Latency-Aware Placement of Data Stream Analytics on Edge Computing

Alexandre da S. Veith

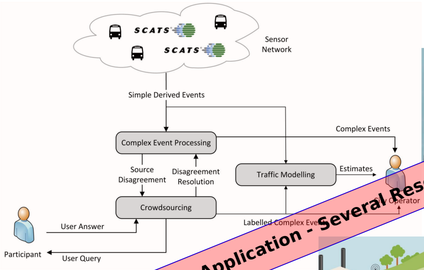
Advised by Marcos D. de Assunção, Laurent Lefèvre

alexandre.veith@ens-lyon.fr

26th June 2018

Data Stream Analytics

Traffic Control



Wearable Cognitive Assistance

A new modality of computing

Entirely new genre of applications

Wearable UI with wireless access to cloudlet

Real-time cognitive engines **on cloudlet**

- scene analysis
- object/person recognition
- speech recognition
- language translation
- planning, navigation
- question-answering technology
- voice synthesis
- real-time machine learning
- ...

Low latency response is crucial

Wide Application - Several Research Fields

The Drawing Assistant: Automated Drawing Guidance and Feedback from Photographs

Emmanuel Iarussi
REVES / Inria Sophia Antipolis

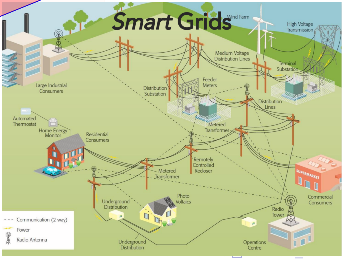
Adrien Bousseau
REVES / Inria Sophia Antipolis

Theophanis Tsandilas
Inria - Univ Paris-Sud & CNRS (LRI)

(a) Interaction setup (b) Model and extracted guides (c) User construction lines and drawing

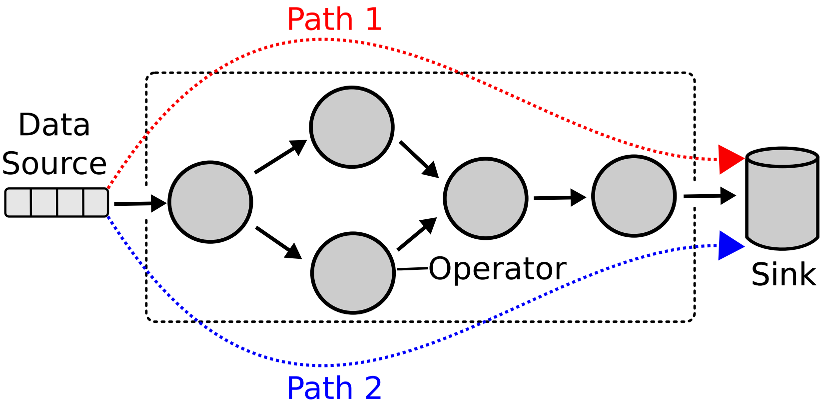
Figure 1. Our drawing assistant provides guidance and feedback over a model photograph that the user reproduces on a virtual canvas (a). We use computer vision algorithms to extract visual guides that enhance the geometric structures in the image (b). In this example, the user first sketched the block-in construction lines (c, blue) before drawing the regions and adding details. This guidance helps users produce more accurate drawings.

Smart Grids

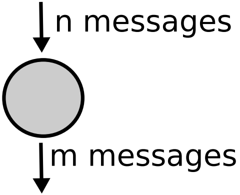


What are latency-aware applications **composed of**?

Applications and their characteristics

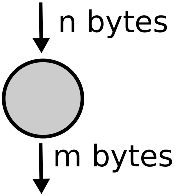


Applications and their characteristics



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

Where are the **latency-aware** applications deployed?

Classic Data Center

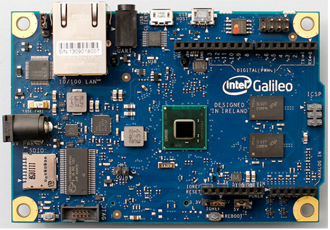
Generating opportunities to deploy applications **closer** where **data are generated**

Micro Data Centers

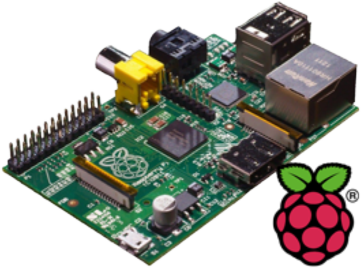
Solutions are exploring computational resources **even closer** where **data are generated**

Edge Devices and Sensors

Galileo



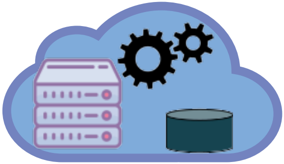
Raspberry Pi 2



How are the applications deployed across **Cloud and Edge Computing**?

CLOUD

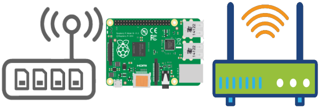
- Data storage
- Batch and stream processing
- Data warehousing
- Business applications



NETWORK

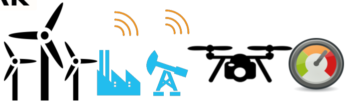
EDGE

- Real-time data processing
- Basic analytics
- Data filtering, optimization
- Data caching, buffering



NETWORK

SENSORS AND CONTROLLERS



Is there any **method** for **deploying** the applications **dynamically** onto edge and cloud?

Problem Statement / Methodology

Minimize metrics such as **end-to-end application latency and energy consumption** by placing operators onto **cloud** and **edge resources**

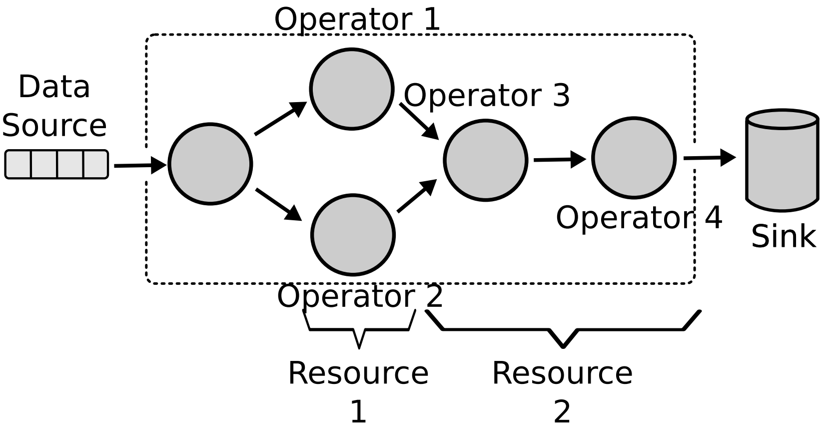
Physical infrastructure capabilities

- CPU and memory
- Network latencies and bandwidth

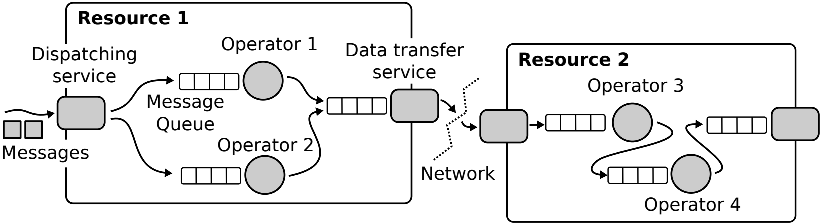
Application requirements

- Selectivity
- Data compression rate
- CPU and Memory
- Data sources and sinks localization

Our proposal model



Our proposal model



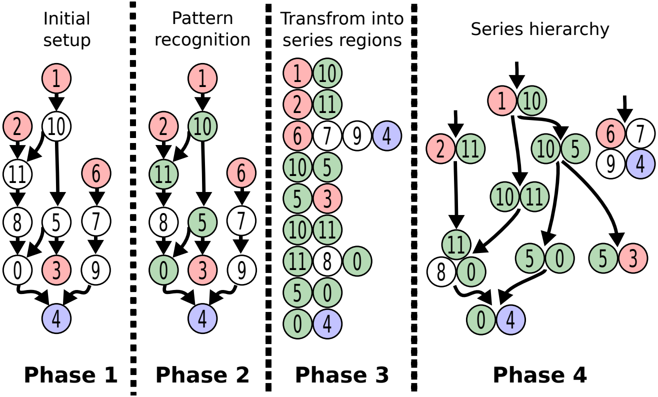
Model is based on **Queueing Theory** - **M/M/1**

Two queues: **Computation** and **Communication**

Response time is equal to the sum of computation and communication into a path.

We aim to **minimize the sum of the response times** (all paths)

Finding Application Patterns



Key behaviors: forks, joins, data sources and sinks

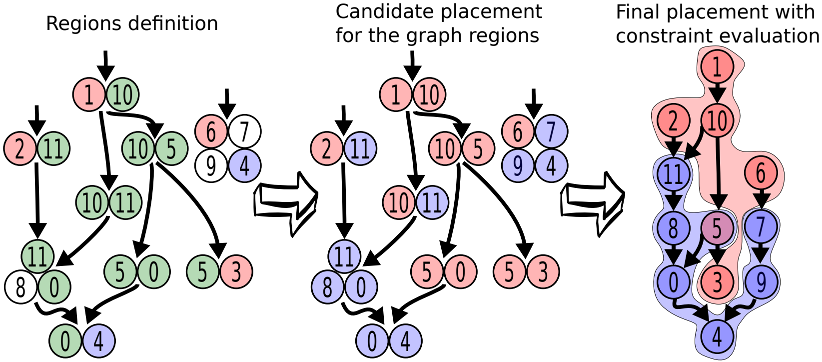
Decompose the application graph following the parallel regions

Response Time Rate (RTR) Strategy

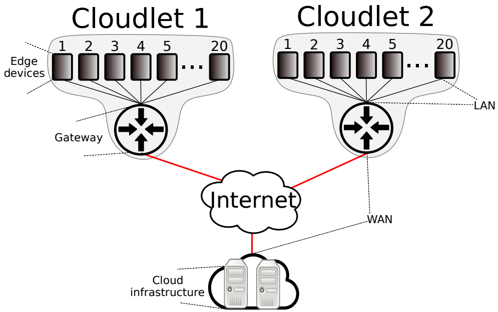
- Response Time Rate for computational resource based on the end-to-end application latency
- Sequentially estimate the operator response time following the upstream(s) and downstream(s) connections
- Evaluate memory, CPU, and bandwidth constraints

Response Time Rate with Region Patterns (RTR+RP) Strategy

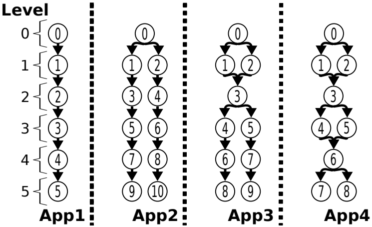
- Split the application graph following the pathways
- Calculate the Response Time Rate only to the edge side



Experimental Setup



Experimental Setup - Microbenchmarks



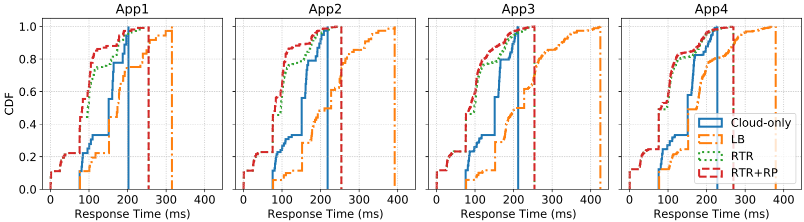
Messages sizes: text - 10 bytes, pictures/objects - 50KB, and voice records - 200KB

Input event rates: Each message size has three input event rates

CPU requirements: 10 bytes - 3.7952 IPS, 50 KB-18976 IPS, and 200 KB - 75904 IPS

Selectivity and data compression rates: 100, 75, 50 and 25

Results on Response Time - Microbenchmarks

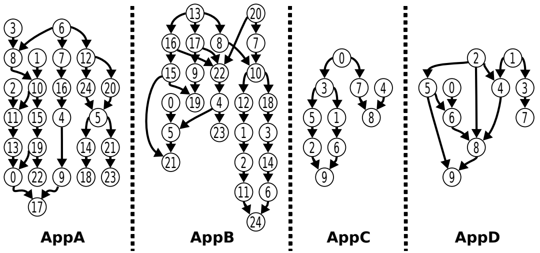


432 experiments (4 selectivities, 4 data compression rates, 3 input event rates, 3 sink locations and 3 input event sizes)

RTR and RTR+RP have shown to be over 95% more efficient than cloud-only approach and LB

Cloud-only achieved 5% better results (when the blue line crosses the red at approx. 200ms)

Experimental Setup - More Complex Applications

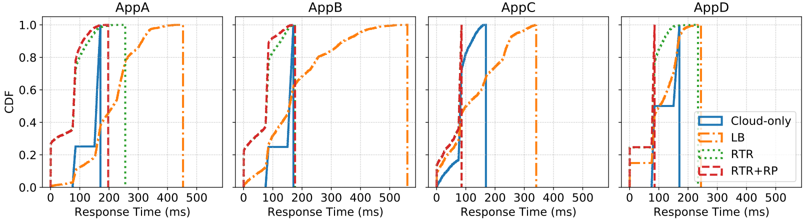


1160 graphs randomly applying multiple selectivities, data compression rates, sink and source locations, input event sizes and rates, memory, and CPU requirements

Large (AppA and AppB) containing 25 operators

Small (AppC and AppD) holding 10 operators.

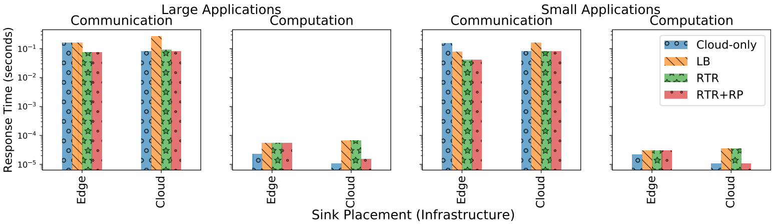
Results on Response Time - More Complex Applications



Our strategies outperformed cloud-only in over **6%** and **50%** under small and large applications, respectively.

Similarly, we improve the response times in over **23%** (small) and **57%** (large applications) compared to the LB approach.

Results on Response Time - More Complex Applications



The communication cost for sinks placed on cloudlets at cloud-only was about **160 ms**, and RTR+RP was **76 ms**.

Our solution outperformed cloud-only in up to **52%**, but sinks on the cloud, RTR+RP had a slight performance loss of **3%**

Conclusions and Future Work

Summary

- A model and the DSP placement problem formalization
- Two strategies to improve the response time
- A performance comparison using a simulated environment

Conclusions

- The key behaviors (forks and joins) of the dataflows directed us to our strategies
- Our strategies using the dataflow aspects allow us to be 50% better in response time

Future Work

- An evaluation using a real environment
- Determine the optimal value and compare with our solutions
- A model to deal with the reconfiguration phase

Questions?