J Supercomput (2012) 61:1–5
DOI 10.1007/s11227-012-0745-2

Editorial: enabling technologies for programming extreme scale systems

Ching-Hsien Hsu

Published online: 2 February 2012

Multi-core architecture presents a new trend, and core-based parallel processing algorithms will continue to become more important. In addition, Grid and Cloud Computing (GCC) has emerged rapidly as an exciting new paradigm that offers a challenging model of computing and poses fascinating problems ranging from parallel architecture to programming models and compiler optimization. Furthermore, the wide-scale deployment of computing nodes, clusters, and embedded processors will continue to fuel core parallel processing and distributed systems algorithms. Thus, the importance of scalable parallel programming will continue to grow. The field of parallel and distributed computing is going through rapid changes. The emerging multi-core commodity processors and advanced networking technologies are allowing for the design of cost-effective parallel and distributed systems. To achieve scalability and high performance in these systems, it is increasingly becoming challenging to design better architecture, software environments, networking protocols, compilers, operating systems, languages, algorithms and applications for designing next generation scalable and high-performance parallel systems. This special issue is intended to foster state-of-the-art research in the area of extreme scale parallel systems including the topics of parallel algorithm design, data flow analysis, virtualization, energy efficient, load balancing, parallel computing model, multi-core programming, GPU implementation and optimization, execution on real-world parallel architecture and novel applications associated with this new paradigm.

The papers included in this special issue demonstrate the effectiveness and efficiency of a variety of technologies and applications in parallel, grid, cloud and extreme scale systems. All of them not only contribute valuable insights and results but

C.-H. Hsu (✉)
Department of Computer Science and Information Engineering, Chung Hua University, Hsinchu, Taiwan
e-mail: chh@chu.edu.tw

also have particular relevance to the high-performance computing community. They also present high-quality results for tackling problems arising from the ever-growing scalable computing technologies.

The first paper by Byoung Uk Kim entitled "Data Flow Analysis for Anomaly Detection and Identification Toward Resiliency in Extreme Scale Systems" presents innovative concepts for anomaly detection and identification, analyzing the duration of pattern transition sequences of an execution window. The author uses a three-dimensional array of features to capture spatial and temporal variability to be used by an anomaly analysis system to immediately generate an alert and identify the source of faults when an abnormal behavior pattern is captured, indicating some kind of software or hardware failure. Experimental results show a detection rate of above 99.9% with no occurrences of false alarms for a wide range of scenarios, and accuracy rate of 100% with short root cause analysis time.

The second paper by Xuanhua Shi, Hai Jin, Hongbo Jiang, Xiaodong Pan, Dachuan Huang and Bo Yu entitled "Toward Scalable Web Systems on Multi-core Clusters: Making Use of Virtual Machines" proposes design of a VM-based Web system on multi-core clusters. The VM-based Web system is scheduled by LVS, and the Web server implementation uses Tomcat. The authors have developed a set of virtual machine management toolkit, called VNIX, that facilitates the administration of virtual machines on clusters. The VNIX can be used to deploy VMs with LVS and Tomcat, and to improve the resource utilization of multi-core cluster. This study conducted extensive experiments to compare the performance of a VM-based Web system and a classical Web system on physical machines. The results show that VM-based Web systems works about three times faster than classical Web systems on the same multi-core clusters, and the error rate of the client requests to the VM-based Web systems is about 20% of that of classical Web systems.

The third paper by Mohsen Sharifi, Hadi Salimi and Mahsa Najafzadeh entitled "Power-Efficient Distributed Scheduling of Virtual Machines using Workload-Aware Consolidation Techniques" proposed an algorithm to schedule the workload of a set of virtual machines (VMs) to a set of physical machines (PMs) in a data center. The goal was to minimize total energy consumption by considering the conflicts between processor and disk utilizations and the costs of migrating VMs. To identify the conflicts, the authors presented four models, namely the target system model, the application model, the energy model and the migration model. Given that workload conflicts in virtualized environments depends on many factors such as the virtualization software, the authors carried out three set of experiments on their configured KVM target infrastructure to measure the performance isolation of VMs in three types of workload: processor intensive workloads, disk intensive workloads, and consolidation of disk and processor intensive workloads. They also presented a consolidation fitness (CF) criterion that the proposed scheduler used to merit the consolidation of a set of VMs on a PM before deciding on any particular scheduling of VMs. The experimental results showed 24.9% power savings compared to two other methods.

The fourth paper, by Wei-Jen Wang, Yue-Shan Chang, Cheng-Hui Wu and Wei-Xiang Kang entitled "A Self-Adaptive Computing Framework for Parallel Maximum Likelihood Evaluation", presents a self-adaptive and parallelized maximum likelihood evaluation (MLE) framework that consists of a master process and a set of worker processes on a distributed environment. The workers are responsible for computing tasks, while the master needs to merge the computing results, to initiate or to terminate another computing iteration, and to decide how to re-distribute the computing tasks to workers. Given a statistics model, the proposed MLE framework can iteratively fine-tune the model parameters to fit the observed data items, and re-distribute the workload of computation to the worker processes. The experimental results show that not only the proposed framework can adapt to environmental changes, but also the proposed framework is effective; even in a stable environment that is dedicated for one application.

The fifth paper by Zhiyong Yuan, Weixin Si, Xiangyun Liao, Zhaoliang Duan, Yihua Ding and Jianhui Zhao entitled "Parallel Computing of 3D Smoking Simulation Based on OpenCL Heterogeneous Platform" introduces OpenCL to implement a real-time smoking simulation in a virtual surgery training simulation system. The Computational Fluid Dynamics (CFD) was adopted to construct the real-time smoking simulation model based on the Navier-Stokes (N-S) equations of incompressible fluid under the condition of normal temperature and pressure. By introducing OpenCL parallel computing model, the authors proposed 3D smoking simulation parallel computing model based on OpenCL and designed the parallel algorithm based on OpenCL according to the GPU platform property. The experimental results show that the proposed smoking simulation model and parallel computing algorithm can meet the requirements of timeliness and visual effect of a real-time physical model.

The sixth paper by Jia-Jhe Li, Chung-Kai Chen, Tung-Yu Wu and Jenq-Kuen Lee entitled "Case Study: Stereo Vision Experiments with Multi-core Software API on Embedded MPSoC Environments" describes a case study on the parallelization of belief propagation for stereo matching using the "Multi-core Software APIs" (MSA) on embedded MPSoC environments. MSA is a library-based middleware providing an asynchronous remote procedure call (RPC) mechanism. It supplies a function-offloading programming model to hide the underlying inter-processor communication and configuration detail from programmers. MSA provides a set of stream-specific APIs for supporting a streaming function remoting mechanism on heterogeneous multi-core architectures. The experiments show that the belief propagation method for stereo matching can be adapted from a single core program to a multi-core one for embedded MPSoC environments rapidly.

The seventh paper by Jingun Hong, Kirak Hong, Bernd Burgstaller and Johann Blieberger entitled "StreamPI: A Stream-Parallel Programming Extension for Object-Oriented Programming Languages" proposed a stream-parallel programming abstraction to extend object-oriented programming languages with stream-programming facilities. It provides a class-hierarchy for actor-specification that is embedded in an object oriented host language. Actors use the underlying run-time system through a language-specific binding. The run-time system itself is language independent and manages actor allocation, stream graph creation and stream program execution on multi-core architectures. Implementation of StreamPI on Intel multi-core architectures for C++ and Ada 2005 shows the effectiveness on an Intel Xeon X7560 server with 2 CPUs and 8 cores per CPU.

The eighth paper by Roberto R. Exposito, Guillermo L. Taboada, Juan Tourino and Ramon Doallo entitled "Design of scalable Java message-passing communications over InfiniBand" presents a scalable and efficient low-level Java message-passing device for communication on InfiniBand systems. The increase in the number of cores per system demands languages with built-in multithreading and networking support, such as Java, as well as scalable and efficient communication middleware that can take advantage of multi-core systems. The developed communication middleware increases Java applications performance on clusters of multi-core processors interconnected via InfiniBand. The performance evaluation on an InfiniBand multi-core cluster has shown that the middleware obtains start-up latencies and bandwidths similar to MPI performance results, obtaining in fact up to 85% start-up latency reduction and twice the bandwidth compared to previous Java middleware on InfiniBand.

The ninth paper by Fanyang, Naixue Xiong and Jong Hyuk Park entitled "A Self-adaptive K Selection Mechanism for Re-Authentication Load Balancing in Large Scale Systems" studied the IEEE 802.16e's re-authentication definition and proposed a self-adaptive K value selection scheme for optimizing load balancing in large scale network system. The proposed mechanism takes the cost of authentication into consideration not only at the server end, but also at the client end. This scheme could minimize the total cost and resolve the limitation in current schemes. The simulation results and relevant analysis demonstrate that such a scheme is effective in terms of the total cost of authentication, master Key renewal, and good security.

The tenth paper by Keqin Li entitled "Optimal Configuration of a Multicore Server Processor for Managing the Power and Performance Tradeoff" studied the problem of power and performance management for a multi-core server processor in a cloud computing environment by optimal server configuration for a specific application environment. This study provides several valuable concluding remarks, such as that cores should be managed in a centralized way to provide the highest performance; for a given server speed constraint, fewer high-speed cores perform better than more low-speed cores; or a given power consumption constraint, there is an optimal selection of server size and core speed; for a given task response time constraint, there is an optimal selection of server size and core speed.

The eleventh paper by Jonathan Cazalas and Ratan Guha entitled "Leveraging Computation Sharing and Parallel Processing in Location-Dependent Query Processing" introduces an efficient and scalable system for monitoring continuous queries by leveraging the parallel processing capability of the Graphics Processing Unit. The authors examine a naive CPU-based solution for continuous range-monitoring queries, and then extend this system using the GPU. In addition, the authors propose a view oriented approach of the location database, thereby reducing computation costs by exploiting computation sharing amongst queries requiring the same view. This study shows that by exploiting the parallel processing power of the GPU, we are able to significantly scale the number of mobile objects, while maintaining an acceptable level of performance.

In the last paper by Victor Malyshkin and Vladislav Perepelkin "Optimization Methods of Parallel Execution of Numerical Programs in the LuNA Fragmented Programming System", methods and algorithms of optimizing the execution of fragmented programs are proposed to be included into the LuNA fragmented programming language, compiler, generator and run-time system. These optimizations provide a hardware undependable representation of numerical algorithms and their portability among a wide range of multi-computers. The numerical algorithms exploit a limited number of the spatial data structures, like vectors, matrices, arrays, 3D meshes. The LuNA supports an explicit declaration of such data structures and implements a number of algorithms to perform the initial distribution and structure-keeping dynamic load balancing on commonly used hardware network topologies, like a 3D torus, a cluster or a complete graph. The experimental results show that the load balancing algorithm works more accurate.

All of the above papers address either technical issues in parallel and distributed computing systems or propose novel application models in the various scalable computing fields. They also trigger further related research and technology improvements in application and services of grid and cloud systems. This special issue serves as a landmark source for education, information, and reference to professors, researchers and graduate students interested in updating their knowledge about or activities in application models for scalable, parallel and distributed computing systems.

The guest editor would like to express sincere gratitude to Prof. Hamid R. Arabnia, the Editor-in-Chief of the *Journal of Supercomputing*, for giving us the opportunity to prepare this special issue. In addition, I am deeply indebted to numerous reviewers for their professional effort, insight and hard work put into commenting on the selected articles which reflect the essence of this special issue. Last but not least, I am grateful to all authors for their contributions and for undertaking a two-cycle revision of their manuscripts, without which this special section could not have been produced.

Finally, we hope that you will enjoy reading these selected papers as much as we did and that you will find this issue informative and helpful in keeping yourselves up-to-date in the fast changing field of the “Scalable Computing Era”.