

## REFERENCES

- [1] A. Shukla and Y. Simmhan, "Benchmarking distributed stream processing platforms for IoT applications," in Proc. Technology Conf. on Performance Evaluation and Benchmarking, Springer, July 2016, pp. 90–106.
- [2] T. H. R. Munige, "Real time stream processing for Internet of Things and sensing environments," in Proc. IEEE Intl. Parallel and Distributed Processing Symposium, Sep. 2016, pp. 1143-1152.
- [3] "Microservices architecture." [Online]. Available: <http://microservices.io/>. [Accessed: 24-Dec-2016].
- [4] S. Perera, "Handling Large Scale CEP Usecase with WSO2 CEP" [online], Available: <http://srinathsvi.blogspot.com/2014/07/handling-large-scale-cep-usecase-with.html> [Accessed: 23-Dec-2014].
- [5] A. Aalto, "Scalability of Complex Event Processing as a part of a distributed Enterprise Service Bus," Ph.D. dissertation, Dept. Science., Aalto University, Espoo, Nov 2012.
- [6] N. P. Schultz-Mller, M. Migliavacca, and P. Pietzuch, "Distributed complex event processing with query rewriting," in Proc. 3<sup>rd</sup> ACM Int. Conf. on Distributed Event-Based Systems, July 2009, pp. 4.
- [7] S. Perera, "Srinath's Blog: My views of the World: How to scale Complex Event Processing (CEP) Systems?" [Online]. Available: <http://srinathsvi.blogspot.com/2012/05/how-to-scale-complex-event-processing.html>. [Accessed: 24-Dec-2016].
- [8] J. Dean and S. Ghemawat. "MapReduce: simplified data processing on large clusters." Communications of the ACM 51, no. 1, Jan. 2008, pp. 107-113.
- [9] V. Gulisano, R. Jimenez-Peris, M. Patino-Martinez, C. Soriente, and P. Valduriez. "StreamCloud: An Elastic and Scalable Data Streaming System,"

in Proc. IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 12, Jan. 2012, pp. 2351-2365.

- [10] “Apache Kafka.” [Online]. Available: <http://kafka.apache.org/>. [Accessed: 29-Sep-2016].
- [11] “Apache Storm.” [Online]. Available: <http://storm.apache.org/>. [Accessed: 29-Sep-2016].
- [12] “Spark Streaming — Apache Spark.” [Online]. Available: <http://spark.apache.org/streaming/>. [Accessed: 29-Sep-2016].
- [13] “Samza.” [Online]. Available: <http://samza.apache.org/>. [Accessed: 29-Sep-2016].
- [14] “Apache Flink: Scalable Batch and Stream Data Processing.” [Online]. Available: <https://flink.apache.org/>. [Accessed: 24-Dec-2016].
- [15] V. Govindasamy and P. Thambidura, “An Efficient and Generic Filtering Approach for Uncertain Complex Event Processing,” In Proc. Intl. Conf. on Data Mining and Computer Engineering, Dec. 2012, pp. 211-216.
- [16] O. Poppe, C. Lei, S. Ahmed, and E. A. Rundensteiner. “Complete Event Trend Detection in High-Rate Event Streams,” In Proc. 2017 ACM Int. Conf. on Management of Data (SIGMOD ‘17). May 2017, pp. 109-124.
- [17] A. K. Leghari, M. Wolf, and Y. Zhou, “Efficient Pattern Detection Over a Distributed Framework,” in Proc. 8<sup>th</sup> Int. Workshop on Business Intelligence for the Real-Time Enterprise, Sep. 2014, pp. 133–149.
- [18] M. Hirzel, “Partition and compose: parallel complex event processing.” In Proc. 6<sup>th</sup> ACM Int. Conf. on Distributed Event-Based Systems (DEBS ‘12), July 2012, pp. 191-200.
- [19] N. K. Pandey, K. Zhang, S. Weiss, H. Jacobsen, and R. Vitenberg. “Distributed event aggregation for content-based publish/subscribe systems,”

In Proc. 8<sup>th</sup> ACM Int. Conf. on Distributed Event-Based Systems (DEBS '14), May 2014, pp. 95-106.

- [20] “Apache Mesos.” [Online]. Available: <http://mesos.apache.org/>. [Accessed: 24-Dec-2016].
- [21] “Production-Grade Container Orchestration," Kubernetes.io. [Online]. Available: <https://kubernetes.io/>. [Accessed: 24-Nov-2018].
- [22] S. Suhothayan, K. Gajasinghe, I. L. Narangoda, S. Chaturanga, S. Perera, and V. Nanayakkara, “Siddhi: A Second Look at Complex Event Processing Architectures,” in Proc. ACM Workshop on Gateway Computing Environments, Nov. 2011, pp. 43-50.
- [23] “wso2/siddhi,” GitHub. [Online]. Available: <https://github.com/wso2/siddhi>. [Accessed: 24-Dec-2016].
- [24] M. A. Fardbastani, and M. Sharifi, “Scalable complex event processing using adaptive load balancing,” Journal of Systems and Software, Mar. 2019, vol. 149, pp. 305-317.
- [25] “Complex Event Processor | WSO2 Inc.” [Online]. Available: <http://wso2.com/products/complex-event-processor/>. [Accessed: 24-Dec-2016].
- [26] M. Cherniack et al., “Scalable Distributed Stream Processing,” in Proc. CIDR Conf., Jan. 2003, vol. 3, pp. 257-268.
- [27] V. Govindasamy, and P. Thambidurai, “Complex Event Processing - A Survey,” Journal of Computing, April 2013, vol. 5.
- [28] K. Vikram, “Finger Lakes: A Distributed Event Stream Monitoring System,” 2007.
- [29] L. Neumeier, B. Robbins, A. Nair, and A. Kesari, “S4: distributed stream computing platform,” in Proc. IEEE Int. Conf. on Data Mining Workshops, Dec. 2010, pp. 170-177.

- [30] “Understanding Scalability,” Docs.oracle.com. [Online]. Available: [https://docs.oracle.com/cd/E23943\\_01/dev.1111/e14301/scalunder.htm#CEPED1975](https://docs.oracle.com/cd/E23943_01/dev.1111/e14301/scalunder.htm#CEPED1975). [Accessed: 07-Dec-2018].
- [31] L. Brenna, J. Gehrke, M. Hong, and D. Johansen. “Distributed event stream processing with non-deterministic finite automata”. In Proc. ACM Int. Conf. on Distributed Event-Based Systems (DEBS '09), July 2009.
- [32] K. H. Lee, Y. J. Lee, H. Choi, Y. D. Chung, and B. Moon, “Parallel Data Processing with MapReduce: A Survey,” ACM SIGMOD Record, vol. 40, no. 4, Jan. 2012, pp. 11–20.
- [33] R. Ananthanarayanan et al., “Photon: Fault-tolerant and Scalable Joining of Continuous Data Streams,” in Proc. ACM SIGMOD Int. Conf. on Management of Data, June 2013, pp. 577–588.
- [34] M. Li et al., “Event Stream Processing with Out-of-Order Data Arrival,” in Proc. 27<sup>th</sup> Int. Conf. on Distributed Computing Systems Workshops, June 2007, pp. 67–74.
- [35] “Siddhi IO TCP,” Wso2-extensions.github.io. [Online]. Available: <https://wso2-extensions.github.io/siddhi-io-tcp/>. [Accessed: 30-Nov-2018].
- [36] “Docker.” [Online]. Available: <https://www.docker.com/>. [Accessed: 29-Sep-2016].