Evaluation of Big Data frameworks using different Machine Learning algorithms

Literature Survey

Jorge Veiga et al. have evaluated the performance evaluation of big data frameworks for extensive data analytics. In this paper, the author has suggested that emerging frameworks like Spark and Flink are replacing MapReduce frameworks such as Hadoop. These frameworks improve both programming APIs and performance. The author proves this by performing a comparative study of Spark, Hadoop and Flink by considering factors like performance and scalability. The results show that there is a significant amount of reduction in execution times by replacing Hadoop with Spark or Flink[1].

P. Jakovits et al. have evaluated several frameworks, including Hadoop (1.0.3) and Spark (0.8.0) on Amazon EC2. Outputs present that Spark outperforms Hadoop up to 48x for the PAM clustering algorithm and up to 99x for the CG linear system solver. But it was also shown that Spark is slower than Hadoop in CLARA k-medoid clustering algorithm due to difficulties in Spark handling a dataset with a large number of small objects[2].

In [3], J. Shi et al. evaluated Hadoop (2.4.0) and Spark (1.3.0) using a set of data analytics workloads on a 4-node cluster. Again results show that Spark performs superiorly to Hadoop by 2.5x for WordCount and 5x for K-Means and PageRank. Authors point out the efficiency of combine and the RDD caching as the main reasons. Hadoop was almost two times faster than Spark in Sort algorithm, showing a more efficient execution model for data shuffling.

In [4], Flink and Spark are compared on a 4-node cluster using real-world datasets by N. Spangenberg et al. Results show that Spark performs exceeding better than Flink by up to 2x for WordCount. At the same time, Flink outperforms Spark by up to 2.5x for PageRank, 3x for K-Means and 5x for a relational query. The authors finally conclude that Flink is more efficient thanks to operators like groupBy or join and the pipelining of data between operators and that the Flink optimizer provides better performance with sophisticated algorithms.

In [5], Flink (0.9.0) and Spark (1.3.1) are evaluated on Amazon EC2 using three workloads from genomic applications over datasets of up to billions of genomic regions by M. Bertoni et al. Results show that Flink outperforms Spark by up to 3x for the Histogram and Mapping to Region workloads, and by up to 4x for the Join workload. Authors conclude that concurrent execution in Flink is more efficient because it produces less sequential stages and that the tuple-based pipelining of data between operators of Flink is more efficient than the block-based Spark counterpart.

In [6], Marc Kaepke has proposed A comparative evaluation of big data frameworks for graph processing. In this paper, the author has evaluated and compared GraphX based on Spark and Gelly based on Flink as two prominent graph processing frameworks. The author has performed some experiments with different graph algorithms and both real-world data and artificially generated data. For this, the author has implemented a new algorithm using both Flink and Gelly.
https://ieeexplore.ieee.org/document/8500067

In [7], the result of experiments with Storm (V 0.9.1), Hadoop (V 1.2.0) and S4 (V 0.6.0) is portrayed about real-time processing which is about analyzing the pages fetched by a web crawler. In the CPU utilization aspect of the experiment, Storm and S4 have steady CPU utilization where Storm uses less CPU, while Hadoop does not have a constant CPU utilization. Additionally, Storm performs better than the other two in analyzing a crawled page.

https://www.mdpi.com/2073-431X/3/4/117/htm

Shan Suthaharan proposed, "Big Data Classification: Problems and Challenges in Network Intrusion Prediction with Machine Learning". The author in this journal discusses system challenges that exist in Big Data problems related predictions of intruders. This journal suggests integrating Hadoop Distributed File Systems and Cloud Technologies technologies with the latest representation-learning techniques and support vector machines to predict intruders on networks through the classification of the Big Data strategy[8].

https://dl.acm.org/doi/10.1145/2627534.2627557

Ananthi Sheshasaayee et al. have proposed a technique for optimizing the tunings needed by Map-Reduce framework to attain best outcomes. Generally, this technique uses a huge amount of time. Data Analysis, Classification and Regression on Big Data using machine learning algorithms in a novel approach used in various science and medical streams. Map Reduce is a widely used framework to parallelize machine learning algorithms. The authors have proposed the Apache Spark-based model. This model is to predict the temperature from existing data by training using tree-based machine learning techniques. This model replaces the map-reduce by Spark for implementing the best prediction result. The prediction outcomes computed are compared to tree-structured ML methods concerning time and space utilization.[9]

Mehdi Assefi et al.[10] have explored the expanding body of the Apache Spark MLlib 2.0 as an open-source, distributed, scalable, and platform independent machine learning library. Specifically, they perform several real world machine learning experiments to examine the qualitative and quantitative attributes of the platform. Furthermore, they highlight current trends in big data machine learning research and provide insights for future work. Their comparative study demonstrates that the Apache Spark MLlib, as expected, is able to be faster in comparison with the Weka components they have utilized, and performing a t-test on the running time matched by either classification algorithms or the clustering method shows statistically significant differences (at $p < 0.01$) between Apache Spark MLlib and Weka.

References

1. J. Veiga, R. R. Expósito, G. L. Taboada, J. Touriño, “Performance evaluation of big data frameworks for large-scale data analytics”
2. P. Jakovits, S. N. Srirama, "Evaluating MapReduce frameworks for iterative scientific computing applications", Proc. of the International Conference on High Performance Computing & Simulation (HPCS'14), pp. 226-233, 2014.
3. J. Shi et al., "Clash of the titans: MapReduce vs. Spark for large scale data analytics", Proc. of the Very Large Data Bases (VLDB) Endowment, vol. 8, no. 13, pp. 2110-2121, 2015.
4. N. Spangenberg, M. Roth, B. Franczyk, "Evaluating new approaches of Big Data analytics frameworks", Proc. of the 18th International Conference on Business Information Systems (BIS'15), pp. 28-37, 2015.
5. M. Bertoni, S. Ceri, A. Kaitoua, P. Pinoli, "Evaluating cloud frameworks on genomic applications", Proc. of the 2015 IEEE International Conference on Big Data (IEEE BigData 2015), pp. 193-202, 2015.
6. M. Kaepke, O. Zunkunft, “A Comparative Evaluation of Big Data Frameworks for Graph Processing”
7. Saeed Shahrivari, “Beyond Batch Processing: Towards Real-Time and Streaming Big Data”
8. Shan Suthaharan, “Big data classification: problems and challenges in network intrusion prediction with machine learning”

Finalized Objectives

Big Data frameworks that will be used for comparison:

- Hadoop
- Spark
- Flink

Algorithms that will be evaluated:

- Word Count (for reference)
- Decision Tree
- Linear regression

Parameters that will be compared:

- Disk Usage
- RAM Usage
- CPU Usage
- Execution time vs Input data size
- Execution time vs Number of nodes(workers)
- Execution time (different algo)