ effective parallelization should be achieved. Compare the values to the case of Taihulight: $P = 11.8$ Gflop/s and $(1 - \alpha_{eef}) = 3.3 \times 10^{-8}$: the limiting top performance is about
4.1 Extrapolating the empirical parameters

One way to derive more accurate estimations for the performance limitations is to utilize the empirical model. Keeping all other parameters constant, the number of processors can be virtually changed. Fig. 9 depicts how the virtual versions of present TOP10 supercomputers will achieve the nominal 1 Eflops/s.

To provide a feeling, how the effective parallelization influences the measurable performance, Fig. 10 depicts what payload performance could be measured on that virtual Taihulight when running benchmark programs having different \((1 - \alpha_{eff})\). This could be crucial when running real-life programs, the need for communication between processing units arises, and especially when they must share some resource.
4.2 Introducing a technical model

Based on the empirical model, some technical meaning can be attributed to the $\alpha_X^{\text{eff}}$ components. Although without considering the technical specifications in details, only the order of magnitude of the contributions can be estimated, it is accurate enough to draw some qualitative conclusions, especially of the limiting values of the different contributions. The total $(1 - \alpha_{\text{eff}})$ is about $3.3 \times 10^{-8}$, so one upper limiting value is known in advance: $(1 - \alpha_{\text{eff}}^\text{sw})$ cannot be higher than that value.

To turn our empirical model to a technical one, data published in [2] are used. The 13,298 seconds benchmark runtime on the 1.45 GHz processors means $2 \times 10^{13}$ clock periods. The absolutely necessary non-parallelizable activity is to start and stop the calculation. If starting and stopping a program on a zero-sized supercomputer without OS could be done in 2 clock periods, then the absolute limit for $(1 - \alpha)$ would be $10^{-13}$.

From the model follows that two of the contributions can be critical when building "big" supercomputers. The OS looping contribution increases linearly with number of processors, and PD contribution linearly increases with the physical size of the computer. As depicted in Fig. [1], these contributions can be
combined in such a way that small contributions from OS are linked to large contributions from PD and vice versa. Anyhow, these two contributions will also provide an upper bound to the absolute performance of supercomputers. Since any of them can be quite small, the limit will be the lower of the two individual bounds.

For considering PD bound, let us consider a cca. 100 meter sized computer having 1 GHz cores: the signal round trip time is cca. $10^{-6}$ seconds, or $10^3$ clock periods. When using high speed internal network, the message length has no considerable contribution and a network message exchange time (including operating time of HW) can be estimated to be of length $10^{-5}$ seconds, or $10^4$ clock periods. So, the absolute limit for $(1 - \alpha)$ of a supercomputer with realistic size, but no operating system, is $10^{-9}$.

An operating system must, however, be used. If one considers context change with its consumed $10^4$ cycles [6], the absolute limit is cca. $10^{-9}$, on a zero-sized supercomputer. In addition, all cores must be manipulated through the system call, which contribution increases linearly with the number of cores and contribution from OS can be dominant at high number of cores.

For the 10 million processors of Taihulight, at least $10^5$ clock cycles must be used. Even when parameters can be passed in one clock cycle, for 10M parameter passings the absolute bound due to OS looping contribution would be in the range of $10^{-6}$. It is surely the dominating contribution for such large number of processors. Is then something wrong with the model? The measurable $(1 - \alpha_{eff})$ for Taihulight must not be lower than any of the contributions, including the one due to looping in OS.

At this point one can understand the role of modularization some supercomputers utilize. In the case of Taihulight, from the 260 cores 4 serves as management processing element (MPE) [3,2], so only the processors (or core groups) rather than individual cores shall be addressed, the rest will be organized by MPEs. This trick reduces the absolute computing performance of a processor only by 2% on one side, but on the other side reduces loop count by about two orders of magnitude, decreasing contribution $(1 - \alpha_{eff}^{OS})$ by two orders of magnitude; in this way enabling to achieve effective parallelization of value $1 \times 10^{-8}$.

4.3 Changing the computing model

Introducing MPEs decreased $(1 - \alpha_{eff})$ and enabled to build supercomputer with 10M processors and at the same time reasonable efficiency. Using MPEs, however, violates computing paradigms: those ”more equal” processors know that some other processors exist. As the above analysis demonstrated, (among others) the presently used Single-Processor Approach (SPA), that is the computing paradigms itself, is a limiting factor in building larger supercomputers. The Explicitly Many-Processor Approach (EMPA) [7,9] enables to use forking-like handling of starting processing units, and in this way the OS looping contribution can be reduced from 10M cycles to 24, in this way eliminating
the most limiting factor from the way of building supercomputers from even more processors.

This is not against Amdahl’s law: if the processors can cooperate, in Eqn. (11) \( f(k) \) should be used instead of \( k \), and the nature of \( f(k) \) enables such drastic changes in the behavior of parallelly working systems. It looks like Amdahl was right with saying: “the organization of a single computer has reached its limits and that truly significant advances can be made only by interconnection of a multiplicity of computers in such a manner as to permit cooperative solution”.

After introducing EMPA, the context change becomes the largest contribution to \( \alpha_{\text{OS}}^{\text{eff}} \). Through introducing a reasonable layering [8], this contribution can be lowered by orders of magnitude; making the propagation time PD the dominating contribution. It can be reduced by decreasing the physical size of supercomputers, say using 3D arrangement. Making all changes mentioned, in principle even Zflop/s supercomputers can be built. However, without making all those changes, even Eflop/s cannot be achieved.

5 Conclusions

The present technical implementations of supercomputers practically reached their technical limits. The reliable database of parameters of supercomputers can be used to draw reliable statistical conclusions on some parameters and limitations of supercomputers. Although the extrapolation of the tendencies enables to make predictions for some future configurations, the careful analysis reveals that the presently exclusively used Single-Processor Approach really forms an upper bound for the performance of supercomputers. The experienced difficulties in building ever-larger supercomputers are of principal rather than technical nature.

References

1. Amdahl, G.M.: Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities. In: AFIPS Conference Proceedings, vol. 30, pp. 483–485 (1967). DOI 10.1145/1465482.1465560
2. Dongarra, J.: Report on the Sunway TaihuLight System. Tech. Rep. Tech Report UT-ECECS-16-742, University of Tennessee Department of Electrical Engineering and Computer Science (2016)
3. Fu, H., Liao, J., Yang, J., Wang, L., Song, Z., Huang, X., Yang, C., Xue, W., Liu, F., Qiao, F., Zhao, W., Yin, X., Hou, C., Zhang, C., Ge, W., Zhang, J., Wang, Y., Zhou, C., Yang, G.: The Sunway TaihuLight supercomputer: system and applications. Science China Information Sciences 59(7), 1–16 (2016). DOI 10.1007/s11432-016-5588-7. URL http://dx.doi.org/10.1007/s11432-016-5588-7
4. Krishnaprasad, S.: Uses and Abuses of Amdahl’s Law. J. Comput. Sci. Coll. 17(2), 288–293 (2001). URL http://dl.acm.org/citation.cfm?id=775359.775386
5. TOP500.org: The top 500 supercomputers. https://www.top500.org/ (2016)
6. Tsafrir, D.: 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). DOI 10.1145/1281700.1281704. URL http://doi.acm.org/10.1145/1281700.1281704
7. Végh, J.: EMPAthY86: A cycle accurate simulator for Explicitly Many-Processor Approach (EMPA) computer. (2016). DOI 10.5281/zenodo.58063). URL https://github.com/jvegh/EMPAthY86
8. Végh, J.: Do we need cross layering activities or reasonable layering in computing systems? IEEE Design & Test p. submitted (2017)
9. Végh, J.: Renewing computing paradigms for more efficient parallelization of single-threads, p. in print. Advances in Parallel Computing. IOS Press (2017)
10. Végh, J., Molnár, P.: How to measure perfectness of parallelization in hardware/software systems. In: 18th Internat. Carpathian Control Conf. ICC, p. paper 121 (2017)