An Experimental Survey on Big Data Frameworks

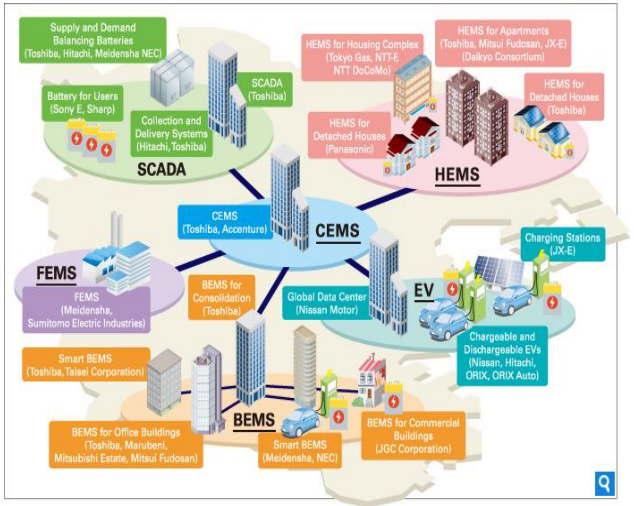
W. Inoubli, S. Aridhi, H. Mezni, M. Maddouri, E. Mephu Nguifo

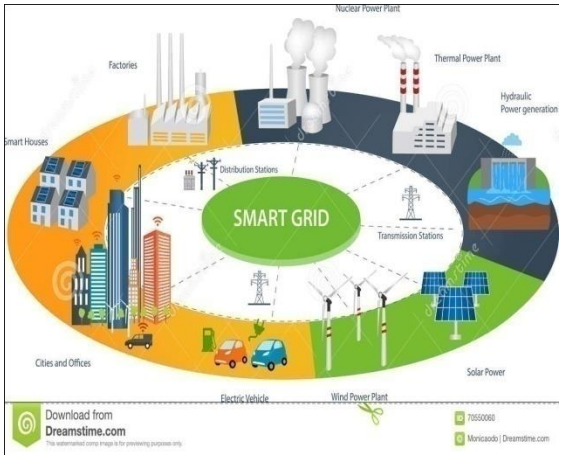
Outline

- Context and motivations
- Overview of Big Data Framewoks
- Experimental study
- Best practices
- Conclusion

Context and motivations(1)

Hearth
Personal Care Services





Context and motivations(2)

Volume of data

- 2.5 trillion bytes of data are generated everyday [7]
- 90% of the data created in the world have been created in the last 2 years [7]

Multiple data sources

- Sensors, social media, images, videos, online shopping, GPS signals

Various data format

- Relational DBMS, structured data (Xml, Json), NOSQL Databases, …

Context and motivations (3)

Big Data problems and challenges:

- Scalability
- Fault tolerance
- Storage

Several Big Data frameworks have been proposed

Context and motivations (4)

Related Works

Related works	Programming Mode	Study Frameworks	Comments
• **Dede et al., 2014,** Future Generation Computer Systems	Batch	•Hadoop •LEMO-MR •Twister	•Comparison of several MapReduce implementations
• **Zhang et al., 2015,** IEEE TKDE	Batch	•Spark	•Linear scaling •Data partitioning •Theorical study
• Veiga et al., **2016,** 2016 IEEE Big Data	Batch	•Flink •Spark •Hadoop	•Several workload
• Zhang et al., **2017,** IEEE ICDE	Stream	•Flink •Spark •Storm	•Several Complex Events Processing (CEP) presented
• Chen et al., **2014,** Information Sciences • Chen et al.,2014, Mobile Networks and Applications	Batch and Stream	•Hadoop •Storm	•Only theoretical study

Related works

Weak points and limits:

- **No** thorough studies in resources consumption for all frameworks
- Impact of the associated parameters on the performance of the studied frameworks **is not** well studied
- **No** best practices provided

Challenges:

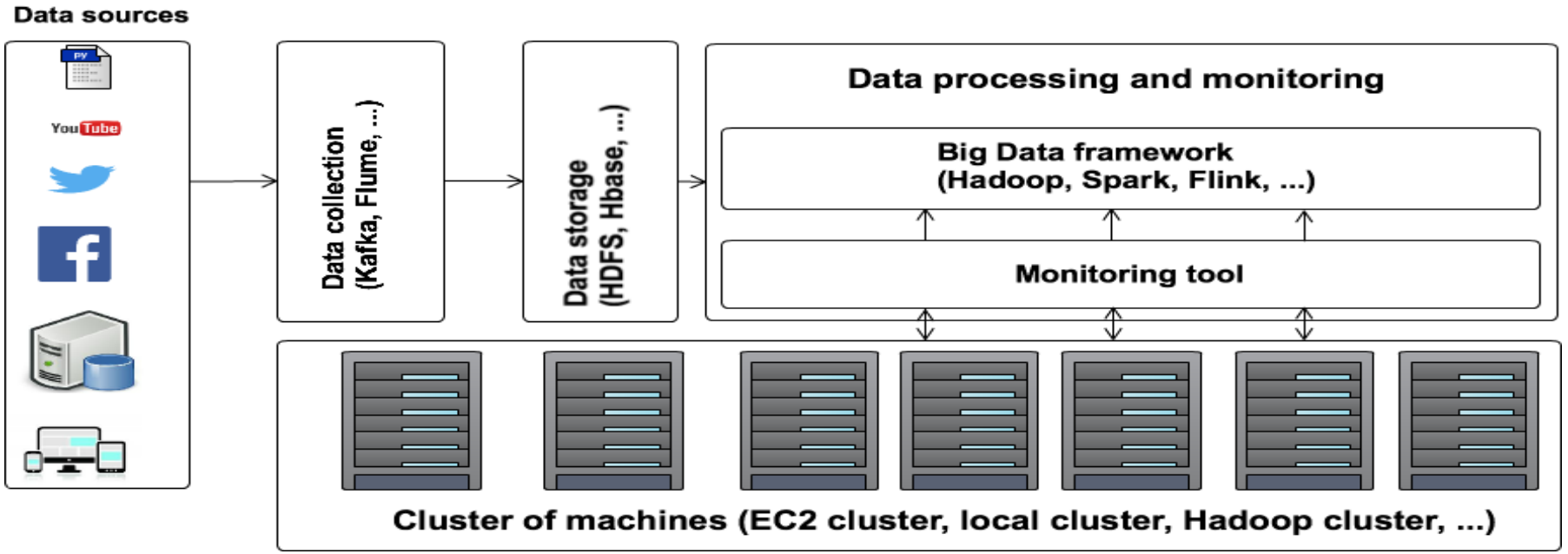
- Categorization of popular Big Data frameworks based on their features
- Experimental study:
  - Batch computing
  - Stream processing
  - Resources consumption
- Best practices based on our theoretical and empirical study

Categorization of popular Big Data frameworks

	Hadoop	Spark	Storm	Flink
Data Format	Key-value	RDD	Key-value	Key-value
Programming mode	Batch	Batch and Stream	Stream	Batch and Stream
Data sources	HDFS	HDFS, DBMS and Kafka	HDFS, HBase and Kafka	Kafka, Kinesis, message queus, socket streams
Programming model	Map and Reduce	Transformation and Action	Topology	Transformation
Pogramming languages	Java	Java, scala and python	Java	Java
Cluster manager	YARN	YARN, mesos, Standalone	Zookeeper	YARN ,Standalone
Comments	Stores large data in HDFS	Gives several APIs to develop interactive applications	Suitable for real-time applications	An extension of MapReduce with graph methods

Experimental protocol (1)

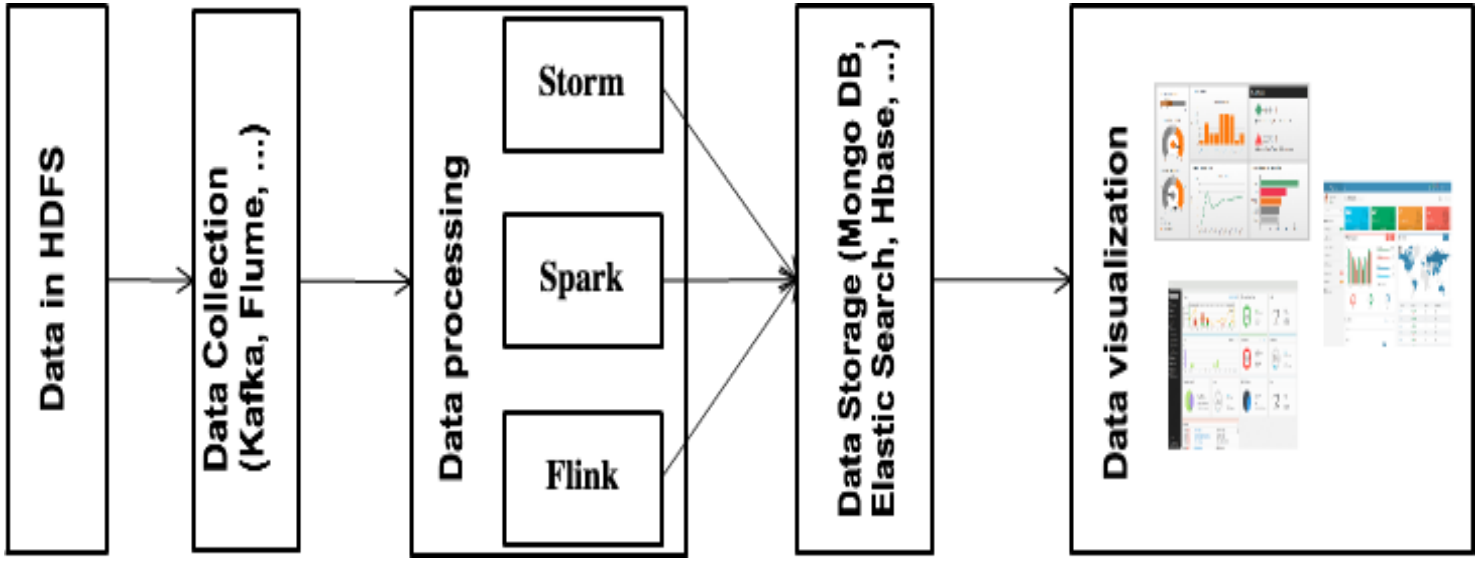
Batch Mode evaluation



- **Workload:** kmeans, WordCount and PageRank.
- **Frameworks:** Hadoop (Mapreduce), Spark and Flink.
- **Features:** Scalabilty, Configuration parameters.

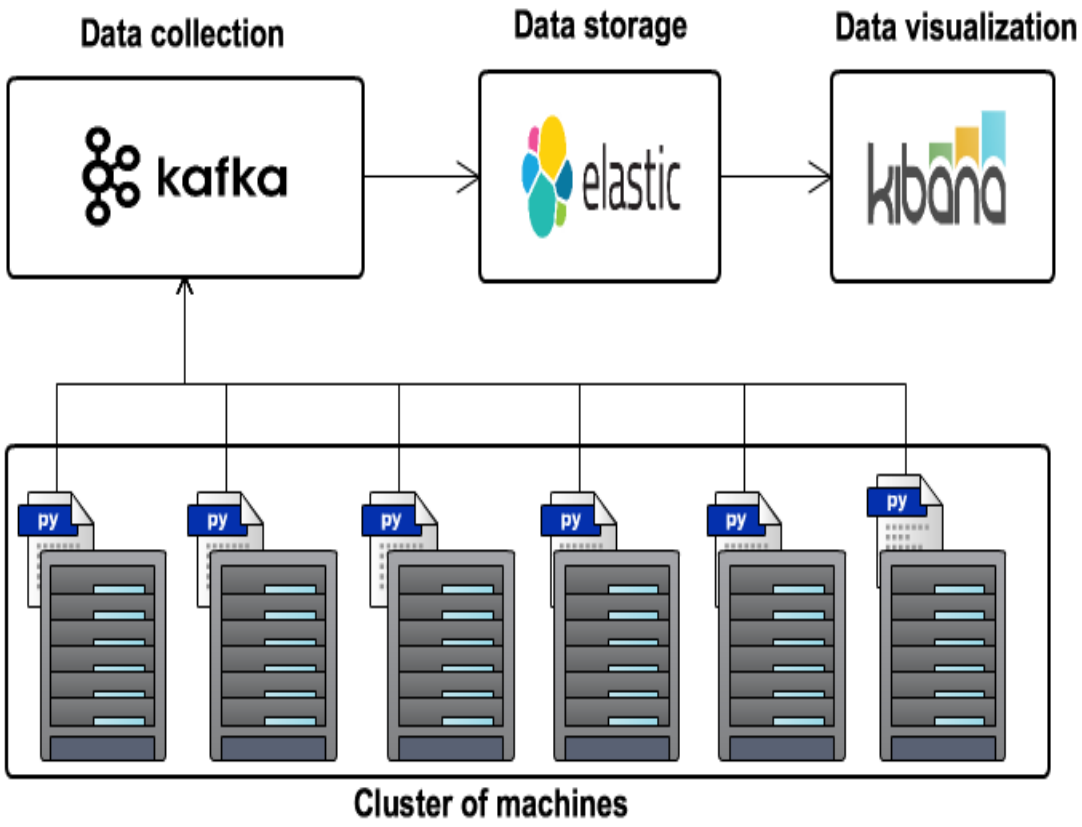
Experimental protocol (2)

StreamMode evaluation



- **Workload:** ETL Workload
- **Frameworks:** Storm, Spark and Flink
- **Features:** Number of processed events

Experimental protocol (3)

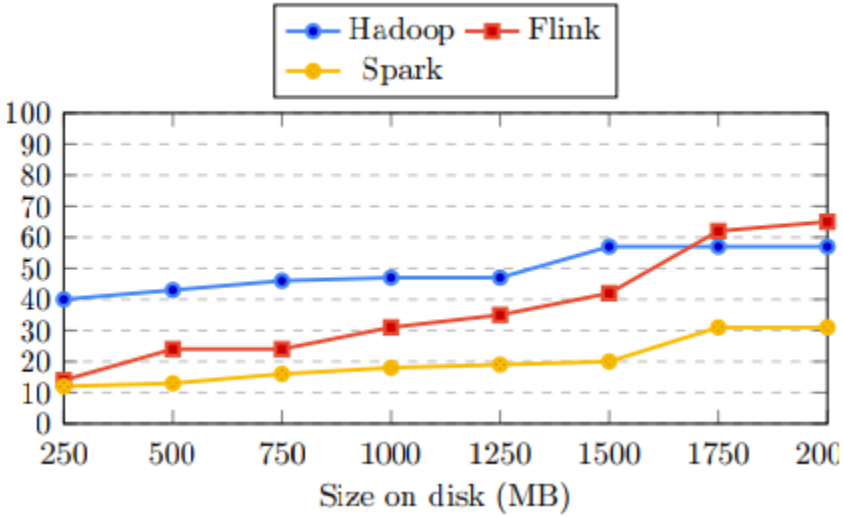


Monitoring Tool:

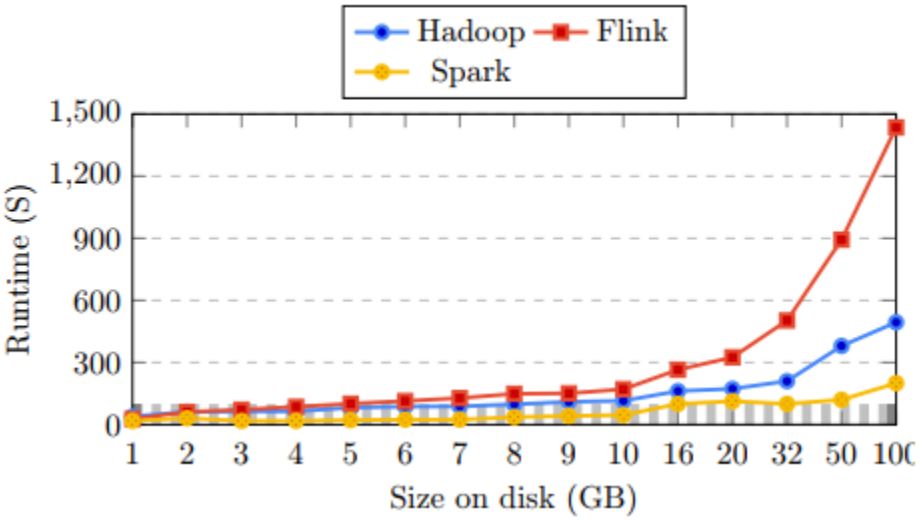
- **Data collection:** Kafka
- **Data Strorage:** ElasticSearch
- **Data Visualisation:** Kibana

Experimental results (1)

Batch Mode results:



(a) Small datasets (WordCount workload)

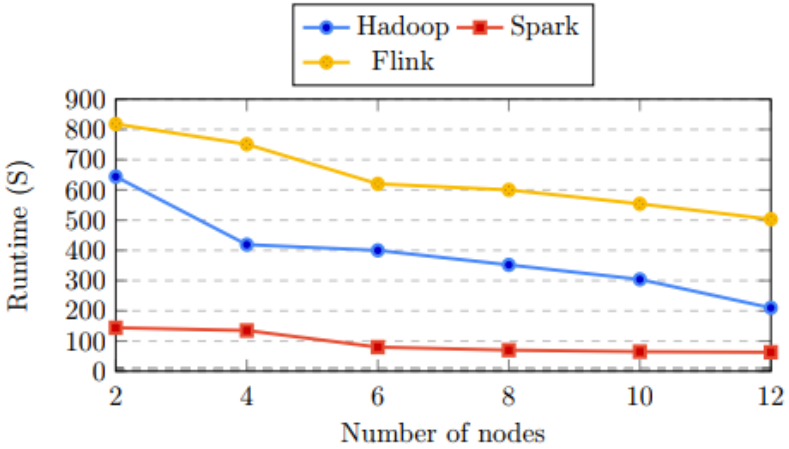


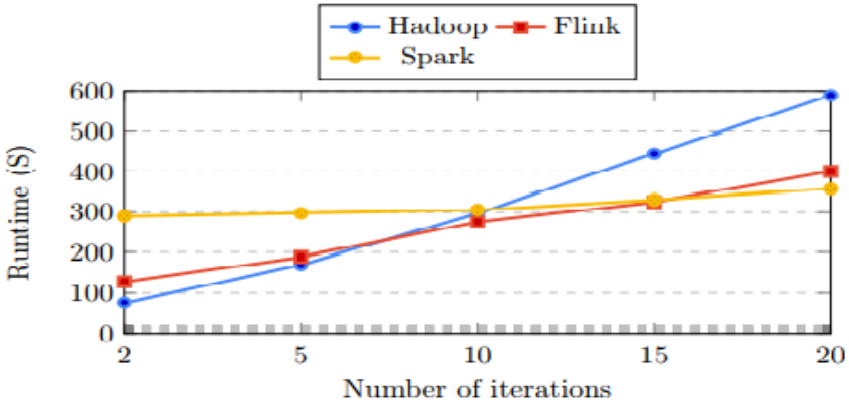
(b) Big datasets (WordCount workload)

- Impact of the size of data on the response time
- Spark is better when we use a small dataset and Hadoop is the best for big datasets

Experimental results (2)

Batch Mode results:



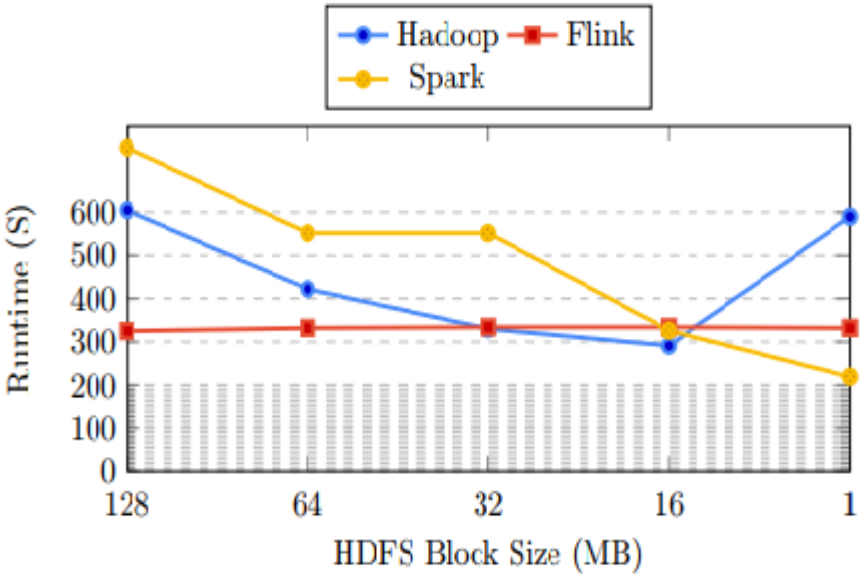


- Vary the number of machines in cluster (Wordcount workload and 50 GB of data)
- Scalability study

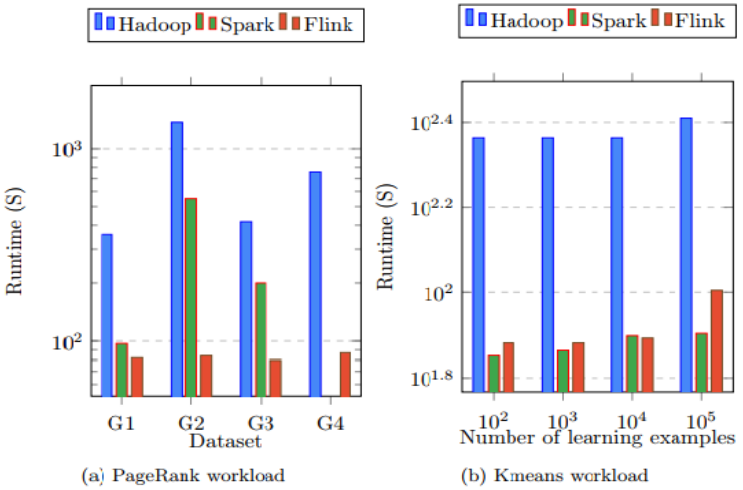
- k-Means workload with10000 training examples
- Vary the number of iterations

Experimental results (3)

Batch Mode results:



- Vary the size of **HDFS** block
- Study the partitioning of data
- WordCount workload and 50 GB of data

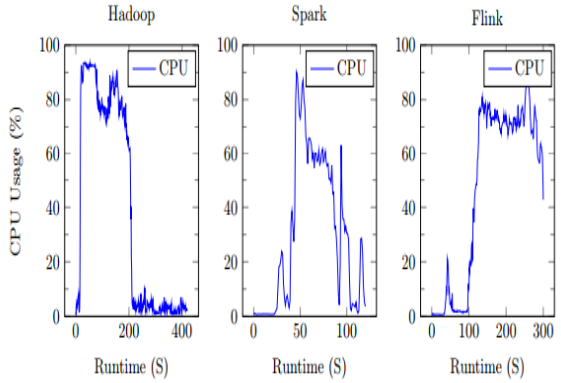


- Iterative workloads (PageRank and Kmeans)
- Vary the number of iterations

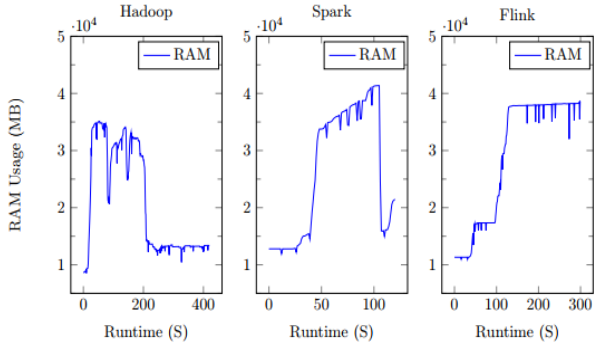
Spark and Hadoop are influenced by data partitioning
Flink perfoms well in the case of In memory applications

Experimental results (4)

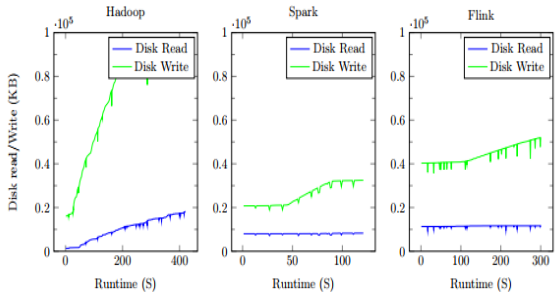
Resources consumption in Batch mode (WordCount workload and 50 GB of data):



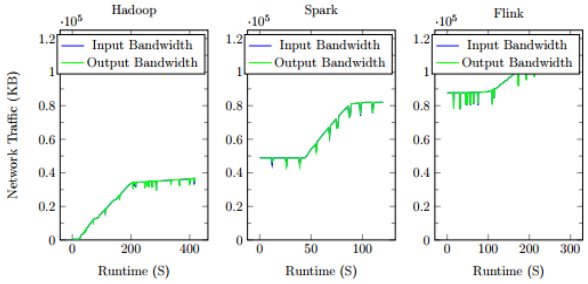
CPU resource usage



RAM resource usage



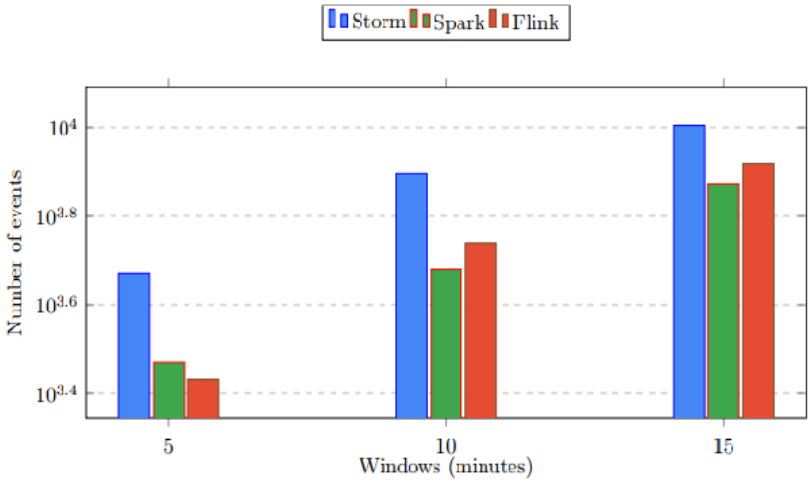
Disk Write/Read resource usage



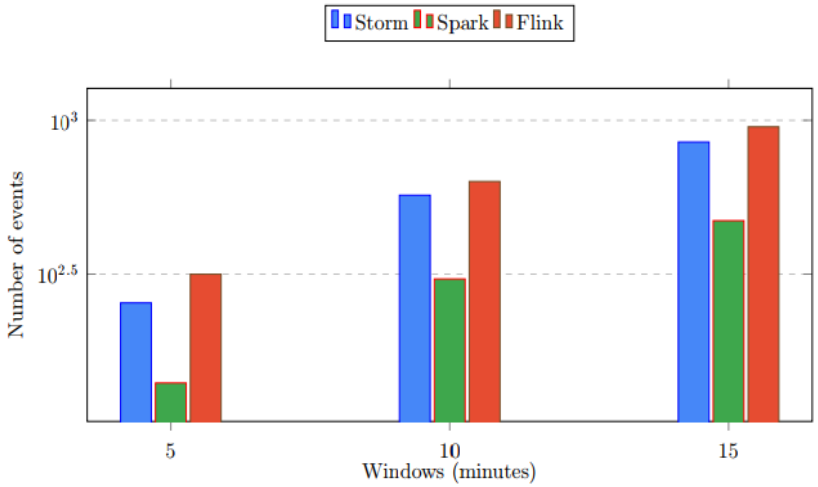
BW resource usage

Experimental results (5)

Stream Mode results :



Event with 100 kb



Event with 500 kb

Impact of the size of messages on the number of processed messages

- An ETL workload is used
- Storm performs better in the case of small datasets
- Flink provides good results in the case of big datasets

BEST PRACTICES

- **Lambda architecture**:
  - **Recommended frameworks:** Spark and Flink
  - Spark and Flink ensure both Batch and Stream processing
- **Micro-batch processing**:
  - **Recommended frameworks:** Spark
  - The features of RDD concept
- **Recommendation systems and big data mining applications**:
  - Recommended frameworks**:** Spark, Flink and Hadoop
  - They provide specific APIs for these use cases
- **Smart cites and social media**:
  - The associated data in mainly stream data
  - Storm and Flink could be used

Conclusion

- An overview of Big Data frameworks (Hadoop, Spark, Storm and Flink)
- We have identified the features of each framework
- An experimental study of the studied frameworks
- Best practices

References

[1] Dede, E., Fadika, Z., Govindaraju, M., & Ramakrishnan, L. (2014). Benchmarking MapReduce implementations under different application scenarios. *Future Generation Computer Systems*, *36*, 389-399.

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

[3] Veiga, J., Expósito, R. R., Pardo, X. C., Taboada, G. L., & Tourifio, J. (2016, December). Performance evaluation of big data frameworks for large-scale data analytics. In *Big Data (Big Data), 2016 IEEE International Conference on* (pp. 424-431). IEEE.

[4] Zhang, S., He, B., Dahlmeier, D., Zhou, A. C., & Heinze, T. (2017, April). Revisiting the design of data stream processing systems on multi-core processors. In *Data Engineering (ICDE), 2017 IEEE 33rd International Conference on* (pp. 659-670). IEEE.

[5] Chen, C. P., & Zhang, C. Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*, *275*, 314-347.

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

[7] Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, *35*(2), 137-144.

Thank you for your attention

Any questions or recommendations?