BigDataBench: A Scalable and Unified Big Data and AI Benchmark Suite

Wanling Gao*†, Jianfeng Zhan*†, Lei Wang*, Chunjie Luo*†, Daoyi Zheng‡, Xu Wen*†, Rui Ren*†, Chen Zheng*, Xiwen He§, Hainan Ye§, Haoning Tang¶, Zheng Cao‖, Shujie Zhang** and Jiahui Dai§
*Institute of Computing Technology, Chinese Academy of Sciences
†University of Chinese Academy of Sciences, China
‡Baidu
§Beijing Academy of Frontier Sciences and Technology
¶Tencent
‖Alibaba
**Huawei

***Abstract*—Several fundamental changes in technology indicate domain-specific hardware and software co-design is the only path left. In this context, architecture, system, data management, and machine learning communities pay greater attention to innovative big data and AI algorithms, architecture, and systems. Unfortunately, complexity, diversity, frequently-changed workloads, and rapid evolution of big data and AI systems raise great challenges. First, the traditional benchmarking methodology that creates a new benchmark or proxy for every possible workload is not scalable, or even impossible for Big Data and AI benchmarking. Second, it is prohibitively expensive to tailor the architecture to characteristics of one or more application or even a domain of applications.**

We consider each big data and AI workload as a pipeline of one or more classes of units of computation performed on different initial or intermediate data inputs, each class of which we call a data motif. We propose a scalable benchmarking methodology that uses the combination of one or more data motifs—to represent diversity of big data and AI workloads. Following this methodology, we present a unified big data and AI benchmark suite—BigDataBench 4.0, publicly available from http://prof.ict.ac.cn/BigDataBench. This unified benchmark suite sheds new light on domain-specific hardware and software co-design: tailoring the system and architecture to characteristics of the unified eight data motifs other than one or more application case by case.

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

The traditional benchmark methodology that creates a new benchmark or proxy for every possible workload is prohibitively costly and hence not scalable, or even impossible for Big Data and AI benchmarking. First, there are many classes of big data and AI applications. Even for Internet services, there are several important application domains, e.g., search engines, social networks, and e-commerce. The value of big data and AI also drives the emergence of innovative application domains. Meanwhile, data (sizes, types, sources, and patterns) have a great impact on workload behaviors and performance significantly [1], [2], so comprehensive and representative real-world data sets should be included.Second, at an earlier stage, it is usually difficult to justify porting a full-scale end-to-end Big data or AI application to a new computer system or architecture simply to obtain a benchmark number [3]; while at a later stage, kernels alone are insufficient to completely assess the performance potential of a new system or architecture on real-world data sets and applications [3]. Meanwhile, the benchmarks should be consistent across different communities for the co-design of software and hardware. Third, the correctness of results and performance figures must be easily verifiable [3]. To some extent, too complex workloads, i.e., full-scale end-to-end Big Data or AI applications raise difficulties in reproducibility and interpretability of performance data [1].

As modern big data and AI workloads are not only diverse, but also fast changing and expanding, it also raises great challenges in domain-specific hardware and software co-design. Even the agile hardware development methodology and tools are adopted [4], it is prohibitively expensive to tailor the architecture to characteristics of one or more application or even a domain of applications, and hence building domain-specific hardware and software systems case by case should be avoided.

This paper presents our joint research efforts on a scalable and unified Big Data and AI benchmarking suite with several industrial partners. On the basis of our previous work [1] that identifies eight data motifs—taking up most of the run time among a wide variety of big data and AI workloads, we propose a scalable benchmarking methodology that uses the combination of one or more data motifs—including *Matrix, Sampling, Transform, Graph, Logic, Set, Sort and Statistic computation* to represent diversity of big data and AI workloads. Our benchmark suite includes micro benchmarks, each of which is a single data motif, the component benchmarks, each of which consists of the combination of one or more data motifs with different weights in terms of runtime, and end-to-end application benchmarks, which are combinations of component benchmarks.

Following this methodology, we present a unified big data and AI benchmark suite—BigDataBench 4.0, publicly avail-

able from http://prof.ict.ac.cn/BigDataBench. BigDataBench 4.0 provides 13 representative real-world data sets and 47 big data and AI benchmarks of seven workload types: online service, offline analytics, graph analytics, AI, data warehouse, NoSQL, and streaming. Also, for each workload type, we provide diverse implementations using state-of-the-art and state-of-the-practise software stacks. Data varieties are considered with the whole spectrum of data types including structured, semi-structured, and unstructured data. Using real data sets as the seed, the data generators [5] are provided to generate the data with a specific scale.

II. BigDataBench 4.0: Big Data and AI Benchmark Suite

Circling around the data motifs identified from these application domains, we define the specifications of micro benchmarks—each of which is a single data motif, component benchmarks—each of which is a combination of data motifs with different weights, and application benchmarks—each of which represents an end-to-end applications. On the basis of the data motif-based benchmarking methodology, we make benchmark decisions and build BigDataBench 4.0. Please note that due to the space limitation, the detailed methodology and decisions about BigDataBench 4.0 are illustrated in our technical report [6].

1) Workloads Diversity: After investigating fundamental components in application domains, we provide a suite of micro benchmarks and component benchmarks. Totally, BigDataBench 4.0 provides 13 representative real-world data sets and 47 big data and AI benchmarks of seven workload types: online service, offline analytics, graph analytics, AI, data warehouse, NoSQL, and streaming.

For big data, we provide diverse workloads covering data mining, machine learning, natural language processing and computer vision techniques. For AI, we identify representative and widely used data motifs in a wide variety of deep learning networks (i.e. convolution, relu, sigmoid, tanh, fully connected, max/avg pooling, cosine/batch normalization and dropout) and then implement each single motif and motif combinations as micro benchmarks and component benchmarks. The AI component benchmarks include Alexnet [7], Googlenet [8], Resnet [9], Inception_Resnet V2 [10], VGG16 [11], DCGAN [12], WGAN [13], Seq2Seq [14] and Word2vec [15], which are important state-of-the-art networks in AI.

2) Representative Real-world Data Set: To cover a full spectrum of data characteristics, we collect 13 representative data sets, including different data sources (text, table, graph, and image), and data types of structured, un-structured, semi-structured. Further, big data generation tools are provided to suit for different cluster scales, including text, table, matrix and graph generators.

3) State-of-the-art Techniques: To perform apple-to-apple comparisons, we provide diverse implementations using the state-of-the-art techniques. For offline analytics, we provide Hadoop, Spark, Flink and MPI implementations. For graph analytics, we provide Hadoop, Spark GraphX, Flink Gelly and GraphLab implementations. For AI, we provide TensorFlow, Caffe and PyTorch implementations. For data warehouse, we provide Hive, Spark-SQL and Impala implementations. For NoSQL, we provide MongoDB and HBase implementations. For streaming, we provide Spark streaming and JStorm implementations.

References

[1] W. Gao, J. Zhan, L. Wang, C. Luo, D. Zheng, F. Tang, B. Xie, C. Zheng, X. Wen, X. He, H. Ye, and R. Ren, "Data motifs: A lens towards fully understanding big data and ai workloads," *Parallel Architectures and Compilation Techniques (PACT), 2018 27th International Conference on*, 2018.

[2] B. Xie, J. Zhan, X. Liu, W. Gao, Z. Jia, X. He, and L. Zhang, "Cvr: Efficient vectorization of spmv on x86 processors," in *2018 IEEE/ACM International Symposium on Code Generation and Optimization (CGO)*, 2018.

[3] D. H. Bailey, E. Barszcz, J. T. Barton, D. S. Browning, R. L. Carter, L. Dagum, R. A. Fatoohi, P. O. Frederickson, T. A. Lasinski, R. S. Schreiber *et al.*, "The nas parallel benchmarks," *The International Journal of Supercomputing Applications*, vol. 5, no. 3, pp. 63–73, 1991.

[4] J. Hennessy and D. Patterson, "A new golden age for computer architecture: Domain-specific hardware/software co-design, enhanced security, open instruction sets, and agile chip development." 2018.

[5] Z. Ming, C. Luo, W. Gao, R. Han, Q. Yang, L. Wang, and J. Zhan, "Bdgs: A scalable big data generator suite in big data benchmarking," *arXiv preprint arXiv:1401.5465*, 2014.

[6] W. Gao, J. Zhan, L. Wang, C. Luo, D. Zheng, R. Ren, C. Zheng, G. Lu, J. Li, Z. Cao *et al.*, "Bigdatabench: A dwarf-based big data and ai benchmark suite," *arXiv preprint arXiv:1802.08254*, 2018.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[10] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." in *AAAI*, 2017, pp. 4278–4284.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[12] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[13] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[14] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.