Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Perspective:

Application software beyond exascale: challenges and possible trends*

Guang-wen YANG‡1,2,3, Hao-huan FU2,3

[1]*Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China*
[2]*MOE Key Laboratory for Earth System Modeling, Department of Earth System Science, Tsinghua University, Beijing 100084, China*
[3]*National Supercomputing Center in Wuxi, Wuxi 214072, China*
E-mail: ygw@tsinghua.edu.cn; haohuan@tsinghua.edu.cn

Received Aug. 2, 2018; Revision accepted Sept. 9, 2018; Crosschecked Oct. 15, 2018

Abstract: With various exascale systems in different countries planned over the next three to five years, developing application software for such unprecedented computing capabilities and parallel scaling becomes a major challenge. In this study, we start our discussion with the current 125-Pflops Sunway TaihuLight system in China and its related application challenges and solutions. Based on our current experience with Sunway TaihuLight, we provide a projection into the next decade and discuss potential challenges and possible trends we would probably observe in future high performance computing software.

Key words: Supercomputing; Exascale; Application
https://doi.org/10.1631/FITEE.1800459 **CLC number:** TP311

1 Introduction

According to current research progress in the USA and China, exascale systems are projected to be usable by scientists by 2021 (approximately). Over the next decade, if the hardware technology develops at the same speed as that in this decade, we will enjoy another improvement of computing power in the scale of two to three orders of magnitude.

Such an enormous increase in computing power would surely involve hardware innovations in many different aspects. As a result, radical changes in both architecture and scale would bring completely different challenges to application software developers (Shalf et al., 2010).

In this study, we start our discussion with the current 125-Pflops Sunway TaihuLight system in China and its related challenges and solutions. Based on the programming experience on Sunway TaihuLight, we discuss the possible challenges in future high performance computing (HPC) software and present our ideas about their developing trend for the next decade.

2 Sunway TaihuLight supercomputer: software challenges and solutions

As announced in the Top500 list on June 20, 2016, the Sunway TaihuLight supercomputer was ranked the fastest supercomputer in the world, with a peak performance of 125.436 Pflops, a sustained linpack performance of 93.015 Pflops, and power efficiency of 6.051 Gflops/W (Fu et al., 2016).

The Sunway TaihuLight system is the first supercomputer with a peak performance greater than

‡ Corresponding author

* Project supported by the National Key Technology R&D Program of China (No. 2016YFA0602200)

ORCID: Guang-wen YANG, https://orcid.org/0000-0002-8673-8254

100 Pflops in the world, and is the first Chinese system that is completely based on homegrown many-core processors. The SW26010 homegrown many-core central processing unit (CPU) includes 260 processing elements in a single chip and provides a peak performance of greater than 3 Tflops. As shown in Fig. 1, similar to the SW1600 CPU, the SW26010 CPU includes four identical core groups (CGs). Each CG includes one management processing element (MPE), one computing processing element (CPE) cluster with 8×8 CPEs, and one memory controller (MC). These four CGs are connected via a network-on-chip (NoC). Each CG has its own memory space, which is connected to the MPE and CPE cluster through the MC. The processor connects to other outside devices through a system interface (SI).

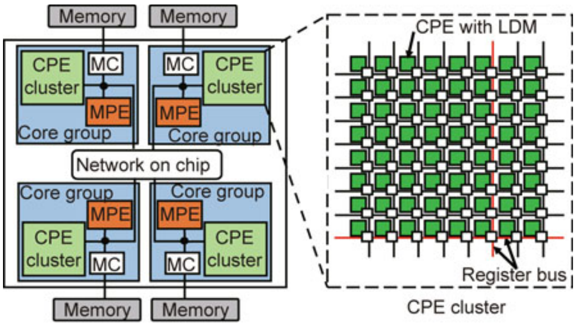

Fig. 1 General architecture of the new Sunway processor. MPE: management processing element; CPE: computing processing element; MC: memory controller; LDM: local data memory

The MPE is a complete 64-bit reduced instruction set computer (RISC) core, which can run in both user and system modes. The MPE completely supports interrupt functions, memory management, superscalar processing, and out-of-order execution. Therefore, the MPE is an ideal core for handling management and communication functions.

In contrast, the CPE is also a 64-bit RISC core, but with limited functions. The CPE can run only in user mode, and it does not support interrupt functions. This element is designed to provide maximum aggregated computing power while minimizing the complexity of the micro-architecture. The CPE cluster is organized as an 8×8 mesh with a mesh network that provides low-latency register data communication among the 8×8 CPEs. The mesh also includes a mesh controller that handles interrupt and synchronization controls. Both the MPE and CPE support 256-bit vector instructions.

In terms of memory hierarchy, each MPE has a 32 KB L1 instruction cache and a 32 KB L1 data cache, with a 256 KB L2 cache for both instructions and data. Each CPE has its own 16 KB L1 instruction cache and a user-controlled scratch pad memory (SPM). The SPM can be configured as either a fast buffer that supports precise user control, or a software-emulated cache that provides automatic data caching. However, as the performance of the software-emulated cache is low, a user-controlled buffering scheme is required to achieve good performance in most cases.

Sunway TaihuLight, while bringing great potential to advance the scientific and engineering capabilities in various domains, also presents major challenges that must be resolved.

The first design challenge is to derive the correct parallelization scheme that maps our target application into the processes and threads that use the over 10 million system cores. Similar to other heterogeneous supercomputers with many-core accelerators, we take a two-level 'MPI+X' approach. Each CG usually corresponds to one message passing interface (MPI) process. We have two different options within each CG. One is Sunway OpenACC, a customized parallel compilation-tool that supports OpenACC 2.0 syntax and targets the CPE cluster. The other is a high-performance and lightweight thread library named Athread, which provides interfaces similar to Pthread to exploit fine-grained parallelism. In our work, we take the Athread approach, which requires more programming effort but exposes more possibilities for tuning both the computation and memory access schemes.

The second challenge is the memory wall in this system. With a byte-to-flop ratio that is five to ten times lower than the other top five systems, extraordinary memory-related innovations are required to scale the simulation capability of the system.

The third challenge is the migration of software to such an architecture with radical changes in both compute and memory hierarchy. For each CPE, instead of a hardware L1 cache, we have a user-controlled 64-KB local data memory (LDM), which completely changes the memory perspective for programmers.

Facing such serious challenges, researchers from different domains, e.g., the Gordon Bell Prize winner and finalists, have exerted tremendous effort into scaling the application performance of the Sunway TaihuLight system.

The 2016 Gordon Bell Prize work (Yang et al., 2016) features a highly scalable fully implicit solver for cloud-resolving atmospheric simulations, which encapsulates novel domain decomposition, multigrid, and incomplete lower-upper (ILU) factorization algorithms to achieve massively parallel computation across 10 million cores.

The 2017 Gordon Bell Prize work (Fu et al., 2017a), in contrast, focuses on the memory constraint issue. The team proposed an elaborate memory scheme that integrates on-chip halo exchange through register communication, optimized blocking configuration guided by an analytic model, and coalesced direct memory access (DMA) with array fusion. On-the-fly compression was used to double the maximum problem size, which further improves performance by 24%.

For migration of complex software, a typical example is the redesign of the community atmosphere model (CAM) for use in the full Sunway TaihuLight system (Fu et al., 2017b), which provides peta-scale climate modeling performance. The complete code is first refactored and optimized using loop transformation tools and OpenACC directives. The Homme dynamic core is redesigned in a finer-grained manner to provide finer memory control and more efficient vectorization. This improves the performance of a 260-core Sunway processor in the range of 28 to 184 Intel CPU cores.

3 Beyond exascale: what to expect for application software

Without a concrete hardware context, it is difficult to discuss potential issues regarding application software. For the anticipated improvement from 100-Pflops systems to exascale systems, meeting both performance and power efficiency requirements for hardware developers is perceived as a significant challenge. With complementary metal oxide semiconductor technology approaching some important physical limits, revolutionary hardware innovations are expected to continually increase the computational performance of supercomputers over the next decade. As we are still far from making predictions about the most likely viable hardware technologies, we offer the following generalized discussions.

While hardware software co-design has already been discussed extensively, customizing hardware for a specific application domain is still a rare case in HPC. As supercomputers supported by national research organizations generally need to serve a wide spectrum of scientific problems with different architecture requirements, new systems can be built only based on improving computing performance and power efficiency. Therefore, we see very few special customized systems (usually with private or commercial funding sources), such as Anton (Shaw et al., 2009) and Anton 2 (Shaw et al., 2014) for molecular dynamics, and various deep learning processors customized for artificial intelligence (AI) applications (Chen et al., 2014).

Over the next decade, general-purpose supercomputers would probably still be the major platform for scientists working in different domains. However, as AI (especially deep learning) applications gradually become common paradigms in many scientific domains, certain AI-oriented cores would likely be integrated as part of the processors.

Most of the following discussions are based on a general-purpose supercomputer assumption. Only trend 3 also addresses architecture.

3.1 Possible trend 1: programming effort shifting from computation to data

In traditional computational algorithms and applications, people care more about the control flow of computational parts. The well-accepted discussion regarding the computational complexities of different algorithms is a typical example. Algorithm designers in the past concentrated on minimizing the amount of computations, while data movement operations gradually become a more substantial consumer of both running time and energy in many scenarios.

In current supercomputers, computing capabilities are clearly going beyond the data movement capability at different levels. Our experience with Sunway TaihuLight has provided many examples of significant performance benefits from careful memory-oriented redesigns and optimizations. In the early stages of the Sunway TaihuLight application development, we observe many complaints regarding the absence of cache in CPEs and the extra effort required to write buffering schemes. However, after the corresponding performance benefits were achieved in various applications, people began viewing the 64-

KB LDM in each CPE as a suitable hardware design strategy to force the integration of buffering strategies during the design and development process.

As mentioned previously, work on extreme-scale earthquake simulations with Sunway (Fu et al., 2017a) is a typical example. In other applications, most of the performance is gained from compute-related tuning or optimization, such as multi-level parallelization schemes and vectorization. In the earthquake simulation problem, nearly all the techniques we adopt (such as array fusion, customized DMA scheme, and on-chip halo exchange scheme) focus on improving memory access efficiency. With memory-related optimizations contributing to most of the performance, we manage to achieve the same computational efficiency as Titan (Cui et al., 2013; Roten et al., 2016) with only half memory bandwidth. Similarly, regarding the fully implicit solver for atmospheric dynamics (Yang et al., 2016), memory-related optimization nearly doubles the computational performance. Even for compute-intensive convolution neural-network kernels, we achieve the required level of performance through the guidance of a memory-oriented performance mode (Fang et al., 2017).

For the exascale and beyond exascale supercomputers in the next decade, unless more revolutionary innovations focus on memory, a greater shift of programming effort from computation to data movement would be expected.

First, more resources should be focused on software bottlenecks. Ideally, software developers, scientists, and mathematicians could place more emphasis on conceiving the flow of data rather than the flow of computation instructions. With such a shifting towards data movement occurring at different levels, the redesigned application could potentially lead to more efficient and aggressive utilization of hardware resources in supercomputers, thus providing a utilization rate at a completely different level.

Second, there has been an increase in the number of data-centered applications in supercomputer platforms. As mentioned above, with deep learning paradigms and other potential approaches that release more power from big data becoming important in many scientific applications, moving the focus from computation to data movement also becomes a natural transition. The launch of the Summit system (Wells et al., 2016), which provides exascale performance in half-precision AI scenarios, is clear evidence of such a trend. Similarly, for exascale systems in China, deep neural networks and other machine learning applications are becoming standard benchmarks as compliments to existing high-performance linpack and graph benchmarks. Such a transition would also help support the convergence of big data applications and extreme-computing applications that many scientific domains are expecting over the next decade.

3.2 Possible trend 2: precision optimization

The precision issue in supercomputers from the last few decades is simply hidden from most users. Users are given only the option to use either 64-bit double or 32-bit single floating-point numbers, and most applications run using an unvarying precision configuration.

In recent years, people have started to accept the adoption of a reduced half-precision for training processes in the domain of deep learning applications, and compressed bits are applied during the inference processes. As a result, more customized hardware architectures have been proposed for deep learning applications and have already demonstrated the potential to increase performance and power efficiency approximately by an order of magnitude.

While most scientific applications are still refusing to sacrifice double precision, we think that the same strategy can be applied in many simulation applications. While the current Sunway processor does not provide hardware support for mixed-precision computation, we explore this technique through compressed precision in memory storage. One example is the compression scheme adopted in the earthquake simulation work on Sunway TaihuLight (Fu et al., 2017a). Storing 16 out of 32 bits doubles the maximum problem size that can be implemented in the system, and further improves the computational performance by 24%. We see similar examples towards other traditional supercomputing domains, such as climate modeling (Palmer, 2015).

Note that bringing precision into this context does not necessarily mean that precision is reduced. Instead, it allows accuracy to be explored during design and optimization. Users can use higher precision for more sensitive variables and processes, and use reduced precision for insensitive variables and processes. Such an optimization process might lead to benefits for both performance and accuracy.

3.3 Possible trend 3: programming hardware instead of software

While it is difficult to support a wide range of scientific applications with a custom chip, programmable hardware chips (e.g., field programmable gate arrays (FPGAs)) are sometimes considered as another potential building block for supercomputers.

Considerable effort has been focused on applying reconfigurable computing technologies to improve computing performance and power efficiency. Prior reported improvements ranged in the scale of one to two orders of magnitude for simple computing kernels and for more complete scientific computing applications in recent years because of chip size limitations (Gan et al., 2013, 2014). Another advantage is the significantly improved power efficiency owing to the lower running frequency.

While reconfigurable FPGAs are not yet broadly accepted by the HPC community, their potential to significantly improve system efficiency renders them a promising candidate over the next decade.

The greatest challenge, which can be termed the greatest benefit, is the change from software programming to hardware programming. Instead of writing a sequence of instructions that process data items, we solve the problem by mapping computation into the corresponding circuits in the chip. There are a number of major advantages associated with the hardware programming approach. Inside the chip, we can easily have thousands of circuit units that form a very deep pipeline and perform computations in parallel. Outside the chip, we can integrate multiple chips into a larger unit with hardware channels. Therefore, instead of practicing extremely complex parallel program design processes, we convert the problem into hardware modules that can be coupled in different ways.

Reconfigurable FPGAs provide hardware programming and communication benefits. Instead of wrapping MPI calls in the software stack, we can provide customized hardware support to further improve communication efficiency and stabilize communication latency in different nodes.

4 Summary

In the supercomputing domain, hardware and software have always been important driving forces, and each has spurred mutual development. As discussed above, even at the current 100-Pflops stage, we have already observed severe challenges relating to parallelization, memory, and programming. In exascale systems and beyond, the continuous demand from scientific simulation and big data analytics will hopefully lead to revolutionary developments in both the way that we build and use systems. Unsurprisingly, these two areas are related to programming, which solves the problem of mapping computation to the underlying hardware system. The other challenge relates to precision, which has long been a missing element during the design of hardware and software. While it is still unknown which factor would really dominate computation over the next decade, we think that some revolutionary technology will likely change the way that we compute and parallelize, and improve computational efficiency.

References

Chen Y, Luo T, Liu S, et al., 2014. DaDianNao: a machine-learning supercomputer. 47th Annual IEEE/ACM Int Symp on Microarchitecture, p.609-622. https://doi.org/10.1109/MICRO.2014.58

Cui Y, Poyraz E, Olsen KB, et al., 2013. Physics-based seismic hazard analysis on peta-scale heterogeneous supercomputers. Int Conf for High Performance Computing, Networking, Storage, and Analysis, p.1-12. https://doi.org/10.1145/2503210.2503300

Fang J, Fu H, Zhao W, et al., 2017. SwDNN: a library for accelerating deep learning applications on Sunway TaihuLight. Int Symp on Parallel and Distributed Processing, p.615-624. https://doi.org/10.1109/IPDPS.2017.20

Fu H, Liao J, Yang J, et al., 2016. The Sunway TaihuLight supercomputer: system and applications. *Sci China Inform Sci*, 59(7):072001. https://doi.org/10.1007/s11432-016-5588-7

Fu H, He C, Chen B, et al., 2017a. 18.9-PFlops nonlinear earthquake simulation on Sunway TaihuLight: enabling depiction of 18-Hz and 8-meter scenarios. Int Conf for High Performance Computing, Networking, Storage, and Analysis, p.1-12. https://doi.org/10.1145/3126908.3126910

Fu H, Liao J, Ding N, et al., 2017b. Redesigning CAM-SE for peta-scale climate modeling performance and ultra-high resolution on Sunway TaihuLight. Int Conf for High Performance Computing, Networking, Storage, and Analysis, p.1-12. https://doi.org/10.1145/3126908.3126909

Gan L, Fu H, Luk W, et al., 2013. Accelerating solvers for global atmospheric equations through mixed-precision data flow engine. 23rd Int Conf on Field Programmable

Logic and Applications, p.1-6.
https://doi.org/10.1109/FPL.2013.6645508

Gan L, Fu H, Yang C, et al., 2014. A highly-efficient and green data flow engine for solving Euler atmospheric equations. 24th Int Conf on Field Programmable Logic and Applications, p.1-6.
https://doi.org/10.1109/FPL.2014.6927462

Palmer T, 2015. Modelling: build imprecise supercomputers. *Nature*, 526(7571):32-33.
https://doi.org/10.1038/526032a

Roten D, Cui Y, Olsen KB, et al., 2016. High-frequency nonlinear earthquake simulations on peta-scale heterogeneous supercomputers. Int Conf for High Performance Computing, Networking, Storage, and Analysis, p.1-12. https://doi.org/10.1109/SC.2016.81

Shalf J, Dosanjh S, Morrison J, 2010. Exascale computing technology challenges. In Conf on High Performance Computing for Computational Science, p.1-25.
https://doi.org/10.1007/978-3-642-19328-6_1

Shaw DE, Dror RO, Salmon JK, et al., 2009. Millisecond-scale molecular dynamics simulations on Anton. Int Conf on High Performance Computing Networking, Storage, and Analysis, p.1-11.
https://doi.org/10.1145/1654059.1654126

Shaw DE, Grossman J, Bank JA, et al., 2014. Anton 2: raising the bar for performance and programmability in a special-purpose molecular dynamics supercomputer. Int Conf for High Performance Computing, Networking, Storage, and Analysis, p.41-53.
https://doi.org/10.1109/SC.2014.9

Wells J, Bland B, Nichols J, et al., 2016. Announcing supercomputer summit. Oak Ridge National Lab, Oak Ridge, TN, USA.
https://www.osti.gov/servlets/purl/1259664

Yang C, Xue W, Fu H, et al., 2016. 10M-core scalable fully-implicit solver for non-hydrostatic atmospheric dynamics. Int Conf for High Performance Computing, Networking, Storage, and Analysis, p.57-68.
https://doi.org/10.1109/SC.2016.5