Bibliography

[1] T. Sterling, M. Anderson, M. Brodowicz. *High Performance Computing. Modern systems and practices*. Cambridge, MA, USA: Morgan Kauffman, 2018 (cit. on pp. 1, 2).

[2] A. Heck, F. Murtagh. «Artificial intelligence applications for Hubble Space Telescope operations». In: *Knowledge-Based Systems in Astronomy*. Heidelberg, Germany: Springer, 1989. Chap. 1, pp. 3–31 (cit. on p. 1).

[3] Kaggle. *2019 Kaggle ML DS Survey*. Tech. rep. Kaggle, 2019 (cit. on p. 2).

[4] International Data Corporation (IDC). «Worldwide Spending on Artificial Intelligence Systems Will Grow to Nearly $35.8 Billion in 2019». In: *Worldwide Semiannual Artificial Intelligence Systems Spending Guide* (2019). URL: https://www.idc.com/getdoc.jsp?containerId=prUS44911419 (cit. on p. 2).

[5] I.H. Witten, E. Frank, M.A. Hall, C.J. Pal. *Data Mining. Practical Machine Learning Tools and Techniques*. Fourth. Cambridge, MA, USA: Morgan Kauffman, 2017. Chap. Appendix B (cit. on p. 2).

[6] M. Zaharia, M. Chowdhury, M.J. Franklin, S. Shenker, I. Stoica. «Spark: Cluster Computing with Working Sets». In: UC Berkley, California, USA, June 2011, p. 1. URL: https://amplab.cs.berkeley.edu/wp-content/uploads/2011/06/Spark-Cluster-Computing-with-Working-Sets.pdf (cit. on p. 3).

[7] E. Sparks, A. Talwalkar. «Spark Meetup: MLbase, Distributed Machine Learning with Spark». In: San Francisco, California, USA, June 2013. URL: http://www.slideshare.net/chaochen5496/mlllib-sparkmeetup8613finalreduced (cit. on p. 3).

[8] G. Ingersoll. «Introducing Apache Mahout». In: *IBM developers Archives* (2009). URL: https://www.ibm.com/developerworks/java/library/j-mahout/ (cit. on p. 3).

[9] *MapReduce*. Wikipedia.org, 2020. URL: https://en.wikipedia.org/wiki/MapReduce (cit. on p. 3).

[10] A. Koliopoulos, P. Yiapanis. F. Tekiner, G. Nenadic, J. Keane. «A Parallel Distributed Weka Framework for Big Data Mining Using Spark». In: *2015 IEEE International Congress on Big Data*. New York, NY, USA, June 2015 (cit. on p. 3).

[11] *Singularity*. URL: `https://singularity.lbl.gov` (cit. on p. 4).

[12] A. Hondroudakis, R. Procter. *The Design of a Tool for Parallel Program Performance Analysis and Tuning*. Edinburgh Parallel Computing Centre. The University of Edinburgh, 1998 (cit. on p. 4).

[13] *TOP500*. URL: `https://www.top500.org` (cit. on p. 5).

[14] *MareNostrum technical information*. URL: `https://www.bsc.es/marenostrum/marenostrum/technical-information` (cit. on p. 5).

[15] *Extrae*. URL: `https://tools.bsc.es/extrae` (cit. on p. 6).

[16] *Paraver: a flexible performance analysis tool*. URL: `https://tools.bsc.es/paraver` (cit. on p. 6).

[17] *PATH and CLASSPATH*. The Java Tutorials, Oracle. URL: `https://docs.oracle.com/javase/tutorial/essential/environment/paths.html` (cit. on p. 11).

[18] U. Joshi. *How Java thread maps to OS thread?* URL: `https://medium.com/@unmeshvjoshi/how-java-thread-maps-to-os-thread-e280a9fb2e06` (cit. on p. 13).

[19] *Java Native Interface Overview*. The Java Native Interface Programmer's Guide and Specification. URL: `https://docs.oracle.com/en/java/javase/11/docs/specs/jni/intro.html#java-native-interface-overview` (cit. on p. 15).

[20] *Java Virtual Machine Tool Interface (JVM TI)*. Oracle Docs. URL: `https://docs.oracle.com/javase/8/docs/technotes/guides/jvmti/` (cit. on p. 15).

[21] *Java Virtual Machine Tool Interface (JVM TI) Reference Guide*. Oracle Docs. URL: `https://docs.oracle.com/javase/8/docs/platform/jvmti/jvmti.html` (cit. on pp. 19, 20).

[22] G. Kiczales, J. Lamping, A. Menhdhekar, C. Maeda, C. Lopes, J. Loingtier, J. Irwin. «Aspect-Oriented Programming». In: Palo Alto, CA, USA, 1997 (cit. on p. 22).

[23] *The Anatomy of an Aspect. The AspectJ Language*. Eclipse Foundation. URL: `https://www.eclipse.org/aspectj/doc/released/progguide/language-anatomy.html#pointcuts` (cit. on p. 22).