

Tradeoffs in Main-Memory Statistical Analytics from Impala to DimmWitted

Victor Bittorf^{1,2}, Marcel Kornacker¹, Christopher Ré², and Ce Zhang²

Cloudera¹ and Stanford University²

{victor.bittorf, marcel}@cloudera.com, {chrismre, czhang}@cs.stanford.edu

Recent years have seen a surge in main-memory SQL-style analytic solutions to quickly deliver business critical information over massive data sets [1, 7, 14]. At the same time, there is an arms race to offer increasingly sophisticated statistical analytics inspired by the success of web search, voice recognition, and image analysis, e.g., Google Brain [8], Facebook [6], and Microsoft’s Adam [2]. This talk describes the first author’s experience porting statistical analytics to Impala via MADlib and observations about research for high-performance main-memory analytics that may be relevant for systems like Impala.

A major motivation for Impala was to enable interactive SQL-analytics queries over data stored in Hadoop. Impala achieves high performance through many techniques including as co-location of computation with data in HDFS, LLVM code generation [13], and aggressive use of SIMD instructions. These optimizations allow Impala to achieve 8x query throughput compared to Shark and Hive for queries in the TPC-DS benchmark [3], and a recent independent benchmark has shown that Impala is about 5 times faster than Hive on MapReduce for TPC-H queries on uncompressed data [10].

We also want high performance statistical analytics in Impala without major changes to its infrastructure. We started with an approach popularized in MADlib, an existing package for in-RDBMS analytics [4]. We ported a subset of MADlib’s statistical models to Impala [5], many of which use the Bismarck architecture [9] that allows statistical analytics via user-defined functions. In particular, the main algorithm is Stochastic Gradient Descent (SGD) a method that has a low memory footprint, rapid convergence, and is a near de facto standard for web-scale learning. SGD captures a wide variety of statistical models including Support Vector Machines (SVMs), Logistic Regression, and Matrix Factorization. Moreover, SGD’s row-wise data access pattern matches the access pattern of User Defined Aggregates [9]. The port has received positive feedback from customers for its scalability, speed, and breadth of machine learning tasks.

While the MADlib port enables some statistical analytics in Impala, it is only a first step: its data layout may be suboptimal, and it may not fully utilize commodity hardware. For example, as we describe SGD, it can be viewed as a row-store access method, and it is natural to wonder if there is a column-store equivalent. Indeed, there is a closely related algorithm call Stochastic Coordinate Descent (SCD). In our recent work, we have described asynchronous versions of both SGD [12] and SCD [11]. These algorithms can be run in massively parallel environments, but it is unclear what the optimal trade-off points and techniques are to run these algorithms at scale. In this VLDB, the tradeoffs of these methods

was studied by a subset of the authors [15] in the context of NUMA systems. In this talk, we describe to what extent these tradeoffs apply and can be used to inform a distributed data-systems like Impala.

References

1. cloudera.com.
2. research.microsoft.com/en-us/news/features/dnnvision-071414.aspx.
3. blog.cloudera.com/blog/2014/05/new-sql-choices-in-the-apache-hadoop-ecosystem-why-impala-continues-to-lead/.
4. madlib.net.
5. blog.cloudera.com/blog/2013/10/how-to-use-madlib-pre-built-analytic-functions-with-impala/.
6. MI2LS: Multi-Instance Learning from Multiple Information Sources. *ACM Conference on Knowledge Discovery and Data Mining (KDD)*.
7. A. Crotty, A. Galakatos, K. Dursun, T. Kraska, U. Cetintemel, and S. Zdonik. Tupeware: Redefining Modern Analytics. *arXiv preprint arXiv:1406.6667*, 2014.
8. J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le, et al. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems*, pages 1223–1231, 2012.
9. X. Feng, A. Kumar, B. Recht, and C. Ré. Towards a unified architecture for in-rdbms analytics. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 325–336. ACM, 2012.
10. A. Floratou, U. F. Minhas, and F. Ozcan. Sql-on-hadoop: Full Circle Back to Shared-Nothing Database Architectures. *Proceedings of the VLDB Endowment*, 7(12), 2014.
11. J. Liu, S. J. Wright, C. Ré, and V. Bittorf. An asynchronous parallel stochastic coordinate descent algorithm. *ICML*, 2014.
12. F. Niu, B. Recht, C. Ré, and S. J. Wright. Hogwild!: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent. In *NIPS*, 2011.
13. S. Wanderman-Milne and N. Li. Runtime Code Generation in Cloudera Impala. *IEEE Data Eng. Bull.*, 37(1):31–37, 2014.
14. R. S. Xin, J. Rosen, M. Zaharia, M. J. Franklin, S. Shenker, and I. Stoica. Shark: Sql and rich analytics at scale. In *SIGMOD*, pages 13–24. ACM, 2013.
15. C. Zhang and C. Ré. DimmWitted: A Study of Main-Memory Statistical Analytics. *PVLDB*, 2014.