Workload Management for Big Data Analytics

Ashraf Aboulnaga
University of Waterloo
ashraf@uwaterloo.ca

Shivnath Babu*
Duke University
shivnath@cs.duke.edu

Categories and Subject Descriptors

H.2.7 [**Database Administration**]: Data warehouse and repository

Keywords

workload management; analytics; parallel database systems; MapReduce

1. INTRODUCTION

Parallel database systems and MapReduce systems (most notably Hadoop) are essential components of today's infrastructure for Big Data analytics. These systems process multiple concurrent workloads consisting of complex user requests, where each request is associated with an (explicit or implicit) service level objective. For example, the workload of a particular user or application may have a higher priority than other workloads. Or, a particular workload may have strict deadlines for the completion of its requests.

The research area of Workload Management focuses on ensuring that the system meets the service level objectives of various requests while at the same time minimizing the resources required to achieve this goal. At a high level, workload management can be viewed as looking beyond the performance of an individual request to the performance of an entire workload consisting of multiple requests.

Questions addressed by workload management research and technologies include: How to implement different priorities for different workloads? How to isolate the performance of one workload from the effect of other workloads? What is the best way to do request scheduling and admission control? What are good mechanisms and policies to control the allocation of resources to workloads statically and dynamically? How to define a workload and associated requests within that workload? How to monitor request performance, resource consumption, and data access patterns to ensure that workload management is effectively achieving its goals? How to ensure that workload management goals are met even in the presence of failures?

This tutorial will discuss the fundamentals of workload management, and present tools and techniques for workload management in parallel databases and MapReduce. Workload management for parallel databases is an established topic, and most parallel database systems have sophisticated workload management tools. The tutorial will present some of these tools as case studies and discuss the underlying techniques that they use. Workload management for MapReduce is still a fledgling research area, and the tutorial will discuss recent advances in this area and future research directions.

*Supported by NSF grants 0917062 and 0964560

2. TUTORIAL OUTLINE

The tutorial will cover the following topics. Some of the papers that will be covered are listed in the next section.

- Introduction
  - Definition of workload management
  - Problems addressed by workload management
  - How are workloads defined?
  - Diversity of workloads
- Workload-level decisions in database systems
  - Physical design
  - Progress monitoring
  - Managing long-running queries
- Performance prediction
  - The role of the query optimizer
  - Machine-learning-based and experiment-driven modeling
  - Modeling query interactions
- Inter-workload interactions
  - Effects of workload interference
  - Multiclass workloads and priorities
  - Mechanisms for workload isolation
- MapReduce workload management
  - Resource sharing and notions of fairness
  - Scheduling and resource allocation

– Mechanisms for workload isolation

- Current trends and research challenges
  – Cloud computing
  – Resource management frameworks
  – Richer service level agreements

3. REFERENCES

[1] M. Ahmad, S. Duan, A. Aboulnaga, and S. Babu. Predicting completion times of batch query workloads using interaction-aware models and simulation. In *EDBT*, pages 449–460, 2011.

[2] M. Akdere, U. Çetintemel, M. Riondato, E. Upfal, and S. B. Zdonik. Learning-based query performance modeling and prediction. In *ICDE*, pages 390–401, 2012.

[3] S. Babu, G. Graefe, and H. A. Kuno. Database Workload Management (Dagstuhl Seminar 12282). *Dagstuhl Reports*, 2(7):73–91, 2012.

[4] K. P. Brown, M. Mehta, M. J. Carey, and M. Livny. Towards automated performance tuning for complex workloads. In *VLDB*, pages 72–84, 1994.

[5] S. Chaudhuri and V. R. Narasayya. Self-tuning database systems: A decade of progress. In *VLDB*, pages 3–14, 2007.

[6] W.-J. Chen, B. Comeau, T. Ichikawa, S. S. Kumar, M. Miskimen, H. T. Morgan, L. Pay, and T. Väättänen. *DB2 Workload Manager for Linux, UNIX, and Windows*. IBM Redbook. IBM, 2008.

[7] Y. Chen, S. Alspaugh, and R. H. Katz. Interactive analytical processing in big data systems: A cross-industry study of MapReduce workloads. *PVLDB*, 5(12):1802–1813, 2012.

[8] R. Cole, F. Funke, L. Giakoumakis, W. Guy, A. Kemper, S. Krompass, H. A. Kuno, R. O. Nambiar, T. Neumann, M. Poess, K.-U. Sattler, M. Seibold, E. Simon, and F. Waas. The mixed workload CH-benCHmark. In *DBTest*, page 8, 2011.

[9] J. Duggan, U. Çetintemel, O. Papaemmanouil, and E. Upfal. Performance prediction for concurrent database workloads. In *SIGMOD*, pages 337–348, 2011.

[10] F. Funke, A. Kemper, S. Krompass, H. A. Kuno, R. O. Nambiar, T. Neumann, A. Nica, M. Poess, and M. Seibold. Metrics for measuring the performance of the mixed workload CH-benCHmark. In *TPCTC*, pages 10–30, 2011.

[11] A. Ganapathi, H. A. Kuno, U. Dayal, J. L. Wiener, A. Fox, M. I. Jordan, and D. A. Patterson. Predicting multiple metrics for queries: Better decisions enabled by machine learning. In *ICDE*, pages 592–603, 2009.

[12] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica. Dominant resource fairness: Fair allocation of multiple resources types. In *NSDI*, 2011.

[13] H. Herodotou and S. Babu. Profiling, what-if analysis, and cost-based optimization of MapReduce programs. *PVLDB*, 4(11):1111–1122, 2011.

[14] B. Hindman, A. Konwinski, M. Zaharia, A. Ghodsi, A. Joseph, R. Katz, S. Shenker, and I. Stoica. Mesos: A platform for fine-grained resource sharing in the data center. In *NSDI*, 2011.

[15] M. Isard, V. Prabhakaran, J. Currey, U. Wieder, K. Talwar, and A. Goldberg. Quincy: Fair scheduling for distributed computing clusters. In *SOSP*, pages 261–276, 2009.

[16] A. C. König, B. Ding, S. Chaudhuri, and V. R. Narasayya. A statistical approach towards robust progress estimation. *PVLDB*, 5(4):382–393, 2011.

[17] S. Krompass, H. A. Kuno, J. L. Wiener, K. Wilkinson, U. Dayal, and A. Kemper. Managing long-running queries. In *EDBT*, pages 132–143, 2009.

[18] S. Krompass, H. A. Kuno, J. L. Wiener, K. Wilkinson, U. Dayal, and A. Kemper. A testbed for managing dynamic mixed workloads. *PVLDB*, 2(2):1562–1565, 2009.

[19] S. Krompass, H. A. Kuno, K. Wilkinson, U. Dayal, and A. Kemper. Adaptive query scheduling for mixed database workloads with multiple objectives. In *DBTest*, 2010.

[20] S. Krompass, A. Scholz, M.-C. Albutiu, H. A. Kuno, J. L. Wiener, U. Dayal, and A. Kemper. Quality of service-enabled management of database workloads. *IEEE Data Eng. Bull.*, 31(1):20–27, 2008.

[21] H. A. Kuno, U. Dayal, J. L. Wiener, K. Wilkinson, A. Ganapathi, and S. Krompass. Managing dynamic mixed workloads for operational business intelligence. In *DNIS*, pages 11–26, 2010.

[22] J. Li, A. C. König, V. R. Narasayya, and S. Chaudhuri. Robust estimation of resource consumption for SQL queries using statistical techniques. *PVLDB*, 5(11):1555–1566, 2012.

[23] J. Li, R. V. Nehme, and J. F. Naughton. GSLPI: A cost-based query progress indicator. In *ICDE*, pages 678–689, 2012.

[24] G. Luo, J. F. Naughton, and P. S. Yu. Multi-query SQL progress indicators. In *EDBT*, pages 921–941, 2006.

[25] L. A. Moakar, T. N. Pham, P. Neophytou, P. K. Chrysanthis, A. Labrinidis, and M. A. Sharaf. Class-based continuous query scheduling for data streams. In *DMSN*, 2009.

[26] K. Morton, M. Balazinska, and D. Grossman. ParaTimer: a progress indicator for MapReduce DAGs. In *SIGMOD*, pages 507–518, 2010.

[27] B. Niu, P. Martin, and W. Powley. Towards autonomic workload management in DBMSs. *J. Database Manag.*, 20(3):1–17, 2009.

[28] B. Niu, P. Martin, W. Powley, P. Bird, and R. Horman. Adapting mixed workloads to meet SLOs in autonomic DBMSs. In *ICDE Workshops*, pages 478–484, 2007.

[29] B. Niu, Y. Xue, P. Martin, and W. Powley. Managing workload importance in enterprise DBMSs. In *CEC*, pages 168–173, 2010.

[30] H. Pang, M. J. Carey, and M. Livny. Multiclass query scheduling in real-time database systems. *IEEE Trans. Knowl. Data Eng.*, 7(4):533–551, 1995.

[31] S. S. Parekh, K. Rose, J. L. Hellerstein, S. Lightstone, M. Huras, and V. Chang. Managing the performance impact of administrative utilities. In *DSOM*, 2003.

[32] W. Wu, Y. Chi, S. Zhu, J. Tatemura, H. Hacıgümüş, and J. F. Naughton. Predicting query execution time: Are optimizer cost models really unusable? In *ICDE*, pages 1081–1092, 2013.

[33] M. Zaharia, D. Borthakur, J. S. Sarma, K. Elmeleegy, S. Shenker, and I. Stoica. Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling. In *EuroSys*, pages 265–278, 2010.

[34] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *NSDI*, 2012.

[35] M. Zaharia, A. Konwinski, A. D. Joseph, R. H. Katz, and I. Stoica. Improving MapReduce performance in heterogeneous environments. In *OSDI*, pages 29–42, 2008.