Received: 12 July 2018 | Accepted: 15 July 2018
DOI: 10.1002/cpe.4926

Check for updates

WILEY

EDITORIAL

Special issue on High performance computing conference (BASARIM-2017)

1 | INTRODUCTION

Numerical operations and calculations have been studied on parallel systems for many years for the purpose of achieving faster results and better performance. In recent years, these studies and their results have reached a certain level and the progress made in this regard has accelerated. The Turkish High Performance Computing Conference has been organized since 2009 for discussing challenges in high performance computing and identifying opportunities in the areas of cloud computing and big data processing for moving ahead. To this end, the integration of big data processing software stack and traditional parallel computing message passing protocols has been addressed in the keynote talk "HPC-enhanced IoT and Data-based Grid" by Geoffrey Charles Fox at the 5th Turkish High Performance Computing Conference (BASARIM-2017).

With the objective of sharing and evaluating scientific research, experience, studies, and results related to high performance computing at this scientific event, this special issue presents the highlights of the program and selected high-quality papers from BASARIM-2017.

2 | SPECIAL ISSUE PAPERS

Aktemur[1] presents a new sparse matrix-vector multiplication (SpMV) implementation, named CSRLenGoto, which is based on complete loop unrolling. The method provides performance improvements especially for matrices whose row length is short. CSRLenGoto incurs an inexpensive preprocessing phase that is compensated in just a few SpMV repetitions. The author parallelized CSRLenGoto and integrated the operation into a state-of-the-art matrix partitioning approach as the kernel operation. The author observed up to 2.46× and on the average 1.29× speedup with respect to highly optimized Intel MKL's SpMV for matrices with short- or medium-length rows.

Topcuoglu et al[2] present a generic private information retrieval (PIR) scheme with parallel multi-exponentiations for multicore architectures. Without revealing any information as to which a data item is requested, PIR allows the data owners to share and/or retrieve data on remote repositories. Homomorphic cryptosystems are commonly exploited for PIR in the literature. Those approaches require not one but many modular exponentiations need to be computed and multiplied to obtain the desired result. The multi-exponentiation operation can be implemented by exponentiating the bases to their corresponding exponents one-by-one and the operation is extremely easy to parallelize. However, when the operation is considered as a whole, it can be computed in a more performance efficient way, but the combined multi-exponentiation is not straightforward to parallelize. Topcuoglu et al propose a generic tensor-based PIR scheme that is efficient and novel to parallelize multi-exponentiations on multicore processors with perfect load balance. The evaluation results demonstrate that the proposed load balancing methods make a parallel multi-exponentiation faster.

Mumcuyan et al[3] have developed a method for optimally bipartitioning sparse matrices with reordering and parallelism. Because communication between tasks can constitute one of the bottlenecks for scalability, an efficient task-to-processor assignment is crucial for performance. A main solution to this problem is to model the tasks as a hypergraph where the pins and nets represent the tasks and the communication among them, respectively. The hypergraph vertices are partitioned into a number of parts, which correspond to processors, in a way that the total number of vertices for each part is balanced and the amount of edges having endpoints in different parts is optimized to be minimal. Recently, for solving sparse matrix bipartitioning, a novel purely combinatorial approach has been proposed. The approach is based on branch-and-bound and can handle hypergraphs that cannot be optimally partitioned by using existing methods because of the problem's complexity. The work of Mumcuyan et al built on top of the previous study with three new ideas. (1) It applies matrix ordering methods to use more information in the earlier branches of the tree, (2) it leverages machine learning to select an ordering based on the matrix features, and (3) it parallelizes the search of an optimal bipartitioning. The results show that their techniques make the bipartitioning sparse matrices significantly faster.

Dikbayir et al[4] have developed two parallel and efficient implementations of broadcast and reduction operations for multidimensional arrays on GPU devices. Reduction and broadcast operations are commonly used in machine learning, especially when performing forward and backward passes in deep neural networks. Existing parallel methods usually developed for scalar reduction but with the increasing size of data and dimensionality, the need for implementations suitable for multidimensional tensors has emerged. The authors first set up a terminology for their methods and define both of operations mathematically in the high-dimensional space. They then analyze and evaluate the mathematical nature of the

 © 2018 John Wiley & Sons, Ltd.

original algorithms in order to exploit any properties. For reduction, they take advantage of its associativity property to reduce multiple dimensions within a single kernel launch to minimize the amount of synchronizations needed in the process. For broadcast, they adapt data reuse in order to avoid replication of the input data in memory. Both of the implementations use coordinate calculation and index translation algorithms in order to map the GPU threads to the corresponding data elements of the multidimensional input tensors. The authors evaluate the algorithms implemented in CUDA on the NVIDIA K40 GPU accelerator. They compare the performance of the methods to the existing implementations in Knet, which is a deep learning framework. The new implementations are able to obtain up to 56x speed up over Knet while reaching %75 of the theoretical limit for memory bandwidth rate of K40 device for test scenarios involving large multidimensional tensors.

Guler and Ozkasap[5] address combining checkpointing techniques with primary-backup replication protocol to further improve efficiency in terms of client blocking time and overall system throughput. For this purpose, they propose an advanced primary-backup replication protocol, which minimizes the failover time by eliminating the recovery process in the event of rollback operation. The authors develop a software framework for a geographically replicated key-value store based on RocksDB and use the PlanetLab overlay network to execute the proposed primary-backup replication protocol. They conduct a thorough analysis of various checkpointing algorithms integrated with primary-backup replication. Using various metrics of interest including blocking time, checkpointing time, checkpoint size, failover time and throughput, and testing with realistic workloads, their findings indicate that the proposed primary-backup replication protocol, supported by Snappy-compressed-periodic-incremental-checkpointing technique, provides significant improvements in the system throughput and reduced blocking times compared to the traditional primary-backup replication protocol.

Soomro et al[6] propose BindMe, which is a thread binding library that supports advanced mapping algorithms. Current multicore machines contain a large number of cores and the number of cores is expected to increase in upcoming exascale multicore machines. Binding parallel tasks to cores according to a placement policy is one of the important performance boosting factors, because it can reduce on-chip communication among parallel threads. Binding also prevents operating system from migrating threads, which improves data locality. However, there is no single mapping policy that works best among all different kinds of applications and platforms because each machine has a different topology and each application exhibits different communication pattern. Determining the best policy for a given application and machine requires extra programming effort. The authors introduce the BindMe, a thread binding library that assists programmer to bind threads to underlying hardware. BindMe incorporates state-of-the-art mapping algorithms, which analyze communication pattern of an application to formulate a task placement policy. The authors also introduce ChoiceMap, a communication aware mapping algorithm that respects mutual priorities of parallel tasks and performs a fair mapping by reducing communication volume among cores. Their results show that choosing a mapping policy that best suits the application behavior can increase its performance and no single policy gives the best performance across different applications.

Fisne and Ozsoy[7] provide a real-time running software defined radio (SDR) with its full pipeline steps running on GPGPUs. Initially, they port the data preprocess step on to GPUs with Big Endian/Little Endian Conversion and Short/Complex Type Conversion. Second, FFT Process for Spectrum and Spectogram is ported on GPU. Signal Detection is achieved afterwards and now the wideband data is ready for down conversion to be reduced to narrowband data. For DDC operation, they have used a different technique then traditional approaches, where using FFT/IFFT blocks for filtering. After downsampling, narrowband data is processed for extracting the sound. FM demodulation and resampling steps are applied for this purpose. For all steps, the algorithms are parallelized both on CPU and GPU, design choices are given, performance results are listed, and analyses are made for reaching to real-time performance. The last demodulation steps are only implemented on CPU since these steps are sufficient enough for real-time requirements. Consequently, their work provides a design of a full running SDR implementation under real-time requirements.

Muhtaroglu et al[8] investigated several design choices for HPC services at different layers of the cloud computing architecture to simplify and broaden its use cases. They compared direct versus iterative parallel linear equation solvers for the platform-as-a-service layer. They observed that several matrix properties, identified before starting long-running solvers, can help HPC services automatically select the amount of computing resources per job, such that the job latency is minimized and the overall job throughput is maximized. They showed that, on top of the 2x-3x speedups gained from parallelization, one can achieve an additional 2x-3x speedup with careful selection of solver types and preconditioner combinations. They explored HPC application performance, load isolation, and deployment issues using application containers (Docker) while also comparing them to physical and virtual machines for the infrastructure-as-a-service layer. They found that Dockerized HPC can be setup and deployed much faster than physical or virtual HPC alternatives and its performance is comparable.

Baeth and Aktas[9] introduced a generic software architecture that can be integrated with existing social media software to enable users to track the dissemination of their data and generate special notifications by using complex event processing. Their solution utilizes social provenance data. The proposed architecture is designed to detect the candidate social media user accounts for copyright violations. They developed a prototype of this proposed architecture. The developed system has a set of extendible facade classes responsible for hiding the complexities of the utilized streaming and complex event processing engines. In turn, this approach makes their implementation highly decoupled. In addition, it adds an extra level of layers segregation by making it much easier to switch to different tools, libraries, and technologies. To facilitate testing of the software architecture, they developed a large-scale synthetic provenance dataset, discussed the details of the prototype implementation, and evaluated its performance. Their prototype performed well with the ingested large number of provenance workflows because the processing overhead is negligible.

Deep learning has emerged as an effective solution to various text mining problems such as document classification and clustering, document summarization, web mining, and sentiment analysis. Karakus et al[10] investigate several deep learning models for binary sentiment classification problem. They report a detailed comparison of the models in terms of accuracy and time performances. Two major deep learning architectures used

in their study are Convolutional Neural Networks and Long Short-Term Memory. Karakus et al built several variants of these models by changing the number of layers, tuning the hyper-parameters, and combining models. They investigated the effect of using the pre-word embeddings with these models. Their experimental results have shown that the use of word embeddings with deep neural networks effectively yields performance improvements in terms of run-time and accuracy.

Cloud computing provides scalable computing resources on demand. Challenges in cloud computing monitoring systems include detecting patterns that might lead to failure of the cloud system, detecting malfunctioning problems within the cloud platform after they occur, and issues related to the fact that existing monitoring solutions are tightly coupled to specific cloud platforms. To address these challenges, Aktas[11] designed a hybrid cloud monitoring software architecture that can work as an add-on layer on top of existing cloud computing platforms. The proposed architecture is designed based on facade software design pattern and utilizes complex event processing concept in which data from various primitive metrics streams are processed to detect previously defined patterns. Prototype applications have been developed to demonstrate the architecture's usability. Performance tests were applied to prototype applications. Computation times required for the operation of the proposed architecture were found to be negligible.

ORCID

Mehmet S. Aktas http://orcid.org/0000-0001-7908-5067

Didem Unat[1]
Mehmet S. Aktas[2]

[1]*Department of Computer Engineering, Koç University, Istanbul, Turkey*
[2]*Department of Computer Engineering, Yildiz Technical University, Istanbul, Turkey*

Correspondence
Mehmet S. Aktas, Department of Computer Engineering, Yildiz Technical University, 34220 Istanbul, Turkey.
Email: aktas@yildiz.edu.tr

REFERENCES

1. Aktemur B. A sparse matrix-vector multiplication method with low preprocessing cost. *Concurrency Computat Pract Exper*. 2018;30:e4701. https://doi.org/10.1002/cpe.4701
2. Topcuoglu C, Kaya K, Savas E. A generic private information retrieval scheme with parallel multi-exponentiations on multicore processors. *Concurrency Computat Pract Exper*. 2018;30:e4685. https://doi.org/10.1002/cpe.4685
3. Mumcuyan A, Usta B, Kaya K, Yenigun H. Optimally bipartitioning sparse matrices with reordering and parallelization. *Concurrency Computat Pract Exper*. 2018;30:e4687. https://doi.org/10.1002/cpe.4687
4. Dikbayir D, Coban EB, Kesen I, Yuret D, Unat D. Fast multidimensional reduction and broadcast operations on GPU for machine learning. *Concurrency Computat Pract Exper*. 2018;30:e4691. https://doi.org/10.1002/cpe.4691
5. Guler B, Ozkasap O. Efficient checkpointing mechanisms for primary-backup replication on the cloud. *Concurrency Computat Pract Exper*. 2018;30:e4707. https://doi.org/10.1002/cpe.4707
6. Soomro PN, Sasongko MA, Unat D. BindMe: a thread binding library with advanced mapping algorithms. *Concurrency Computat Pract Exper*. 2018;30:e4692. https://doi.org/10.1002/cpe.4692
7. Fisne A, Ozsoy A. Design and implementation of real time wideband software defined radio applications with GPGPUs. *Concurrency Computat Pract Exper*. 2018;30:e4791. https://doi.org/10.1002/cpe.4791
8. Muhtaroglu N, Ari I, Kolcu B. Democratization of HPC cloud services with automated parallel solvers and application containers. *Concurrency Computat Pract Exper*. 2018;30:e4782. https://doi.org/10.1002/cpe.4782
9. Baeth MJ, Aktas MS. An approach to custom privacy policy violation detection problems using big social provenance data. *Concurrency Computat Pract Exper*. 2018;30:e4690. https://doi.org/10.1002/cpe.4690
10. Karakus BA, Talo M, Hallac IR, Aydin G. Evaluating deep learning models for sentiment classification. *Concurrency Computat Pract Exper*. 2018;30:e4783. https://doi.org/10.1002/cpe.4783
11. Aktas MS. Hybrid cloud computing monitoring software architecture. *Concurrency Computat Pract Exper*. 2018;30:e4694. https://doi.org/10.1002/cpe.4694