

Guest Editorial: The Parallel Storage, Processing and Analysis for Big Data

Maozhen Li¹ · Zhuo Tang²

Received: 17 October 2016 / Accepted: 5 November 2016 / Published online: 16 November 2016
© Springer Science+Business Media New York 2016

Big data has received a significant amount of attention from both the academia and industry since the US government announced the Big Data Initiative in 2012. Big data is featured with a number of Vs include high volume, velocity, variety and veracity. The purpose of the special issue is to collate a selection of representative articles that were primarily presented at the 11th International Conference on Natural Computation (ICNC) on 15–17 August 2015 in Zhangjiajie, China. The special issue was also made open to public submissions for a wide inclusion.

MapReduce [1–3] has become a major computing model in support of big data applications especially in dealing with data of huge volumes in an offline mode. As a result, MapReduce has been widely employed in parallelization of data intensive applications. In [4], Wang et al. employ MapReduce to parallel a spatio-temporal convolutional neural network for automatic recognition of action events in surveillance videos. The Map and Reduce processes are implemented on a single computer with multiple CPU cores. Liu et al. [5] further evaluate the performance of MapReduce in computation in comparison with Spark [6], an in-memory computing technology which can be deployed on MapReduce. A back propagation neural network (BPNN) is parallelized with MapReduce and Spark respectively and the speedup of the parallel BPNN using Spark is significantly faster than the one parallelized with MapReduce. Alternatively, Phan et al. [7] present an approach to speeding up the computation of neural networks inspired by the parallel circuits found in the human retina. Wu

✉ Maozhen Li
Maozhen.Li@brunel.ac.uk

¹ Department of Electronic and Computer Engineering, Brunel University London, Uxbridge UB8 3PH, UK

² School of Information Science and Technology, Hunan University, Changsha 410082, China

and Wang [8] extends MapReduce in support of applications with data dependencies facilitating coarse-grained parallelism.

Modern computers nowadays are equipped with multiple CPU cores. A number of papers are included in this special issue targeting at multi-core computing platforms. In [9], Wang et al. present a partition algorithm for multi-dimensional loop applications on heterogeneous multicore processors. In this work the authors take into account memory access pattern information and fully consider the heterogeneity of processors to achieve high processor utilization. Gu et al. [10] aim to discover the utilization of CPU resources by mining usage patterns. In [11], Liu et al. analyze deadlock conditions in concurrent programs which normally are deployed on multi-core platforms. A Petri net is employed to model concurrent programs and the correlation between the Petri net model and sub-process nets is analyzed. Ouyang et al. [12] look at hardware/software partitioning taking into account communication overhead on heterogeneous multiprocessor system-on-chip platforms.

High performance storage systems are of vital important in support of big data applications. Ou et al. [13] fully exploit flash memory to target at high-performance PCIe SSD for large capacity storage. Deduplication plays an increasing role in eliminating replicas and saving space and network bandwidth in various storage systems. For this purpose, Deng et al. [14] focus on deduplication over small-scale storage systems with adequate bandwidth in between and propose a deduplication system with request-aware placement policy. Du et al. [15] take into account data migration in reduction of power consumption of embedded systems that are empowered with emerging scathed-pad memory and non-volatile memory.

Cloud computing providers such as Amazon still relied on Web services in support of big data applications. In [16], Du et al. look at big data issues in the form of Web services and employ service clusters to deal with the complexity in computation. Sun et al. [17] research the problem of repairing workflow models by employing a mirroring matrix based on the variant of footprints and parallel programs.

We hope that the perspectives presented in this special issue would be of a great interest to the readers. We also encourage the readers to contribute to this exciting and fast growing research area.

Acknowledgements We would like to thank Ms. Renuka Nidhi, an Assistant of the Journals Editorial Office of Springer, for her great support in publication of the special issue.

References

1. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. *Commun. ACM* **51**(1), 107–113 (2008)
2. Jiang, D., Ooi, B.C., Shi, L., Wu, S.: The performance of MapReduce: an in-depth study. *Proc. VLDB Endow.* **3**(1), 472–483 (2010)
3. Panda, B., Herbach, J.S., Basu, S., Bayardo, R.J.: PLANET: massively parallel learning of tree ensembles with MapReduce. *Proc. VLDB Endow.* **2**(2), 1426–1437 (2009)
4. Wang, Q., Zhao, J., Gong, D., Shen, Y., Li, M., Lei, Y.: Parallelizing convolutional neural networks for action event recognition in surveillance videos. *Int. J. Parallel Prog.* (2016). doi:[10.1007/s10766-016-0451-4](https://doi.org/10.1007/s10766-016-0451-4)
5. Liu, Y., Xu, L., Li, M.: The parallelization of back propagation neural network in MapReduce and Spark. *Int. J. Parallel Prog.* (2016). doi:[10.1007/s10766-016-0401-1](https://doi.org/10.1007/s10766-016-0401-1)

6. Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S., Stoica, I.: Spark: cluster computing with working sets. In: Proceedings of the 2nd USENIX Workshop on Hot Topics in Cloud Computing (HotCloud), Boston (2010)
7. Phan, K.T., Maul, T.H., Vu, T.T.: An empirical study on improving the speed and generalization of neural networks using a parallel circuit approach. *Int. J. Parallel Prog.* (2016). doi:[10.1007/s10766-016-0435-4](https://doi.org/10.1007/s10766-016-0435-4)
8. Wu, H.H., Wang, C.M.: Generalization of large-scale data processing in one MapReduce job for coarse-grained parallelism. *Int. J. Parallel Prog.* (2016). doi:[10.1007/s10766-016-0444-3](https://doi.org/10.1007/s10766-016-0444-3)
9. Wang, Y., Li, K., Li, K.: Partition scheduling on heterogeneous multicore processors for multi-dimensional loops applications. *Int. J. Parallel Prog.* (2016). doi:[10.1007/s10766-016-0445-2](https://doi.org/10.1007/s10766-016-0445-2)
10. Gu, Z., He, L., Chang, C., Sun, J., Chen, H., Huang, C.: Developing an efficient pattern discovery method for CPU utilizations of computers. *Int. J. Parallel Prog.* (2016). doi:[10.1007/s10766-016-0439-0](https://doi.org/10.1007/s10766-016-0439-0)
11. Liu, W., Wang, L., Du, Y., Li, M.: Deadlock property analysis of concurrent programs based on Petri Net structure. *Int. J. Parallel Prog.* (2016). doi:[10.1007/s10766-016-0440-7](https://doi.org/10.1007/s10766-016-0440-7)
12. Ouyang, A., Peng, X., Liu, J., Sallam, A.: Hardware/software partitioning for heterogeneous MPSoC considering communication overhead. *Int. J. Parallel Prog.* (2016). doi:[10.1007/s10766-016-0466-x](https://doi.org/10.1007/s10766-016-0466-x)
13. Ou, Y., Xiao, N., Liu, F., Chen, Z., Chen, W., Wu, L.: Gemini: a novel hardware and software implementation of high-performance PCIe SSD. *Int. J. Parallel Prog.* (2016). doi:[10.1007/s10766-016-0449-y](https://doi.org/10.1007/s10766-016-0449-y)
14. Deng, M., Chen, W., Xiao, N., Yu, S., Hu, Y.: GLE-Dedup: a globally–locally even deduplication by request-aware placement for better read performance. *Int. J. Parallel Prog.* (2016). doi:[10.1007/s10766-016-0450-5](https://doi.org/10.1007/s10766-016-0450-5)
15. Du, J., Li, R., Xiao, Z., Tong, Z., Zhang, L.: Optimization of data allocation on CMP embedded system with data migration. *Int. J. Parallel Prog.* (2016). doi:[10.1007/s10766-016-0436-3](https://doi.org/10.1007/s10766-016-0436-3)
16. Du, Y., Wang, L., Qi, M.: Constructing service clusters based on service space. *Int. J. Parallel Prog.* (2016). doi:[10.1007/s10766-016-0437-2](https://doi.org/10.1007/s10766-016-0437-2)
17. Sun, Y., Du, Y., Li, M.: A repair of workflow models based on mirroring matrices. *Int. J. Parallel Prog.* (2016). doi:[10.1007/s10766-016-0438-1](https://doi.org/10.1007/s10766-016-0438-1)