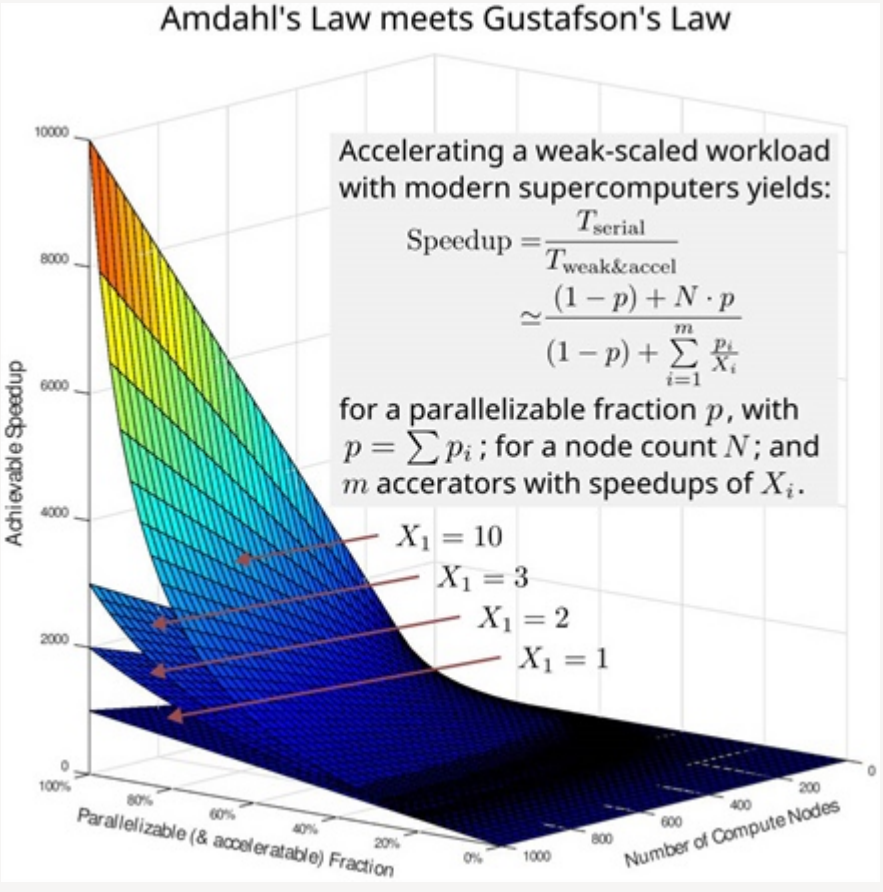
Research | September 06, 2022

Life after Fugaku: What Have We Learned and How Do We Proceed as the End of Moore's Law Approaches?

By Satoshi Matsuoka and Jens Domke

High-performance computing (HPC) has seen great success over the past several decades. It rode the wave of persistent transistor shrinking that Gordon Moore observed and focused on finding technological solutions to the ever-growing need for supercomputing performance. Various types of accelerators made their way into HPC systems, including floating-point accelerators, vector coprocessors, and general-purpose graphics processing units (GPUs) [4]. More recently, the community has been experimenting with matrix accelerators, field-programmable gate arrays, and quantum computers [9]. These devices follow an evolutionary trend wherein the best survive — not just as coprocessors that are connected by input/output busses like Peripheral Component Interconnect Express or integrated as system-on-chips (SoCs), but also as special function units inside or close to the central processing unit (CPU). Famous examples include 64-bit floating point units, Advanced Vector Extensions and cryptography instruction sets, and accelerated processing units. All of these devices and features have helped to accelerate workloads under Amdahl's law [1].

The supercomputing community typically regards Amdahl's law as the strong-scaling law under which the use of more compute nodes accelerates a given parallelizable fraction of the workload and reduces the time to solution. But this law also applies to accelerators, and the potential speedup is bound by the ratio of accelerated and non-accelerable fractions of the algorithm. Furthermore, a second



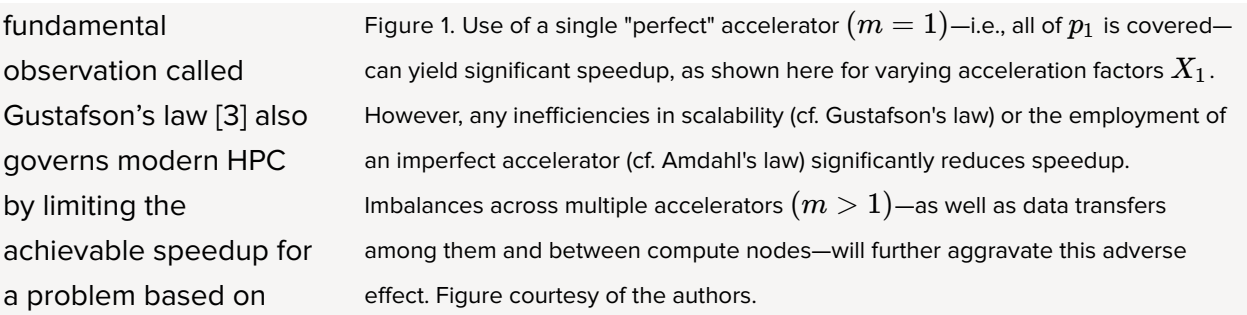


Figure 1. Use of a single "perfect" accelerator $(m = 1)$—i.e., all of p_1 is covered—can yield significant speedup, as shown here for varying acceleration factors X_1. However, any inefficiencies in scalability (cf. Gustafson's law) or the employment of an imperfect accelerator (cf. Amdahl's law) significantly reduces speedup. Imbalances across multiple accelerators $(m > 1)$—as well as data transfers among them and between compute nodes—will further aggravate this adverse effect. Figure courtesy of the authors.

fundamental observation called Gustafson's law [3] also governs modern HPC by limiting the achievable speedup for a problem based on how well the parallelizable or accelerable fraction can be weak scaled onto many nodes; one accomplishes this by increasing the overall workload and maintaining a constant amount of work per node. Weak scaling overcomes a problem's bottlenecks—slowdowns due to communication issues with the interconnection network or inherent imbalances in the workload distribution—that become evident when one strong-scales to the same number of compute nodes. Fully understanding the implications of Amdahl and Gustafson's laws is necessary to advance the future of supercomputing (see Figure 1).

HPC is currently at a crossroads, and the community is increasingly focusing on domain-specific accelerators and extreme heterogeneity as possible solutions to the imminent technological slowdown [10]. Yet accelerators are only a means to an end. Let us consider a few aspects of accelerated computing, scalability, and high-end supercomputing [6]. We can split the current and imaginable HPC landscape into four categories. The first is (1) *homogeneous systems with respect to node selection and homogeneous node designs*, which only use general-purpose processors. Supercomputer Fugaku at Japan's Riken Center for Computational Science is a famous example of this principle. The next category consists of (2) *homogeneous systems with heterogeneous nodes*, such as the upcoming Frontier and Aurora supercomputers that host multiple GPUs of the same type alongside the CPU. A third pragmatic configuration is (3) *heterogeneous systems (typically island-based) with homogeneous nodes within partitions*, like NASA's Pleiades supercomputer. Researchers usually deploy this category for convenience—i.e., accessing a common file system—but not to accelerate a single application across the entire supercomputer. Lastly, the fourth category is (4) *extreme heterogeneity in the system design*, with a mix of different nodes (not necessarily in an island enclosure) and a mix of accelerators in each node. One could likewise argue that machine-like designs [2] deserve their own category. But unless someone can solve the HPC-critical latency problem for these systems, we will omit them from further discussion.

From the users' point of view, the perfect tool for performing their research duties and advancing science and society is a uniform, complexity-free system that is similar to daily devices like laptops and smartphones. This reason is also responsible for the success of commercial clouds — their complexity can be hidden because workloads tend to be embarrassingly parallel or elastic in nature. Unfortunately, our football-field-sized supercomputers are naturally bigger than pocket devices. Therefore, we must expose some complexity to achieve scalability of real parallel programs that process massive datasets. The second-best tool is hence one that fits category (1). Our data that pertains to submissions for Fugaku usage indicate that many users rapidly adopt the A64FX CPU *and* Fugaku due to their

ease of use and programming. We have seen unexpectedly high system utilization both during installation [7] and after general operation commenced in 2021, which supports our statement about category (1) systems. Any category thereafter increases the users' efforts. The community has painfully experienced this fact throughout the last decade when migrating to GPU-based category (2) systems, which are now adjacent to (1) since GPUs have also become general-purpose compute engines alongside CPUs.

Still, the lessons from this recent GPU-accelerated approach are twofold. Achieving a balanced use of CPU and GPU for the same workload is extremely difficult to orchestrate, especially with changing workload balances among different nodes during an execution. The trend hence involves having more GPUs per node and more memory on the GPU in order to fit the entire workload without moving it between main memory and GPU memory to ultimately avoid slowdown under Amdahl's law. Scientific programs also tend to use either CPUs or GPUs, but not both at the same time due to Gustafson's law; any imbalance between CPU and GPU will increase the likelihood that the application in question will not scale to many compute nodes.

The second lesson is that the success of any processor or accelerator depends primarily on the software ecosystem, not just the raw hardware performance [5]. We have seen several positive examples—such as x86, Arm, CUDA, and so forth—all of which have very large user bases outside of HPC. Negative examples like Intel Itanium, IBM Cell, and the plethora of recent chips for artificial intelligence (AI) from startups exist as well; these lack software ecosystems to broadly exploit the theoretical peak that they offer.

Another argument for category (1) and (2) systems comes from outside of the HPC field. The economics of scale dictate success in a capitalistic market, and the most prosperous and disruptive companies in the world utilize one simple trick: standardize, mass produce, and replicate — with an emphasis on the latter. A car manufacturer that attempted to customize each car for every consumer would spend more resources and produce products at a much slower pace, ultimately losing out to the competition. Similarly, an army of programmers who try to orchestrate a parallel application onto a sea of heterogeneous compute nodes in a single supercomputer will face an enormous productivity slowdown. This result is not due to the attempt's impossibility or laziness within the community; rather, it occurs because the complexity increases to unsustainable levels.

Extreme heterogeneity has found success in embedded areas such as smartphone SoCs. However, smartphones follow the economics of scale and have huge ecosystems from the standpoint of parallelization and speedup. Additionally, they are only subject to Amdahl's law. Ignoring Gustafson's law commonly leads to the incomplete conclusion that category (4) systems will serve as successful "general purpose" supercomputers. The only exception is a domain-specific system that solely executes one large workflow so that the balancing and ratio of accelerators is predetermined. Any other scientific workload or complex workflow on this system would suffer from both the underutilization of the system's plethora of accelerators as well as imbalances under Gustafson's law, since the problem's imbalances do not match the system's hardware imbalances. Furthermore, even if one assumes that building one domain-

specific category (4) system for a single workflow makes economic sense, such a problem would then have likely reached its weak scalability. As a result, other supercomputers would quickly outclass it and it would transition to even higher customization with application-specific integrated circuits (as we see for customized AI accelerators) [8]. While it is hence theoretically possible to build category (4) systems, such systems will hamper users' productivity, suffer from poor scalability under Gustafson's law, and face poor efficiency due to memory movement overhead among many intra-node accelerators — thereby limiting performance gains under Amdahl's law.

In conclusion, the fundamental principles of accelerated supercomputers are Amdahl's law and Gustafson's law; any scientific workloads that attempt to exploit the system's performance at scale must satisfy both laws. To maximize the acceleration under Amdahl's law, one must perform the dominant processing on the same accelerator since any intra-node data movement will nullify potential performance gains. This action thus requires the same accelerator on every compute node. To maximize the acceleration under Gustafson's law, the parallelizable fraction of the workload must be uniformly load-balanced. The preferable and least complex approach is thus the single program, multiple data technique, which distributes the workload over uniform CPUs or accelerators. In addition, any scalable and efficient scientific application must minimize the general parallelization overhead (e.g., communication and imbalances) as much as possible. Nodes, processors, and accelerators must therefore be tightly coupled. This coupling should preferably occur on chip or on package, since large-distance segregation entails costly (in terms of time and energy) data movement — typically across bottlenecks like on-node and especially off-node interconnection networks.

Acceleration, while temporarily beneficial and performant, is a means to an end and not a purpose in itself for the future of high-end supercomputing. All successful large-scale systems —regardless of acceleration status—must follow the principle of a uniform compute node configuration across the entire HPC system, coupled with robust interconnect to sustain maximum bandwidth among nodes and accelerators. The discussed laws and principles are mathematical and physical in nature and will therefore apply to future systems in the same way that they applied to previous generations of supercomputers; they simply cannot be avoided. HPC providers and users must keep these principles in mind to accelerate science and support the future prosperity of society.

References

[1] Amdahl, G.M. (1967). Validity of the single processor approach to achieving large scale computing capabilities. In *Proceedings of the April 18-20, 1967, spring joint computer conference (AFIPS'67)* (pp. 483-485). Atlantic City, NJ. Association for Computing Machinery.

[2] Faraboschi, P., Keeton, K., Marsland, T., & Milojicic, D. (2015). Beyond processor-centric operating systems. In *15th workshop on hot topics in operating systems (HotOS XV)*. Kartause Ittingen, Switzerland. USENIX Association.

[3] Gustafson, J.L. (1988). Reevaluating Amdahl's law. *Commun. ACM*, *31*(5), 532-533.

[4] Matsuoka, S. (2018). Cambrian explosion of computing and big data in the post-Moore era. In *Proceedings of the 27th international symposium on high-performance parallel and distributed computing (HPDC '18)* (pp. 105). Tempe, AZ.

Association for Computing Machinery.

[5] McInnes, L.C., Heroux, M.A., Draeger, E.W., Siegel, A., Coghlan. S., & Antypas, K. (2021). How community software ecosystems can unlock the potential of exascale computing. *Nat. Comput. Sci.*, *1*, 92-94.

[6] Research Organization for Information Science and Technology. (2019). *Research achievements using the HPCI system including K computer (Vol. I-VI)*. Retrieved from https://www.hpci-office.jp/pages/e_hpci_booklet.

[7] Research Organization for Information Science and Technology. (2021). *Fugaku annual report 2020*. Retrieved from https://www.r-ccs.riken.jp/fugaku/fugaku-annual-reports/2020/index.

[8] Reuther, A., Michaleas, P., Jones, M., Gadepally, V., Samsi, S., & Kepner, J. (2021). AI accelerator survey and trends. In *2021 IEEE high performance extreme computing conference (HPEC)* (pp. 1-9). Institute of Electrical and Electronics Engineers.

[9] Shalf, J. (2020). The future of computing beyond Moore's Law. *Philos. Trans. R. Soc. A: Math. Phys. Eng. Sci.*, *378*(2166).

[10] Vetter, J.S., Brightwell, R., Gokhale, M., McCormick, P., Ross, R., Shalf, J., ... Wilke, J. (2018). *Extreme heterogeneity 2018 - Productive computational science in the era of extreme heterogeneity: Report for DOE ASCR workshop on extreme heterogeneity*. Washington, DC: USDOE Office of Science Advanced Scientific Computing Research.

Satoshi Matsuoka is director of the RIKEN Center for Computational Science in Japan. He previously led the TSUBAME series of supercomputers, currently leads various major supercomputing research projects, and is a major driving force behind the development of the next-generation flagship supercomputer of Japan: Supercomputer Fugaku. His research interests include high-performance systems and software, large-scale resilience, green computing, and convergence of big data and artificial intelligence with high-performance computing. Matsuoka received a Ph.D. in information science from the University of Tokyo. He is a fellow of the Association for Computing Machinery and the Japan Society for Software Science and Technology and a member of the Institute of Electrical and Electronics Engineers.

Jens Domke is the team leader of the Supercomputing Performance Research Team at the RIKEN Center for Computational Science in Japan. He received his Ph.D. in 2017 from the Technische Universität Dresden, where he studied high-performance computing routing algorithms and interconnects. Domke contributed the DFSSSP and Nue routing algorithms to the subnet manager of InfiniBand and built the first large-scale HyperX prototype at the Tokyo Institute of Technology. His research interests include system co-design, performance evaluation, and extrapolation, as well as the modeling, interconnect networks, and optimization of parallel applications and architectures.