

Assessing the Impact of Network Bandwidth and Operator Placement on Data Stream Processing for Edge Computing Environments

Alexandre Veith, Marcos Dias de Assunção, Laurent Lefèvre



alexandre.veith@ens-lyon.fr

23rd June 2017

Overview

- 1 Stream Processing Context
 - Introduction
 - Data Stream Processing Engines
 - Application Graph
 - Cloud Computing
 - Fog Computing
 - Contributions
 - Goal
- 2 Experiments Setup
 - Assumptions
 - Infrastructure Configuration
 - Analysed Application
 - Evaluated Scenarios
- 3 Results
- 4 Conclusion and Future Work

Data are generated all the time!



[Atzori et al., 2010]

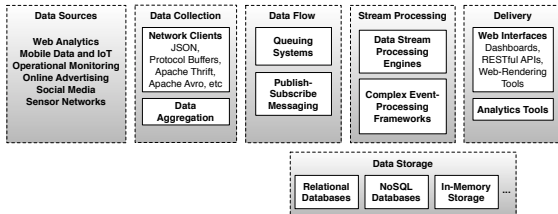
- Today's instruments and services are producing **ever-increasing amounts of data**
- Data requires processing to provide **insights or assist in decision making**
- Big Data poses challenges regarding **data transfer, storage, and processing**
- **MapReduce** has been very popular for **batch processing** (**High latencies**)
- Data generated is received in (near) real-time and requires **quick analysis** (**Low latencies**)

Data Stream Processing Engines

The interest in **processing data events as they arrive** has led to the emergence of several Data Stream Processing Engine (DSPE)



DSPEs are part of larger architectures that comprise multiple tiers



Grow at different paces and accommodate changes

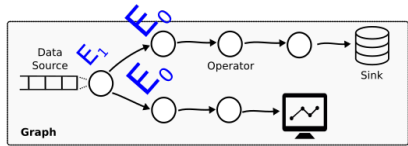
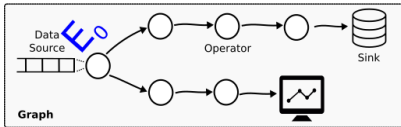
- + end-to-end latency
- + communication cost

Stream Processing Application Characteristics

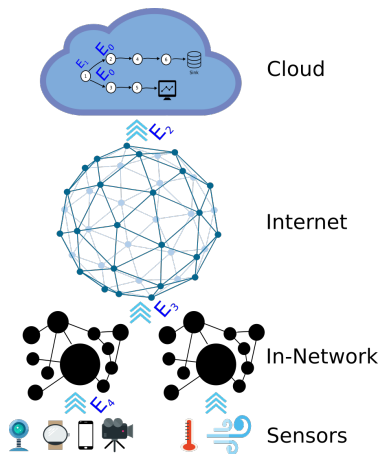
Vertices are operators that execute a function over the incoming data and **edges define** how **data flows** between them

Data Source generates the data events and **Sink** consumes them

Data Events arrive continuously (It is not possible to predict when they ceased)



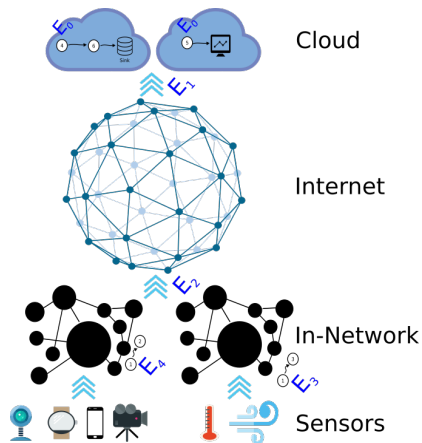
Cloud Computing



Clouds are often the target infrastructure for deploying such engines

- Pay-as-you-go business model
- Scalability
- Elasticity

Fog Computing

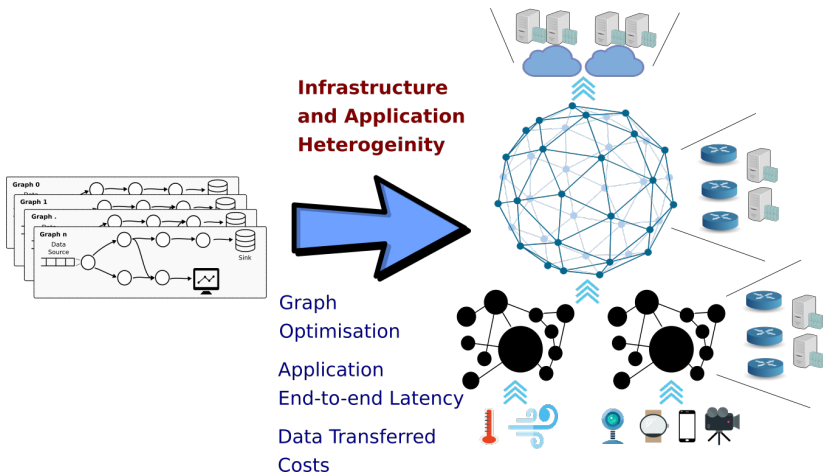


More modern solutions intend to **exploit the edges of the Internet**

- Reduce the **application end-to-end latency** and **communication costs**
- Enable services to **react locally**
- **Offload processing** from the cloud

[Chan, 2016]

Research Goals!

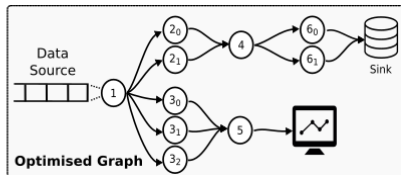
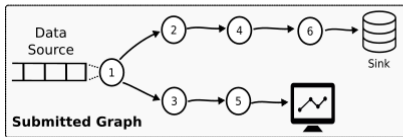


What do we really want to solve?

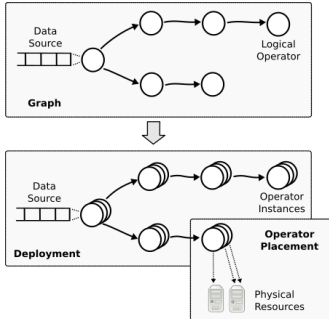
APPLICATION AUTO-PARALISATION

- Profitability mechanism

- 1 Throughput
- 2 Transferred Data



Are there benefits in deploying operators in Fog Computing?



We aim to evaluate **multiple combinations of operators deployment**, spreading them across the **cloud and edge network**

Environment restrictions

Heterogeneity

The heterogeneity taken into account for this first effort is the **network bandwidth between the edge-to-cloud**

Application's operators

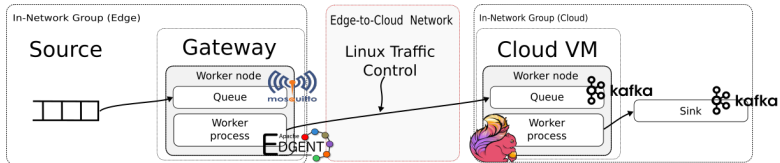
We consider **stateless operators**

Application's structure

We considered **pipelined applications**

Roles, Frameworks and Tools

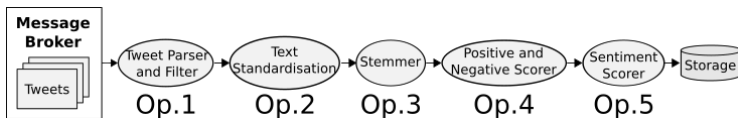
4 R410 Dell servers - Intel® Xeon® Processor E5506 4M Cache, 2.13 GHz, 4.80 GT/s Intel® QPI



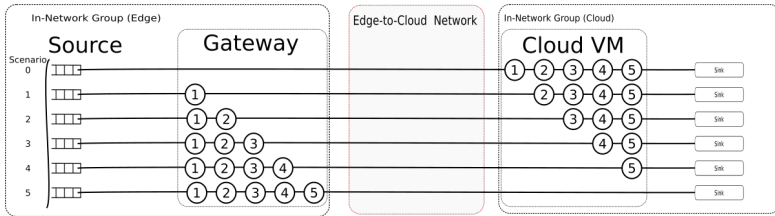
- Python psutil 5.2.0 used for measuring **resource consumption**, **CPU utilisation**, **memory usage** and **network I/O**
- Linux Network Time Protocol (NTP) for **synchronizing the time**
- **Off-the-shelf frameworks** for DSPE and Queues

Sentiment Analysis

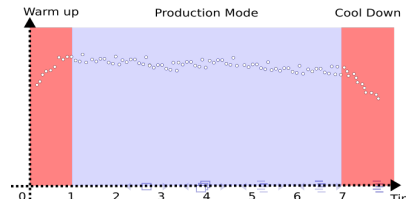
- Compute the **polarity** of tweets
- Dataset with **1 GB** of tweets
- A simple **Natural Language Processing (NLP)** technique
- Tweets are **JSON** dictionaries



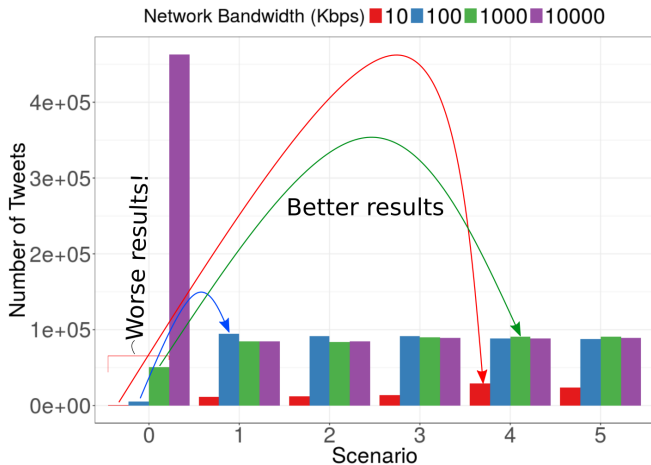
How the evaluations were carried out?



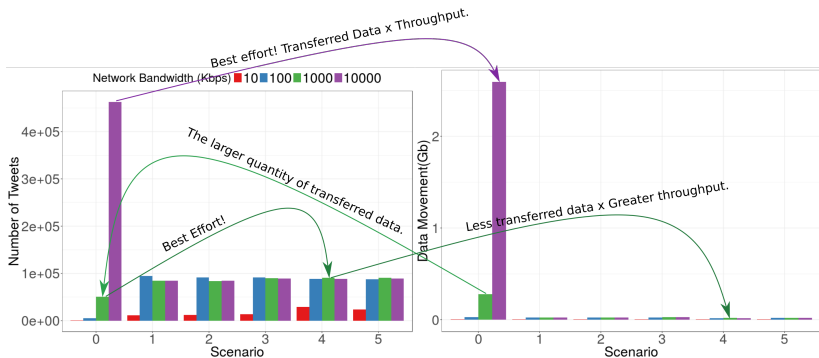
- 1 **Setup Edge-to-Cloud network bandwidth** (10, 100, 1000, 10000 Kbps)
- 2 **7 minutes** considering the testbed specification
- 3 Eliminate **warm up** and **cool down** effects



What really happened?



What really happened?



Conclusion

In this work we evaluated the **impact in terms of number of tweets** handled by a stream processing application when **varying the network bandwidth** and the **deployment of operators across cloud and edge Computing** resources.

- **graph behaviour** has a high impact on the placement, mainly points with **data reduction**
- placing operators **both in cloud and edge** network brings some **important benefits**
- depending on the **network bandwidth**, the edge deployment **improves** the number of **processed events**

Future Work – Thank you for your attention – Questions!

- Consider **resources heterogeneity**
- Isolate operators processes to achieve a certain level of **determinism**
- Look for **infrastructures** that handle FOG Computing
- Improve our model to **reduce its complexity**

References



Atzori et al. (2010)

The internet of things: A survey.

Computer Networks 54(15):2787–2805.



Chan, S. (2010)

Apache quarks, watson, and streaming analytics: Saving the world, one smart sprinkler at a time.

Bluemix Blog.



Uckelmann et al. (2011)

An architectural approach towards the future internet of things

In Architecting the internet of things pages 1–24. Springer.