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:

Exploring high-performance processor architecture beyond the exascale*

Xiang-hui XIE‡, Xun JIA

State Key Laboratory of Mathematical Engineering and Advanced Computing, Wuxi 214125, China

E-mail: xie.xianghui@meac-skl.cn; jia.xun@meac-skl.cn

Received July 11, 2018; Revision accepted Sept. 7, 2018; Crosschecked Oct. 10, 2018

Abstract: The ever-increasing need for high performance in scientific computation and engineering applications will push high-performance computing beyond the exascale. As an integral part of a supercomputing system, high-performance processors and their architecture designs are crucial in improving system performance. In this paper, three architecture design goals for high-performance processors beyond the exascale are introduced, including effective performance scaling, efficient resource utilization, and adaptation to diverse applications. Then a high-performance many-core processor architecture with scalar processing and application-specific acceleration (Massa) is proposed, which aims to achieve the above three goals by employing the techniques of distributed computational resources and application-customized hardware. Finally, some future research directions regarding the Massa architecture are discussed.

Key words: High-performance computing; Beyond the exascale; Processor architecture; Application-customized hardware; Distributed computational resources

https://doi.org/10.1631/FITEE.1800424 **CLC number:** TP303

1 Introduction

Scientific computation and engineering applications have an insatiable need for computing power. Delivering a performance beyond the exascale (10^{18} double-precision floating-point operations per second) during 2020–2030 will help tackle the most challenging problems in research areas like energy, climate, and astrophysics. As a fundamental building block of a supercomputing system, high-performance processors and their architectures are crucial to system performance, energy efficiency, and applications. Current high-performance processor architectures rely on multi-core and single instruction multiple data (SIMD) techniques to improve performance. Heterogeneous designs, which integrate a general-purpose multi-core processor for irregular computation and a many-core coprocessor for high-throughput computation, are leveraged to achieve a trade-off between performance and power consumption. The tightly coupled on-chip heterogeneous architecture (SW26010 (Xu et al., 2017)) and the loosely coupled inter-chip heterogeneous architecture (Xeon + Xeon Phi/GPU) have become important parts of today's high-performance computing landscape (Schulte et al., 2015).

Existing high-performance processors can deliver teraflop performance (10^{12} floating-point operations per second). With a continuous evolution of processor technology and application of innovative semiconductor devices and materials (Shalf and Leland, 2015), this performance level is expected to scale up by an order of magnitude in the future, addressing the performance needs of supercomputing systems beyond the exascale. However, there are

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 91430214 and 61732018)

ORCID: Xiang-hui XIE, http://orcid.org/0000-0002-2661-0179

several challenges associated with high-performance processor architecture designs beyond the exascale. Enabling a sustainable performance that scales proportionally with the peak performance is challenging. Energy efficiency of current high-performance processors is nowhere near the 50 Gflops/W target; this will be the greatest challenge for processor architecture design beyond the exascale (Hemmert, 2016). In addition, specific performance needs from emerging applications such as data analytics and deep learning are creating additional challenges for high-performance processor architecture designs. Compared to existing processors, architectures beyond the exascale must not only improve performance substantially, but also achieve a balance among performance, energy-efficiency, and application needs.

We propose an inter-chip heterogeneous architecture consisting of a scalar many-core processor and an application-specific coprocessor. It leverages distributed computational resources and application-customized hardware techniques to address high-performance processor architecture design beyond the exascale. The visions and insights presented in this paper may help inspire further research on high-performance, high-efficiency processor architectures in the future.

2 Goals of architecture design for high-performance processor beyond the exascale

Based on the on-chip heterogeneous many-core processor SW26010, the Sunway TaihuLight supercomputing system delivers a sustainable performance of 93 Pflops on the Linpack benchmark, the second highest performance among world-wide supercomputing systems (Fu et al., 2016). Applications like atmospheric dynamics (Yang et al., 2016) and earthquake simulation (Fu et al., 2017) developed on this system have been awarded the Gordon Bell prize. Based on the SW26010 and Sunway TaihuLight, we analyze the processor architecture and applications executed on the system, leading to the following three goals of architecture design for high-performance processors beyond the exascale: effective performance scaling, efficient resource utilization, and adaptation to diverse applications.

The first goal of architecture design for high-performance processors beyond the exascale is effective performance scaling. Scientific computation and engineering applications based on numerical computations represent the dominant applications in supercomputing, and would be major applications for future supercomputers. Therefore, improving numerical computation performance, especially the double-precision floating-point processing, should be the primary target of processor architecture designs beyond the exascale. Current high-performance processors employ the SIMD technique to boost and further scale the peak performance, with the SW26010 and Intel Xeon Phi processor supporting four-way and eight-way double-precision floating-point vector processing, respectively, as an example. Performance acceleration contributed by vector processing from the SW26010 is demonstrated in Table 1. Among the five large-scale scientific computation and engineering applications executed on Sunway TaihuLight, two of them benefit from vector processing optimization, with an acceleration of 1.31 at most. The other three applications remain unaffected due to discrete memory access, transcendental functions, and limited vectorizable computation. Obviously, the vector processing unit in the SW26010 suffers from low utilization, which results in a limited sustainable performance for real-life applications. Similar results occur for Intel processors, because increasing the vector length from 128 bits (SSE) to 256 bits (AVX) and 512 bits (AVX-512) does not translate into much improved performance for most benchmarks in NPB and SPEC CPU2006 (Zhao et al., 2015). Therefore, high-performance processor architecture designs beyond the exascale need a more effective performance scaling technique.

Table 1 Performance acceleration with vector processing optimization

Application	Application area	Speedup
Swe2d	Explicit solver for atmosphere shallow water equation	1.31
3d_fdm	3D seismic wave forward modeling by pseudospectrum method	1.01
Opencfd	Computational fluid dynamics	1.20
Wrf	Numerical weather prediction	1.00
GKUA	Solver for complex hypersonic flow in various regimes	1.00

The second goal of architecture design for high-performance processors beyond the exascale is efficient resource utilization. With dark silicon

getting more and more exacerbated, 50% of a fixed-size chip fabricated on the future 8-nm process technology must be powered off (Esmaeilzadeh et al., 2011), which implies that resource utilization is an important factor for future high-performance processor architecture designs. The current many-core processor architecture integrates a large number of less-complicated general-purpose processing cores on-chip and exploits the application's inherent parallelism and locality through pipelined instruction executions. However, aggressive architecture optimizations, such as multi-level caches, branch prediction, and out-of-order executions, incur prohibitive hardware overheads. Taking the SW26010 many-core processor as an example, computation accounts for only 10% of the total power consumption (Zheng et al., 2014). Moreover, the loss of computational efficiency usually occurs for specific kernels executing on state-of-the-art high-performance many-core processors, due to the underlying general-purpose architecture. According to Table 2, among the representative many-core processors, the SW26010 achieves the highest computational efficiency for double-precision floating-point matrix multiplication, a widely used computation-intensive kernel in scientific computation and engineering applications. However, this efficiency can be further improved within the same cost of computation and storage resources (Jia et al., 2017). Useless power consumption and less-than-optimal computational efficiency indicate that current many-core processor architectures fail to use on-chip resources efficiently, and a more efficient resource utilization technique needs to be considered for high-performance processor architecture designs beyond the exascale.

Table 2 Efficiency of double-precision matrix multiplication on many-core high-performance processors

Many-core processor	Efficiency (%)
PEZY-SC2	89
Intel KNL	90
NVIDIA P100	93
SW26010	95

The last goal of architecture design for high-performance processors beyond the exascale is adaptation to diverse applications. The pillar applications for high-performance computing and their corresponding computation characteristics are summarized in Table 3. Emerging applications, such as data analytics and deep learning, exhibit totally different computation characteristics compared with traditional scientific computation and engineering applications, and create different performance needs for supercomputing systems and the constituent high-performance processors. Apparently, these needs cannot be satisfied by current processor architectures, which are oriented toward double-precision floating-point numerical computation capabilities. Taking breadth-first search (BFS), a typical memory-intensive kernel for data analytics applications, as an example, because of the low and irregular memory access, the system-scale computational throughput of Sunway TaihuLight (Lin et al., 2017) is 23 755.7 giga traversed edges per second (GTEPS), corresponding to only 0.019% of the system's peak performance. As for deep learning, SW26010's performance in convolutional neural network training based on the swDNN library (Fang et al., 2017) is two orders of magnitude lower than that of Google TPU2, due to the processor's lack of abundant memory bandwidth and support for limited numerical precision. The gap between the capability of current high-performance processors and specific performance needs of the emerging applications remains to be bridged by a processor architecture that is adaptive to diverse applications beyond the exascale.

Based on the above analysis, we conclude that high-performance processor architecture design beyond the exascale must achieve an appropriate balance among computational capability, resource utilization, and application needs.

3 High-performance many-core processor architecture with scalar processing and application-specific acceleration (Massa)

In effort to achieve the aforementioned goals of architecture design for high-performance processors beyond the exascale, a many-core processor architecture with scalar processing and application-specific acceleration, referred to as Massa, is proposed in this study. As illustrated in Fig. 1, the Massa architecture is an inter-chip heterogeneous design, consisting of a host processor and a coprocessor. The host processor is based on a many-core architecture,

Table 3 Applications and the corresponding computation characteristics

Application	Computational characteristics
Scientific computation and engineering	Numerical computation on real numbers, especially double-precision floating-point
Data analytics	Graph computation on correlations
Deep learning	Tensor computation on single- or half-precision floating-point data and integers

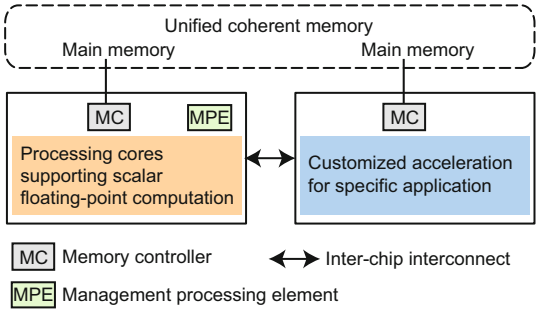

Fig. 1 The high-performance many-core processor architecture with scalar processing and application-specific acceleration (Massa)

such as the on-chip deeply fused many-core architecture (Zheng et al., 2015) featured in SW26010, with each processing core supporting scalar double-precision floating-point processing instead of vector processing. The coprocessor provides acceleration for specific applications, such as traditional scientific computation and engineering applications, and emerging applications including data analytics and deep learning. Furthermore, the host processor and the coprocessor in Massa communicate through an inter-chip interconnect, and these two chips share a unified coherent memory space and programming framework. It is worth noting that the Massa architecture is generally applicable to any supercomputing system, although this architecture is originated from the SW26010 processor and the Sunway TaihuLight system.

The computing processing elements (CPEs), i.e., the processing cores of the host processor in Massa, support scalar double-precision floating-point processing and thread-level parallelism. Modest performance needs for numerical computations from major scientific computation and engineering applications, represented by 3d_fdm, Opencfd, and GKUA in Table 1, and other applications, are simply met by the host processor running alone, while extreme performance needs from specific applications can be addressed by the coprocessor in Massa. In this case, the number of processing cores in the host processor should be carefully chosen to provide a sufficient but not excessive performance, which needs to be further explored in the future. Compared with the performance scaling technique based on vector processing, Massa incorporates performance needs more appropriately in the processor architecture design through a differentiated distribution of computational resources between the host processor and the coprocessor. Also, the application-specific acceleration is more user-friendly and efficient than the fine-grained scheduling and execution of SIMD instructions, which is ultimately beneficial in improving the sustainable performance of supercomputing systems. Distributed computational resources combined with application-specific acceleration lead to an effective performance scaling technique for high-performance processor architectures beyond the exascale.

The application-customized hardware technique is exploited by the coprocessor in Massa to provide acceleration for demanding applications. Given a targeted application, the type and order of computations to be performed are determined. By customization, data and control paths to realize these computations can be directly fixed in the hardware, which simplifies the implementation of the coprocessor considerably. Thus, the extra hardware costs generated by the multi-level cache, instruction issuing, and scheduling in traditional general-purpose processor cores are obviated. Recently, application-customized hardware has become an important technique for designing high-performance, high-efficiency computational architectures. Coprocessor or accelerator design based on this technique has attracted significant research interest from both academic and industrial communities, and various accelerators have been proposed. Those for matrix computations in scientific computation and engineering applications (Pedram et al., 2011), graph computation in data analytics (Ozdal et al., 2016), and neural network training in deep learning (Jouppi et al., 2017) have all achieved at least an order of magnitude better performance per unit area than traditional general-purpose processors, contributing to an efficient resource

utilization technique for high-performance processor architectures beyond the exascale.

The inter-chip interconnect in Massa provides a dedicated fabric for the host processor and the co-processor to communicate with each other, and its capability matches the performance of the two chips. Based on a simplified yet efficient communication protocol, the specific performance needed for numerical, graph, or tensor computation can be handled by the many-core host processor coupled with different application-specific coprocessors. Compared with integrating a general-purpose multi-core host processor with a many-core coprocessor, as represented by Xeon Phi or GPU, a much larger variety of performance needs from emerging applications can be met by the Massa architecture at much lower communication overhead and hardware costs, while the realizable advantage of the inter-chip heterogeneous architecture is retained. 'The Machine' architecture proposed by Hewlett-Packard Enterprise also features a dedicated interconnect and customized accelerators for various applications (Williams, 2017). However, the memory-driven computing paradigm of 'The Machine' will cause a fundamental change in conventional programming models, and the complexity of application programming on this system will inevitably increase. In contrast, the Massa architecture's adaption to diverse applications is enabled without sacrificing existing software ecosystems.

Distributed computational resources and application-customized hardware are fundamental in the Massa architecture to achieve the goals of architecture design for high-performance processors beyond the exascale. Applications of these two techniques also bring new research opportunities, where a mechanism that enables a unified coherent memory shared between the host processor and the coprocessor in Massa is the most critical for further exploration. Academic research and open consortiums in industrial communities such as OpenCAPI, Gen-Z, and CCIX (Silbertstein, 2017) have made extensive efforts toward this goal. Mechanisms based on software demand paging and hardware cache coherence have been proposed (García-Flores et al., 2017), but shortcomings of redundant data transfer and excessive hardware cost remain. Future development of a light-weight, coordinated software-hardware mechanism is desirable to solve this problem.

4 Conclusions

Effective performance scaling, efficient resource utilization, and adaptation to diverse applications are three goals of architecture design for high-performance processors beyond the exascale. The Massa architecture proposed in this paper aims to achieve these goals by coupling a scalar processing many-core processor with application-specific coprocessors and employing the techniques of distributed computational resources and application-customized hardware. Future research efforts will be made to explore an alternative unified coherent memory mechanism, design a customized coprocessor for scientific computation and engineering applications, and demonstrate the promise of the Massa architecture by building a prototype system.

Acknowledgements

The authors would like to thank Dr. Gui-ming WU and Dr. Dong WU from the State Key Laboratory of Mathematical Engineering and Advanced Computing, and Dr. Xin LIU from the National Supercomputing Center in Wuxi for their insightful suggestions on this paper.

References

Esmaeilzadeh H, Blem E, Amant RS, et al., 2011. Dark silicon and the end of multicore scaling. 38th Annual Int Symp on Computer Architecture, p.365-376. https://doi.org/10.1145/2000064.2000108

Fang JR, Fu HH, Zhao WL, et al., 2017. swDNN: a library for accelerating deep learning applications on Sunway TaihuLight. 31st Int Parallel and Distributed Processing Symp, p.615-624. https://doi.org/10.1109/IPDPS.2017.20

Fu HH, Liao JF, Yang JZ, et al., 2016. The Sunway TaihuLight supercomputer: system and applications. *Sci China Inform Sci*, 59(7):1-15. https://doi.org/10.1007/s11432-016-5588-7

Fu HH, He CH, Chen BW, et al., 2017. 18.9-Pflops nonlinear earthquake simulation on Sunway TaihuLight: enabling depiction of 18-Hz and 8-meter scenarios. 30th Int Conf for High Performance Computing, Networking, Storage and Analysis, p.1-12. https://doi.org/10.1145/3126908.3126910

García-Flores V, Ayguade E, Peña AJ, 2017. Efficient data sharing on heterogeneous systems. Proc 46th Int Conf on Parallel Processing, p.121-130. https://doi.org/10.1109/ICPP.2017.21

Hemmert S, 2016. Green HPC: from nice to necessity. *Comput Sci Eng*, 12(6):8-10. https://doi.org/10.1109/MCSE.2010.134

Jia X, Wu GM, Xie XH, 2017. A high-performance accelerator for floating-point matrix multiplication. 15th Int Symp on Parallel and Distributed Processing with

Applicatons, p.396-402.
https://doi.org/10.1109/ISPA/IUCC.2017.00063

Jouppi NP, Young C, Patil N, et al., 2017. In-datacenter performance analysis of a tensor processing unit. 44th Annual Int Symp on Computer Architecture, p.1-12.
https://doi.org/10.1145/3079856.3080246

Lin H, Tang XC, Yu BW, et al., 2017. Scalable graph on Sunway TaihuLight with ten million cores. 31st Int Parallel and Distributed Processing Symp, p.635-645.
https://doi.org/10.1109/IPDPS.2017.53

Ozdal MM, Yesil S, Kim T, et al., 2016. Energy efficient architecture for graph analytics accelerators. 43rd Int Symp on Computer Architecture, p.166-177.
https://doi.org/10.1109/ISCA.2016.24

Pedram A, Gerstlauer A, van de Geijn RA, 2011. A high-performance, low-power linear algebra core. 22nd Int Conf on Application-specific System, Architecture and Processors, p.35-42.
https://doi.org/10.1109/ASAP.2011.6043234

Schulte MJ, Ignatowski M, Loh GH, et al., 2015. Achieving exascale capabilities through heterogeneous computing. *IEEE Micro*, 35(4):26-36.
https://doi.org/10.1109/MM.2015.71

Shalf JM, Leland R, 2015. Computing beyond Moore's law. *Computer*, 48(12):14-23.
https://doi.org/10.1109/MC.2015.374

Silbertstein M, 2017. OmniX: an accelerator-centric OS for omni-programmable systems. 16th Workshop on Hot Topics in Operating Systems, p.69-75.
https://doi.org/10.1145/3102980.3102992

Williams RS, 2017. What's next? [The end of Moore's law] *Comput Sci Eng*, 19(2):7-13.
https://doi.org/10.1109/MCSE.2017.31

Xu ZG, Lin J, Matsuoka S, 2017. Benchmarking SW26010 many-core processor. 31st Int Conf on Parallel and Distributed Processing Symp Workshops, p.743-752.
https://doi.org/10.1109/IPDPSW.2017.9

Yang C, Xue W, Fu HH, et al., 2016. 10m-core scalable fully-implicit solver for nonhydrostatic atmospheric dynamics. 29th Int Conf for High Performance Computing, Networking, Storage and Analysis, p.57-68.
https://doi.org/10.1109/SC.2016.5

Zhao B, Gao W, Zhao RC, et al., 2015. Performance evaluation of NPB and SPEC CPU2006 on various SIMD extensions. 1st Int Conf on Big Data Computing and Communications, p.257-272.
https://doi.org/10.1007/978-3-319-22047-5_21

Zheng F, Zhang K, Wu GM, et al., 2014. Architecture techniques of many-core processor for energy-efficient in high performance computing. *Chin J Comput*, 37(10):2176-2186 (in Chinese).
https://doi.org/10.3724/SP.J.1016.2014.02176

Zheng F, Li HL, Lv H, et al., 2015. Cooperative computing techniques for a deeply fused and heterogeneous many-core processor architecture. *J Comput Sci Technol*, 30(1):145-162.
https://doi.org/10.1007/s11390-015-1510-9