Impacts of the Stream-Processing Model on Multi Geo-spatial Environments

Paulo Souza Junior
Institute of Informatics
Federal University of Rio Grande do Sul

September 4, 2017

INSTITUTO DE INFORMÁTICA UFRGS

Agenda

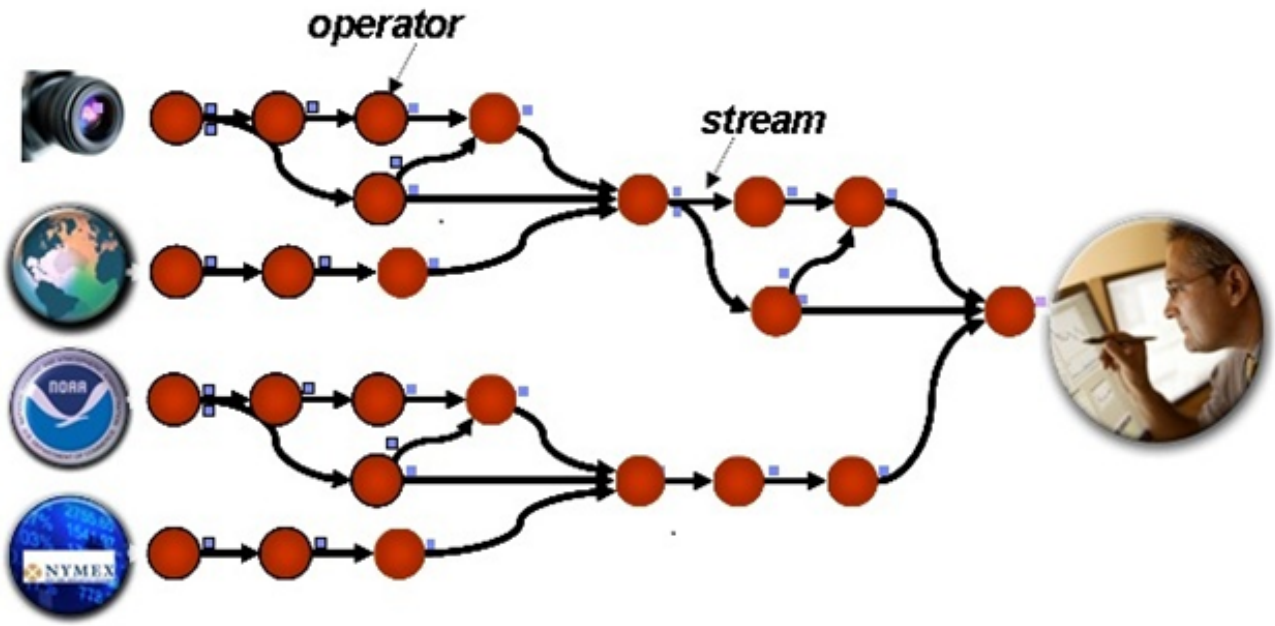
Motivation

- The most different technologies are converging to exchange data, as a consequence Internet of Things (IoT) is taking a vital place on the Big Data Analytics.
- Most of Big Data applications uses batch analytics (i.e., Batch-Processing). [1]
- There is a need for "fast" analytics.

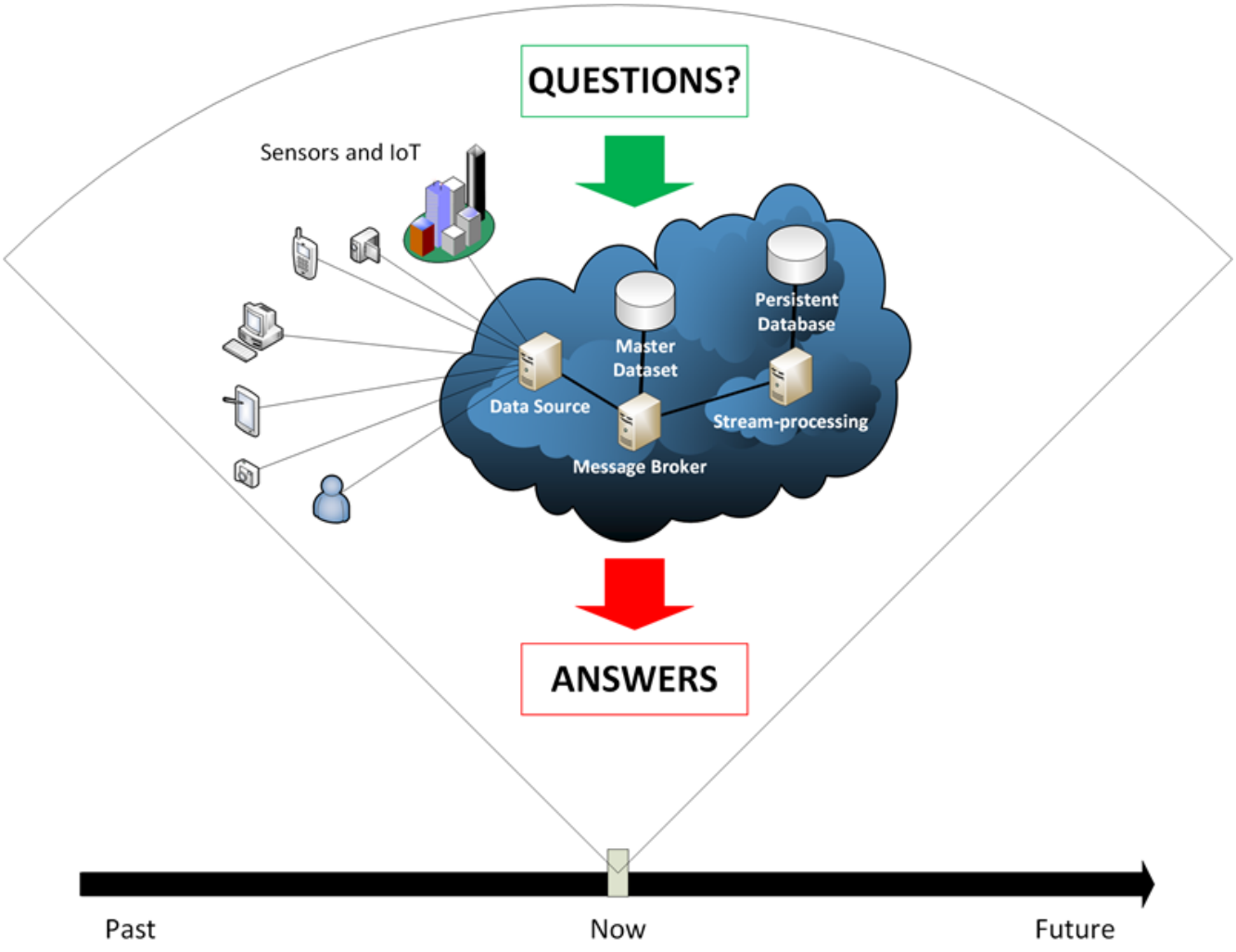
Motivation

- Batch-Processing is not structured to have low latencies in the processing of the batches. Thus, a new model was introduced – Stream-Processing.
- Stream-processing growth in recent years (i.e., real-time and near-real-time data streams) [2].

Stream Processing



Problems


QUESTIONS?
Sensors and IoT
Persistent Database
Master Dataset
Data Source
Stream-processing
Message Broker
ANSWERS
Past
Now
Future

Problems

Some limitations

- Memory
  - Over 3.3 millions posts on Facebook
  - Over 423 thousands Tweets on Twitter[a]
- Storage
  - 40 PB per annun CERN LHC [b]
- CPU
  - Limitations on Cloud suppliers [2]

[a] http://www.smartinsights.com/
[b] http://www.idgenterprise.com/

Problems

- So, how to avoid those restrictions?

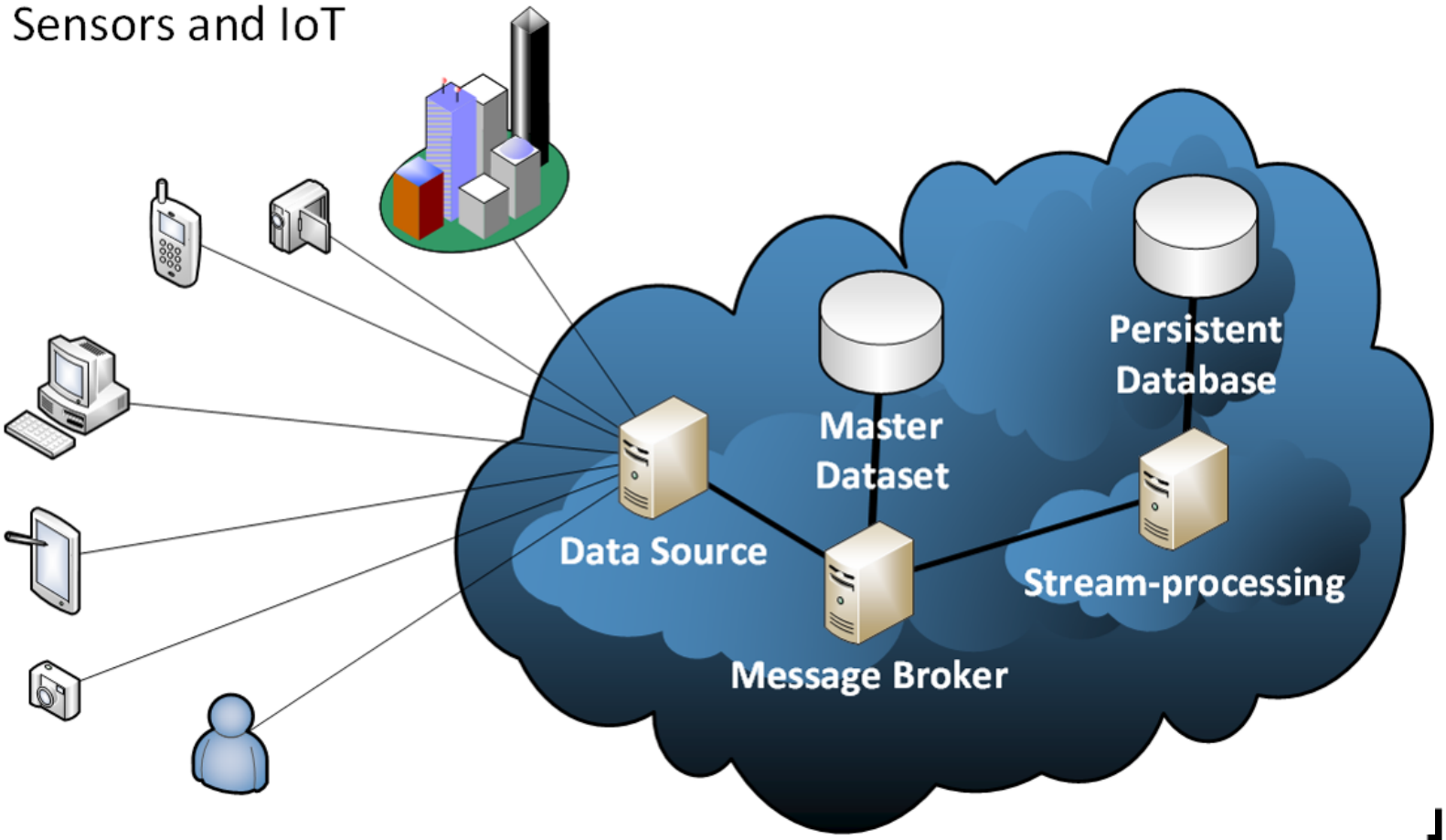
Alternatives

- Increase local capacity?
  - ...physical restrictions
- Distribute it geographically?
  - Data transfer limitations!

Problems

- How to optimize data communication in stream processing applications that make use of geographically distributed infrastructures?
- Identify a variability (quantity) in the transmission of events in a dynamic network (Internet), where there is no control over the environment. Also, test flow processing methods by identifying what best suits the scenario.

Model


Sensors and IoT
Persistent Database
Master Dataset
Data Source
Stream-processing
Message Broker

Model

Compose by

- Data Source
  - Produces data to the Message Broker
  - Filters and organizes data by removing noise
- Message Broker
  - Store and organize messages
  - Queues with replicas and consistency
- Stream-Processing Engine
  - Store and organize messages
  - Queues with replicas and consistency

Decoupled scenario

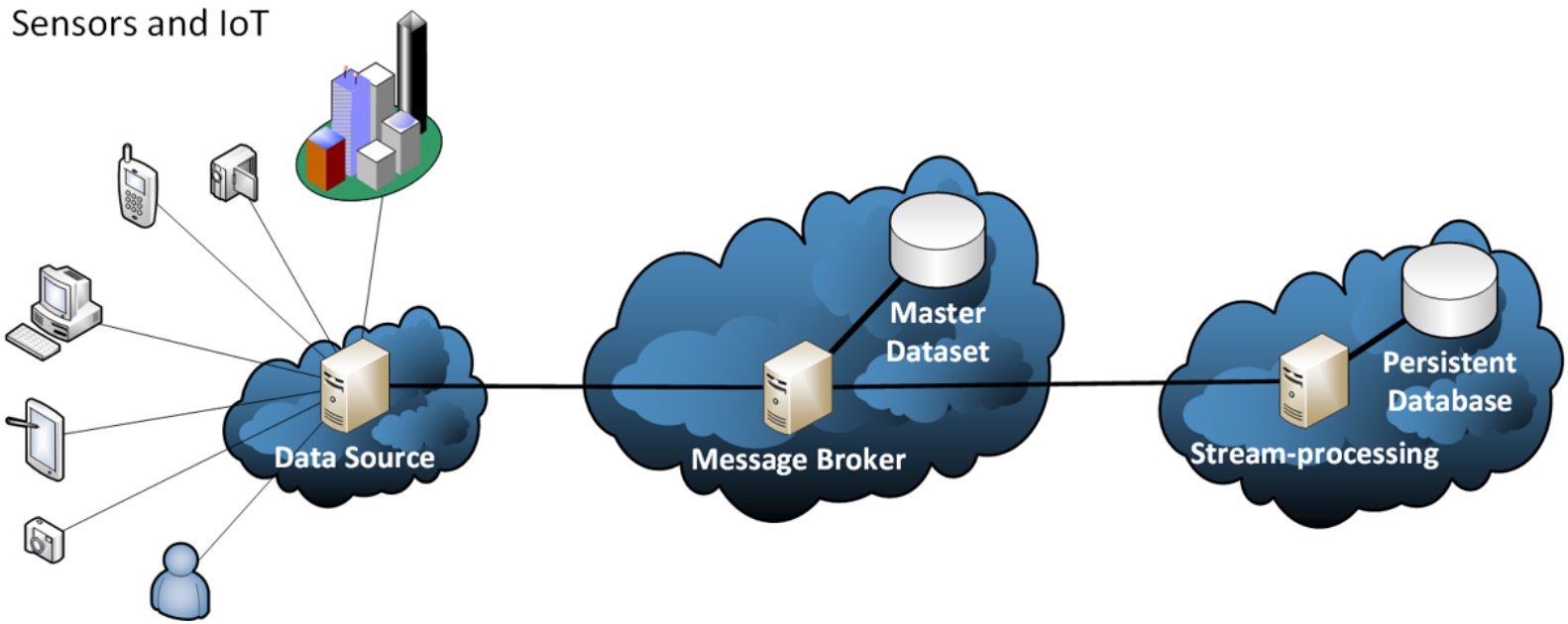


Figure: Decoupled Scenario

Stream Processing

- So, how to increase throughput considering the latency across datacenters?

Stream Processing

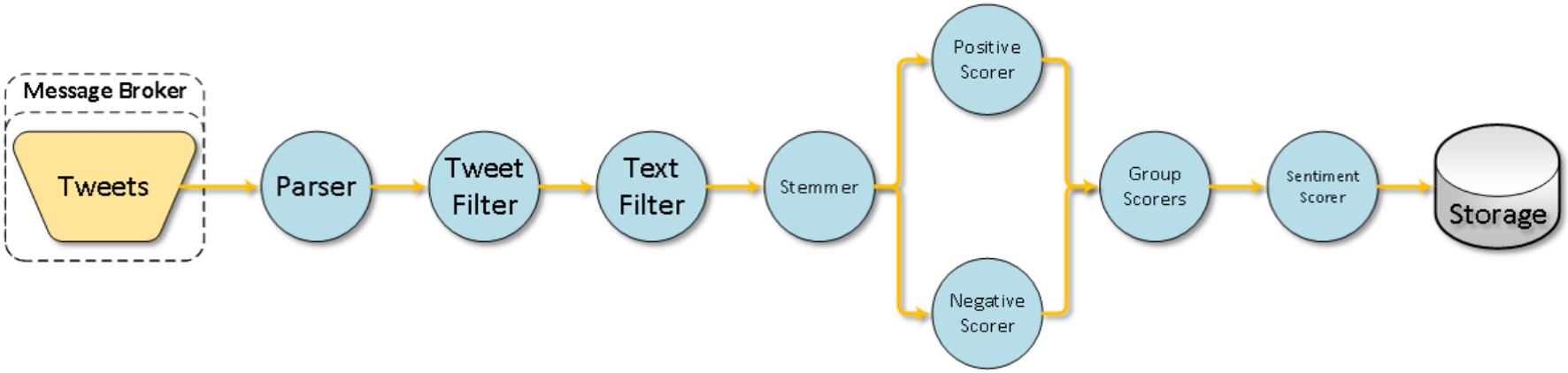
- The method of grouping data and processing in small batches was used to obtain low latency for the processing of events [3] [4]
  - Amount of events
  - Time
  - Size of events

Prototype

Software

- Apache Kafka 0.10.0 ← Message Queue framework
- Apache Flink 1.0.3 ← Stream-processing framework
  - Sentiment Analysis application for tweets
- Java 8
- Apache Maven 3.0.5

Application



- Natural Language Processing
- Classification over two classes
- Synthetic dataset, previously collected from Twitter[1]

[1] http://twitter4j.org/en/

Prototype

Software

- Apache Kafka 0.10.0 ← Message Queue
- Apache Flink 1.0.3 ← Stream-processing engine
  - Sentiment Analysis application for tweets
- Java 8
- Apache Maven 3.0.5
- Ganglia
- Hperf
- Flink's loggers

Prototype

Datacenters

- Data Source → West of the United States
  - x1 A10 8 Intel® Xeon® E5-2670 @ 2.6 GHz 56 GB DDR3-1600 MHz
- Message Broker → East Japan
  - x1 A10 8 Intel® Xeon® E5-2670 @ 2.6 GHz 56 GB DDR3-1600 MHz
- Stream-Processing Engine → Northern Europe
  - x3 A10 8 Intel® Xeon® E5-2670 @ 2.6 GHz 56 GB DDR3-1600 MHz

Evaluation

Initial experiment was conduct considering

- Batched-sized by amount of events
  - One-at-a-time [3]
  - Batch-based: 2, 10, 250, 500 and 1000 amount of events
- Replication factor equals to 20

Metrics

- Execution Time
- Time per event (throughput)

Results

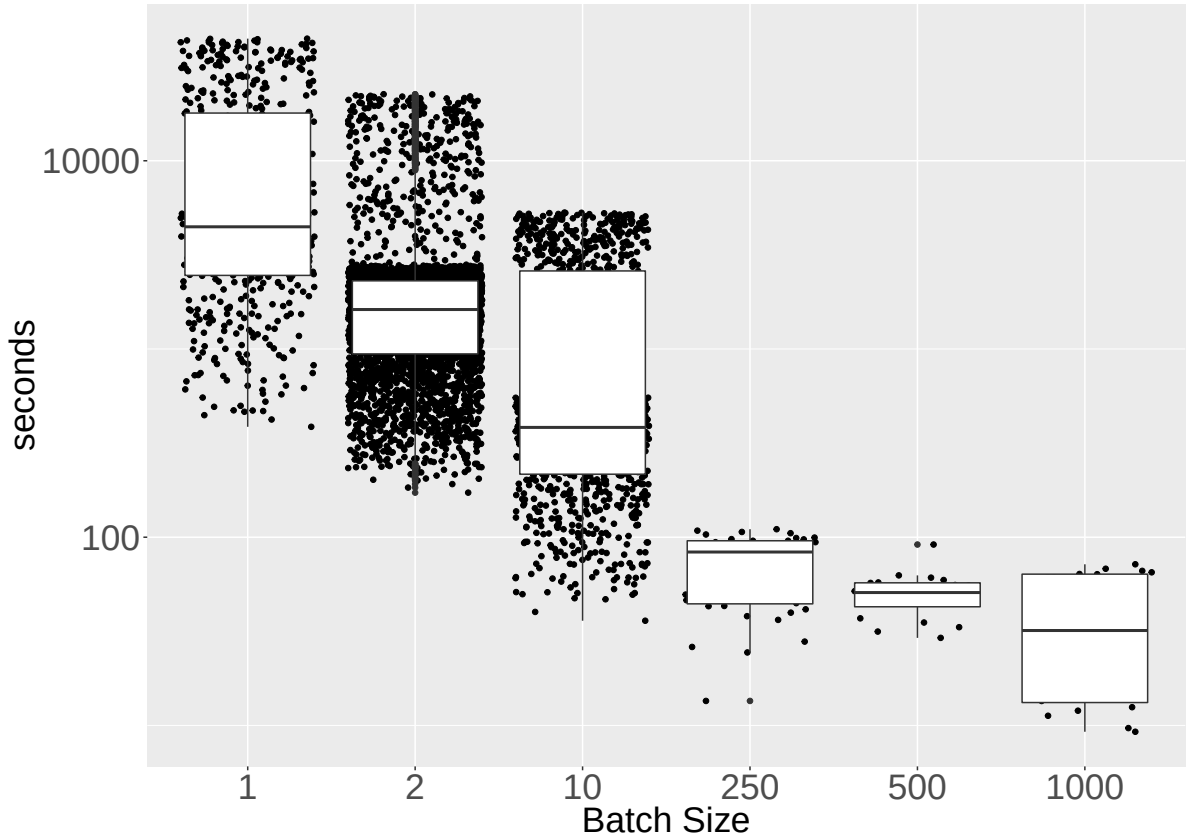


Figure: Execution time

Results

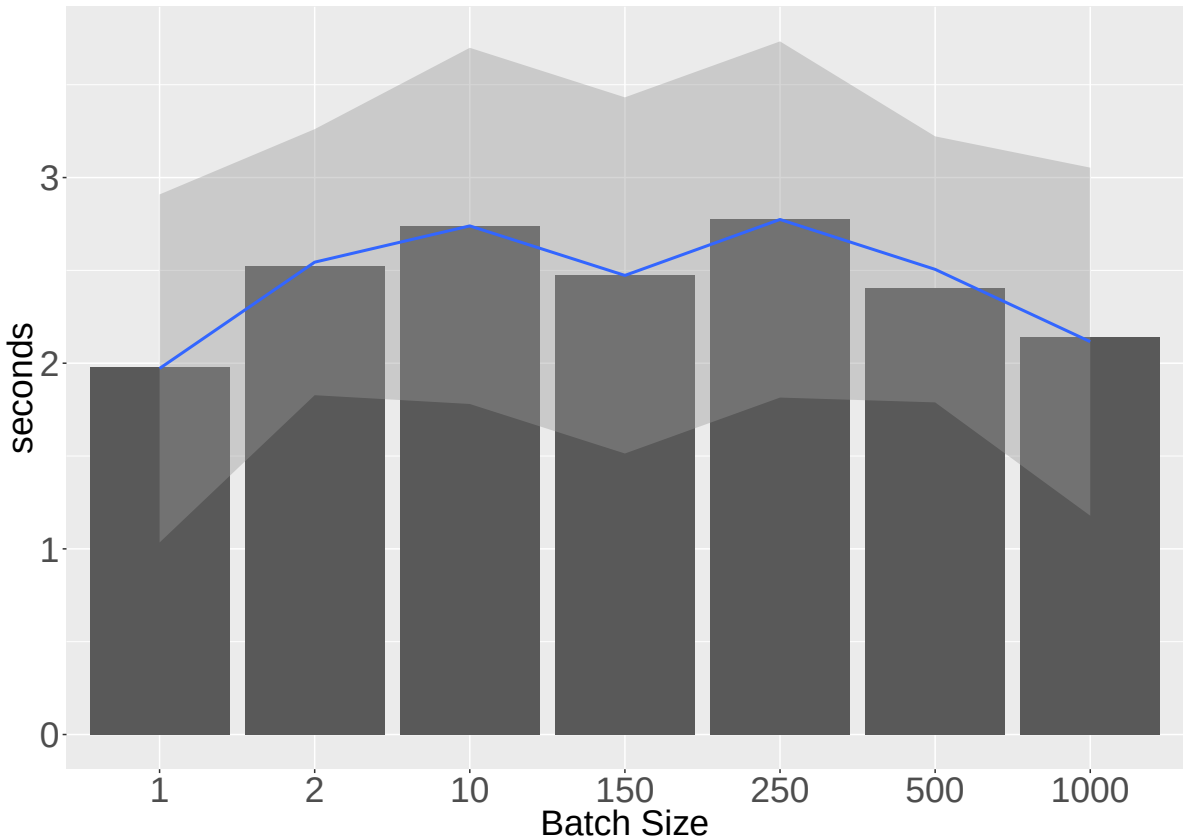


Figure: Event time on Flink

Results

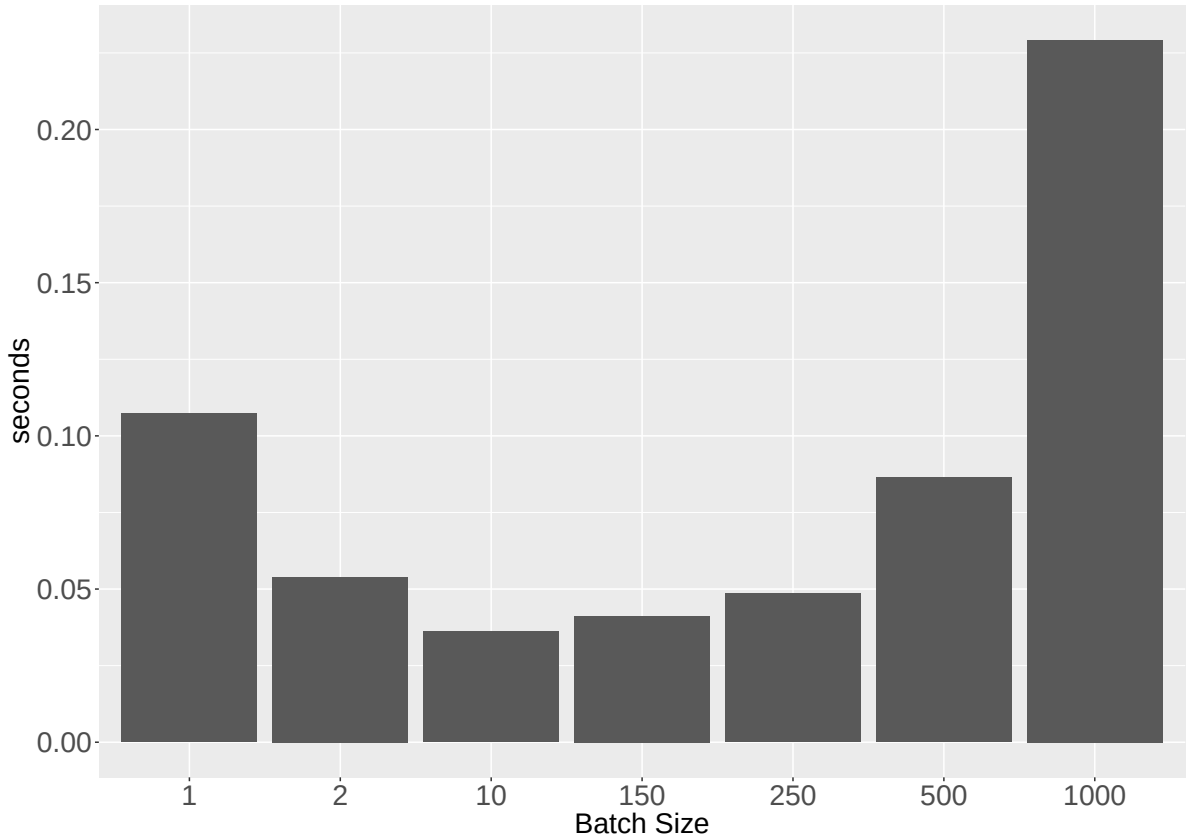


Figure: Time per event

Conclusions

- The approach resulted a time decrease of 66% from the common scenarios.
- Our findings demonstrate that it is necessary to evaluate the environment and the techniques used for sending batches, since it may vary according to network latency and environmental characteristics.
- There's a relation between network latency and throughput of events.
- Extreme importance of incorporating communication in the decisions of data movements.

.

Future work

- Dynamic batch-based size for each scenario, adapting to latency variations.
- Usage of Multi path TCP with dynamic batch sized.
- Simulating Stream-processing systems.

.


M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI'12. Berkeley, CA, USA: USENIX Association, 2012, pp. 2–2. [Online]. Available: http://dl.acm.org/citation.cfm?id=2228298.2228301


R. Tudoran, A. Costan, and G. Antoniu, "Overflow: Multi-site aware big data management for scientific workflows on clouds," *IEEE Transactions on Cloud Computing*, vol. 4, no. 1, pp. 76–89, Jan 2016.


T. Das, Y. Zhong, I. Stoica, and S. Shenker, "Adaptive stream processing using dynamic batch sizing," in *Proceedings of the ACM Symposium on Cloud Computing*, ser. SOCC '14. New

York, NY, USA: ACM, 2014, pp. 16:1–16:13. [Online]. Available: http://doi.acm.org/10.1145/2670979.2670995

X. Liao, Z. Gao, W. Ji, and Y. Wang, "An enforcement of real time scheduling in spark streaming," in *IGSC*. IEEE, 2015, pp. 1–6.