参考文献

[1] Facebook 大数据：每天处理逾 25 亿条内容和 500TB 数据，http://www.leiphone.com/news/201406/0823-danice-facebooks-data.html

[2] Google Processing 20,000 Terabytes A Day, And Growing. http://techcrunch.com/2008/01/09/google-processing-20000-terabytes-a-day-and-growing/

[3] A Comprehensive List of Big Data Statistics. https://www.bigdatanews.datasciencecentral.com/profiles/blogs/a-comprehensive-list-of-big-data-statistics

[4] A Conversation On The Role Of Big Data In Marketing And Customer Service. http://www.mediapost.com/publications/article/173109/a-conversation-on-the-role-of-big-data-in-marketin.html

[5] 为决策支持带来价值大数据的 4V 理论. http://server.51cto.com/News-281578.htm

[6] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. In *6th Symposium on Operating System Design and Implementation (OSDI)*, pages 137–150, 2004.

[7] S. Ghemawat, H. Gobioff, and S. Leung. The google file system. In *Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP)*, pages 29–43, 2003.

[8] Apache Hadoop. http://hadoop.apache.org/

[9] M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly. Dryad: distributed data-parallel programs from sequential building blocks. In *Proceedings of the 2007 EuroSys Conference (EuroSys)*, pages 59–72, 2007.

[10] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauly, M. J. Franklin, S. Shenker, and I. Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 15–28, 2012.

[11] Apache Spark. https://spark.apache.org/

[12] Apache Pig. http://pig.apache.org

[13] Apache Hive. http://hive.apache.org/

[14] Apache Mahout. http://mahout.apache.org/

[15] J. E. Gonzalez, R. S. Xin, A. Dave, D. Crankshaw, M. J. Franklin, and I. Stoica. Graphx: Graph processing in a distributed dataflow framework. In *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pages 599–613, 2014.

[16] MLlib: Apache Spark's scalable machine learning library. http://spark.apache.org/mllib/

[17] Spark SQL: Spark's module for working with structured data. http://spark.apache.org/sql/

[18] Hadoop File System. https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsUserGuide.html

[19] Apache HBase. http://hbase.apache.org/

[20] M. Zaharia, T. Das, H. Li, T. Hunter, S. Shenker, and I. Stoica. Discretized streams: fault-tolerant streaming computation at scale. In *ACM SIGOPS 24th Symposium on Operating Systems Principles (SOSP)*, pages 423–438, 2013

[21] Apache Flink. https://flink.apache.org/

[22] G. Ananthanarayanan, A. Ghodsi, A. Warfield, D. Borthakur, S. Kandula, S. Shenker, and I. Stoica. Pacman: Coordinated memory caching for parallel jobs. In *Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 267–280, 2012.

[23] H. Li, A. Ghodsi, M. Zaharia, S. Shenker, and I. Stoica. Tachyon: Reliable, memory speed storage for cluster computing frameworks. In *Proceedings of the ACM Symposium on Cloud Computing (SoCC)*, pages 6:1–6:15, 2014.

[24] Alluxio - Data Orchestration for the Cloud. https://www.alluxio.io/

[25] C. Chambers, A. Raniwala, F. Perry, S. Adams, R. R. Henry, R. Bradshaw, and N. Weizenbaum. Flumejava: easy, efficient data-parallel pipelines. In *Proceedings of the 2010 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, pages 363– 375, 2010.

[26] Cascading. http://www.cascading.org.

[27] R. Pike, S. Dorward, R. Griesemer, and S. Quinlan. Interpreting the data: Parallel analysis with sawzall. *Scientific Programming*, 13(4):277–298, 2005.

[28] B. Chattopadhyay, L. Lin, W. Liu, S. Mittal, P. Aragonda, V. Lychagina, Y. Kwon, and M. Wong. Tenzing A SQL implementation on the mapreduce framework. *PVLDB*, 4(12):1318–1327, 2011.

[29] Y. Yu, M. Isard, D. Fetterly, M. Budiu, U´. Erlingsson, P. K. Gunda, and J.

Currey. Dryadlinq: A system for general-purpose distributed data-parallel computing using a high-level language. In *8th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pages 1–14, 2008.

[30] R. Chaiken, B. Jenkins, P. Larson, B. Ramsey, D. Shakib, S. Weaver, and J. Zhou. SCOPE: easy and efficient parallel processing of massive data sets. *PVLDB*, 1(2):1265–1276, 2008.

[31] Z. Guo, X. Fan, R. Chen, J. Zhang, H. Zhou, S. McDirmid, C. Liu, W. Lin, J. Zhou, and L. Zhou. Spotting code optimizations in data-parallel pipelines through periscope. In *10th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pages 121–133, 2012.

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

[33] A. Verma, L. Cherkasova, and R. H. Campbell. ARIA: automatic resource inference and allocation for mapreduce environments. In *Proceedings of the 8th International Conference on Autonomic Computing (ICAC)*, pages 235–244, 2011.

[34] A. Verma, L. Cherkasova, and R. H. Campbell. Resource provisioning framework for mapreduce jobs with performance goals. In *ACM/IFIP/USENIX 12th International Middleware Conference (Middleware)*, pages 165–186, 2011.

[35] J. Tan, A. Chin, Z. Z. Hu, Y. Hu, S. Meng, X. Meng, and L. Zhang. DynMR: dynamic mapreduce with reducetask interleaving and maptask backfilling. In *Ninth Eurosys Conference (EuroSys)*, pages 2:1–2:14, 2014.

[36] T. Condie, N. Conway, P. Alvaro, J. M. Hellerstein, K. Elmeleegy, and R. Sears. Mapreduce online. In *Proceedings of the 7th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 313–328, 2010.

[37] L. Xu, J. Liu, and J. Wei. FMEM: A fine-grained memory estimator for mapreduce jobs. In *10th International Conference on Autonomic Computing (ICAC)*, pages 65–68, 2013.

[38] Y. Kwon, M. Balazinska, B. Howe, and J. A. Rolia. Skewtune: mitigating skew in mapreduce applications. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 25–36, 2012.

[39] M. Zaharia, B. Hindman, A. Konwinski, A. Ghodsi, A. D. Joseph, R. H. Katz, S. Shenker, and I. Stoica. The datacenter needs an operating system. In *3rd USENIX Workshop on Hot Topics in Cloud Computing (HotCloud)*, 2011.

[40] V. K. Vavilapalli, A. C. Murthy, C. Douglas, S. Agarwal, M. Konar, R. Evans, T. Graves, J. Lowe, H. Shah, S. Seth, B. Saha, C. Curino, O. O'Malley, S. Radia, B. Reed, and E. Baldeschwieler. Apache hadoop YARN: yet another resource negotiator. In *ACM Symposium on Cloud Computing (SoCC)*, pages 5:1–5:16, 2013.

[41] B. Hindman, A. Konwinski, M. Zaharia, A. Ghodsi, A. D. Joseph, R. H. Katz, S. Shenker, and I. Stoica. Mesos: A platform for fine-grained resource sharing in the data center. In *Proceedings of the 8th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2011.

[42] Open MPI: Open Source High Performance Computing. http://www.open-mpi.org/

[43] A. Rasmussen, V. T. Lam, M. Conley, G. Porter, R. Kapoor, and A. Vahdat. Themis: an i/o-efficient mapreduce. In *ACM Symposium on Cloud Computing (SoCC),* page 13, 2012.

[44] K. Nguyen, K. Wang, Y. Bu, L. Fang, J. Hu, and G. H. Xu. FACADE: A compiler and runtime for (almost) object-bounded big data applications. In *Proceedings of the Twentieth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pages 675–690, 2015.

[45] Y. Bu, V. R. Borkar, G. H. Xu, and M. J. Carey. A bloat-aware design for big data applications. In *International Symposium on Memory Management (ISMM)*, pages 119–130, 2013.

[46] L. Lu, X. Shi, Y. Zhou, X. Zhang, H. Jin, C. Pei, L. He, and Y. Geng. Lifetime-based memory management for distributed data processing systems. *PVLDB*, 9(12):936–947, 2016.

[47] L. Xu, T. Guo, W. Dou, W. Wang, and J. Wei. An experimental evaluation of garbage collectors on big data applications. *PVLDB*, 12(5):570–583, 2019.

[48] K. Nguyen, L. Fang, G. H. Xu, B. Demsky, S. Lu, S. Alamian, and O. Mutlu. Yak: A high-performance big-data-friendly garbage collector. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pages 349–365, 2016.

[49] Project Tungsten: Bringing Apache Spark Closer to Bare Metal. https://databricks.com/blog/2015/04/28/project-tungsten-bringing-spark-closer-to-bare-metal.html

[50] K. Morton, A. L. Friesen, M. Balazinska, and D. Grossman. Estimating the progress of mapreduce pipelines. In *Proceedings of the 26th International Conference on Data Engineering, (ICDE)*, pages 681–684, 2010.

[51] K. Morton, M. Balazinska, and D. Grossman. Paratimer: a progress indicator for mapreduce dags. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 507–518, 2010.

[52] N. Khoussainova, M. Balazinska, and D. Suciu. Perfxplain: Debugging mapreduce job performance. *PVLDB*, 5(7):598–609, 2012.

[53] S. Li, H. Zhou, H. Lin, T. Xiao, H. Lin, W. Lin, and T. Xie. A characteristic study on failures of production distributed data-parallel programs. In *35th International Conference on Software Engineering (ICSE)*, pages 963–972, 2013.

[54] S. Kavulya, J. Tan, R. Gandhi, and P. Narasimhan. An analysis of traces from a production mapreduce cluster. In *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid)*, pages 94–103, 2010.

[55] L. Xu, W. Dou, F. Zhu, C. Gao, J. Liu, H. Zhong, and J. Wei. Experience report: A characteristic study on out of memory errors in distributed data-parallel applications. In *26th IEEE International Symposium on Software Reliability Engineering (ISSRE)*, pages 518–529, 2015.

[56] M. Interlandi, K. Shah, S. D. Tetali, M. A. Gulzar, S. Yoo, M. Kim, T. D. Millstein, and T. Condie. Titian: Data provenance support in spark. *PVLDB*, 9(3):216–227, 2015.

[57] M. A. Gulzar, M. Interlandi, S. Yoo, S. D. Tetali, T. Condie, T. D. Millstein, and M. Kim. Bigdebug: debugging primitives for interactive big data processing in spark. In *Proceedings of the 38th International Conference on Software Engineering (ICSE)*, pages 784–795, 2016.

[58] L. Xu, W. Dou, F. Zhu, C. Gao, J. Liu, and J. Wei. Characterizing and diagnosing out of memory errors in mapreduce applications. *Journal of Systems and Software*, 137:399–414, 2018.

[59] L. Fang, K. Nguyen, G. H. Xu, B. Demsky, and S. Lu. Interruptible tasks: treating memory pressure as interrupts for highly scalable data- parallel programs. In *Proceedings of the 25th Symposium on Operating Systems Principles (SOSP)*, pages 394–409, 2015.

[60] H. Yang, A. Dasdan, R. Hsiao, and D. S. P. Jr. Map-reduce-merge: simplified relational data processing on large clusters. In *Proceedings of the ACM SIGMOD*

International Conference on Management of Data (SIGMOD), pages 1029–1040, 2007.

[61] T. Condie, N. Conway, P. Alvaro, J. M. Hellerstein, K. Elmeleegy, and R. Sears. Mapreduce online. In *Proceedings of the 7th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 313–328, 2010.

[62] Y. Bu, B. Howe, M. Balazinska, and M. D. Ernst. Haloop: Efficient iterative data processing on large clusters. *PVLDB*, 3(1):285–296, 2010.

[63] R. Power and J. Li. Piccolo: Building fast, distributed programs with partitioned tables. In *9th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pages 293–306, 2010.

[64] Structured Streaming Programming Guide. https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html

[65] Spark SQL, DataFrames and Datasets Guide. https://spark.apache.org/docs/latest/sql-programming-guide.html

[66] B. Chambers and M. Zaharia. Spark: the definitive guide: big data processing made simple. " O'Reilly Media, Inc.", 2018.

[67] 朱锋、张韶全、 黄明.《Spark SQL 内核剖析》，电子工业出版社，2018.

[68] Spark Cluster Mode Overview. https://spark.apache.org/docs/latest/cluster-overview.html

[69] RDD Programming Guide. http://spark.apache.org/docs/latest/rdd-programming-guide.html#transformations

[70] White T. Hadoop 权威指南. 清华大学出版社, 2010.

[71] M. Armbrust, R. S. Xin, C. Lian, Y. Huai, D. Liu, J. K. Bradley, X. Meng, T. Kaftan, M. J. Franklin, A. Ghodsi, and M. Zaharia. Spark SQL: relational data processing in spark. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 1383–1394, 2015.

[72] 周志华. 机器学习. 清华大学出版社, 2016.

[73] Simple Techniques for Improving SGD. http://www.cs.cornell.edu/courses/cs6787/2017fa/Lecture2.pdf

[74] M. Li, D. G. Andersen, J. W. Park, A. J. Smola, A. Ahmed, V. Josifovski, J. Long, E. J. Shekita, and B. Su. Scaling distributed machine learning with the parameter server. In *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pages 583–598, 2014.

[75] E. P. Xing, Q. Ho, W. Dai, J. K. Kim, J. Wei, S. Lee, X. Zheng, P. Xie, A. Kumar, and Y. Yu. Petuum: A new platform for distributed machine learning on big data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1335–1344, 2015.

[76] Spark on Angel: A Flexible and Powerful Parameter Server for large-scale machine learning. https://github.com/Angel-ML/angel

[77] H. Cui, J. Cipar, Q. Ho, J. K. Kim, S. Lee, A. Kumar, J. Wei, W. Dai, G. R. Ganger, P. B. Gibbons, G. A. Gibson, and E. P. Xing. Exploiting bounded staleness to speed up big data analytics. In *2014 USENIX Annual Technical Conference (USENIX ATC)*, pages 37–48, 2014.

[78] Generalized linear model. https://en.wikipedia.org/wiki/Generalized_linear_model

[79] Subgradient method. https://en.wikipedia.org/wiki/Subgradient_method

[80] (Sub)gradient Descent. http://www.cs.umd.edu/class/spring2017/cmsc422/slides0101/lecture12.pdf

[81] Subgradient Method. http://www.stat.cmu.edu/~ryantibs/convexopt/lectures/sg-method.pdf

[82] Linear Methods - RDD-based API. https://spark.apache.org/docs/latest/mllib-linear-methods.html

[83] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1-7):107–117, 1998.

[84] 卢昌海, 谷歌背后的数学. https://www.changhai.org/articles/technology/misc/google_math.php

[85] S. Verma, L. M. Leslie, Y. Shin, and I. Gupta. An experimental comparison of partitioning strategies in distributed graph processing. *PVLDB*, 10(5):493–504, 2017.

[86] G. Malewicz, M. H. Austern, A. J. C. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski. Pregel: a system for large-scale graph processing. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 135–146, 2010.

[87] Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, C. Guestrin, and J. M. Hellerstein. Distributed graphlab: A framework for machine learning in the cloud. *PVLDB*, 5(8):716–727, 2012.

[88] J. E. Gonzalez, Y. Low, H. Gu, D. Bickson, and C. Guestrin. Powergraph:

Distributed graph-parallel computation on natural graphs. In *10th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pages 17–30, 2012.

[89] GraphX Programming Guide. https://spark.apache.org/docs/latest/graphx-programming-guide.html

[90] H. Fu, D. K J Lin, and H. Tsai. Damping factor in Google page ranking. *Applied Stochastic Models in Business and Industry (2006)*: 431-444.

[91] A. Roy, I. Mihailovic, and W. Zwaenepoel. X-stream: edge-centric graph processing using streaming partitions. In *ACM SIGOPS 24th Symposium on Operating Systems Principles (SOSP)*, pages 472–488, 2013.

[92] PowerGraph: A framework for large-scale machine learning and graph computation. https://github.com/jegonzal/PowerGraph

[93] Bulk synchronous parallel – Wikipedia. https://en.wikipedia.org/wiki/Bulk_Synchronous_Parallel

[94] Gelly: Flink Graph API. https://ci.apache.org/projects/flink/flink-docs-stable/dev/libs/gelly/

[95] Iterative Graph Processing. https://ci.apache.org/projects/flink/flink-docs-release-1.9/dev/libs/gelly/iterative_graph_processing.html

[96] Spark Configuration. https://spark.apache.org/docs/latest/configuration.html

[97] Quadratic probing – Wikipedia. https://en.wikipedia.org/wiki/Quadratic_probing

[98] Cache replacement policies – Wikipedia. https://en.wikipedia.org/wiki/Cache_Replacement_Policies

[99] [SPARK-14289] Support multiple eviction strategies for cached RDD partitions. https://issues.apache.org/jira/browse/SPARK-14289

[100] Hadoop DistributedCache. https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/DistributedCacheDeploy.html

[101] 范斌，顾荣，《Alluxio：大数据统一存储原理与实践》，电子工业出版社，2019

[102] T. Xiao, J. Zhang, H. Zhou, Z. Guo, S. McDirmid, W. Lin, W. Chen, and L. Zhou. Nondeterminism in mapreduce considered harmful? an empirical study on non-commutative aggregators in mapreduce programs. In *36th International Conference on Software Engineering (ICSE)*, pages 44–53, 2014.