An Experimental Survey on Big Data Frameworks

Wissem Inoubli[1], Sabeur Aridhi[2], Haithem Mezni[3], Mondher Maddouri[4] and Engelbert Mephu Nguifo[5]

1 University of Tunis El Manar, Faculty of sciences of Tunis, LIPAH, 1060, Tunis, Tunisia

2 University of Lorraine, LORIA, Campus Scientifique BP 239, Vandoeuvre-lès-Nancy, France

3 University of Jendouba, Avenue de l'Union du Maghreb Arabe, Jendouba 8189, Tunisia

4 University Clermont Auvergne, CNRS, LIMOS, F-63000 Clermont-Ferrand, France

Context and motivations

- Big Data problems lead to several research questions (scalability problems, fault tolerance and data management).
- The 4 V's of Big Data: Volume, Variety, Veracity and Velocity.
- Several Big Data frameworks have been proposed.
- Only few studies on evaluating Big Data frameworks.

Popular Big Data frameworks

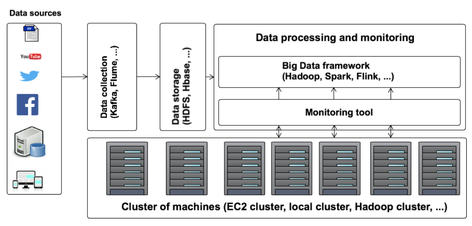
	Hadoop	Spark	Storm	Flink
Data format	Key-value	RDD	Key-value	Key-value
Processing mode	Batch	Batch and stream	Stream	Batch and stream
Data sources	HDFS	HDFS, DBMS and Kafka	HDFS, HBase and Kafka	Kafka socket
Programming model	Map and Reduce	Transformation and Action	Topology	Transformation
Cluster manager	YARN	Standalone, YARN and Mesos	YARN or zookeeper	zookeeper
Iterative computation	Yes (by running multiple MapReduce jobs)	Yes	Yes	Yes

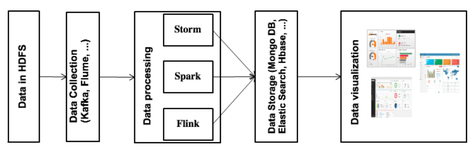
Experimental protocol

We consider two scenarios according to the data processing mode. For each scenario, we measure the performance of the presented frameworks.

Batch Mode scenario

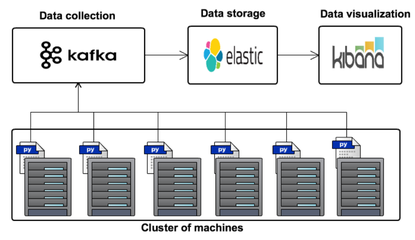
- Evaluated frameworks: Hadoop, Spark and Flink
- Workloads: WordCount, Kmeans and PageRank
- Features: Size of data, Scalability, Configuration parameters





Stream Mode scenario

- Evaluated frameworks: Spark, Storm and Flink
- Workload: ETL workload
- Features: Number of processed events



Monitoring

- Data collection: Kafka
- Data Storage: ElasticSearch
- Data Visualization: Kibana

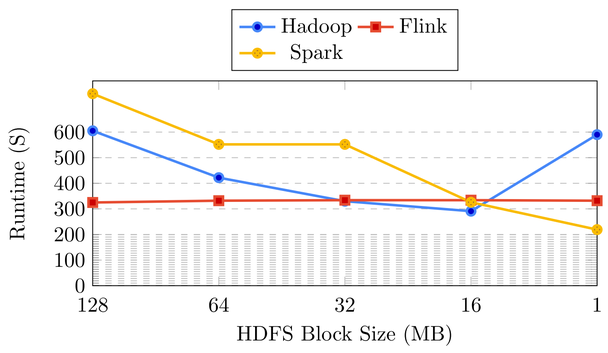
Experimental results

Batch mode processing

The results of our experiments are presented in Table 1:

Feature/Framework	Hadoop	Spark	Flink
Size of data	★★	★★	★
Iterative computing	★	★★	★★
Scalability	★★★	★★	★

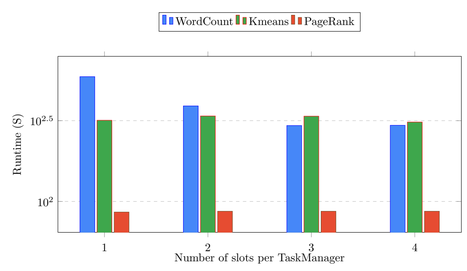
Table 1: Batch mode result

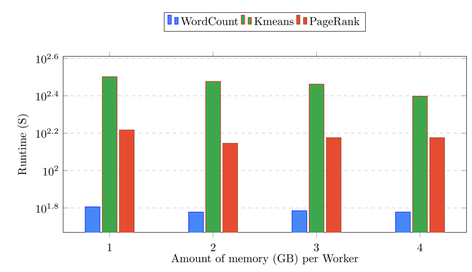


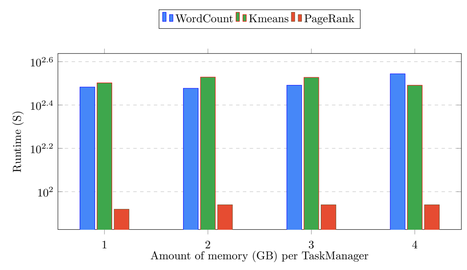
Availability

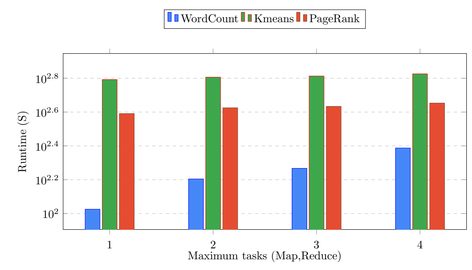
`https://members.loria.fr/SAridhi/files/software/bigdata/`

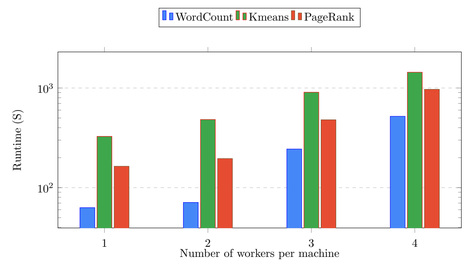
Impact of configuration parameters

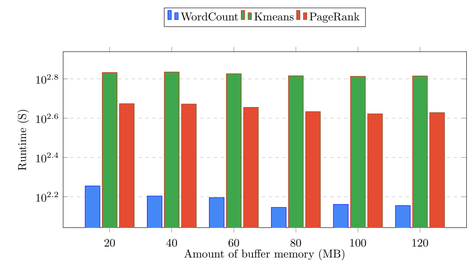










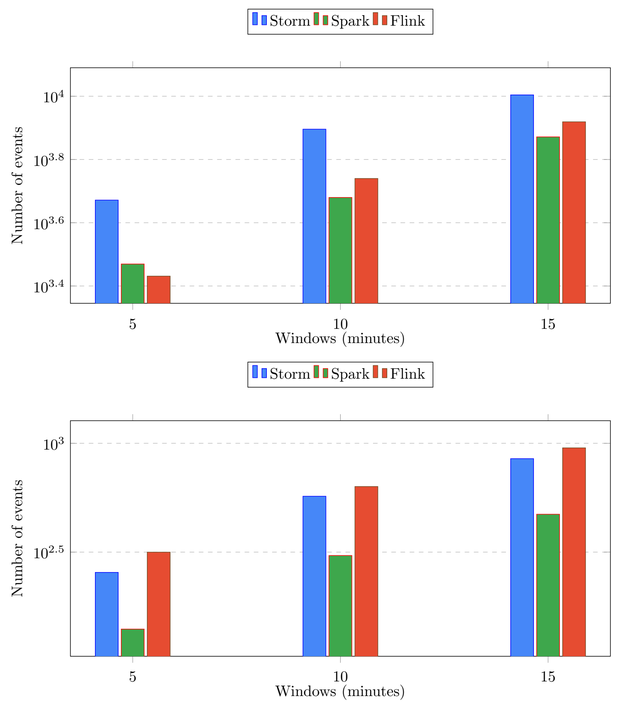


Resources consumption

Features/Frameworks	Hadoop	Spark	Flink
CPU	★★★	★★	★★
RAM	★	★★★	★★
bandwidth	★	★★	★★★
Disk	★★★	★★	★★

Table 2: Monitoring results

Stream mode processing



References

[1] S. Aridhi and E. M. Nguifo. Big graph mining: Frameworks and techniques. *Big Data Research*, 6(C):1 – 10, 2016. ISSN 2214-5796.

[2] C. P. Chen and C.-Y. Zhang. Data-intensive applications, challenges, techniques and technologies: A survey on big data. *Information Sciences*, 275:314–347, 2014.

[3] M. Chen, S. Mao, and Y. Liu. Big data: A survey. *Mobile Networks and Applications*, 19(2):171–209, 2014.

[4] D. García-Gil, S. Ramírez-Gallego, S. García, and F. Herrera. A comparison on scalability for batch big data processing on apache spark and apache flink. *Big Data Analytics*, 2(1):1, 2017.

[5] H. Zhang, G. Chen, B. C. Ooi, K.-L. Tan, and M. Zhang. In-memory big data management and processing: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 27(7):1920–1948, 2015.