Bibliography

Abdelkader, D.M. and Omara, F., (2012). Dynamic task scheduling algorithm with load balancing for heterogeneous computing system. *Egyptian Informatics Journal*, *13*(2), pp.135-145.

Agrawal, D., Bernstein, P., Bertino, E., Davidson, S., Dayal, U., Franklin, M., Gehrke, J., Haas, L., Halevy, A., Han, J. and Jagadish, H.V., (2011). Challenges and Opportunities with big data 2011-1.

Ahmad, S.G., Liew, C.S., Munir, E.U., Ang, T.F. and Khan, S.U., (2016). A hybrid genetic algorithm for optimization of scheduling workflow applications in heterogeneous computing systems. *Journal of Parallel and Distributed Computing*, *87*, pp.80-90.

Ahmad, S.G., Munir, E.U. and Nisar, W., (2012). PEGA: A performance effective genetic algorithm for task scheduling in heterogeneous systems. In *High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICESS), 2012 IEEE 14th International Conference on* (pp. 1082-1087). IEEE.

Anon, (2018). [online] Available at: https://github.com/fluxroot/hadaps. [Accessed 18 Oct. 2018].

Arabnejad, H. and Barbosa, J.G., (2014). List scheduling algorithm for heterogeneous systems by an optimistic cost table. *IEEE Transactions on Parallel and Distributed Systems*, *25*(3), pp.682-694.

Archive.ics.uci.edu. (2019). *UCI Machine Learning Repository: Bag of Words Data Set*. [online] Available at: https://archive.ics.uci.edu/ml/datasets/bag+of+words [Accessed 9 Apr. 2019].

Barbosa, J.G. and Moreira, B., (2011). Dynamic scheduling of a batch of parallel task jobs on heterogeneous clusters. *Parallel computing*, *37*(8), pp.428-438.

Braam, P.J. and Zahir, R., (2002). Lustre: A scalable, high performance file system. *Cluster File Systems, Inc.*

Cardellini, V., Grassi, V., Lo Presti, F. and Nardelli, M., (2015). Distributed QoS-aware scheduling in Storm. In *Proceedings of the 9th ACM International Conference on Distributed Event-Based Systems* (pp. 344-347). ACM.

Chaudhuri, S., Dayal, U. and Narasayya, V., (2011). An overview of business intelligence technology. Communications of the ACM, 54(8), pp.88-98.

Chen, M., Mao, S. and Liu, Y., (2014). Big data: A survey. Mobile networks and applications, 19(2), pp.171-209.

Choudhury, P., Chakrabarti, P.P. and Kumar, R., (2012). Online scheduling of dynamic task graphs with communication and contention for multiprocessors. *IEEE Transactions on Parallel and Distributed Systems*, *23*(1), pp.126-133.

Coulouris, G.F., Dollimore, J. and Kindberg, T., (2005). Distributed systems: concepts and design. pearson education.

Coursera. (2018). *3.4. MapReduce Fault-Tolerance - Week 1: Orientation, Introduction to Clouds, MapReduce | Coursera*. [online] Available at: https://www.coursera.org/learn/cloud-computing/lecture/cWOfn/3-4-mapreduce-fault-tolerance [Accessed 15 Dec. 2018].

Dai, W., Ibrahim, I. and Bassiouni, M., (2017), June. An improved replica placement policy for Hadoop Distributed File System running on Cloud platforms. In *Cyber Security and Cloud Computing (CSCloud), 2017 IEEE 4th International Conference on* (pp. 270-275). IEEE.

Dean, J. and Ghemawat, S., (2008). MapReduce: simplified data processing on large clusters. Communications of the ACM, 51(1), pp.107-113.

Denny, L., Herrero, R., Levin, C. and Kim, J. (2015). Cervical Cancer. *Disease Control Priorities, Third Edition (Volume 3): Cancer*, pp.69-84.

Depardon, B., Le Mahec, G. and Séguin, C., (2013). Analysis of six distributed file systems.

Dharanipragada, J., Padala, S., Kammili, B. and Kumar, V., (2017). Tula: A disk latency aware balancing and block placement strategy for Hadoop. In *Big Data (Big Data), 2017 IEEE International Conference on* (pp. 2853-2858). IEEE.

Docs.gluster.org. (2018). *Home - Gluster Docs*. [online] Available at: https://docs.gluster.org/en/latest/ [Accessed 21 Nov. 2018].

El-Rewini, H., Lewis, T. G., & Ali, H. H. (1994). *Task scheduling in parallel and distributed systems*. Prentice-Hall, Inc..

Fahmy, M.M., Elghandour, I. and Nagi, M., (2016), December. CoS-HDFS: co-locating geo-distributed spatial data in hadoop distributed file system. In *Proceedings of the 3rd IEEE/ACM International Conference on Big Data Computing, Applications and Technologies* (pp. 123-132). ACM.

Ghemawat, S., Gobioff, H. and Leung, S.T., (2003). *The Google file system* (Vol. 37, No. 5, pp. 29-43). ACM.

Ghodsi, A., Zaharia, M., Hindman, B., Konwinski, A., Shenker, S. and Stoica, I., (2011). Dominant Resource Fairness: Fair Allocation of Multiple Resource Types. In Nsdi (Vol. 11, No. 2011, pp. 24-24).

Ghosh, S., (2014). Distributed systems: an algorithmic approach. Chapman and Hall/CRC.

Gilbert, S. and Lynch, N., 2002. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *Acm Sigact News*, *33*(2), pp.51-59.

Grid5000.fr. (2018). Grid5000. [online] Available at: https://www.grid5000.fr/mediawiki/index.php/Grid5000:Home [Accessed 10 Dec. 2018].

Guan, P., Kuhl, M., Li, Z. and Liu, X., (2000). A Survey of Distributed File Systems. University of California, San Diego.

Hadoop.apache.org. (2018a). Hadoop – Apache Hadoop 2.7.2. [online] Available at: https://hadoop.apache.org/docs/r2.7.2/index.html [Accessed 18 Oct. 2018].

Hadoop.apache.org. (2018b). *Apache Hadoop 2.7.2 – HDFS Architecture*. [online] Available at: https://hadoop.apache.org/docs/r2.7.2/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html#NameNode_and_DataNodes [Accessed 7 Dec. 2019].

Hadoop.apache.org. (2018c). Apache Hadoop 2.7.2 – Hadoop: Capacity Scheduler. [online] Available at: https://hadoop.apache.org/docs/r2.7.2/hadoop-yarn/hadoop-yarn-site/CapacityScheduler.html [Accessed 13 Nov. 2018].

Hadoop.apache.org. (2018d). Apache Hadoop 2.7.2 – Hadoop: Fair Scheduler. [online] Available at: https://hadoop.apache.org/docs/r2.7.2/hadoop-yarn/hadoop-yarn-site/FairScheduler.html [Accessed 13 Nov. 2018].

Hadoop.apache.org. (2018e). Apache Hadoop 2.7.2 – YARN Node Labels. [online] Available at: https://hadoop.apache.org/docs/r2.7.2/hadoop-yarn/hadoop-yarn-site/NodeLabel.html [Accessed 18 Oct. 2018].

Hadoop.apache.org. (2018f). *Apache Hadoop 2.7.2 – HDFS Architecture*. [online] Available at: https://hadoop.apache.org/docs/r2.7.2/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html#aMoving_Computation_is_Cheaper_than_Moving_Data [Accessed 10 Dec. 2018].

Hadoop.apache.org. (2019). *Apache Hadoop 2.7.2 – Apache Hadoop YARN*. [online] Available at: https://hadoop.apache.org/docs/r2.7.2/hadoop-yarn/hadoop-yarn-site/YARN.html [Accessed 2 May 2019].

He, Y., Liu, J. and Sun, H., (2011). Scheduling functionally heterogeneous systems with utilization balancing. In *Parallel & Distributed Processing Symposium (IPDPS), 2011 IEEE International* (pp. 1187-1198). IEEE.

Herodotou, H., Lim, H., Luo, G., Borisov, N., Dong, L., Cetin, F.B. and Babu, S., (2011), January. Starfish: a self-tuning system for big data analytics. In *Cidr* (Vol. 11, No. 2011, pp. 261-272).

Howard, J.H., Kazar, M.L., Menees, S.G., Nichols, D.A., Satyanarayanan, M., Sidebotham, R.N. and West, M.J., (1988). Scale and performance in a distributed file system. *ACM Transactions on Computer Systems (TOCS)*, *6*(1), pp.51-81.

Hsiao, H.C., Chung, H.Y., Shen, H. and Chao, Y.C., (2013). Load rebalancing for distributed file systems in clouds. *IEEE transactions on parallel and distributed systems*, *24*(5), pp.951-962.

Hwang, K. and Briggs, F.A., (1985). Computer architecture and parallel processing. McGraw-Hill.

I2.wp.com. (2018). [online] Available at: https://i2.wp.com/opensourceforu.com/wp-content/uploads/2018/03/Table-1-Comparison-of-teh-best-Big-Data-frameworks-2.jpg?ssl=1 [Accessed 7 Dec. 2018].

IBM Big Data & Analytics Hub. (2018). *The Four V's of Big Data*. [online] Available at: https://www.ibmbigdatahub.com/infographic/four-vs-big-data [Accessed 22 Nov. 2018].

Ibm.com. (2019a). [online] Available at: https://www.ibm.com/downloads/cas/XKBEABLN [Accessed 23 Feb. 2019].

Ibm.com. (2019b). *IBM Knowledge Center*. [online] Available at: https://www.ibm.com/support/knowledgecenter/en/SSZUMP_7.2.1/mapreduce_integration/map_reduce_terasort_example.html [Accessed 10 Apr. 2019].

Issues.apache.org. (2019a). *[HADOOP-3445] Implementing core scheduler functionality in Resource Manager (V1) for Hadoop - ASF JIRA*. [online] Available at: https://issues.apache.org/jira/browse/HADOOP-3445 [Accessed 12 Apr. 2019].

Issues.apache.org. (2019b). *[HADOOP-3746] A fair sharing job scheduler - ASF JIRA*. [online] Available at: https://issues.apache.org/jira/browse/HADOOP-3746 [Accessed 12 Apr. 2019].

Jacobson, R. (2018). 2.5 quintillion bytes of data created every day. How does CPG & Retail manage it? - IBM Consumer Products Industry Blog. [online] IBM Consumer Products Industry Blog. Available at: https://www.ibm.com/blogs/insights-on-

business/consumer-products/2-5-quintillion-bytes-of-data-created-every-day-how-does-cpg-retail-manage-it/ [Accessed 15 Nov. 2018].

Kanemitsu, H., Hanada, M. and Nakazato, H., (2016). Clustering-based task scheduling in a large number of heterogeneous processors. IEEE Transactions on Parallel and Distributed Systems, 27(11), pp.3144-3157.

Khaldi, D., Jouvelot, P. and Ancourt, C., (2015). Parallelizing with BDSC, a resource-constrained scheduling algorithm for shared and distributed memory systems. *Parallel Computing*, *41*, pp.66-89.

Kwok, Y.K. and Ahmad, I., (1999). Static scheduling algorithms for allocating directed task graphs to multiprocessors. *ACM Computing Surveys (CSUR)*, *31*(4), pp.406-471.

Labrinidis, A. and Jagadish, H.V., (2012). Challenges and opportunities with big data. Proceedings of the VLDB Endowment, 5(12), pp.2032-2033.

Laney, D., (2001). 3D data management: Controlling data volume, velocity and variety. *META group research note*, *6*(70), p.1.

Lapkin, A., (2012). Hype Cycle for Big Data. *2012 Gartner Group*.

Li, K., Tang, X., Veeravalli, B. and Li, K., (2015). Scheduling precedence constrained stochastic tasks on heterogeneous cluster systems. *IEEE Transactions on computers*, *64*(1), pp.191-204.

Liu, Q., Cai, W., Shen, J., Fu, Z., Liu, X. and Linge, N., (2016). A speculative approach to spatial-temporal efficiency with multi-objective optimization in a heterogeneous cloud environment. *Security and Communication Networks*, *9*(17), pp.4002-4012.

Liu, Y., Jing, W., Liu, Y., Lv, L., Qi, M. and Xiang, Y., (2017). A sliding window-based dynamic load balancing for heterogeneous Hadoop clusters. *Concurrency and Computation: Practice and Experience*, *29*(3), p.e3763.

Lustre.org. (2018). *About the Lustre File System | Lustre*. [online] Available at: http://lustre.org/about/ [Accessed 23 Nov. 2018].

Meng, L., Zhao, W., Zhao, H. and Ding, Y., (2015). A Network Load Sensitive Block Placement Strategy of HDFS. *KSII Transactions on Internet & Information Systems*, *9*(9).

Munir, E.U., Mohsin, S., Hussain, A., Nisar, M.W. and Ali, S., (2013). SDBATS: a novel algorithm for task scheduling in heterogeneous computing systems. In *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2013 IEEE 27th International* (pp. 43-53). IEEE.

Muthukkaruppan, K., Ranganathan, K. and Tang, L., Facebook Inc, (2016). *Placement policy*. U.S. Patent 9,268,808.

Ncbi.nlm.nih.gov. (2019a). *Homo sapiens (ID 51) - Genome - NCBI*. [online] Available at: https://www.ncbi.nlm.nih.gov/genome/?term=human [Accessed 9 Apr. 2019].

Ncbi.nlm.nih.gov. (2019b). *Human papillomavirus type 16 (ID 5607) - Genome - NCBI*. [online] Available at: https://www.ncbi.nlm.nih.gov/genome/?term=Human+papillomavirus+type+16 [Accessed 9 Apr. 2019].

Padole, M. and Shah, A., (2018). Comparative Study of Scheduling Algorithms in Heterogeneous Distributed Computing Systems. In Advanced Computing and Communication Technologies (pp. 111-122). Springer, Singapore.

Park, D., Kang, K., Hong, J. and Cho, Y., (2016). An efficient Hadoop data replication method design for heterogeneous clusters. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing* (pp. 2182-2184). ACM.

Qu, K., Meng, L. and Yang, Y., (2016). A dynamic replica strategy based on Markov model for hadoop distributed file system (HDFS). In *Cloud Computing and Intelligence Systems (CCIS), 2016 4th International Conference on* (pp. 337-342). IEEE.

Qureshi, F., Muhammad, N. and Shin, D.R., (2016). RDP: A storage-tier-aware Robust Data Placement strategy for Hadoop in a Cloud-based Heterogeneous Environment. *KSII Transactions on Internet & Information Systems*, *10*(9).

Sandberg, R., Goldberg, D., Kleiman, S., Walsh, D. and Lyon, B., (1985). Design and implementation of the Sun network filesystem. In *Proceedings of the Summer USENIX conference* (pp. 119-130).

Satyanarayanan, M., (1989). A survey of distributed file systems. Annual Review of Computer Science, 4(1), pp.73-104.

Satyanarayanan, M., Kistler, J.J., Kumar, P., Okasaki, M.E., Siegel, E.H. and Steere, D.C., (1990). Coda: A highly available file system for a distributed workstation environment. *IEEE Transactions on computers*, *39*(4), pp.447-459.

Schwan, P., (2003). Lustre: Building a file system for 1000-node clusters. In *Proceedings of the 2003 Linux symposium*(Vol. 2003, pp. 380-386).

Shah, A. and Padole, M. (2018). Load Balancing through Block Rearrangement Policy for Hadoop Heterogeneous Cluster. *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*.

Shah, A. and Padole, M. (2019). Performance Analysis of Scheduling Algorithms in Apache Hadoop. *Data, Engineering and Applications*, pp.45-57.

Shvachko, K., Kuang, H., Radia, S. and Chansler, R., (2010). The hadoop distributed file system. In Mass storage systems and technologies (MSST), 2010 IEEE 26th symposium on (pp. 1-10). IEEE.

Subhash, G. (2018). *Hadoop scheduler*. [online] Available at: https://www.slideshare.net/subhaskghosh/hadoop-scheduler [Accessed 14 Nov. 2018].

Tanenbaum, A. and Steen, M. (2007). *Distributed systems*. Upper Saddle River, N.J.: Prentice Hall.

Tanenbaum, A.S. and Van Steen, M., (2007). Distributed systems: principles and paradigms. Prentice-Hall.

Tang, Z., Jiang, L., Zhou, J., Li, K. and Li, K., (2015). A self-adaptive scheduling algorithm for reduce start time. *Future Generation Computer Systems*, *43*, pp.51-60.

Team, D. (2018). Data locality in Hadoop: The Most Comprehensive Guide – DataFlair. [online] Data-flair.training. Available at: https://data-flair.training/blogs/data-locality-in-hadoop-mapreduce/ [Accessed 18 Oct. 2018].

Team, D. (2018). *Rack Awareness in Hadoop HDFS – An Introductory Guide – DataFlair*. [online] Data-flair.training. Available at: https://data-flair.training/blogs/rack-awareness-hadoop-hdfs/ [Accessed 18 Oct. 2018].

Thanh, T.D., Mohan, S., Choi, E., Kim, S. and Kim, P., (2008). A taxonomy and survey on distributed file systems. In Networked Computing and Advanced Information Management, 2008. NCM'08. Fourth International Conference on (Vol. 1, pp. 144-149). IEEE.

Thekkath, C.A., Mann, T. and Lee, E.K., (1997). *Frangipani: A scalable distributed file system* (Vol. 31, No. 5, pp. 224-237). ACM.

Trujillo, G., Kim, C., Jones, S., Garcia, R. and Murray, J., (2015). *Virtualizing hadoop: how to install, deploy, and optimize hadoop in a virtualized architecture*. VMware Press.

Ullman, J. (1975). NP-complete scheduling problems. *Journal of Computer and System Sciences*, 10(3), pp.384-393.

Vavilapalli, V.K., Murthy, A.C., Douglas, C., Agarwal, S., Konar, M., Evans, R., Graves, T., Lowe, J., Shah, H., Seth, S. and Saha, B., (2013). Apache hadoop yarn: Yet another resource negotiator. In Proceedings of the 4th annual Symposium on Cloud Computing (p. 5). ACM.

Vineetha, V., Biji, C. and Nair, A. (2019). SPARK-MSNA: Efficient algorithm on Apache Spark for aligning multiple similar DNA/RNA sequences with supervised learning. *Scientific Reports*, 9(1).

Wang, G., Wang, Y., Liu, H. and Guo, H., (2016). HSIP. *Scientific Programming*, *2016*, p.19.

Weil, S.A., Brandt, S.A., Miller, E.L., Long, D.D. and Maltzahn, C., (2006). Ceph: A scalable, high-performance distributed file system. In *Proceedings of the 7th symposium on Operating systems design and implementation* (pp. 307-320). USENIX Association.

Xie, J., Yin, S., Ruan, X., Ding, Z., Tian, Y., Majors, J., Manzanares, A. and Qin, X., (2010). Improving mapreduce performance through data placement in heterogeneous hadoop clusters. In *Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on* (pp. 1-9). IEEE.

Zaharia, M., Borthakur, D., Sen Sarma, J., Elmeleegy, K., Shenker, S. and Stoica, I., (2010). Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling. In *Proceedings of the 5th European conference on Computer systems* (pp. 265-278). ACM.

Zaharia, M., Konwinski, A., Joseph, A.D., Katz, R.H. and Stoica, I., (2008). Improving MapReduce performance in heterogeneous environments. In *Osdi* (Vol. 8, No. 4, p. 7).

Zhang, Y. and Zhang, Y. (2018). YARN Node Labels: Label-based scheduling and resource isolation - Hadoop Dev. [online] Hadoop Dev. Available at: https://developer.ibm.com/hadoop/2017/03/10/yarn-node-labels/ [Accessed 13 Nov. 2018].

Zheng, W. and Sakellariou, R., (2013). Stochastic DAG scheduling using a Monte Carlo approach. *Journal of Parallel and Distributed Computing*, *73*(12), pp.1673-1689.