High-Performance Computing

Hochleistungsrechnen

Hans-Joachim Bungartz*

* Correspondence author: bungartz@in.tum.de

Today, the field of Computational Science and Engineering (CSE) is generally considered as a key technology for science and industry. In Germany, for example, institutions such different as the Science Council (Wissenschaftsrat), the German Research Foundation (DFG), or the International Association of Applied Mathematics and Mechanics (GAMM) are currently thinking about implementing CSE as a discipline or section, respectively, of its own – or are already doing so. In CSE, the word "computational" is much more specific than "computer-based" or "computer-aided", the latter almost being a matter of course given the level of IT penetration today. "Computational" rather means that the whole process of research and development, possibly aiming at very different goals from getting fundamental insight into our world to improving a very concrete product, is essentially based on computations. Eventually, different names appear, such as "Simulation-based Engineering", "Simulation Science", or "Simulation Technology" (putting the focus on the classical computational task of numerical simulations), or "Computational and Data-driven Science and Engineering" (thus emphasizing the data or information behind the mere numbers) – they all are facets under the CSE umbrella. And the crucial part of CSE for science and industry is obvious – one just has to look at the increasing number of computation-driven publications in leading scientific journals on the one side or at the need for faster product development cycles on the other side.

But now for the role of High-Performance Computing (HPC), the field of large-scale computations on high-end parallel computer systems and the topic of this special issue of *it*. Concerning the type of computations in modern CSE, not all of them, of course, are large-scale ones, and not all of them use or even need high-end systems. A lot of simulation, analysis, or optimization work has become routine and can be done with commercial software on standard desktop systems. However, this is not at all a contradiction to the ever-growing relevance of HPC for CSE, it rather underlines the success of HPC and depicts the path how leading-edge technology of today with a, maybe, seemingly narrow scope will influence everyday work of tomorrow. Let me start our discussion of HPC with some core statements which may help to understand the essence of HPC.

First, HPC is intrinsically cross-disciplinary. It gathers a broad spectrum of methodologies from informatics and mathematics, thus enabling various areas from science and engineering to exploit the potential of supercomputers and to, hence, answer their most urgent research questions. And the family of HPC users keeps growing, with life sciences, medicine, or even social sciences entering the stage more frequently.

Second, dealing with the computationally most challenging tasks, HPC has always been focused on "performance", but this notion turned out to be much more multi-faceted than the hardware-centered peak performance entries in the Top500 lists (http://www.top500.org), showing the 500 currently leading installations world-wide every six months, suggest. At the beginning, there were just algorithmic complexity (greetings from O(N) ...) and floating point operations per second (or, briefly, Flops), but meanwhile, researchers look at memory efficiency, in particular cache efficiency, at communication-avoiding schemes, at scalability issues, and at energy aspects. Flops are still the dominant male of metrics, but there are more: operations per Watt for example, or how many floating point operations can be performed before the next byte from the memory is needed; or even a bit esoteric ones, such as number of publications per kilowatt, showing how successfully an HPC system is used.

Third, the second issue shows that the times when hardware was developed independently, when algorithms were developed merely following complexity or accuracy

This special issue was recommended by Editor Prof. J. Teich and Editor in Chief Prof. Dr. P. Molitor.

arguments, and when, afterwards, the user communities just had to select suitable algorithms for their models from the literature and to port them to the systems available are mostly gone. Models, algorithms, software, and hardware – everything is closely interwoven now. This is the reason that the notion of co-design has become a frequent visitor also at HPC conferences and journals. What we need is probably not that much an application-system co-design, but rather an application-algorithm one that links numerical algorithms and mathematical models as the heart of the applications, as well as an algorithm-system co-design that takes into account the interplay of the structure of efficient algorithms (typically involving heterogeneous patterns such as hierarchy, dynamics, or adaptivity) and the structure of parallel systems.

Fourth, the HPC world (or better the computer systems world) is going parallel, and it is doing that in a really massive way. No way to hide behind a small number of extremely strong processors as we have learned to love them since the good old times of vector processors: Which farmer would not prefer the strength of two oxen to that of thousands of chicken when having to plough his field, which aircraft designer would not prefer four strong jet engines to 100 000 hairdryers? Among other challenges, making all hairdryers to blow at the same time in the same direction may be tedious… Nevertheless, clock rates are basically frozen for technological reasons, hence Moore's law can only survive if we turn the classical "we are getting stronger" into a "we are getting more" – and that is exactly what has been visible now for a couple of years already. In the not that far future, we will see core numbers beyond a million and above, no longer organized as flat arrays, but as increasingly complex hierarchical and hybrid constellations. The variety occurring – play stations and graphics cards suddenly entered the HPC stage, and commodity chip manufacturers are striking back – may appear somewhat surprising and, actually, results in some pioneering spirit concerning the future of many-core architectures. To some extent, however, this development leads us back to the roots: HPC as one of the main users, but also as one of the main drivers of chip technology.

Fifth, the breathtaking speed with which large-scale simulation results can be delivered raises the question of how to explore the numbers to get the desired insight (greetings from Richard Hamming this time). And there are more questions coming along with the data: How to deal with uncertainties in the data? How to derive general statements from single-case runs (patient-specific, e. g.)? How to efficiently combine data from completely different sources (such as experiments and computations)?

Sixth, we see a dramatic change in the role of software also in HPC. High-end simulation software is no longer a single program written by a heroic PhD student within a 3-year life cycle – we encounter huge simulation software systems, sometimes with hundreds of thousands or even millions of lines of code. And as application software, the system software and the tools landscape are not well prepared neither – which, for example, means that several boxes that had already been more or less closed (run-time system or compiler support, special parallel languages, etc.) in informatics research have to be reopened quickly. Actually, it is obvious that the software issue will be the roadblock for HPC, much more threatening than hardware or algorithmic challenges. And this means that buying and operating supercomputers is not enough – we need well educated people for the HPC software ecosystem, or shortly: We need brains in addition to the racks.

My number seven has just been mentioned – education. Of course, the above "we need brains" also is a funding issue, but it is, first of all, an educational one. We have to ensure that there will be experts with the required spectrum of knowledge and expertise, and we have to tackle this challenge both via special offers such as HPC lab courses in traditional study programs or for professionals and via specialized HPC-oriented programs.

Eighth, we also need a new mind-setting within the involved communities, assigning all HPC-related research activities the scientific value and merits they deserve. The times of "let the most brilliant people do the modelling, let the average brains do the algorithms, and let the remaining ones write the programs" are gone forever!

Ninth, we will see a broader range of HPC usage models. Beside the classical batch job, additional scenarios such as interaction in computational steering or urgent computing, e. g., will gain in importance.

My tenth and final statement refers to funding. The German and European HPC landscape have got well working structures – the key words Gauss Alliance, Gauss Centre, or PRACE will be filled with life in one of this issue's contributions – and significant investments in high-end systems have been made. Compared with that, the funding schemes for software are less developed. But there are remarkable and promising efforts of how problems can be addressed. Two prominent examples shall be mentioned here: the funding program "HPC Software for Scalable Parallel Systems" by the German Federal Ministry for Education and Research (BMBF) with its first three calls in 2007, 2010, and 2012, and the Priority Program "Software for Exascale Computing" by the German Research Foundation (DFG) which has just started its first 3-year funding period.

In the following, you are invited to have a look at the state of the art of HPC on its way to the exascale era from different perspectives. While James Reinders takes the hardware and systems perspective, Harald Köstler and Ulrich Rüde focus on algorithms and software. Then, two contributions discuss the application point of view, Martin Horsch, Christoph Niethammer, Jadran Vrabec, and Hans Hasse addressing scientific applications, and Christoph Gümbel reporting requirements of and chances for industry. The HPC landscape (infrastructure, operation, usage, and funding) is the topic of Arndt Bode and Wolfgang Nagel, before “The Miracle, Mandate, and Mirage of HPC” by David Keyes provides us with some thoughtful remarks. So enjoy this special issue on High-Performance Computing!

Hans-Joachim Bungartz

Prof. Dr. Hans-Joachim Bungartz holds the Chair of Scientific Computing at TUM’s informatics department. He has degrees in both informatics and mathematics. His research and teaching interest and activities are in CSE and HPC. Hans-Joachim Bungartz is TUM’s Graduate Dean and a board member of Leibniz Supercomputing Centre (LRZ). Furthermore, he currently acts as chairman of both DFG’s commission for IT infrastructure (KfR) and the German National Research and Education Network (DFN).

Address: Technische Universität München, Department of Informatics Chair of Scientific Computing, Boltzmannstrasse 3, 85748 Garching, Deutschland, e-mail: bungartz@in.tum.de