Quality of Service Aware Mechanisms for (Re)Configuring Data Stream Processing Applications on Highly Distributed Infrastructure

Alexandre da Silva Veith

alexandre.veith@ens-lyon.fr

19th August 2019

About me

- I am Brazilian and I live in France.
- I work in the LIP at École Normale Supérieure de Lyon (ENS de Lyon) as a Ph.D. student.
- LIP is associated with CNRS and INRIA.
- My thesis supervisors are:

Laurent Lefèvre

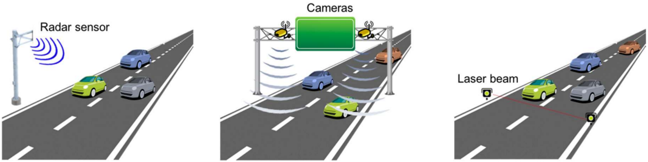
Marcos Dias de Assunção

- International collaborations:
  - University Carlos III of Madrid, Spain - Manuel F. Dolz
  - Rutgers University, US - Eduard Renart and Manish Parashar
  - Federal University of Rio Grande do Sul, Brazil - Julio Anjos and Claudio Geyer
  - IBM Brazil - Carlos Cardonha

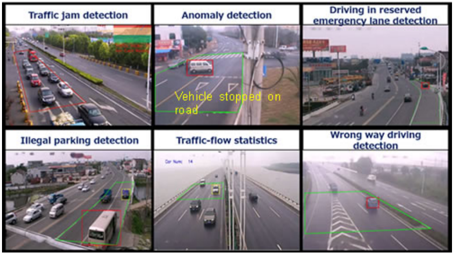
Motivation

- Society has become more interconnected producing **ever-increasing amounts of data** as a result of:
  - instrumented business process;
  - monitoring of user activities;
  - wearable assistance;
  - among other reasons.
- A large part of the data is most valuable when it is analysed, **as it is generated**.
- Under IoT and smart cities scenarios **continuous data streams** must be processed in short delays to provide insights or support the decision-making.
- For instance, sensors are suitable to transform transportation more intelligent.

Motivation



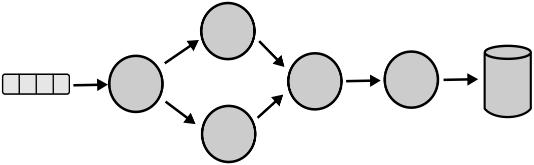
Source (Guerrero-Ibáñez, Zeadally and Contreras-Castillo, 2018)



Source (Guerrero-Ibáñez, Zeadally and Contreras-Castillo, 2018)

Data Stream Processing (DSP) Application

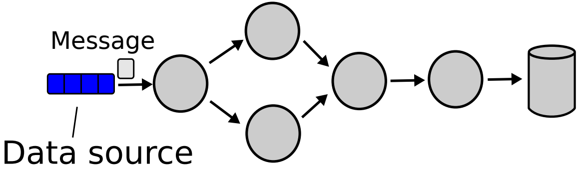
Data Stream Processing supplies the environment for processing data in a timely manner and under most frameworks the application is structured as a **dataflow**.



Data Stream Processing (DSP) Application

Data Stream Processing supplies the environment for processing data in a timely manner and under most frameworks the application is structured as a **dataflow**.

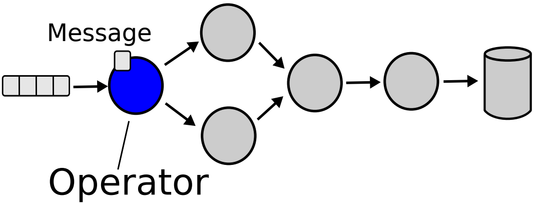
- Data sources



Data Stream Processing (DSP) Application

Data Stream Processing supplies the environment for processing data in a timely manner and under most frameworks the application is structured as a **dataflow**.

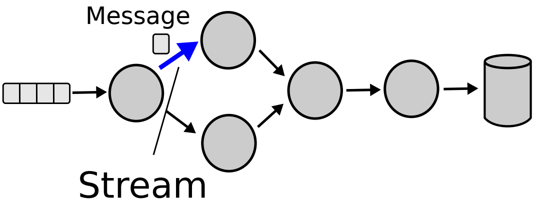
- Data sources
- Operators (stateless and stateful, selectivity, data compression/expansion)



Data Stream Processing (DSP) Application

Data Stream Processing supplies the environment for processing data in a timely manner and under most frameworks the application is structured as a **dataflow**.

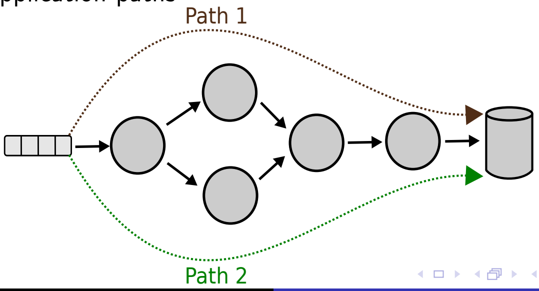
- Data sources
- Operators
- Streams



Data Stream Processing (DSP) Application

Data Stream Processing supplies the environment for processing data in a timely manner and under most frameworks the application is structured as a **dataflow**.

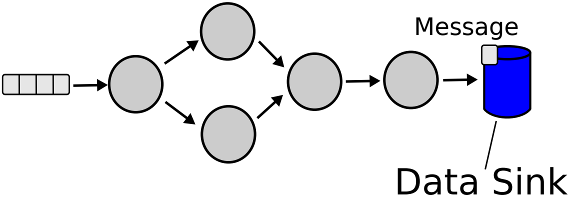
- Data sources
- Operators
- Streams
  - Application paths



Data Stream Processing (DSP) Application

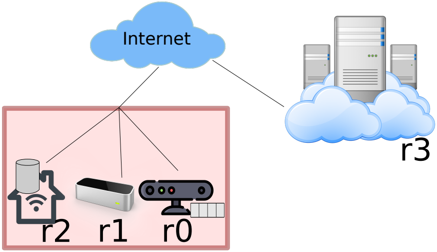
Data Stream Processing supplies the environment for processing data in a timely manner and under most frameworks the application is structured as a **dataflow**.

- Data sources
- Operators
- Streams
- Data sinks



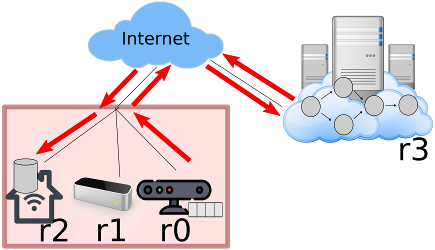
Life-cycle of DSP Applications

- **Initial operator placement (application configuration)**



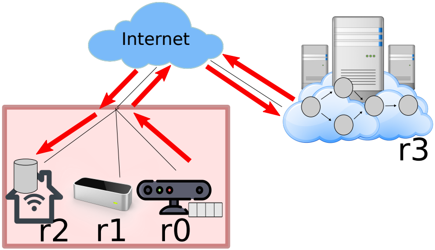
Life-cycle of DSP Applications

- **Initial operator placement (application configuration)**
  - The whole application dataflow is placed on a single cloud service provider (traditional).
    - DSP frameworks were conceived to run on clusters of **homogeneous computing resources** or single cloud service provider with **virtually unlimited computing resources**.



Life-cycle of DSP Applications

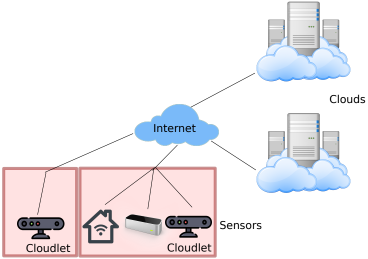
- **Initial operator placement (application configuration)**
  - The whole application dataflow is placed on a single cloud service provider (traditional).



High communication overhead (Hu et al., 2016) - hard to achieve (near) real-time data analytics.

Life-cycle of DSP Applications

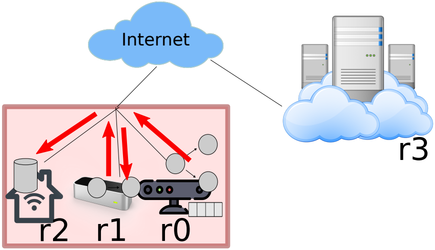
Edge computing infrastructure is the combination of constrained devices and cloud service providers (IoT scenario).



- Sensors continuously produce data, forward to gateways and switches on network cloudlet by **low latency links**.
- Cloudlets comprises devices, with low, but **non-negligible memory and CPU capabilities** grouped according to their location or network latency.

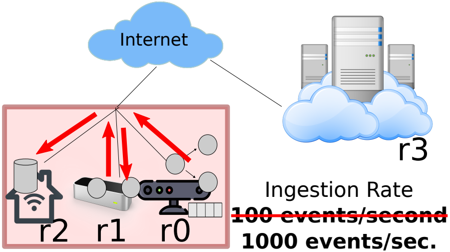
Life-cycle of DSP Applications

- **Initial operator placement (application configuration)**
    - The whole application dataflow is placed on a single cloud service provider (traditional)
    - Application placement on edge devices
        - Devices have limited CPU, memory and bandwidth capabilities



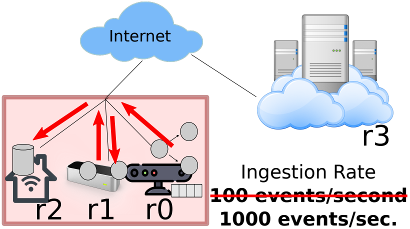
Life-cycle of DSP Applications

- Initial operator placement (application configuration)
- **Application reconfiguration**
  - Workload changes, infrastructure changes, ...



Life-cycle of DSP Applications

- Initial operator placement (application configuration)
- **Application reconfiguration**
  - Workload changes, device failures, ...
  - Two main methods to reconfigure:
    - Parallel track: an operator replica is created and then the **replicas run concurrently** until to synchronise their states.
    - Pause-and-resume: the upstream operator is **paused**, the migration happens, and then the upstream is resumed.



How to handle the life-cycle of DSP applications on edge computing?

Initial Operator Placement

Problem

How to dynamically establish the operator placement on edge computing devices by respecting their limitations and meeting the performance requirements?

Challenges

- Memory, CPU and network bandwidth limitations on edge devices
- Heterogeneity of computing resources (CPU and memory capabilities) and operators (patterns and computing requirements)
- QoS requirements (end-to-end latency, monetary cost, ...)

Contributions

Alexandre da Silva Veith, Marcos Dias de Assunção and Laurent Lefèvre. "Latency-Aware Placement of Data Stream Analytics on Edge Computing". In: *International Conference on Service-Oriented Computing (ICSOC).* Ed. by Claus Pahl, Maja Vukovic, Jianwei Yin and Qi Yu. Cham: Springer International Publishing, 2018, pp. 215–229. ISBN: 978-3-030-03596-9.

Eduard Gibert Renart, Alexandre Da Silva Veith, Daniel Balouek-Thomert, Marcos Dias de Assuncao, Laurent Lefevre and Manish Parashar. "Distributed Operator Placement for IoT Data Analytics Across Edge and Cloud Resources". In: *19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID 2019).* Larnaca, Cyprus, May 2019, pp. 459–468.

Application Reconfiguration

Problem

How to dynamically reorganise or migrate operators on edge computing devices by respecting their limitations and meeting the performance requirements?

Challenges

- Same challenges than the application configuration.
- Reducing the application downtime for migrating operators (pause-and-resume).
- Managing replicas synchronisation (parallel-track).

Contributions

Alexandre da Silva Veith, Felipe Rodrigo de Souza, Marcos Dias de Assunção, Laurent Lefevre and Julio C.S. dos Anjos. "Multi-Objective Reinforcement Learning for Reconfiguring Data Stream Analytics on Edge Computing". In: *48th International Conference on Parallel Processing (ICPP 2019)*. Kyoto, Japan, Aug. 2019.

Alexandre da Silva Veith, Marcos Dias de Assunção and Laurent Lefevre. "Monte-Carlo Tree Search and Reinforcement Learning for Reconfiguring Data Stream Processing on Edge Computing". In: *The International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*. Campo Grande, Brazil, Oct. 2019.

Application Reconfiguration Motivations

Reconfiguration challenges

- Data sources and sinks can be **located** both on the **cloud** and **cloudlets** (IoT scenario).
- **Multiple patterns to operators** such as selectivity, data compression/expansion, stateful and stateless.
- **Edge devices limitations** when defining the application reconfiguration.
- Application reconfiguration follows a **pause-and-resume** method because of the edge devices computing limitations.
- **Large search spaces** due to applications with large number of operators and infrastructures with large number of computing resources.

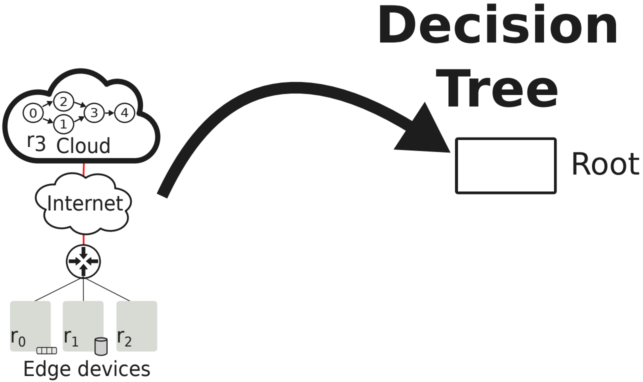
Application Reconfiguration Motivations

- The combination between **Monte-Carlo Tree Search algorithm (MCTS) and Reinforcement Learning (RL)** has demonstrated to be **efficient with large search spaces problems** in board games such as Alpha Go, Tic-Tac-Toe, among others.
- RL enables a scheduling policy agent to **learn** how to make decisions by **interacting with an environment**.
- Commonly the RL environment is **modelled as a Markov Decision Process** (MDP).

Our approach

Our proposed solution comprises an **MDP framework for the application reconfiguration** and an MCTS implementation.

MDP Framework and MCTS Implementation



- The **root** node contains:
  - The state of the current application deployment.
  - Application and infrastructure statistics.

Monitored Statistics

- **Response time**
  end-to-end latency from the time events are generated to the time they reach the sinks.

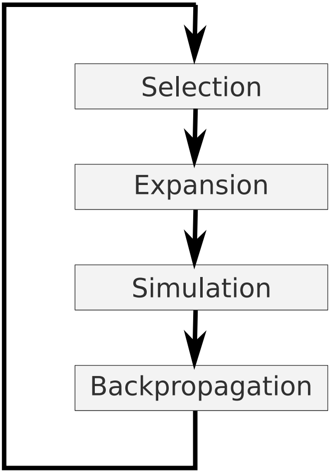
Monitored Statistics

- **Response time**
  end-to-end latency from the time events are generated to the time they reach the sinks.
- **WAN traffic**
  data volume crossing WAN network links.

Monitored Statistics

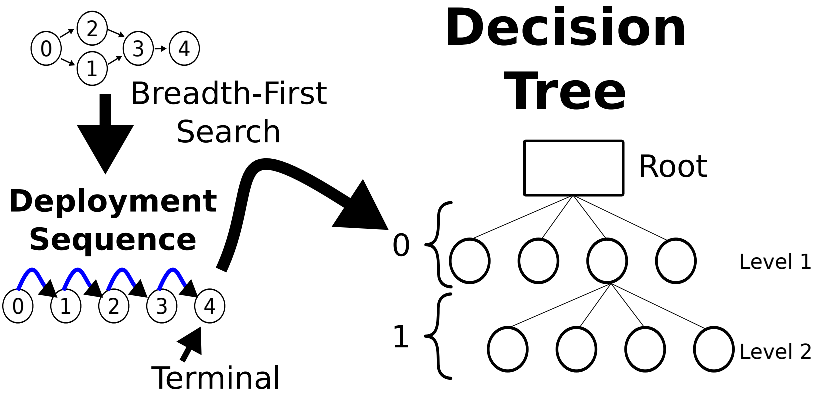
- **Response time**
  end-to-end latency from the time events are generated to the time they reach the sinks.
- **WAN traffic**
  data volume crossing WAN network links.
- **Monetary cost**
  calculates the monetary cost using the number of connections and exchanged messages across the cloud and the cloudlets, and vice-versa (*Azure IoT Hub* 2019; *AWS IoT Core* 2019).

MDP Framework and MCTS Implementation



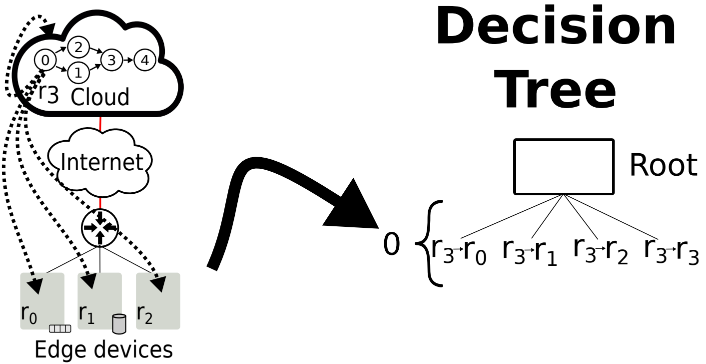
- A **budget** within a **number of iterations** is used to execute the MCTS loop.
- The algorithm **builds a decision-tree** with possible reconfiguration deployments.

MDP Framework and MCTS Implementation



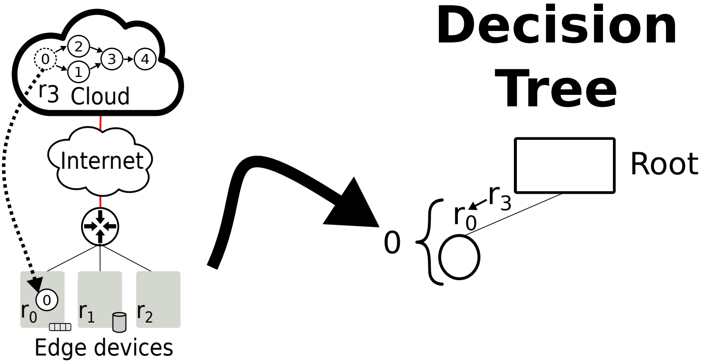
- The **deployment sequence** is used to create the decision-tree levels.

MDP Framework and MCTS Implementation



- **Decision-tree edges** are the available actions to move or not an operator from one resource to another.

MDP Framework and MCTS Implementation



- **Decision-tree node** comprises the state of the reconfiguration deployment, and variables for supporting future decisions (*e.g.*, **reward**).

Quality of Service Metrics

For estimating the reward, the following four QoS metrics are **simulated** by using a proposed **Queueing Theory model**:

- **Response time**
- **WAN traffic**
- **Monetary cost**

Reconfiguration Overhead

The total downtime incurred by migrating operator code and operator states.

Single aggregate cost based on **Simple Additive Weighting method** (Yoon et al., 1995) covers the four QoS metrics:

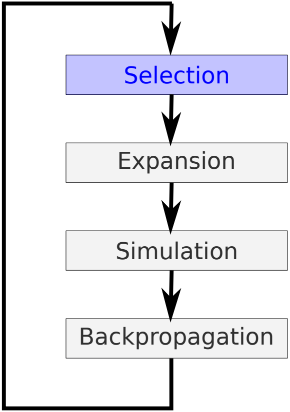
$w_0 \times$ **Response time** $+ w_1 \times$ **WAN traffic**$+$
$w_2 \times$ **Monetary cost** $+ w_3 \times$ **Reconfiguration Overhead**

where w corresponds to weight assigned to the metric.

Reward

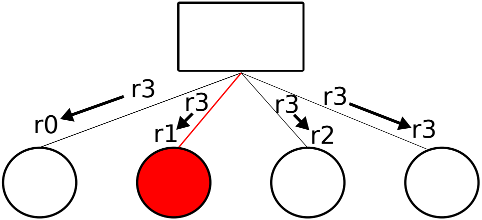
The Reward is the difference between the **single aggregate cost** of the current application deployment and the **single aggregate cost** of the reconfiguration deployment.

MDP Framework and MCTS Implementation

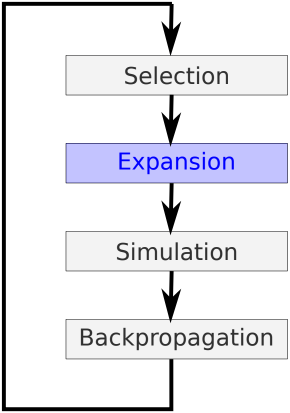



```
while node is not terminal do
    if node is fully expanded then
        Get the most promising node;
    else
        return node;
    end
end
```

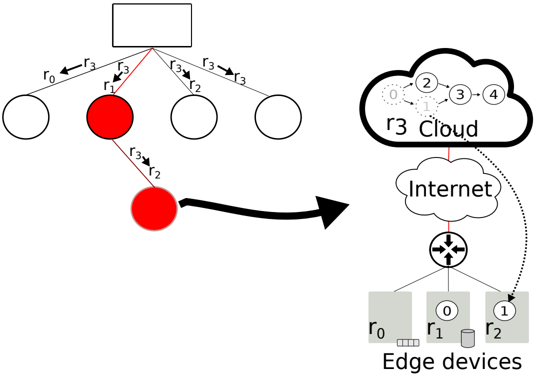
Upper Confidence Bound for Trees (UCT) - UCB1 algorithm (Sutton and Barto, 2018) provides the most promising node.



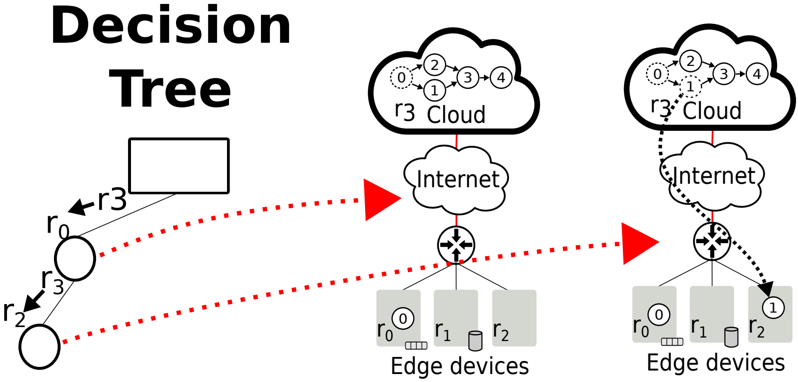
MDP Framework and MCTS Implementation



1 choose an untried action;
2 add a new child to the selected node;
3 initiate the decision variables;
4 return new node;

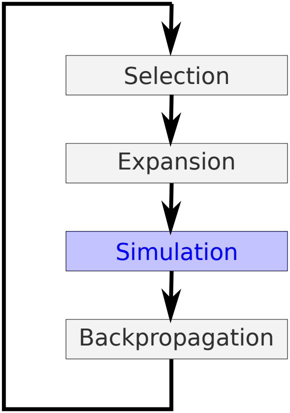


MDP Framework and MCTS Implementation

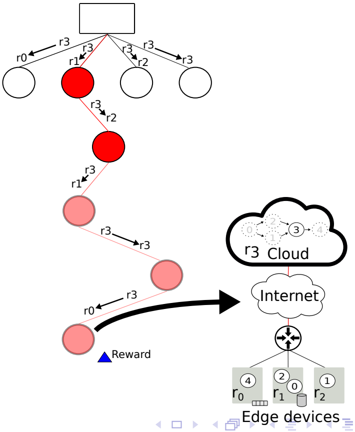


- The node deployment is always derived from the parent node.

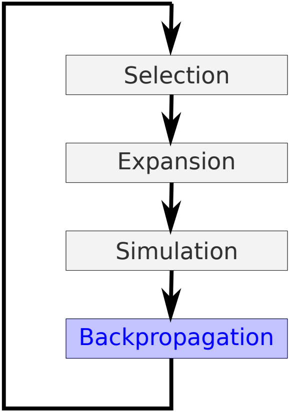
MDP Framework and MCTS Implementation



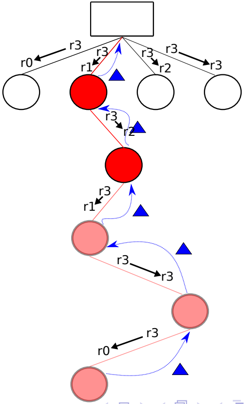
while *node is not terminal* **do**
  Choose an action randomly;
end
Simulate the new placement and determine its reward;



MDP Framework and MCTS Implementation

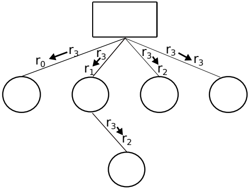



```
node ← current node;
while node is not the root do
    update node decision variables;
    node ← parent of node;
end
```

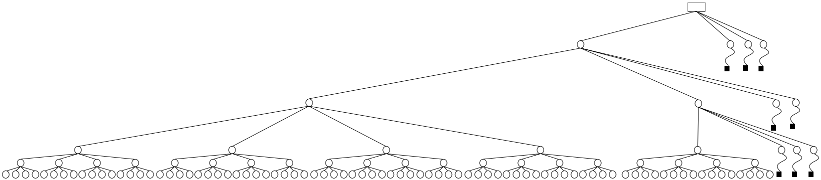


MDP Framework and MCTS Implementation

Decision-tree after the explained iteration.

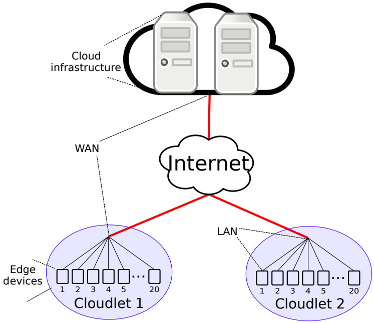


Decision-tree after consuming the iteration budget.



- Simulation tool developed atop **OMNET++**.
- One iteration of a Monitoring, Analysis, Planning and Execution (**MAPE**) loop is considered for operator reconfiguration:
  - Analysed when the execution reaches **300** seconds or all application paths have processed **500** messages; whichever comes last.
  - RL algorithms with a computational budget of **10,000** iterations.

Experimental Setup



- Edge: Two sites with 20 **Raspberry PI 2** (4,74 MIPS at 1GHz and 1GB of RAM);
- Cloud: Two **AMD RYZEN 7 1800x** (304,51 MIPS at 3.6GHz and 1TB of RAM);
- LAN (Hu et al., 2016): Latency **U(0.015-0.8)ms** and bandwidth equal to 100 Mbps;
- WAN (Hu et al., 2016): Latency **U(65-85)ms** and bandwidth equal to 1 Gbps.

Implementations of Our Proposed MDP Framework

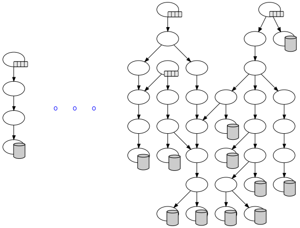
- **MCTS-UCT** (Sutton and Barto, 2018) basic version of the Monte-Carlo Tree Search with UCT.
- **TDTS-Sarsa(λ)** (Vodopivec, Samothrakis and Ster, 2017) creates intermediary rewards for each operator movement and employs them when estimating the reward.

Performance Comparison

- **Traditional**, which deploys the whole application dataflow on the cloud.
- **LB** (Taneja and Davy, 2017), which evaluates edge devices capabilities and operator requirements to offload operators from cloud to edge devices for reducing the response time.

Evaluated Applications

Example of application graphs

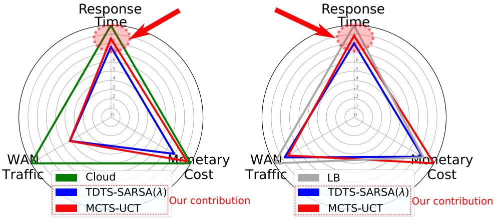


Parameter	Value	Unit
cpu	1000-10000	Instructions per second
Data compression rate	0-90	%
mem	100-7500	bytes
Input event size	100-2500	bits/second
Selectivity	0-90	%
Input event rate	1000-10000	Number of messages
ws (window size)	1-100	Number of messages

- **Eleven application graphs** with single and multiple data paths are considered.
- Application graph obtained by using a Python library (*Generic grahs* 2019).
- **20%** of the whole application are **stateful** operators.
- Data sources and sinks are placed on the cloudlets, except for the sink of the longest dataflow path (cloud).

Evaluation of Response Time

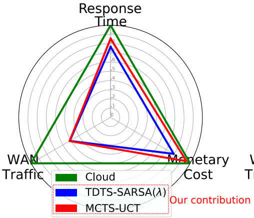
Scenario 1: Response time with weight equal to 1.

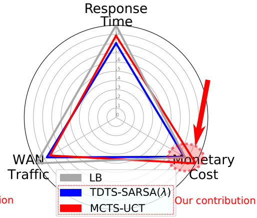


RL algorithms achieved over **20% better response time**, and reduce the WAN traffic by over **50%** and the monetary cost by **15%** when comparing to Cloud approach.

Evaluation of Response Time

Scenario 1: Response time with weight equal to 1.

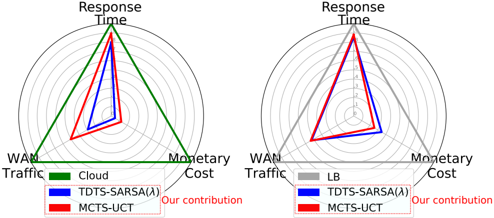




Single-criterion optimisation fails in bringing **monetary cost and WAN traffic guarantees**. Monetary cost increased by over **15%**.

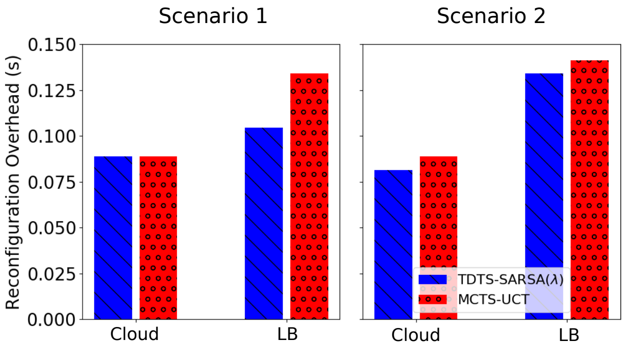
Evaluation of Monetary Cost and WAN Traffic Guarantees

Scenario 2: Cloud and LB with weights for response time, monetary cost and WAN traffic equal to 0.33.



RL algorithms **offloaded operators from cloud to closer data sources and sinks** placed on cloudlets reducing the WAN traffic and monetary cost.

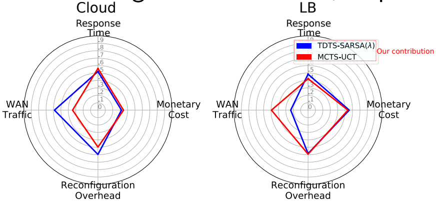
Overhead of the Reconfiguration Decisions



RL algorithms achieved lower reconfiguration overhead in over **40%** when starting from the Cloud solution.

Evaluation Including Reconfiguration Overhead

Scenario 3: 0.4, 0.2, 0.2, and 0.2 weights for response time, monetary cost, WAN traffic and reconfiguration overhead, respectively.



- The reconfiguration overhead weight resulted in migrating **operators without states or small states**.
- The attributed WAN traffic and monetary cost weights **offloaded operators from cloud to cloudlets**.
- The **combination of the metric weight raised the response time** because big part of the operator migrations moved the operators closer to data sources and sinks.

Conclusions

- By considering only response time as a QoS metric, it is not possible to guarantee WAN traffic, monetary cost, and reconfiguration overhead.
- Multiple QoS metric evaluation provided a holistic view of the environment (infrastructure + application).
- The suggested WAN traffic, monetary cost, and reconfiguration overhead metrics increased response time reduction.
- Our proposed approach demonstrated to be efficient for establishing reconfiguration deployments.

Future Work

Possible collaborations:

- Investigation of machine learning mechanisms.
- Implementation of reconfiguration solutions in a real-life framework.
- Investigation of operator replication.
- Models for device failures, energy consumption, ...

Questions?

List of Publications

Alexandre da Silva Veith, Marcos Dias de Assunção and Laurent Lefèvre. "Latency-Aware Placement of Data Stream Analytics on Edge Computing". In: *International Conference on Service-Oriented Computing (ICSOC)*. Ed. by Claus Pahl, Maja Vukovic, Jianwei Yin and Qi Yu. Cham: Springer International Publishing, 2018, pp. 215–229. ISBN: 978-3-030-03596-9.

Eduard Gibert Renart, Alexandre Da Silva Veith, Daniel Balouek-Thomert, Marcos Dias de Assuncao, Laurent Lefevre and Manish Parashar. "Distributed Operator Placement for IoT Data Analytics Across Edge and Cloud Resources". In: *19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID 2019)*. Larnaca, Cyprus, May 2019, pp. 459–468.

Alexandre da Silva Veith, Felipe Rodrigo de Souza, Marcos Dias de Assunção, Laurent Lefevre and Julio C.S. dos Anjos. "Multi-Objective Reinforcement Learning for Reconfiguring Data Stream Analytics on Edge Computing". In: *48th International Conference on Parallel Processing (ICPP 2019)*. Kyoto, Japan, Aug. 2019.

Alexandre da Silva Veith, Marcos Dias de Assunção and Laurent Lefevre. "Monte-Carlo Tree Search and Reinforcement Learning for Reconfiguring Data Stream Processing on Edge Computing". In: *The International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*. Campo Grande, Brazil, Oct. 2019.

Paulo de Souza, Kassiano Matteussi, Julio dos Anjos, Jorge dos Santos, Claudio Geyer and Alexandre da Silva Veith. "Aten: A Dispatcher for Big Data Applications in Heterogeneous Systems". In: *2018 International Conference on High Performance Computing Simulation (HPCS)*. July 2018, pp. 585–592.

Julio C.S. dos Anjos, Kassiano Matteussi, Paulo De Souza, Alexandre da Silva Veith, Gilles Fedak, Jorge Barbosa and Carlos Geyer. "Enabling Strategies for Big Data Analytics in Hybrid Infrastructures". In: *2018 International Conference on High Performance Computing Simulation (HPCS)*. July 2018, pp. 869–876.

Alexandre da Silva Veith and Marcos Dias de Assuncao. "Apache Spark". In: *Encyclopedia of Big Data Technologies*. Ed. by Sherif Sakr and Albert Y. Zomaya. Cham: Springer International Publishing, 2019, pp. 77–81. ISBN: 978-3-319-77525-8.

Guerrero-Ibáñez, Juan, Sherali Zeadally and Juan Contreras-Castillo. "Sensor Technologies for Intelligent Transportation Systems". In: *Sensors* 18.4 (2018). ISSN: 1424-8220. DOI: 10.3390/s18041212. URL: https://www.mdpi.com/1424-8220/18/4/1212.

Hu, Wenlu, Ying Gao, Kiryong Ha, Junjue Wang, Brandon Amos, Zhuo Chen, Padmanabhan Pillai and Mahadev Satyanarayanan. "Quantifying the Impact of Edge Computing on Mobile Applications". In: *7th ACM SIGOPS Asia-Pacific Wksp on Systems*. APSys '16. Hong Kong, Hong Kong: ACM, 2016, 5:1–5:8. ISBN: 978-1-4503-4265-0.

Alexandre da Silva Veith, Marcos Dias de Assunção and Laurent Lefèvre. "Latency-Aware Placement of Data Stream Analytics on Edge Computing". In: *International Conference on Service-Oriented Computing (ICSOC)*. Ed. by Claus Pahl, Maja Vukovic, Jianwei Yin and Qi Yu. Cham: Springer International Publishing, 2018, pp. 215–229. ISBN: 978-3-030-03596-9.

Eduard Gibert Renart, Alexandre Da Silva Veith, Daniel Balouek-Thomert, Marcos Dias de Assuncao, Laurent Lefevre and Manish Parashar. "Distributed Operator Placement for IoT Data Analytics Across Edge and Cloud Resources". In: *19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID 2019)*. Larnaca, Cyprus, May 2019, pp. 459–468.

Alexandre da Silva Veith, Felipe Rodrigo de Souza, Marcos Dias de Assunção, Laurent Lefevre and Julio C.S. dos Anjos. "Multi-Objective Reinforcement Learning for Reconfiguring Data Stream Analytics on Edge Computing". In: *48th International Conference on Parallel Processing (ICPP 2019)*. Kyoto, Japan, Aug. 2019.

Alexandre da Silva Veith, Marcos Dias de Assunção and Laurent Lefevre. "Monte-Carlo Tree Search and Reinforcement Learning for Reconfiguring Data Stream Processing on Edge Computing". In: *The International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*. Campo Grande, Brazil, Oct. 2019.

Azure IoT Hub. 2019. URL: https://azure.microsoft.com/en-us/services/iot-hub/.

AWS IoT Core. 2019. URL: https://aws.amazon.com/iot-core/pricing/.

Yoon, K.P., P.K. Yoon, C.L. Hwang, SAGE. and inc Sage Publications. *Multiple Attribute Decision Making: An Introduction*. Multiple Attribute Decision Making: An Introduction. SAGE Publications, 1995.

Sutton, Richard S. and Andrew G. Barto. *Reinforcement Learning: An introduction*. MIT press, 2018.

Vodopivec, Tom, Spyridon Samothrakis and Branko Ster. "On Monte Carlo Tree Search and Reinforcement Learning". In: *Journal of Artificial Intelligence Research* 60 (2017), pp. 881–936.

Taneja, M. and A. Davy. "Resource aware placement of IoT application modules in Fog-Cloud Computing Paradigm". In: *IFIP/IEEE Symp. on Integrated Net. and Service Management (IM)*. May 2017, pp. 1222–1228.

Generic grahs. 2019. URL: https://gist.github.com/bwbaugh/4602818.

Paulo de Souza, Kassiano Matteussi, Julio dos Anjos, Jorge dos Santos, Claudio Geyer and Alexandre da Silva Veith. "Aten: A Dispatcher for Big Data Applications in Heterogeneous Systems". In: *2018 International Conference on High Performance Computing Simulation (HPCS)*. July 2018, pp. 585–592.

Julio C.S. dos Anjos, Kassiano Matteussi, Paulo De Souza, Alexandre da Silva Veith, Gilles Fedak, Jorge Barbosa and Carlos Geyer. "Enabling Strategies for Big Data Analytics in Hybrid Infrastructures". In: *2018 International Conference on High Performance Computing Simulation (HPCS)*. July 2018, pp. 869–876.

Alexandre da Silva Veith and Marcos Dias de Assuncao. "Apache Spark". In: *Encyclopedia of Big Data Technologies*. Ed. by Sherif Sakr and Albert Y. Zomaya. Cham: Springer International Publishing, 2019, pp. 77–81. ISBN: 978-3-319-77525-8.