Big Data Technologies circa 2012

Vinayak Borkar*
University of California, Irvine
vborkar@ics.uci.edu

Michael J. Carey†
University of California, Irvine
mjcarey@ics.uci.edu

1. INTRODUCTION

The growth of the World Wide Web has led to an astronomical amount of data being generated. More recently, the amount of user-generated content has seen tremendous expansion thanks to social media like Facebook and Twitter. Enterprises, researchers, and even governments consider this data to be an invaluable source of insight into people's behavior, creating a race to analyze as much data as possible. This race has driven virtually everyone, ranging from Web companies to brick and mortar businesses, into a "Big Data" frenzy. On the systems side, traditional relational databases have proven to be un-scalable, too expensive, too rigid, and/or too heavy-weight for dealing with current Big Data problems. As a result, there has been an explosion in the number of systems being developed, both within industry as well as in academia, to manage massive amounts of data.

Traditionally, data management systems were classified broadly into Online Transaction Processing (OLTP) systems and Decision Support Systems (DSS). Key-Value stores [13] have become the system of choice in the Big Data universe to perform short, single-record "transactions" at scale, playing the role of OLTP systems, albeit with limited functionality and weaker transaction guarantees. On the analytics side, MapReduce [17] and Hadoop [5] have dominated the space for scalable data analyses. There has also been an emergence of specialized systems for Big Data problems that are not naturally solved by MapReduce (those involving iterations, for example).

2. BIG DATA BACKGROUND

Google, being at the forefront of the Big Data "revolution", was forced to take matters into its own hands to stay competitive in the search engine space. Falling costs of commodity hardware made it evident that the only way to reign in the growing data was to use many computers in parallel. In 2004, Google proposed the MapReduce [17] system in conjunction with the Google File System [22] as a way to perform computation at massive scale using commodity computers. The MapReduce framework greatly simplified parallel computation for programmers by letting them avoid parallel programming. Programmers simply had to implement simple single-threaded code in the form of "Map" and "Reduce" functions which was invoked by the MapReduce infrastructure in parallel on different instances of data spread across Google's distributed file system. In addition to being able to index the entire web in reasonable amounts of time, the MapReduce system allowed programmers at Google to perform massive data processing tasks quickly using a simple programming model. Yahoo!, motivated by the MapReduce system from Google, implemented Hadoop [5] (and the Hadoop Distributed File system) and released it as open-source software.

The MapReduce paper marked the beginning of a new era of Big Data technologies. High-level layers were soon developed on top of MapReduce, further increasing programmer productivity for domain-specific tasks. Sawzall [29] and (much later) Tenzing [25] were two systems built by Google using the MapReduce layer as a runtime and parallelizing framework for text-processing and SQL execution, respectively. Outside of Google, Hadoop has become the de-facto standard for scaling data-processing and Yahoo! created the PigLatin [27] language and the Pig [28] system to run on top of Hadoop. Facebook released Apache Hive [6], a SQL-like language that also uses Hadoop as the runtime layer. Besides the Hadoop/MapReduce family of systems, alternate large-scale data-processing frameworks were proposed by various companies as well as research groups at universities. Some examples of alternative technologies include Dryad [24], DryadLINQ [32], and SCOPE [14] from Microsoft, Nephele/PACTs [7] from TU Berlin, Hyracks [10] and ASTERIX [8] from the UC Irvine, and Spark [33] from UC Berkeley.

While the MapReduce approach has been successful at analyzing large datasets that are rarely modified, there was also a need for systems to store large amounts of data and perform quick inserts, updates, and deletes of records identified by a key. This requirement led to the introduction of Key-Value stores into the Big Data ecosystem. Google developed BigTable [15], Amazon created Dynamo [18], Facebook created the Dynamo clone, Cassandra [1], and Yahoo! created the BigTable clone, HBase [2] as well as a new system, PNUTS [16] to satisfy this growing need.

*Presenter

†Co-author of tutorial content, but not presenting at the conference

The 18th International Conference on Management of Data (COMAD), 14th-16th Dec, 2012 at Pune, India.

Today's Big Data systems also include specialized platforms for solving niche problems. Pregel [26] and its open-source clones (Giraph [4] and GoldenOrb [23]) are used for parallel computation over large graphs. Similar in spirit to the MapReduce programming model, Pregel provides a simple API for programmers to express complicated graph algorithms using single-threaded code (the logic for a single vertex) which is then parallelized by the Pregel infrastructure. Machine-Learning has been another domain that has seen the emergence of specialized systems based on the Iterative-Map-Reduce-Update model [12]. Vowpal Wabbit [3] is a system custom built at Yahoo! for solving Machine Learning problems involving aggregation trees.

No list of Big Data Technologies can be considered complete without the mention of Parallel Databases, a heavily researched [20, 9, 21] area in the period from the early 1980s to the mid 1990s. Commercially, Teradata [30] and NonStop SQL [31] were tremendous successes in the parallel database space. DeWitt and Gray [19] describe important principles surrounding partitioned-parallel data computation using shared-nothing computers; the very same principles govern the operation of all the "modern" Big Data systems mentioned earlier in this section. A longer discussion of Big Data technologies can be found in [11].

3. TUTORIAL OUTLINE

The outline for the tutorial is as follows:

1. Background: Parallel Database Systems
   - Shared Everything vs. Shared Disk vs. Shared Nothing Systems
   - Three Forms of Parallelism in Parallel Database Systems: Pipelined Parallelism, Partitioned Parallelism, and Independent Parallelism
   - Parallelization Metrics: Speedup and Scaleup
   - A Case Study: Gamma
2. MapReduce and Hadoop
   - The MapReduce Programming Model
   - The Hadoop Platform
     - Hadoop Distributed File System (HDFS)
     - Fault-Tolerance in MapReduce
   - Examples
     - Word Count
     - Join and Aggregate Processing
3. High-Level Languages for Big Data
   - PigLatin
   - HiveQL
   - ASTERIX Query Language (AQL)
4. Alternative Data-Parallel Platforms
   - Overview of the Space of Big Data Platforms
   - Case Studies
     - Hyracks
     - Stratosphere (Nephele/PACTs)
5. Key-Value Stores
   - Key Value API
   - Consistency in Key-Value Stores
   - Case Studies
     - Cassandra
     - HBase
     - PNUTS
6. Specialized Systems
   - Pregel
   - Iterative-Map-Reduce-Update

4. PRESENTER BIO

Vinayak Borkar is a PhD. candidate and a Research Scientist at the University of California, Irvine in the Computer Science department. His research focuses on the efficient use of large clusters in solving Big Data problems. He was the primary developer of the Hyracks data-parallel platform. Prior to his affiliation with UCI, he developed various data-management products for close to ten years at Informatica Inc., BEA Systems Inc., and several startups. He received his Masters in Computer Science and Engineering from the Indian Institute of Technology, Bombay in 2001.

5. REFERENCES

[1] Apache Cassandra website. `http://cassandra.apache.org`.

[2] Apache HBase website. `http://hbase.apache.org`.

[3] Vowpal wabbit. `http://hunch.net/ vw/`.

[4] Giraph: Open-source implementation of Pregel. http://incubator.apache.org/giraph/.

[5] Hadoop: Open-source implementation of MapReduce. http://hadoop.apache.org.

[6] The Hive Project. http://hive.apache.org/.

[7] Dominic Battré, Stephan Ewen, Fabian Hueske, Odej Kao, Volker Markl, and Daniel Warneke. Nephele/PACTs: a Programming Model and Execution Framework for Web-Scale Analytical Processing. In *SoCC*, pages 119–130, New York, NY, USA, 2010. ACM.

[8] Alexander Behm, Vinayak R. Borkar, Michael J. Carey, Raman Grover, Chen Li, Nicola Onose, Rares Vernica, Alin Deutsch, Yannis Papakonstantinou, and Vassilis J. Tsotras. Asterix: towards a scalable, semistructured data platform for evolving-world models. *Distrib. Parallel Databases*, 29:185–216, June 2011.

[9] H. Boral, W. Alexander, L. Clay, G. Copeland, S. Danforth, M. Franklin, B. Hart, M. Smith, and P. Valduriez. Prototyping Bubba, A Highly Parallel Database System. *IEEE Trans. on Knowl. and Data Eng.*, 2(1):4–24, March 1990.

[10] Vinayak R. Borkar, Michael J. Carey, Raman Grover, Nicola Onose, and Rares Vernica. Hyracks: A Flexible and Extensible Foundation for Data-Intensive Computing. In *ICDE*, pages 1151–1162, 2011.

[11] Vinayak R. Borkar, Michael J. Carey, and Chen Li. Inside "Big Data Management": Ogres, Onions, or Parfaits? In *EDBT*, 2012.

[12] Yingyi Bu, Vinayak Borkar, Michael J. Carey, Joshua Rosen, Neoklis Polyzotis, Tyson Condie, Markus Weimer, and Raghu Ramakrishnan. Scaling datalog for machine learning on big data. Technical report, CoRR. URL: http://arxiv.org/submit/427482 or http://isg.ics.uci.edu/techreport/TR2012-03.pdf, 2012.

[13] Rick Cattell. Scalable SQL and NoSQL data stores. *SIGMOD Rec.*, 39:12–27, May 2011.

[14] Ronnie Chaiken, Bob Jenkins, Per A. Larson, Bill Ramsey, Darren Shakib, Simon Weaver, and Jingren Zhou. SCOPE: Easy and Efficient Parallel Processing of Massive Data Sets. *Proc. VLDB Endow.*, 1(2):1265–1276, 2008.

[15] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber. Bigtable: A distributed storage system for structured data. *ACM Trans. Comput. Syst.*, 26(2):4:1–4:26, June 2008.

[16] Brian F. Cooper, Raghu Ramakrishnan, Utkarsh Srivastava, Adam Silberstein, Philip Bohannon, Hans-Arno Jacobsen, Nick Puz, Daniel Weaver, and Ramana Yerneni. Pnuts: Yahoo!'s hosted data serving platform. *Proc. VLDB Endow.*, 1(2):1277–1288, August 2008.

[17] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified data processing on large clusters. In *OSDI '04*, pages 137–150, December 2004.

[18] Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, and Werner Vogels. Dynamo: amazon's highly available key-value store. *SIGOPS Oper. Syst. Rev.*, 41(6):205–220, October 2007.

[19] David DeWitt and Jim Gray. Parallel Database Systems: The Future of High Performance Database Systems. *Commun. ACM*, 35:85–98, June 1992.

[20] David J. DeWitt, Robert H. Gerber, Goetz Graefe, Michael L. Heytens, Krishna B. Kumar, and M. Muralikrishna. GAMMA - a high performance dataflow database machine. In *VLDB*, pages 228–237, 1986.

[21] Shinya Fushimi, Masaru Kitsuregawa, and Hidehiko Tanaka. An Overview of The System Software of a Parallel Relational Database Machine GRACE. In *Proceedings of the 12th International Conference on Very Large Data Bases*, VLDB '86, pages 209–219, San Francisco, CA, USA, 1986. Morgan Kaufmann Publishers Inc.

[22] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The Google File System. In *Proc. 19th ACM Symp. on Operating Systems Principles*, SOSP '03, New York, NY, USA, 2003. ACM.

[23] GoldenOrb: Open-source implementation of Pregel. http://www.raveldata.com/goldenorb/.

[24] Michael Isard, Mihai Budiu, Yuan Yu, Andrew Birrell, and Dennis Fetterly. Dryad: Distributed Data-Parallel Programs from Sequential Building Blocks. In *EuroSys*, pages 59–72, 2007.

[25] Liang Lin, Vera Lychagina, and Michael Wong. Tenzing A SQL Implementation on the MapReduce Framework. *Proceedings of the VLDB Endowment*, 4(12):1318–1327, 2011.

[26] Grzegorz Malewicz, Matthew H. Austern, Aart J.C Bik, James C. Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski. Pregel: a system for large-scale graph processing. In *Proceedings of the 2010 international conference on Management of data*, SIGMOD '10, pages 135–146, New York, NY, USA, 2010. ACM.

[27] Christopher Olston, Benjamin Reed, Utkarsh Srivastava, Ravi Kumar, and Andrew Tomkins. Pig Latin: A Not-so-Foreign Language for Data Processing. In *SIGMOD Conference*, pages 1099–1110, 2008.

[28] Pig Website. `http://hadoop.apache.org/pig`.

[29] Rob Pike, Sean Dorward, Robert Griesemer, and Sean Quinlan. Interpreting the Data: Parallel Analysis with Sawzall. *Scientific Programming*, 13(4):277–298, 2005.

[30] J. Shemer and P. Neches. The Genesis of a Database Computer. *Computer*, 17(11):42 –56, Nov. 1984.

[31] The Tandem Database Group. Nonstop SQL: A distributed, high-performance, high-availability implementation of SQL. *Second International Workshop on High Performance Transaction Systems*, September 1987.

[32] Yuan Yu, Michael Isard, Dennis Fetterly, Mihai Budiu, Úlfar Erlingsson, Pradeep Kumar Gunda, and Jon Currey. DryadLINQ: A System for General-Purpose Distributed Data-Parallel Computing Using a High-Level Language. In Richard Draves and Robbert van Renesse, editors, *OSDI*, pages 1–14. USENIX Association, 2008.

[33] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. Spark: cluster computing with working sets. HotCloud'10, page 10, Berkeley, CA, USA, 2010.