Leveraging Resource Bottleneck Awareness and Optimizations for Data Analytics Performance

Tiago B. G. Perez
Dept. of Computer Science
University of Colorado
Colorado Springs, USA

Xiaobo Zhou (Advisor)
Dept. of Computer Science
University of Colorado
Colorado Springs, USA

Abstract

Spark is among the most popular parallel and distributed data analytics frameworks in use today. But just like its peers and predecessors, it still suffers from performance degradation caused by resource bottlenecks. The challenge is finding multiple methods that leverage awareness of resource bottlenecks to improve performance.

Among the research progress we have made so far, includes the proposal and development of two systems. The first, *PETS*, is a tuning system that is capable of tuning multiple parameters at the same time in a few iterations, taking advantage of a resource awareness feedback system. The second, *MRD*, is a cache memory management system which utilizes reference-distance information, to evict the furthest and least likely data in the cache to be used, while aggressively prefetching the nearest and most likely data that will be needed.

Both systems have shown promising results, with significant performance gains over default systems, as well as improvements in related recent work.

1. MOTIVATION

Spark [9] is among the most popular parallel and distributed data analytics frameworks in use today. But just like its peers and predecessors [2], it still suffers from performance degradation caused by resource bottlenecks. Detecting and identifying a bottleneck is already hard for a single system that is composed of multiple resources (e.g. CPU, memory, etc.), the problem is made even harder as it scales and deals with the intricacies of large parallel and distributed systems. Finding simple ways of detecting, identifying and mitigating these bottlenecks is an important step in improving the performance issue faced by these frameworks.

The goal is to find multiple methods, such as: tuning, memory management, performance modeling and task scheduling; that encompass the measurement, analysis, identification, and mitigation of multiple resource bottlenecks in a parallel and distributed system. Leveraging the awareness of resource bottlenecks with the addition of optimizations in various system parameters and resource management, to improve the performance and counter bottlenecks that may occur in a single or multiple nodes of a multi-node cluster. Combining both offline and online methods to optimize and improve system performance.

ICPP '18, August 13 - 16 2018, Eugene, OR, USA

2. BOTTLENECK AWARENESS AND OPTIMIZATIONS

Among the research progress we have made so far, includes the proposal and development of two systems. The first, a tuning system called *PETS*, which stands for *Parameter Ensemble Table for Spark*, was motivated from the challenges and time consuming effort of tuning Spark with its dozens of parameters that impact performance. Since current techniques rely on trial-and-guess or the use of scarce expert knowledge that few posses, and most previous tuning works for other frameworks are not to adaptable to Spark [3, 4], this posed as an opportunity to close a gap in this research field. Also, previous works [1, 5] ignored the issue of resource bottlenecks, and did not take advantage of a resource awareness feedback system to the tuning process. PETS allowed us have significant performance gains by adjusting multiple parameters at the same time, in just a few iterations. A simplified view of the architecture can be seen in Figure 1. Evaluation from our implementation resulted in promising results, which maintain significant speedups with fast convergence speeds across multiple types of workloads, data size and clusters. A summary of our results can be seen in Figure 2.

Secondly, a cache memory management system called *MRD*, which stands for *Most Reference Distance*, was motivated from Spark's usage of the popular Least Recently Used (LRU) policy, which however is oblivious to the data dependency information available in the application's directed acyclic graph (DAG [6]). Recent research [8, 7] have made great strides in exploiting this information, but opportunities to further extend its usage exist. MRD utilizes reference-distance information, defined as the distance between the current stage in the workflow and the usage of the data, to evict the furthest and least likely data in the cache to be used, while aggressively prefetching the nearest and most likely data that will be needed. A simplified view of the architecture can be seen in Figure 3. Results are very promising and showed significant improvements in performance and cache hit ratio over the standard LRU policy and recent DAG-aware research alternatives. A summary of our results can

be seen in Figure 4.

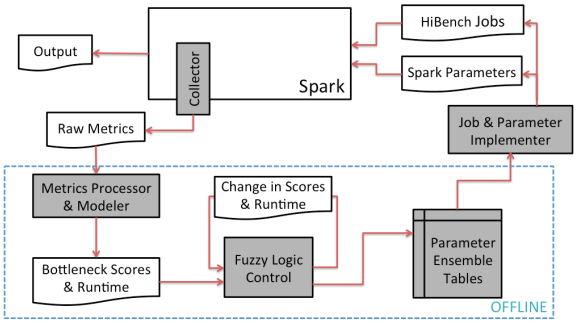
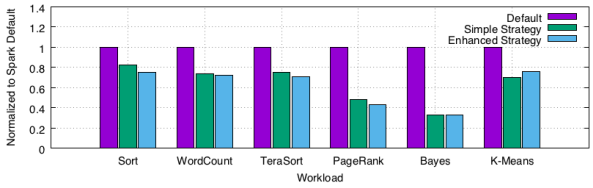
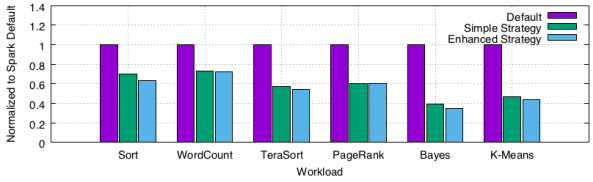


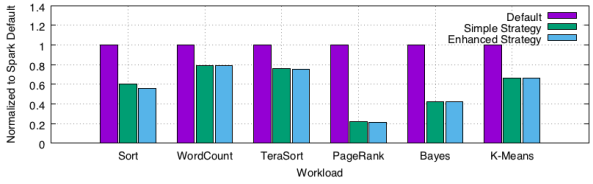
Figure 1: PETS system architecture.



(a) Homogeneous cluster.



(b) Heterogeneous cluster 1.



(c) Heterogeneous cluster 2.

Figure 2: PETS result summary.

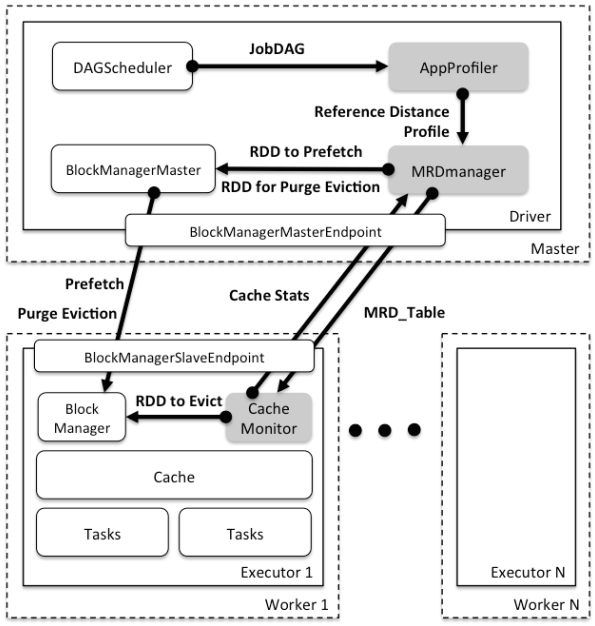


Figure 3: MRD system architecture.

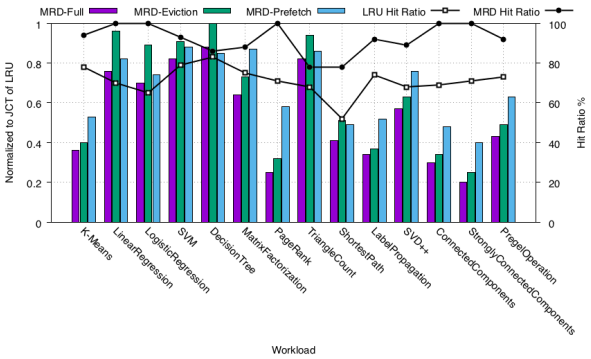


Figure 4: MRD result summary.

3. CONCLUSION

With the development of PETS and MRD, multiple aspects of bottlenecks have been addressed, while obtaining performance gains. Yet, there are currently many gaps which call for further research, such as developing a reliable bottleneck performance modeling and scheduling system, which is a direction for future work.

Acknowledgement

This research was supported in part by U.S. NSF grant CNS-1422119 and CAPES scholarship BEX 13424-13-0.

4. REFERENCES

[1] Cheng, D., Rao, J., Guo, Y., and Zhou, X. Improving mapreduce performance in heterogeneous environments with adaptive task tuning. In *Proc. of ACM Middleware* (2014).

[2] Hadoop, A. Apache hadoop, 2009. http://hadoop.apache.org/.

[3] Herodotou, H., Lim, H., Luo, G., Borisov, N., Dong, L., Cetin, F. B., and Babu, S. Starfish: A self-tuning system for big data analytics. In *Proc. of CIDR* (2011).

[4] Li, M., Zeng, L., Meng, S., Tan, J., Zhang, L., Butt, A. R., and Fuller, N. Mronline: Mapreduce online performance tuning. In *Proc. of ACM HPDC* (2014).

[5] Liao, G., Datta, K., and Willke, T. L. Gunther: Search-based auto-tuning of mapreduce. In *Proc. of Springer Europar* (2013).

[6] Wang, L. Directed acyclic graph. In *Encyclopedia of Systems Biology*. Springer, 2013, pp. 574–574.

[7] Xu, L., Li, M., Zhang, L., Butt, A. R., Wang, Y., and Hu, Z. Z. Memtune: Dynamic memory management for in-memory data analytic platforms. In *In Proc. of IEEE IPDPS* (2016).

[8] Yu, Y., Wang, W., Zhang, J., and Letaief, K. B. Lrc: Dependency-aware cache management for data analytics clusters. In *Proc. of IEEE INFOCOM* (2017).

[9] Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., and Stoica, I. Spark: Cluster computing with working sets. In *Proc. of USENIX HOTCLOUD* (2010).