Latency-Aware Strategies for Placing Data Stream Analytics onto Edge Computing
Alexandre da Silva Veith, Marcos Dias de Assuncao, Laurent Lefèvre

To cite this version:
Alexandre da Silva Veith, Marcos Dias de Assuncao, Laurent Lefèvre. Latency-Aware Strategies for Placing Data Stream Analytics onto Edge Computing. HotEdge’18 - USENIX Workshop on Hot Topics in Edge Computing, Jul 2018, Boston, United States. pp.1, 2018. hal-01875999

HAL Id: hal-01875999
https://inria.hal.science/hal-01875999
Submitted on 18 Sep 2018
Latency-Aware Strategies for Placing Data Stream Analytics onto Edge Computing

Alexandre da Silva Veith, Marcos Dias de Assunção, Laurent Lefèvre
Avalon, LIP, ENS Lyon, University of Lyon • Lyon, France
alexandre.veith@ens-lyon.fr

Acknowledgment: This work was performed within the framework of the LABEX MILYON (ANR-10-LABX-0070) of Université de Lyon, within the program “Investissements d’Avenir” (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR).

Scenarios
- Monitoring of operational infrastructure
- Anomaly detection, fraud detection
- Social networks
- Internet of Things (IoT)

Applications
- Directed dataflows whose vertices are operators that execute a function over the incoming data and edges define how data flows between them

Operator Characteristics
- Paths and location: multiple sources and sinks distributed across cloud and/or edge
- Fork/ Split: messages can be replicated or scheduled to downstream operators
- Merge/ Join: merges the outcome of upstream operators
- Selectivity: The ratio of number of input messages to output messages
- Data compression/ expansion factor: The ratio of the size of input events to the size of output events

Response time (RT): the total time taken for a message to traverse a path

Latency-Aware Decisions
- Minimize the response times in all paths

Resource and Application Models
- Example of operator placement
- Resource 1: Cloudlet 1, Cloudlet 2
- Resource 2: Internet

Finding Application Patterns
- Initial setup
- Pattern recognition
- Transform
- Series hierarchy

Response Time Rate (RTR) Strategy
- Regions definition
- Candidate placement

RTR with Region Patterns (RTR+RP) Strategy
- Candidate placement = destination infrastructure of the message
- Only cloud = cloud
- Edge and/or cloud = edge

Response time estimation - only for edge candidates

Experimental setup
- Cloudlet 1: Cloudlet 2
- Cloud: Two AMD RYZEN 7 1800x (304,51 MIPS at 3.6GHz and 1TB of RAM)
- LAN:
  - Latency: U(0.015-0.8)ms
  - Bandwidth: 100 Mbps
- WAN:
  - Latency: U(65-85)ms
  - Bandwidth: 1 Gbps

Microbenchmarks - Evaluate the impact of fork/join operators

Analysis
- Our approach: Microbenchmarks
- 95% better
- Complex Applications
- 50% faster in average
- 52% of reduction in the communication for sinks located in the edge

Conclusions
- Dynamically movement of operators across edge and cloud
- 50% faster in generic and complex dataflows

Reference

Complex Applications - Investigate the outcomes for generic and multiple path applications

Response Time (ms)

0 100 200 300 400 500

Cloud placements

0.0 0.2 0.4 0.6 0.8 1.0

Norm. CDFs

AppA

AppB