Self-Adaptive Data Stream Processing in Geo-Distributed Computing Environments

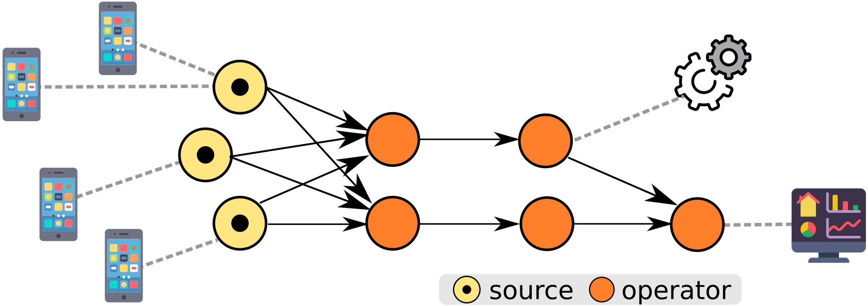
Gabriele Russo Russo,
Valeria Cardellini, Francesco Lo Presti
University of Rome Tor Vergata

Introduction: new trends for Big Data

New pervasive services enabled by real-time Big Data analytics
(e.g., Smart City)

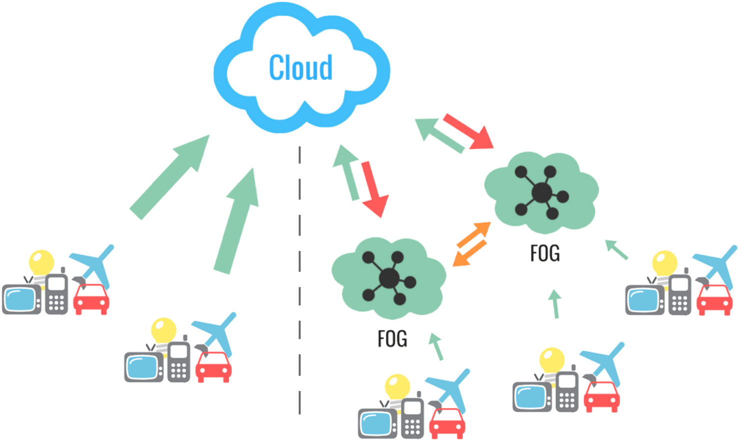
Data Stream Processing (DSP)

- Continuous processing of unbounded sequences: **data streams**
- Data processed "on the fly"
- Applications represented as DAGs (operators + streams)

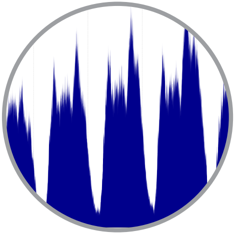


DSP from Cloud to Fog

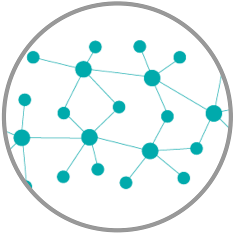
- Latency requirements to support real-time services
- Idea: moving computation towards data sources and consumers



DSP in the fog: old and new challenges

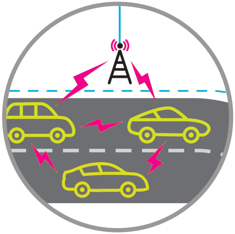


Adapting to variable conditions



Decentralized control

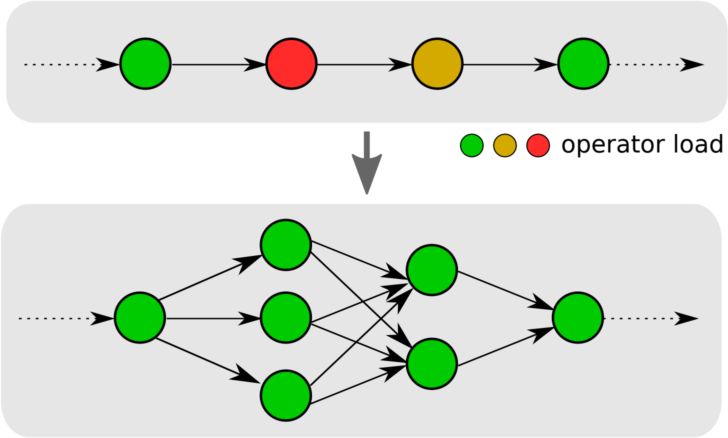
Model uncertainty



Mobility

Self-adaptive DSP: Elasticity

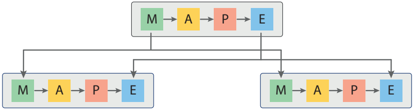
- Parallel replicas of operators to face higher data rates
- Elastic parallelism allows to avoid over- and under-provisioning
- Goal: decentralized elasticity, accounting for model uncertainty

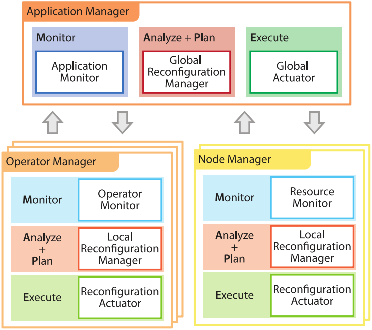


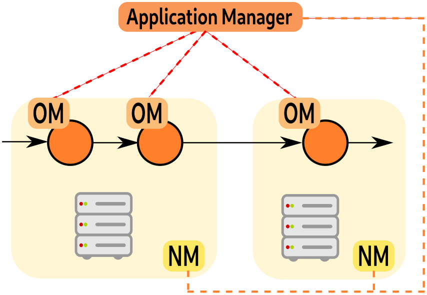
EDF: a framework for Decentralized Elasticity

Based on Hierarchical MAPE:

- Centralized Application Manager
- Decentralized Operator Managers and Node Managers

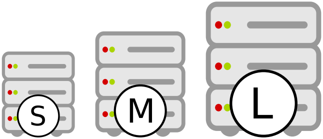






Elasticity Policy for the Operator Manager

- Number of parallel replicas adapted to input data rate
- **Heterogeneous infrastructure**: several types of computing resources available to run the replicas



Operating costs for a single operator

- resources cost: depends on amount and type of used resources
- adaptation cost: performance degradation due to reconfiguration
- SLO violation: paid whenever response time (or throughput) violates a given threshold

$\rightarrow$ would like to minimize all of them in the long-term

$\rightarrow$ problem formulated as a Markov Decision Process

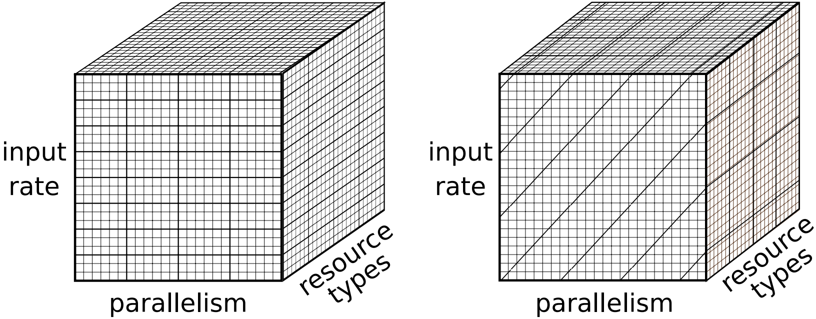
Function Approximation for MDPs

- Problem: standard MDP resolution techniques rely on "Q table" $\rightarrow$ do not scale
- Idea: replacing the Q table with a parametric function $\hat{Q}(s, a, \theta)$
- Need to store (and compute) only the parameters θ
- We focus on linear Function Approximation:
$$\hat{Q}(s, a, \boldsymbol{\theta}) = \sum_i \phi_i(s, a)\theta_i$$
- Weights $\boldsymbol{\theta}$: updated using Stochastic Gradient Descent
- Features ϕ: critical choice for good accuracy!

Defining features: Tile Coding

Tile Coding: cover the state space with "tilings"

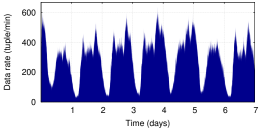
- "similar" states covered by a single tile (i.e., a single feature)
- different number and shape of tiles
- multiple overlapping tilings combined for increased accuracy

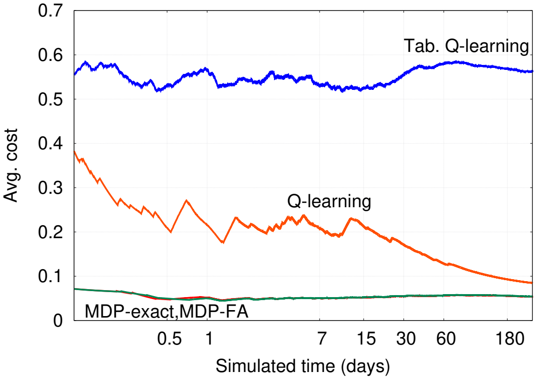


G. Russo Russo, V. Cardellini, F. Lo Presti, "Reinforcement learning based policies for elastic stream processing on heterogeneous resources", *Proc. ACM DEBS 2019*, Darmstadt, Germany, 24-28 June 2019.

Results

- We compare the average cost achieved by various resolution algorithms by simulation
- To deal with model uncertainty: reinforcement learning





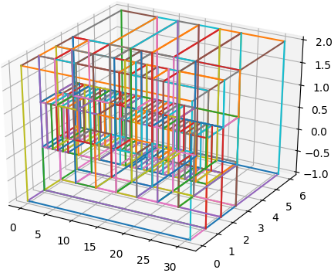
MDP resolution:
exact, FA

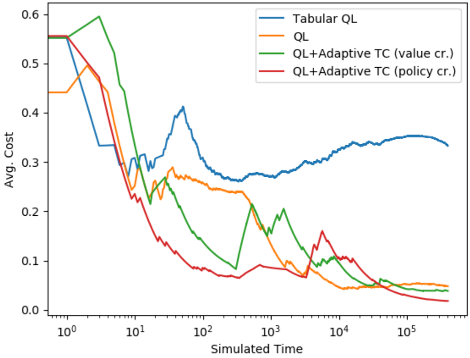
Reinforcement learning:
Tabular Q-learning
Q-learning with FA

Adaptive Tile Coding

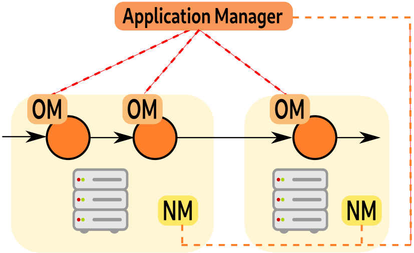
- Tile Coding still requires expertise to choose size/shape of tiles
- If the problem changes, may need new tilings

- Adaptive Tile Coding: identify best partitioning in an automated way
- Start with one large tile, then iteratively split to increase accuracy





Elasticity: the Application Manager

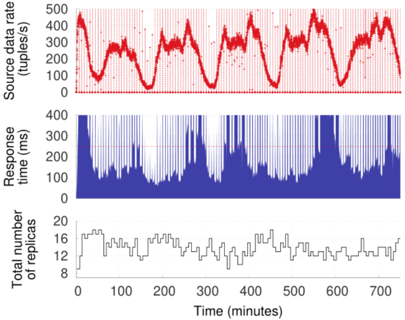


- Application Manager should coordinate local decisions of OMs
- First issue to tackle: adaptation overhead
- A heuristic based on a token bucket
  - OMs adaptation decisions must be accepted by AM
  - Each adaptation requires a token
  - Different tokens generated based on observed performance

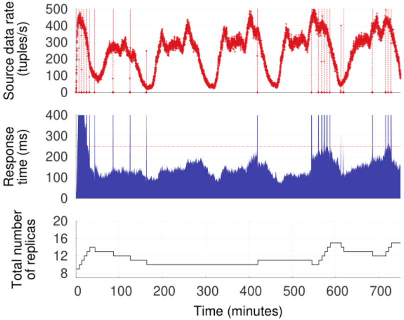
V. Cardellini, F. Lo Presti, M. Nardelli, G. Russo Russo,
"Decentralized self-adaptation for elastic data stream processing",
Future Generation Computing Systems, Vol. 87, pp. 171-185, October 2018.

Results

- EDF implemented on top of Apache Storm
- With token bucket, much less adaptations and negligible performance degradation



Q-learning

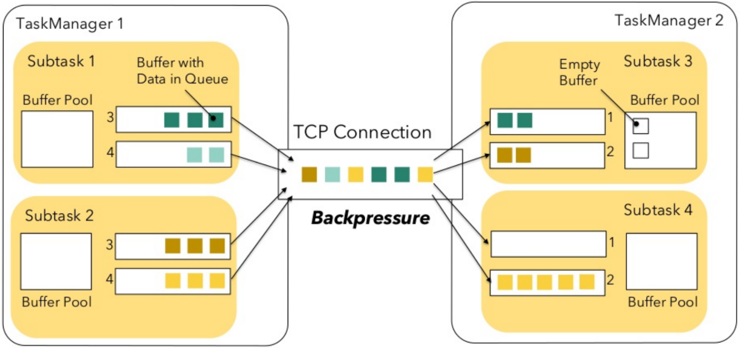


Q-learning, with token bucket

Open challenges

Controlling performance of modern DSP frameworks require models to account for additional factors, e.g.:

- Load distribution among stateful parallel replicas may not be balanced
- Operators are not independent: backpressure



Thanks for your attention!

russo.russo@ing.uniroma2.it
www.ce.uniroma2.it/~russorusso