Control Flow Behavior of Cloud Workloads

Jee Ho Ryoo, Michael LeBeane, Muhammad Faisal Iqbal and Lizy K. John
The University of Texas at Austin
{jr45842, mlebeane, faisaliqbal}@utexas.edu, ljohn@ece.utexas.edu

Abstract

With massive amounts of information on the web, cloud applications are rapidly emerging as one of the main-stream domains in modern computing, yet very little is known about their behavior. To our knowledge, this paper presents the first detailed study of control flow behavior in cloud workloads. We characterize branch predictability behavior of cloud and big data benchmarks, and compare against those of widely known CPU workloads based on profiling and simulation. Our in-depth branch analysis of workloads present striking differences in terms of higher prevalence of indirect branches, larger offsets in branch targets, abundance of multi-target branches and low BTB hit-rates. We identify performance bottlenecks involving branch predictability and provide suggestions that can be incorporated in future datacenter oriented processor designs. We perform Principal Component Analysis and clustering techniques to understand similarity/dissimilarity between cloud and CPU workloads.

1. Introduction

Datacenter oriented computing has become prominent with the rise of cloud based services. Everyday, millions of users across the world rely on web search, social network services, video streaming and cloud storage services for business and personal related purposes [6]. The recent interest in cloud computing has spurred the creation and analysis of many cloud benchmarking suites. Jia et al. [4] use hardware performance counters to propose and characterize the DCBench suite for big data benchmarking. Apache Hadoop applications are evaluated using system-level metrics using the HiBench benchmarking suite [3]. Ferdman et al. present CloudSuite [2] and provide the Simics images used in this paper. BigDataBench [13] is a recently proposed benchmarking suite that contains the greatest breadth of all publicly available benchmarks. It was created as a collaboration project between academia and various industry partners. Yahoo released a benchmarking framework and datasets specifically targeted towards the cloud as part of the YCSB package [1]. YCSB is distributed as a java application that allows for variable workload and system configurations. Li et al. introduced a framework called CloudCmp that enables different public cloud providers to be reasonably compared to each other in a fair and consistent manner [5].

This paper uses a mixture of microarchitecture independent and dependent characteristics while most of the prior research largely employs microarchitecture dependent characteristics. The Principal Component Analysis (PCA) and clustering techniques used in this paper are similar to techniques used in [9] [10], but to the best of our knowledge, we are the first to apply these techniques to cloud and big data workloads.

2. Workload Profiling and Simulation

This work uses profiling and simulation tools to extract instruction traces in order to analyze control flow behavior. CloudSuite benchmarks [2] are used as representative cloud workloads and SPEC CPU2006 integer suite is used for CPU workloads [11]. We perform our experiments on a cycle-accurate out-of-order processor simulator that closely models today's high performance server. For cycle accurate timing simulation, we used Flexus [14]. The instruction trace generation was performed on SIMICS [8]. We ran the SPEC CPU2006 INT workloads in rate mode [11].

3. Evaluation

Branches are common control flow features that limit performance if incorrect speculation is made. In most SPEC INT programs, control flow instructions take approximately 20% of total dynamic instructions [10]. In our study, branches are defined as stated in SPARCv9 Architecture Manual. Branches in the SPARC v9 ISA are categorized into three groups: conditional branches, direct branches and indirect branches.

Sophisticated branch predictors have enabled microprocessor vendors to achieve high performance and there are several metrics that can speculate how well branch predictors perform. Various branch prediction schemes are used to evaluate the accuracy of branch prediction in our workloads. The accuracy of each branch predictor is measured in terms of mispredictions per 1K instructions. Figure 1 shows how various branch predictors perform on each workload [15] [7]. Cloud workloads have higher misprediction rate than CPU workloads on average in all predictor schemes. The PAs predictor has a smaller PHT entries because some are allocated for BHSR entries when we constraint that all predictors have the same total number of entries. Therefore, we observe that the misprediction is higher with PAs predictor, which indicates that the cloud workloads will suffer from high branch misprediction due to PHT index aliasin. A gshare predictor is able to reduce the misprediction by removing some of its aliasing as it randomizes the PHT. However, it still shows a noticeably higher misprediction rate. In this new era of cloud computing, there can be future research areas where a branch predictor is designed to consider these workloads with a very large instruction footprint.

We have now studied different control flow characteristics of cloud workloads and compared against CPU workloads. We would like a statistically rigorous method of incorporating our metrics to cluster the workloads together. Towards this

978-1-4799-6454-3/14/$31.00 ©2014 IEEE

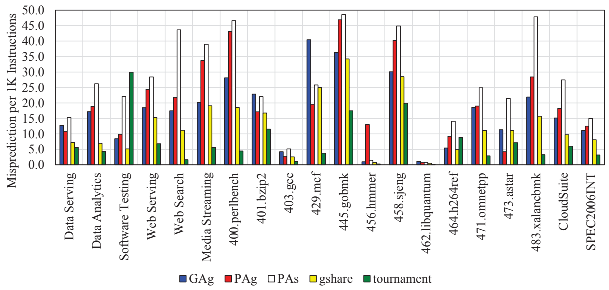

Figure 1: Branch Prediction Analysis

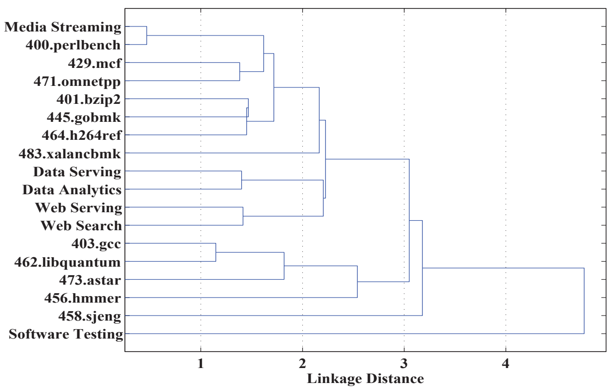

Figure 2: Similarity Dendrogram

end, we employ PCA. Out of all characteristics (PC's), we only choose PC's whose eigenvalues are greater than 1 based on Kaiser Criterion. Then, we apply PCA and reduce our dimensionality to 4 PC's, which covers 93% of variance.

We now use the principal components to cluster workloads into a set of groups in a hierarchical fashion using a dendrogram [10]. This process uses the Euclidean distance between each workload and clusters workloads that are close to each other. The dendrogram is presented in Figure 2. Workloads that are connected together near the leafs exhibit similar control flow behavior. For example, if we would like to cluster workloads into a cluster of 4 groups, we draw a vertical line that such that this line contains 4 intersections. In our figure, the vertical line at the linkage distance 3 lets us to choose 4 clusters. We see that *Data Serving*, *Data Analytics*, *Web Serving*, and *Web Search* are placed next to each other and form their own cluster at a very small linkage distance. However, *Software Testing* requires a very large linkage distance before it is merged with other clusters, indicating dissimilarity to the other benchmarks. Overall, these cluster groups show that cloud workloads exist in a different space with unique control flow characteristics. Although we have presented different control flow features independently, the statistical clustering verifies our results that cloud workloads do have their unique characteristics.

4. Conclusion

Growing interests in cloud applications have demanded the development of efficient hardware and software for cloud computing systems. This paper takes the initiatives in evaluating effectiveness of conventional control flow optimization techniques in the new domain of cloud computing. In this paper, the control transfer characteristics of cloud workloads are compared against traditional CPU workloads. The similarity/dissimilarity study using PCA shows that cloud workloads fall under a different category in terms of control flow behavior. We believe that deeper understanding of distinct branch behaviors provide useful insights to both software programmers and hardware vendors in cloud computing environment.

5. Acknowledgement

This work has been supported and partially funded by National Science Foundation under grant numbers1337393 and 1117895. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] B. F. Cooper *et al.*, "Benchmarking cloud serving systems with ycsb," in *Proceedings of the 1st ACM Symposium on Cloud Computing*, ser. SoCC '10. New York, NY, USA: ACM, 2010, pp. 143–154. http://doi.acm.org/10.1145/1807128.1807152

[2] M. Ferdman *et al.*, "Clearing the clouds: A study of emerging scale-out workloads on modern hardware," in *Proceedings of the Seventeenth International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS XVII. New York, NY, USA: ACM, 2012, pp. 37–48. http://doi.acm.org/10.1145/2150976.2150982

[3] S. Huang *et al.*, "The hibench benchmark suite: Characterization of the mapreduce-based data analysis," in *Data Engineering Workshops (ICDEW), 2010 IEEE 26th International Conference on*, March 2010, pp. 41–51.

[4] Z. Jia *et al.*, "Characterizing data analysis workloads in data centers," in *Workload Characterization (IISWC), 2013 IEEE International Symposium on*, Sept 2013, pp. 66–76.

[5] A. Li *et al.*, "Cloudcmp: Comparing public cloud providers," in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '10. New York, NY, USA: ACM, 2010, pp. 1–14. http://doi.acm.org/10.1145/1879141.1879143

[6] J. Lin *et al.*, "What is cloud computing?" *IT as a Service*, vol. 11, no. 2, p. 10, 2009.

[7] G. H. Loh and D. S. Henry, "Predicting conditional branches with fusion-based hybrid predictors," in *Parallel Architectures and Compilation Techniques, 2002. Proceedings. 2002 International Conference on*. IEEE, 2002, pp. 165–176.

[8] P. Magnusson *et al.*, "Simics: A full system simulation platform," *Computer*, vol. 35, no. 2, pp. 50–58, Feb 2002.

[9] A. Phansalkar *et al.*, "Measuring program similarity: Experiments with spec cpu benchmark suites," in *Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software, 2005*, ser. ISPASS '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 10–20. http://dx.doi.org/10.1109/ISPASS.2005.1430555

[10] A. Phansalkar, A. Joshi, and L. K. John, "Analysis of redundancy and application balance in the spec cpu2006 benchmark suite," in *Proceedings of the 34th Annual International Symposium on Computer Architecture*, ser. ISCA '07. New York, NY, USA: ACM, 2007, pp. 412–423. http://doi.acm.org/10.1145/1250662.1250713

[11] SPEC, "SPEC CPU2006." http://www.spec.org/cpu2006.

[12] L. M. Vaquero *et al.*, "A break in the clouds: Towards a cloud definition," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 50–55, Dec. 2008. http://doi.acm.org/10.1145/1496091.1496100

[13] L. Wang *et al.*, "Bigdatabench: a big data benchmark suite from internet services," in *The 20th IEEE International Symposium On High Performance Computer Architecture (HPCA-2014)*, February 2014.

[14] T. Wenisch *et al.*, "Simflex: Statistical sampling of computer system simulation," *Micro, IEEE*, vol. 26, no. 4, pp. 18–31, July 2006.

[15] T.-Y. Yeh and Y. Patt, “A comparison of dynamic branch predictors that use two levels of branch history,” in *Computer Architecture, 1993., Proceedings of the 20th Annual International Symposium on*, May 1993, pp. 257–266.