nature reviews physics

https://doi.org/10.1038/s42254-024-00750-z

Perspective

Check for updates

The co-evolution of computational physics and high-performance computing

Jack Dongarra[1,2,3] & David Keyes[4,5]

Abstract

High-performance computational physics has been instrumental in advancing scientific research by regularly providing breakthroughs in speed, accuracy and modelling fidelity. This Perspective highlights the contributions of physicists to the development of high-performance computing infrastructure, algorithms and applications from the early days of computing to the exascale era. We recall the pioneering work of Fermi and von Neumann, who set directions and laid foundations for computational science and examine the ongoing impact of physicists in overcoming current challenges in high-performance computing, such as energy consumption and data storage. As we celebrate milestones such as exascale computing and generative artificial intelligence, it is inspiring to recognize the enduring influence of physicists in driving technological innovations and ensuring the future progress of computational science.

Sections

[1]University of Tennessee, Oak Ridge, TN, USA. [2]Oak Ridge National Laboratory, Oak Ridge, TN, USA. [3]University of Manchester, England, UK. [4]Applied Mathematics and Computational Sciences, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia. [5]Applied Physics and Applied Mathematics, Columbia University, New York, NY, USA. e-mail: dongarra@icl.utk.edu; david.keyes@kaust.edu.sa

Introduction

Seventy years ago, Fermi wrote a letter to Avanzi, rector of the University of Pisa, advocating for computational modelling with an emphasis on performance, stating, “Experience shows that the possibility of executing complex calculations with great speed and accuracy quickly creates an enormous demand for these services, which would soon be beyond the capacity of just one machine.” Fermi went on to argue that “An electronic computer would constitute a research instrument from which all science and research activities would profit, in a way that is currently inestimable”[1]. Extrapolating from the world-changing influence of early computers in neutron transport computations at Los Alamos, Fermi foresaw opening up computational access to scientific modellers of all disciplines and training the next generation of scientists in the use of these tools. His advice was followed and a digital computer known as CEP (Calcolatrice Elettronica Pisana, meaning Pisa Electronic Calculator), offering 6.7 Kflop s^{-1} on 4K 36-bit magnetic core words, was installed in Pisa the following year[1]. Fermi foresaw the dawning of computational science although he did not directly contribute to it in terms of the ‘three pillars’ of computational architecture, algorithms or applications. Other physicists, however, pioneered many advances in high-performance computational modelling and simulation. von Neumann contributed directly to all three pillars: architecture (the design of EDVAC, one of the earliest electronic computers, and the ‘von Neumman architecture’)[2], algorithms (linear algebra in floating-point arithmetic)[3] and applications (the modelling of hydrodynamic shocks)[4]. He also devised theory to undergird their use, for example, the ‘von Neumann stability analysis’ of finite difference discretizations of partial differential equations[5]. At Los Alamos, von Neumann used mechanical devices no more powerful than punch cards and solenoids, but even that led to, as Fermi suggested, “an enormous demand” for computational services that as of 2024 shows no sign of being sated.

What began with the Electronic Numerical Integrator and Computer, the first fully electronic, general-purpose computer designed to calculate the ballistics of artillery shells was quickly recognized as an instrument for scientific discovery in many physics applications. As soon as it was completed, von Neumann used it for thermonuclear calculations and it had an important role in the development of Monte Carlo methods. This Perspective highlights roles physicists have in driving advances in high-performance computing (HPC), lest they be forgotten amid our celebrations of exascale and generative artificial intelligence. Our intention is to be inspirational, rather than comprehensive, as we ask: as HPC hits walls of energy and storage, will physicists in pursuit of their own applications again come to the fore with generalizable solutions? (For more details on the history of the synergies of physics and computing, see refs. 6–8, and on general contemporary high-performance computing, see refs. 9–11).

What is HPC?

HPC uses supercomputers to tackle complex problems at scales that – dare we say – not even von Neumann could have envisioned. With parallel architectures and parallel algorithms as the sources of its unparalleled power, HPC has revolutionized diverse applications ranging from science and engineering to finance and health care. Modelling and simulation, big data analytics and machine learning have joined theory, observation and experiment as fundamental modalities of scientific discovery and have largely surpassed them as modalities of engineering design.

Today’s HPC represents the culmination of nearly eight decades of innovation and technological advancement in the digital domain, embodying essentially five key components: processing power, memory, networking, algorithms and software.

Processing power

At the heart of every HPC system lies an array of processing cores, ranging up to ten million in today’s largest systems, each executing computational tasks in parallel. In the aggregate, this parallelism enables systems to deliver today up to 10^{18} operations per second in 64-bit double precision (Fig. 1), or more, depending upon the data type and the structure of the operations. Special units such as tensor cores supplied by various vendors can perform as many as 64 double-precision operations in one cycle, in the form of updating a 4×4 matrix with the product of two other 4×4 matrices. Tensor operations in half-precision execute at nearly 1 Pflop s^{-1} (989 Tflop s^{-1}) on a single NVIDIA H100 SXM[12]. Such rates are close to achievable in many practical problems that use matrix–matrix multiplications, from block matrix factorizations to frequency domain wave problems to machine learning convolutions. The Google tensor processing unit is an extreme case, processing 65,536 multiply–add operations for 8-bit integers on every cycle[13].

However, memory-limited routines, such as solving a system of linear equations with an iterative method such as conjugate gradients, typically operate at just one percent of peak performance, even on the most advanced systems. An important contemporary trend is a hybrid computational node of low-latency central processing units with large memories but limited memory bandwidth and throughput-oriented graphics processing units with high bandwidth for smaller memories.

Storage and fast memory

Key to supporting the operation of HPC processing is the provision of vast memory capacities and rapid delivery of data on the critical path, achieved through the efficient use of a multilevel memory hierarchy to stash next-in-use data in registers accessible by the processors, in a choreography that is a combination of the skills of the programmer and the compiler. Contemporary processors offer a variety of memory types with different sizes and streaming rates. Fast memory is the primary factor driving up the cost of HPC systems, whereas transferring data through memory hierarchies and across a distributed memory network is the main contributor to their energy consumption and the time required for operations. The fidelity of representation of the physical state of complex systems is ultimately limited by the memory capacity, the memory access latency and the memory bandwidth. Memories are triply finite meaning that they have a finite number of words, each of which can represent only a finite number of quantities that can be loaded and stored at only a finite rate. No currently imaginable supercomputer has a number of addressable bits that comes within one millionth of Avogadro’s number, for example.

Networking

To bring to bear the processing and memory of many nodes on a single physical system being modelled, HPC systems are intricately interconnected through high-speed networks, facilitating data exchange and awareness of state between individual computing nodes. These networks enable computational tasks with data interdependencies distributed across the system to be coordinated and, wherein unavoidable, synchronized. Contemporary networks, such as Slingshot 11, which powers four of the May 2024 Top 10 systems on the ‘Top 500’, are rich enough in topological complexity to enable any pair of nodes

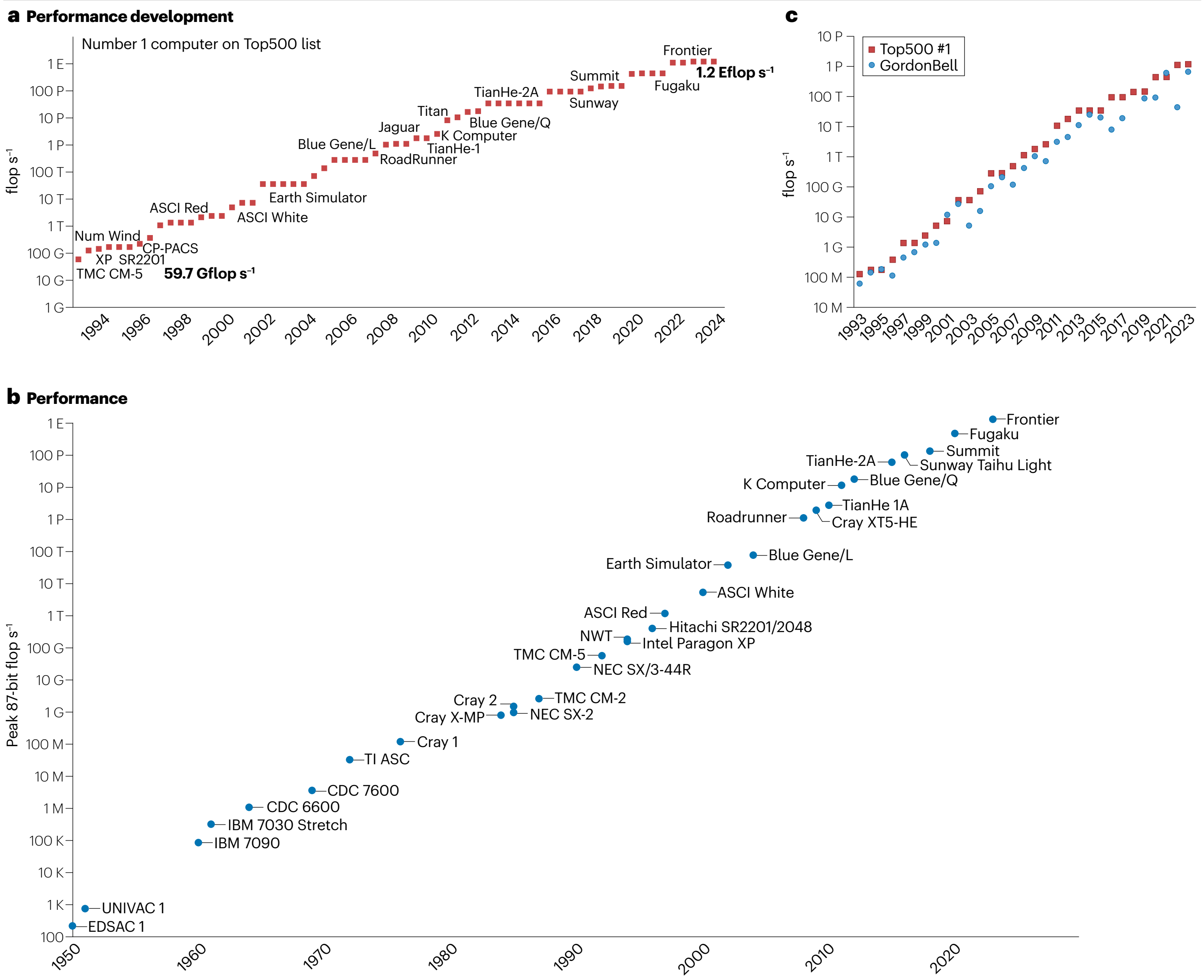

Fig. 1 | Plots of logarithm of performance in floating-point operations per second (flop s^{-1}) versus time in years. a, Three decades of top-ranked systems in High Performance Linear algebra (HPL). **b**, Theoretical peak performance of leading scientific computers over seven decades. **c**, The tracking of the HPL 500 benchmark by Gordon Bell Prize peak performance winners.

to exchange data with at most a few hops, while being much more time-efficient and energy-efficient and manufacturable than the 'ideal' of a crossbar switch, with all-to-all connections in hardware.

Algorithms

The hardware 'body' of processors, memory and networking is useless without the algorithmic 'mind' that orchestrates the computation. Improvements in the hardware typically change the constants in the time and space complexity models of high-performance computations, whereas algorithms can change the exponents and are, thus, the true linchpins of HPC, enabling users to harness the full potential of hardware resources. Every decade since the 1960s has seen at least one profound source of complexity reduction in operations, data motion and data storage in core computations and representations of system state owing to algorithmic breakthroughs. Table 1 lists several algorithms that shaved a fractional power or more of the problem size N off of the previously best known computational complexity for some core computational task, while delivering an equivalent or tunably acceptable accuracy in the result. Progress in HPC under Amdahl's law requires 'weak scaling' in which additional performance is invested in proportionally larger problems so that the useful computation dwarfs the non-parallelizable overheads. Successful weak scaling using nodes with fixed memory-to-processing ratios, requires 'optimal' computational complexity: the operation count cannot be allowed to outpace the memory volume by more than a poly-log factor. Such algorithms are typically recursively hierarchical.

Table 1 | Complexity reductions in a core computational task enabled by the introduction of a new optimal algorithm, relative to its lowest complexity predecessor

Task	Complexity reduction
Fast Fourier transform (1960s)	$N^2 \rightarrow N\log N$
Multigrid (1970s)	$N^{4/3}\log N \rightarrow N$
Fast multipole (1980s)	$N^2 \rightarrow N$
Sparse grids (1990s)	$N^d \rightarrow N(\log N)^{d-1}$
H matrices (2000s)	$N^3 \rightarrow k^2 N(\log N)^2$
Multilevel Monte Carlo (2000s)	$N^{3/2} \rightarrow N(\log N)^2$
Randomized matrix algorithms (2010s)	$N^3 \rightarrow N^2\log k$

N is the problem size. For hierarchical and randomized matrix algorithms, *k* represents the effective maximum rank of a block of the matrix, typically much less than *N* in problems that derive from physically causal systems. For sparse grids, *d* is the spatial dimension of the grid.

Software

The near-optimal-in-*N* algorithms mentioned in the Table 1 and many others are implemented in software that encapsulates expertise so that it is reliably transferable from expert designers to non-expert users (more precisely, to users that are expert in something else). Scientific software engineering is a discipline of its own, which has over the past few decades evolved the following elements:

- Verification is the process of checking that the model is being correctly computed. If fundamentally discrete, it can be in principle be checked directly. If the model is discretized from the continuum, it can be verified in a limit of some discrete parameters such as mesh size, time step size, particle number and ensemble number. Because software is constantly under improvement in capability or portability, this generally requires regular regression testing. The 'method of manufactured solutions', which posits a function and adds terms to the equation to make it the solution, is often used.
- Validation is the process of checking that the outputs of the verified computation are consistent with the physics being modelled.
- Uncertainty quantification establishes bounds within which the outputs of the computation can be trusted, given all of the uncertainties of its inputs, including the governing equations, boundary and initial conditions, values of parameters, and system 'noise'. One such type of noise is floating-point rounding error. This perspective unifies the larger burden of uncertainty quantification with classical numerical error analysis, with which it shares many probabilistic tools.
- Repeatability, replicability and reproducibility are related concepts, which the Association for Computing Machinery has defined in a nested way[14]. Repeatability considers whether the same team running the same code in the same computational environment gets the same result. Replicability considers whether a different team running the same code in the same computational environment gets the same result, that is, if the description of how to use the code and system is sufficiently complete. Reproducibility considers whether a different team can reimplement the computation in new code and a new environment and get the same result.
- Standardization and pre-competitive industry roadmaps have been essential to the portability and planned interoperability of HPC software and, thus, to the productivity of HPC developers and users.
- Benchmark problems with accepted results (such as the 'driven cavity' in fluid mechanics[15]) can endow users with confidence in new methods and developers with credibility in presenting them.
- Performance benchmarks, such as the High Performance Linear algebra (HPL) 500 (ref. 16), the HPCG 500 (ref. 17), the HPL-MxP[18], the Graph500 (ref. 19), the Green500 (ref. 20) and the IO500 (ref. 21), allow users to compare hardware–software environments for their cost-effectiveness and time-effectiveness and are an essential part of high-performance computing, beyond the general requirements of software for computational science and engineering.
- Performance modelling allows hardware and software architects to design systems that better support a class of computations by asking 'what if' questions.
- Hourglass reusability refers to the ability of software at the narrow waist of an hourglass to connect many applications above to many architectures below through a uniform application programming interface (API), leveraging software development over many requirements and many resources.

The components of processing, memory, networking, algorithms and software make up a five-dimensional Borromean complex: take away any one ring and the others fall apart. This is illustrated in Fig. 2 for HPC as a whole on the left for three rings, wherein architecture encompasses by itself processing, memory and networking, as further unrolled on the right. Note, for example, that the software ring lies on top of the architecture ring, but that the algorithms ring passes above the software ring and below the architecture ring, making all three mutually inseparable.

Physicists driving the HPC infrastructure

Physicists have important roles in the earliest developments in computing and have continued to contribute to advances in computational infrastructure and applications into the current exascale era. For example, the open science high-performance computational facility with the largest user base, the National Energy Research Scientific Computing Center (NERSC), which originated in the Center for Thermonuclear Research Computer Center (CTRCC), was established by Trivelpiece in 1974 at the Lawrence Livermore National Laboratory[22]. CTRCC was subsequently renamed the National Magnetic Fusion Energy Computer Center (NMFECC) and became NERSC in 1990. Within a year of founding, a remote access system allowed fusion energy scientists at Oak Ridge National Laboratory, Los Alamos National Laboratory and General Atomics to communicate with the centralized computers. Trivelpiece recommended that the Magnetic Fusion Energy Network (MFEnet) of the Department of Energy be combined with the High Energy Physics network (HEPnet) to become ESnet (Energy Sciences Network) in 1986. Hence, fusion and high energy physicists pioneered not only multi-user supercomputer facilities but also networks to make these universally accessible.

Meanwhile, astrophysicists have contributed greatly to the culture of open-source scientific software with versatile software frameworks, from the Zeus codes (beginning in the late 1970s) of Norman and Stone[23], the eponymous Barnes and Hut tree code (1989)[24], the Cactus framework (1995) of Seidel and collaborators[25] and the Flash code (early 2000s) from the University of Chicago ASCI centre[26], to name a few. These physics application frameworks paralleled the development of open-source linear algebra codes from the numerical analysis community, such as LINPACK and EISPACK from the early 1970s, all of which set the standard for the creation of today's virtuous cycles of software development and best practices in long-term software maintenance for user communities.

There are numerous other examples of physicists driving HPC infrastructure. Proceeding from network architecture to data architecture, the world wide web originated at CERN from a proposal[27] by physicist Berners-Lee in March 1989. By the end of that year, to productively share experimental data, particle physicists had given the world html, http and URLs and created the first web server, web browser and web editor, leading to such a dominant source of demand for these newly imagined computational services that as of today five of the six most valuable companies in the world by market capitalization are IT companies[28].

Culminating a distinguished history of contributions to architecture from lattice gauge theorists, including the Caltech quantum chromodynamics (QCD) project (which led to the Intel Paragon and ASCI Red), the European APE computers and the Japanese QCDPAX/CP-PACS, the QCD-on-a-Chip (QCDOC) machine was built at Columbia University in 2004. Engineered by Christ and collaborators[29], QCDOC delivered 10 Tflop s^{-1}, sustaining about 30% of its theoretical peak performance on lattice gauge computations. This was achieved through stripping down the operating system and focusing on efficient network operations such as neighbour exchange and global reductions. International Business Machines's (IBM's) subsequent recruitment of four particle physicists to engage in the design of this revolutionary system soon led to the commercial success of the Blue Gene series of supercomputers (known, respectively, as 'L', 'P' and 'Q'), which, for 15 years, held anywhere from one to four positions in the Top 10 of the HPL 500, topping the list for four consecutive years (2004–2007), as shown in Fig. 1a. The performance of systems on the HPL Top 500 benchmark closely tracks the theoretical peak performance of such systems, often achieving as high as 85% of the peak. Peak double-precision performance of leading scientific computers over seven decades is shown in Fig. 1b.

Thus, particle physicists ushered in a new era of performance and helped to pioneer the co-design of hardware and applications. The culture of high performance had begun much earlier through the zealotry of Cray and was enthroned in the semi-annual ranking of supercomputers in Top 500 (ref. 16) (also referred to as the HPL 500), but features of the Blue Gene architecture, such as providing two interprocessor networks, one optimized for local data exchange and the other specialized for global reductions to address a key bottleneck of the iterative solvers of lattice gauge theory, provided new inspiration for architectures specialized to the demands of simulating the physical world.

In 2008, the US National Medal of Technology and Innovation was awarded to IBM for the above-mentioned Blue Gene systems for "new science, unsurpassed speed and unparalleled energy efficiency"[30]. On new science, for three consecutive years (2005–2007) following their introduction, Blue Gene/L systems, originally designed for QCD, served instead to take the Gordon Bell Peak Performance Prize in macroscopic materials applications, boosting previous Bell Prize-capturing flop s^{-1} rates by more than an order of magnitude in the process (note the jump on the graph in Fig. 1c between 2004 and 2006).

An expanded discussion of the success of physicists with the Gordon Bell Prize follows in the next section. The extension to a wide variety of applications of an architecture originally motivated by a particular application is a recurring success story. It is mirrored by the successes in modelling and simulation applications of many architectures motivated by artificial intelligence, from the Connection Machine CM-2 (1989) to the Cerebras CS-2 (2023). Ironically, a QCD application was awarded the Gordon Bell Prize only once (in 1995, not on a Blue Gene system; see the section 'Physicists and the Gordon Bell Prize') but five Gordon Bell Prizes were won on Blue Gene systems for other applications. Historically, physicists have been instrumental in the development of the technologies and culture of HPC discussed in this section. A convenient way to demonstrate this is to look at the dominance of physicist-led teams in history of the Gordon Bell Prize.

Physicists and the Gordon Bell Prize

The history of advances in HPC architecture and infrastructure, algorithms, software and applications pioneered by physicists of various stripes is reflected in the advances in physics applications that have been recognized with the Gordon Bell Prize. Established in 1987, the Gordon Bell Prize is awarded annually by the Association for Computing Machinery in conjunction with the Institute of Electrical and Electronics Engineers Computer Society. The prize aims to showcase innovations in parallel computing and highlight groundbreaking research that pushes the boundaries of what is possible with HPC systems. The prize is awarded to individuals or research teams who have demonstrated exceptional performance and scalability in solving real-world problems using parallel computing techniques. Submissions inevitably focus on applications that require massive computational resources. (See a history of the prize, including insights from the founder, in ref. 31). Winning entries are selected based on several criteria, including the importance of the problem addressed, the innovation and creativity of the approach, the scalability and efficiency of the parallel computing techniques used, and the impact of the results on scientific or engineering research. By focusing on progress in scaling the performance of bona fide computational applications, the Gordon Bell Prize complements kernel benchmarks such as the Top 500. The categories of Gordon Bell Prizes and the criteria for winning them have evolved over its 37 years, as HPC and its capabilities and applications reach have evolved. Each year, a committee of Gordon Bell Prize judges has latitude to adapt the prize, as did the late Gordon Bell himself in its earliest years, with the overall goal of steering HPC in practically useful directions — rewarding what is most meritorious for advancing the field's directions. This periodic tuning is necessary because, for example, as Olympic records evolve incrementally, by seconds or centimetres per decade, HPC records evolve by ratios as much as three orders of magnitude per decade and have been doing so for several decades, requiring constant re-evaluation of what is an important new capability.

One aspect of the Gordon Bell Prize that has been a regular component over its history is the award for peak performance on a real application, of which there is typically only one first prize. In some

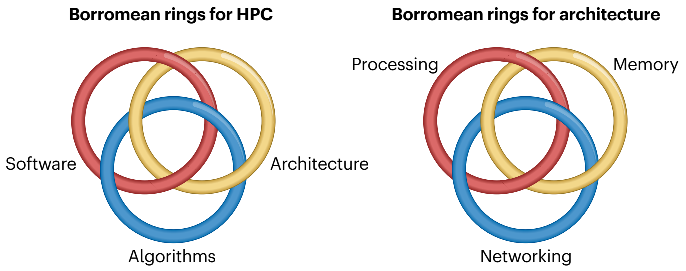

Fig. 2 | Borromean rings highlighting the interdependence of software, architecture and algorithms, and the three interdependent components of hardware architecture. Borromean rings are topologically linked such that by cutting any one ring, all the other rings fall apart. The idea illustrates the interdependence of high-performance computing (HPC) and hardware components.

years, peak performance on a real application is the only prize awarded. It is, therefore, interesting to note that teams led by physicists have led in this prize category for 30 of the 36 times that this prize has been awarded (Table 2), including in shared first prizes in 1996 and 2000.

Relevance to the advancement of the field of applications through HPC is a fundamental requirement of the prize whereas advancement of HPC, itself, is not. Nevertheless and unsurprisingly, advancement of HPC is often a consequence of the endeavour to extract new capabilities from supercomputers. For example, the first Gordon Bell Prize for a physics application was awarded to astrophysicists Michael S. Warren and John Salmon for the cosmological application of self-gravitating particles in 1992. This team was again awarded the peak performance prize with a new advance in 1997. Noteworthy HPC contributions in their work were the use of adaptively refined octree data structures and Hilbert space-filling curves to enhance memory locality[32].

A few Gordon Bell awards resulted in the co-design of new hardware, not widely available commercially. This includes the MDM molecular dynamics simulator in 2000, the GRAPE gravitational pipeline machines of 1995, 1996, 2000, 2001 and 2003, and the Anton-2 molecular dynamics simulator in 2014 (Table 2).

How do physicists use HPC?

The motivation of von Neumann for computing could not have been to obtain quantitative predictions – memories were too small for high-fidelity representations of complex systems and processing rates too slow. Instead, he sought qualitative understanding of phenomena, mainly nonlinear, that were not approachable through classical analytical means – whether 'exact' (a euphemism for practically unsummable infinite series), asymptotic or probabilistic (such as the Metropolis algorithm).

By contrast, following the pronouncement of American Physical Society President Langer in 1999 that "the computer literally is providing a new window through which we can observe the natural world in exquisite detail"[33], the aspiration of high-performance computational science and engineering has been quantitative results. In many applications, computational error bounds are now narrower than experimental error bounds. This was a strong motivation for the Scientific Discovery through Advanced Computing (SciDAC) programmes of the US Department of Energy (2001–present).

Table 2 | Years in which the Gordon Bell Prize in peak performance was awarded for a physics application, grouped by topic and ordered by year of first appearance

Field	Year
Cosmology	1992, 1996[a], 1997, 2012
Fluid dynamics	1993, 1996[a], 1999, 2007, 2010, 2013
Quantum chromodynamics	1995
Condensed matter physics	1998, 2005, 2006, 2008, 2009, 2011, 2019, 2023
Molecular dynamics	2000[a], 2014, 2020
Astrophysics	2000[a], 2001
Atmospheric dynamics	2002, 2016
Solid earth geophysics	2003, 2004, 2015, 2017
Quantum circuit	2021
Laser physics	2022

[a]Two teams tied for first place in 1996 and 2000.

Simulations can be reliable substitutions for experiments that are controversial, dangerous, prohibited, impossible, difficult to instrument or simply too expensive. Some big experiments (for example, the Superconducting Supercollider) have been left incomplete when their evolving price tags became too high. Others (for example, ITER) exceed by more than an order of magnitude the cost of the largest supercomputer of the world, which makes a simulation environment dedicated to their development comparably inexpensive. In addition, simulations can be a 'time machine' for experiments that take too long to complete naturally before their results are required. Simulations can also narrow the parameter space of expensive or time-consuming experiments, steering scientists to the most useful experiments to build and perform.

Though physicists have made decades of pioneering contributions to HPC, their categories of use of HPC are not notably different from other user communities. Major use cases have been enshrined in Accelerated Strategic Computing Initiative (ASCI) and SciDAC programmes of the US Department of Energy. Topical motivations throughout physics for the SciDAC programme were described in the Science-based Case for Large-scale Simulation (SCaLES) report[34], contributed to by 315 US Department of Energy-funded scientists, including accelerator designers, astrophysicists, climate modellers, materials scientists, nanoscientists, plasma physicists and lattice gauge theorists, and fluid dynamicists.

Whereas a telescope is for astronomers, an accelerator for particle physicists and an electron microscope for condensed matter or biophysicists, a supercomputer is for all, as foreseen by Fermi. It is like the proverbial office water cooler, wherein serendipitous conversations occur between scientists of all specializations, which have many computational tools in common, whether the corresponding phenomena span angstroms or parsecs, femtoseconds or millennia.

Why will physics continue to drive HPC?

Physicists tend first to dissect the phenomena they study in every possible way to isolate the inevitably computational components to use the computer as efficiently as possible. However, what is left usually still remains computationally challenging. Computational physics is a realm of the 'multis': the phenomena remaining to be effectively modelled and understood are usually multidimensional, multi-scale, multi-rate and multi-physics in nature. Computing such systems may require integrating multiple models with different fidelities and multiple nodes consisting of multiple cores executing multiple threads of multilevel hierarchical algorithms designed and implemented by multidisciplinary teams. Most systems are also fundamentally nonlinear, putting them beyond the range of many analytical tools based on linearity and superposition. It is a reliable prediction that physics will continue to drive computing at the high-end, as it has for over eight decades.

Clouds are currently forming over both capability and capacity in the realm of HPC. The end of Dennard scaling[35], relating the increase of processing rates to miniaturization at constant power per unit area on a CMOS devices, and the slowing of Moore's law[36], relating to the density of devices per unit area, have dampened the slope of forecast increases of computational power available from tightly coupled supercomputers at the same time that raw electrical power demands, with concomitant environmental consequences, have injected caution into the proliferation of loosely coupled farms of servers. So far, processor and memory technologies more energy-efficient than CMOS do not come close to competing with CMOS in speeds or feeds.

As new component technologies (such as optical computing) ultimately emerge as competitive, among the challenges they will face are the time and energy costs of converting back and forth to the CMOS-based devices that are slower to be replaced. Now, more than ever fresh ideas from physicists are needed to solve these challenges.

Since the 1981 lecture of Feynman[37], quantum computing has beckoned as a new mode of computation, especially for systems that are themselves quantum mechanical in nature. Physicists will probably be both the early enablers and the early users of this emerging technology, whose first practical appearances will be in integration with classical supercomputing.

Beyond traditional simulation, which produces data from models, machine learning now produces models from data but often with questionable efficiency in terms of parameter size and training time. When the data come from the physical world, there are major gains to be made by exploiting physical structure to prune the machine models and in combining data with mathematical models and deriving a hybrid computation that is superior to either. Artificial intelligence is also used to interpret outcomes of experiments that produce large amounts of data, such as astronomical observations and particle colliders. As Perlmutter described it in his work, which required identifying supernovae in tens of thousands of galaxies, "This was a perfect job for a computer to do."[38] Many such jobs lie ahead.

Published online: 23 August 2024

References

1. De Marco, G., Mainetto, G., Pisani, S. & Savino, P. The early computers of Italy. *IEEE Ann. Hist. Comput.* **21**, 28–36 (1999).
2. Von Neumann, J. First draft of a report on the EDVAC. *IEEE Ann. Hist. Comput.* **15**, 27–75 (1993).
3. Von Neumann, J. & Goldstine, H. H. Numerical inverting of matrices of high order. *Bull. Amer. Math. Soc.* **53**, 1021–1099 (1947).
4. Von Neumann, J. & Richtmyer, R. D. A method for the numerical calculation of hydrodynamic shocks. *J. Appl. Phys.* **21**, 232–237 (1950).
5. Charney, J. G., Fjörtoft, R. & Von Neumann, J. Numerical integration of the barotropic vorticity equation. *Tellus* **2**, 237–254 (1950).
6. Metropolis, N., Howlett, J. & Rota, G.-C. (eds) *A History of Computing in the Twentieth Century* (Academic, 1980).
7. Adler, B. (ed.) *Special Purpose Computers* (Academic, 1988).
8. Battimelli, G., Ciccotti, G. & Greco, P. *Computer Meets Theoretical Physics* (Springer, 2020).
9. Fox, G., Williams, R. & Messina, P. *Parallel Computing Works* (Morgan-Kaufmann, 1994).
10. Vetter, J. *Contemporary High Performance Computing* (Chapman and Hall/CRC, 2017).
11. Eijkhout, V. *The Art of HPC* Volumes 1–4 https://theartofhpc.com/ (The Art of HPC, 2022).
12. NVIDIA. NVIDIA H100 Tensor Core GPU. *NVIDIA* https://www.nvidia.com/en-us/data-center/h100/ (2024).
13. Google. An in-depth look at Google's first tensor processing unit (TPU). *Google* https://cloud.google.com/blog/products/ai-machine-learning/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu (2017).
14. ACM. Artifact review and badging. *Association for Computing Machinery* https://www.acm.org/publications/policies/artifact-review-badging (2020).
15. Park, S. Direct numerical simulation for lid-driven cavity under various Reynolds numbers in fully staggered grid. *Phys. Fluids* **35**, 115110 (2023).
16. Strohmaier, E., Meuer, H. W., Dongarra, J. & Simon, H. D. The TOP500 list and progress in high-performance computing. *Computer* **48**, 42–49 (2015).
17. Dongarra, J., Heroux, M. A. & Luszczek, P. High-performance conjugate-gradient benchmark: a new metric for ranking high-performance computing systems. *Int. J. High Perform. Comput. Appl.* **30**, 3–10 (2016).
18. Dongarra, J. & Luszczek, P. HPL-MxP https://hpl-mxp.org/ (2024).
19. Graph500 https://graph500.org/ (2024).
20. Green500 https://top500.org/lists/green500/ (2024).
21. Io500 https://io500.org/ (2024).
22. NERSC. Al Trivelpiece and the origins of NERSC. *National Energy Research Scientific Computing Center* https://www.nersc.gov/news-publications/nersc-news/nersc-center-news/nersc-40th-anniversary/al-trivelpiece-and-the-origins-of-nersc/ (2014).
23. Stone, J. M. & Norman, M. L. ZEUS-2D: a radiation magnetohydrodynamics code for astrophysical flows in two space dimensions. I — the hydrodynamic algorithms and tests. *Astrophys. J. Suppl. Ser.* **80**, 753–790 (1992).
24. Barnes, J. E. & Hut, P. Error analysis of a tree code. *Astrophys. J. Suppl. Ser.* **70**, 389–417 (1989).
25. Goodale, T. et al. The Cactus framework and toolkit: design and applications. In *Vector and Parallel Processing — VECPAR'2002, 5th International Conference, Lecture Notes in Computer Science* 197–227 (Springer, 2003).
26. Dubey, A. et al. Evolution of FLASH, a multi-physics scientific simulation code for high-performance computing. *Int. J. High Perform. Comput. Appl.* **28**, 225–237 (2013).
27. Berners-Lee, T. Information management: a proposal. *w3 archive* https://www.w3.org/History/1989/proposal.html (2024).
28. CompaniesMarketCap. Largest companies by market cap. *CompaniesMarketCap* https://companiesmarketcap.com/ (2024).
29. Boyle, P. et al. QCDOC: a 10 teraflops computer for tightly-coupled calculations. In *SC '04: Proceedings of the 2004 ACM/IEEE Conference on Supercomputing* 40 (2004).
30. NSTMF. National medal of technology and innovation. *National Science and Technology Medals Foundation* https://nationalmedals.org/laureate/ibm-corporation-2/ (2024).
31. Bell, G., Bailey, D. H., Dongarra, J., Karp, A. H. & Walsh, K. A look back on 30 years of the Gordon Bell Prize. *Int. J. High Perform. Comput. Appl.* **31**, 469–484 (2017).
32. Warren, M. & Salmon, J. A parallel hashed Oct-Tree N-body algorithm. In *Supercomputing '93: Proceedings of the 1993 ACM/IEEE Conference on Supercomputing* 12–21 (ACM, 1993).
33. Schweber, S. & Wächter, M. Complex systems, modelling and simulation. *Stud. Hist. Philos. Sci. B* **31**, 583–609 (2000).
34. Keyes, D., Colella, P., Dunning Jr, T. & Gropp, W. A science-based case for large-scale simulation (US DOE, 2003).
35. Dennard, R. et al. Design of ion-implanted MOSFET's with very small physical dimensions. *IEEE J. Solid State Circuits* **9**, 256–268 (1974).
36. Moore, G. E. Cramming more components onto integrated circuits, reprinted from *Electronics*, volume 38, number 8, April 19, 1965, pp.114 ff. *IEEE Solid State Circuits Soc. Newslett.* **11**, 33–35 (2006).
37. Feynman, R. P. Simulating physics with computers. *Int. J. Theor. Phys.* **21**, 467–488 (1982).
38. Chirigati, F. The universe's expansion in the eyes of computers. *Nat. Comput. Sci.* **2**, 545–547 (2022).

Acknowledgements
The authors thank Iulia Georgescu for historical observations and her motivation for this article.

Author contributions
The authors contributed equally to all aspects of the article.

Competing interests
The authors declare no competing interests.

Additional information
Peer review information *Nature Reviews Physics* thanks Junichiro Makino and the other, anonymous, referee(s) for their contribution to the peer review of this work.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This is a U.S. Government work and not under copyright protection in the US; foreign copyright protection may apply 2024