The Science of Cloud Computing – PI Meeting Application

Magdalena Balazinska

University of Washington
Department of Computer Science and Engineering
Box 352350, Seattle, WA 98195-2350
magda@cs.washington.edu
http://www.cs.washington.edu/homes/magda/

Office (206) 616-1069
Fax: (206) 543-2969

1 Research Interest Areas

(3) Data Portability, Consistency, and Management
(8) Cloud Self-Monitoring and Autonomic Control

2 Current Research Activities

Data Intensive Scalable Computing In our Nuage project [21], we are looking at challenges related to efficiently analyzing massive scale datasets using parallel data processing engines (parallel relational database management systems or MapReduce [10] type systems) running on either private or public clouds. In collaboration with domain scientists on campus, we have ported several real scientific analysis tasks onto relational databases (single-node and parallel) and Hadoop [12]. We found that porting the analysis logic to these engines was only half the problem. Extracting high-performance from these engines was the bigger challenge [15, 17, 27].

Based on our experience above, we have developed several extensions to MapReduce. Our goal is to help users more easily get high-performance from parallel data processing systems without having to rely on experts. As part of our research, we developed an automated technique for handling skew [15, 16], a fault-tolerance optimizer [26], a new framework for automating recursive and cyclic analyses [7], a time-remaining progress indicator [20, 19], and an instrumentation to collect more accurate intermediate running time statistics. Our extensions enable users to more easily achieve better performance thanks to optimizations of recursive analytics, semi-automated skew management, and more efficient failure handling. They also help users better understand what is happening to their queries through a more accurate time-remaining progress estimator.

As a separate effort, we are part of the SciDB team [23] whose goal is to build a new parallel data management system for processing multidimensional array data [22], common in science. Our key contributions to the project so far include C++ language bindings [9] and a new storage manager [24].

The PI also has extensive prior experience building other distributed systems [1, 6, 3, 4, 5, 8, 11, 13, 14].

Cloud Economics An important benefit of cloud systems is their "pay-as- you-go" charging mechanism: each user pays exactly for what she consumes. However, cloud systems in general and data-management-as-a-service systems (*e.g.*, SQL Azure [18], Amazon S3 [2], etc.) in particular are used increasingly in collaborative settings, where multiple users access common data sets. This creates a major new challenge: how to properly price optimizations in the face of collaborations, where multiple users access the same dataset, and one optimization can benefit multiple users. There is a cost to each optimization (cost of building an index or replicating data), and if each optimization benefits some but not all users, it is not clear at all what optimizations to implement and how to share their costs. In recent work, we addressed the challenge of offering and pricing optimizations in a collaborative, data-management-as-a-service system through the use of Mechanism Design. We developed a mechanism that achieves higher utility than existing techniques for pricing optimization in the cloud, while ensuring that all optimization costs are recovered by the cloud provider [25].

3 Future Research Problems

Data Intensive Scalable Computing On the topic of data intensive scalable computing, we will continue working on helping non-expert users perform efficient analytics on large-scale clusters. As above, we will target small research groups and individuals who do not have access to a team of experts to help them use these systems efficiently. For those individuals, MapReduce and similar remain intriguing case-studies but are not widely adopted in daily work. To address this challenge, we will continue to focus on helping users express their analysis needs in a way that leads to a fast execution. As examples of concrete problems to address, we have work in progress on skew handling for arbitrary user-defined functions, lazy evaluation of parallel query workflows, and more.

The challenge of efficiently using a shared-nothing cluster is already great when the cluster is a private cloud. When the cloud is public, users face the additional challenge of deciding how many resources to use in the cloud (what size cluster should I use to run my queries?) and how to manage these resources (should I add/drop machines over time? Should I use spare cycles in some other way?). We will address this problem of helping users manage their resources in the cloud.

We plan to create a *toolkit that will help non-expert users cost-effectively exploit cloud resources for data analytics*. Given an analytics workload, possibly only partly defined (*i.e.*, a user knows that she will run a batch query followed by a series of interactive queries but the details of the latter depend on the output of the former), our toolkit will assist users before, during, and after performing their analysis. Before starting, our toolkit will help users figure-out what resources to reserve from a parallel data processing system running in a cloud: The toolkit will help users decide what size cluster to use and how to run the workload within this cluster to meet the user's desired trade-off between performance, cost, and predictability. During execution, our toolkit will help users efficiently exploit the reserved resources by either changing the cluster size dynamically or running various tasks to exploit any spare cycles in a fixed-size cluster. Finally, once the workload completes, our toolkit will help users understand the performance they got from the cloud system and tune it for their next workload. We already have ongoing preliminary work building this toolkit.

We posit that enabling non-expert users such as domain scientists to more easily extract high-performance from parallel data processing systems, better understand these engines, and more easily manage cloud resource, will help lift these tools from their current ”experimental” status into production-quality tools in daily science use.

Cloud Economics There are many challenges at the intersection of data-management-as-a-service systems and economics. In our recent work described above, we studied the problem of tuning data-management-as-a-service systems and passing the costs of these tunings fairly onto users. In future work, we will study additional challenges in the area of database-service economics.

As a first such challenge, we plan to develop and study the idea of a *relational data market in the cloud*. Integrating datasets within and across domains is recognized as a critical tool for scientific advancement. Cloud computing platforms have emerged as particularly well-suited to support such sharing because they provide a single logical location for storing data, support users in managing it, and enable easy access to that data from anywhere in the world. However, while today's cloud computing systems offer simple pricing schemes for storage and compute resources, the economics of data sharing are poorly understood and only coarsely supported. For example, a user who uploads her data to the cloud must pay associated storage and network utilization costs. Can this user then charge other users small amounts of money for accessing her data in order to recoup her costs? If so, can we make this pricing fine-grained and flexible (*e.g.*, Can some subsets of the data be more expensive than others)? If a user purchases multiple datasets and combines them, can that user re-sell the data for a profit? Should the original data providers be able to get some fraction of that profit? To answer these questions, we propose to build and study a relational data market in the cloud. Our system will enable users to sell their data in the cloud, choosing how to price their data using fine-grained and flexible pricing schemes. We will also study how best to support users in managing pricing schemes, computing query prices, and cost-effectively evaluating queries over priced data. We will also study properties such as fairness and stability of our proposed relational data market.

References

[1] D. Abadi, Y. Ahmad, M. Balazinska, U. Çetintemel, M. Cherniack, J. Hwang, W. Lindner, A. Maskey, A. Rasin, E. Ryvkina, N. Tatbul, Y. Xing, and S. Zdonik. The design of the Borealis stream processing engine. In *Proc. of the Second Biennial Conf. on Innovative Data Systems Research (CIDR)*, Jan. 2005.

[2] Amazon Simple Storage Service (Amazon S3). `http://www.amazon.com/gp/browse.html?node=16427261`.

[3] M. Balazinska. *Fault-Tolerance and Load Management in a Distributed Stream Processing System*. PhD thesis, Massachusetts Institute of Technology, Feb. 2006.

[4] M. Balazinska, H. Balakrishnan, S. Madden, and M. Stonebraker. Fault-tolerance in the Borealis distributed stream processing system. In *SIGMOD'05: Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, pages 13–24, June 2005.

[5] M. Balazinska, H. Balakrishnan, S. R. Madden, and M. Stonebraker. Fault-tolerance in the Borealis distributed stream processing system. *ACM Transactions on Database Systems*, 33(1):13–24, Mar. 2008.

[6] M. Balazinska, H. Balakrishnan, and M. Stonebraker. Contract-based load management in federated distributed systems. In *Proc. of the First Symp. on Networked Systems Design and Implementation (NSDI)*, Mar. 2004.

[7] Y. Bu, B. Howe, M. Balazinska, and M. D. Ernst. HaLoop: Efficient iterative data processing on large clusters. *Proc. of the VLDB Endowment*, 3(1):285–296, 2010.

[8] M. Cherniack, H. Balakrishnan, M. Balazinska, D. Carney, U. Çetintemel, Y. Xing, and S. Zdonik. Scalable distributed stream processing. In *Proc. of the First Biennial Conf. on Innovative Data Systems Research (CIDR)*, Jan. 2003.

[9] P. Cudre-Mauroux, H. Kimura, K.-T. Lim, J. Rogers, R. Simakov, E. Soroush, P. Velikhov, D. L. Wang, M. Balazinska, J. Becla, D. DeWitt, B. Heath, D. Maier, S. Madden, J. Patel, M. Stonebraker, and S. Zdonik. A demonstration of SciDB: A science-oriented DBMS (demonstration). In *Proc. of the 35th Int. Conf. on Very Large DataBases (VLDB)*, 2009.

[10] J. Dean and S. Ghemawat. MapReduce: simplified data processing on large clusters. In *Proc. of the 6th USENIX Symp. on Operating Systems Design & Implementation (OSDI)*, 2004.

[11] R. Geambasu, M. Balazinska, S. D. Gribble, and H. M. Levy. HomeViews: Peer-to-peer middleware for personal data sharing applications. In *SIGMOD'06: Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, June 2007.

[12] Hadoop. `http://hadoop.apache.org/`.

[13] J. Hwang, M. Balazinska, A. Rasin, U. Çetintemel, M. Stonebraker, and S. Zdonik. High-availability algorithms for distributed stream processing. In *Proc. of the 21st Int. Conf. on Data Engineering (ICDE)*, Apr. 2005.

[14] Y. Kwon, M. Balazinska, and A. Greenberg. Fault-tolerant stream processing using a distributed, replicated file system. In *Proc. of the 34th Int. Conf. on Very Large DataBases (VLDB)*, Aug. 2008.

[15] Y. Kwon, M. Balazinska, B. Howe, and J. Rolia. Skew-resistant parallel processing of feature-extracting scientific user-defined functions. In *ACM Symposium on Cloud Computing (SOCC)*, 2010.

[16] Y. Kwon, D. Nunley, J. P. Gardner, M. Balazinska, B. Howe, and S. Loebman. Scalable clustering algorithm for N-body simulations in a shared-nothing cluster. In *22nd international conference on scientific and statistical database management (SSDBM)*, 2010.

[17] S. Loebman, D. Nunley, Y. Kwon, B. Howe, M. Balazinska, and J. P. Gardner. Analyzing massive astrophysical datasets: Can Pig/Hadoop or a relational DBMS help? In *Proc. of the Workshop on Interfaces and Architectures for Scientific Data Storage (IASDS'09)*, Aug. 2009.

[18] Microsoft SQL Azure. `http://www.microsoft.com/windowsazure/sqlazure/`.

[19] K. Morton, M. Balazinska, and D. Grossman. ParaTimer: A progress indicator for MapReduce DAGs. In *SIGMOD'10: Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, June 2010.

[20] K. Morton, A. Friesen, M. Balazinska, and D. Grossman. Estimating the progress of MapReduce pipelines. In *Proc. of the 26th Int. Conf. on Data Engineering (ICDE)*, Mar. 2010.

[21] Nuage project: Scientific data management in the cloud. http://nuage.cs.washington.edu/.

[22] J. Rogers, R. Simakov, E. Soroush, P. Velikhov, M. Balazinska, D. DeWitt, B. Heath, D. Maier, S. Madden, J. Patel, M. Stonebraker, S. Zdonik, A. Smirnov, K. Knizhnik, and P. G. Brown. Overview of SciDB: Large scale array storage, processing and analysis. In *SIGMOD'10: Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, 2010.

[23] SciDB. http://www.scidb.org/.

[24] E. Soroush, M. Balazinska, and D. Wang. ArrayStore: A storage manager for complex parallel array processing. In submission.

[25] P. Upadhyaya, M. Balazinska, and D. Suciu. How to price shared optimizations in the cloud. In submission.

[26] P. Upadhyaya, Y. Kwon, and M. Balazinska. A latency and fault-tolerance optimizer for parallel data processing systems. Technical report, University of Washington, 2010.

[27] K. Wiley, A. Connolly, J. Gardner, S. Krughoff, M. Balazinska, B. Howe, Y. Kwon, and Y. Bu. Astronomy in the cloud: Using MapReduce for image coaddition. In submission.