



Editorial for the special issue on operating systems and programming systems for HPC

Xiaobing Feng¹ · Minyi Guo²

Published online: 13 November 2020
© China Computer Federation (CCF) 2020

With the coming of exascale computing era, programming systems and operating systems (including runtime systems) are facing several challenges. In aspect of architecture, increasing deeper level of parallelism, heterogeneity, and the adoption of diverse domain specific accelerators raise the urgent need for programmability, performance optimization and portability. On the other side, big data analytics and machine learning applications demand to be ported and optimized on modern HPC systems. This issue focuses on the novel ideas, methods, as well as efforts of system software development for resolving the above challenges, and to fill the gap between applications and the underlying hardware systems.

We have eight invited papers selected for this special issue based on a peer-review procedure, which cover a number of different aspects that relate to programming systems, operating or runtime systems challenges mentioned above.

The first part of the special issue focuses on the improvements of programming systems for contemporary large scale HPC systems. We have four papers that discuss programming system innovations covering traditional HPC applications and deep learning area, tackling inter-node parallel scalability and intra-node processor heterogeneity, addressing user programmability and performance challenges.

- The first paper written by Li Chen et al. presents AceMesh, a task-based data-driven programming language targeting legacy MPI applications. The new language features not only relieve the programmer from

tedious refactoring efforts but also provide possibility for structured execution of complex task graphs, data locality exploitation, and less runtime overhead. Related compiling and runtime optimizations are also presented, and evaluation on two supercomputing platforms shows its performance superiority to existing programming models.

- The second paper written by Libo Zhang et al. proposes an automatic mapping technique for OpenACC kernel codes on heterogeneous, deeply fused many-core architecture. Static compiling analysis is integrated with dynamic feedbacks. Experimental results show that the approach gets similar performance to the manual annotated approach.
- The third paper written by Zihan Liu et al. focuses on how to exploit the performance potential of deep learning inference accelerators in a compiler tool chain. It introduces an operator level SDK for a specialized hardware accelerator (Cambricon) and proposes a middle layer compiler tool-chain. The tool chain not only provides enough abstraction level, but also exposes major optimization knobs. Experiments show its great potential in optimization compared to the existing runtime.
- The last paper written by GuoFeng Lv et al. centers around how to optimize distributed DNN (deep neural networks) training on the world-leading supercomputer, Sunway TaihuLight. Several distributed algorithms are implemented upon swFlow, different communication strategies are analyzed, multi-severs and hierarchical ring all-reduce models are introduced. With a benchmark from deep learning-based cancerous region detection algorithm, good parallel efficiency are obtained for at most 1024 processors, revealing the great opportunity for joint combination of deep learning and HPC system.

✉ Xiaobing Feng
fxb@ict.ac.cn

Minyi Guo
myguo@sjtu.edu.cn

¹ State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

² Shanghai Jiaotong University, Shanghai, China

The second part of the special issue, consisting of two research papers, focuses on the runtime system and operating system support for HPC applications. Customized

heterogeneous architectures and characterized execution models call for innovative runtime techniques to fully exploit parallelism from applications and map them upon hardwares. The heterogeneity of many-core processor also brings about new techniques to the operating system.

- The first paper written by Qingxiang Chen et al. focuses on runtime system support for specialized graph processing accelerators. Targeting a heterogeneous architecture with both dataflow and control-flow execution models, a novel runtime system is designed to adaptively offload each subgraph to an appropriate processor, accompanied by a frontier-sensitive graph partition approach to maximize the parallelism. Results show that the approach achieves higher throughput compared to a state-of-art design.
- The second paper written by Yan Zheng et al. focuses on the memory management challenges in the homegrown heterogeneous many-core processor. It put forwards a segment-page-combined memory management technique to overcome the related obstacles. Evaluation shows its performance advantage.

The last part of the special issue is two survey papers on programming models and runtime system support. Heterogeneous many-core processors are now an integral part of modern computing systems, and this brings about plentiful research efforts on programming models (or frameworks), code transformation tools and optimizing techniques. Hardware transactional memory which provides optimistic concurrency control schemes, calls for runtime innovations on effective fallback mechanism.

- The paper written by Jianbin Fang and his colleagues in National University of Defense Technology, and Prof. Zheng Wang from University of Leeds, provides a comprehensive survey for parallel programming models or systems for heterogeneous many-core architectures, reviews the related compiling and optimization techniques, and also points out challenges and potential research directions.
- The paper written by Zhenwei Wu and his colleagues in National University of Defense Technology, discusses software-side efforts that enforcing progress guarantees for commodity best-effort hardware transactional mem-

ory, the profiling and performance tuning techniques, and research efforts about joint usage of HTM and non-volatile memory.

We would like to take this chance to thank all the authors and the reviewers for their brilliant contribution to this special issue of CCF THPC. Only with their great efforts, we are able to put together the eight research papers that discuss different topics, and present different ideas that help to bridge HPC applications and the underlying parallel architectures.



Xiaobing Feng is currently a professor of Institute of Computing Technology, Chinese Academy of Sciences and a professor of University of Chinese Academy of Sciences. His present research interests include programming model, programming analysis, compiler optimization. He is a distinguished member of CCF.



Minyi Guo is currently Zhiyuan Chair professor of Shanghai Jiao Tong University (SJTU), China. Before joined SJTU, Dr. Guo had been a professor of the school of computer science and engineering, University of Aizu, Japan. Dr. Guo received the national science fund for distinguished young scholars from NSFC in 2007. His present research interests include parallel/distributed computing, compiler optimizations, big data and cloud computing. He has more than 400 publications in major journals

and international conferences in these areas. He received seven best/highlight paper awards from international conferences including ALS-POS 2017 and ICCD 2018. He is now on the editorial board of *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Cloud Computing* and *Journal of Parallel and Distributed Computing*. Dr. Guo is a member of Academy of Europe, a fellow of IEEE, a fellow of CCF, and a distinguished member of ACM.