Deploying data streaming applications in the Fog

Valeria Cardellini

cardellini@ing.uniroma2.it

University of Rome Tor Vergata

2nd Workshop "Through the Fog" – Pisa, Italy

Data stream processing (DSP)

- A variety of low-latency and location-aware applications in diverse domains:
  - Situation-aware applications (e.g., intelligent urban transport, surveillance, and traffic congestion)
  - Social data mining

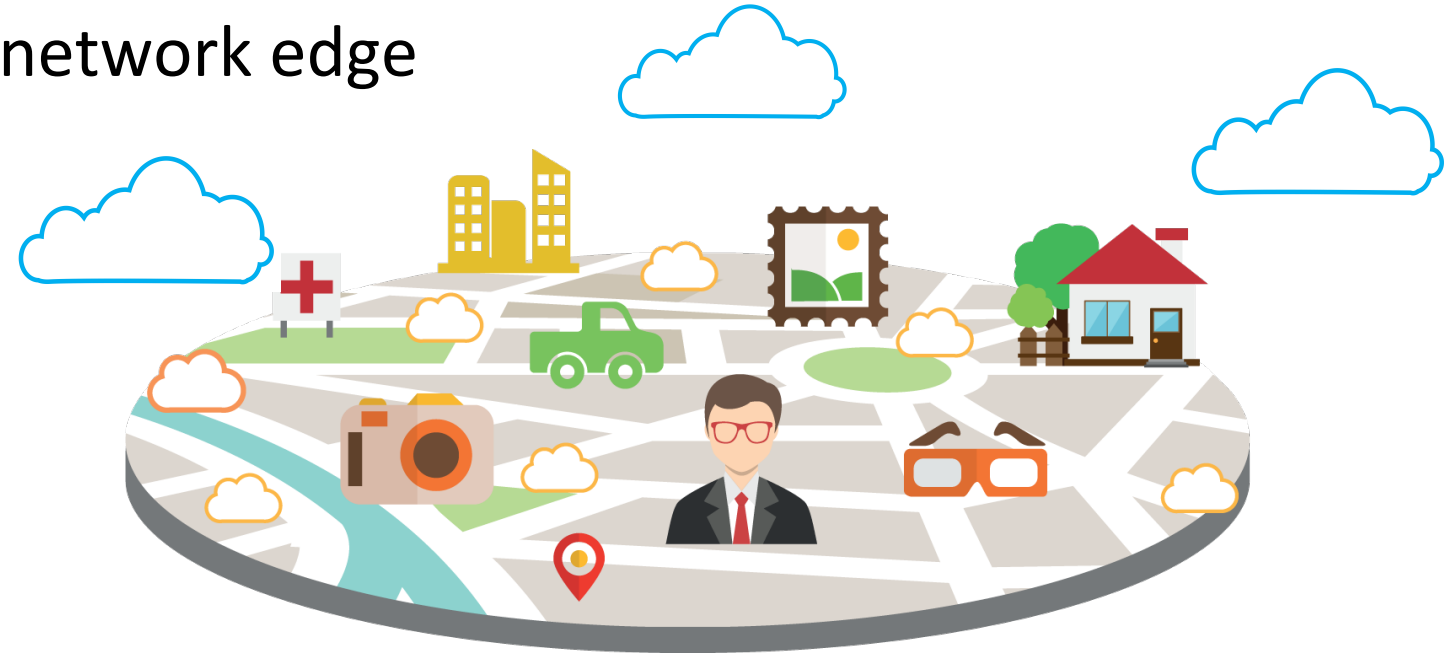
- Require
  - Continuous **real-time processing** of unbounded **data streams** generated by multiple, distributed sources
  - To extract valuable information in a timely and reliable manner

In a new distributed environment

- To increase scalability and availability, reduce latency, network traffic, and power consumption

  Exploit distributed and near-edge computation

  – **Edge/fog computing** ("the cloud close to the ground"): many micro data centers located at the network edge

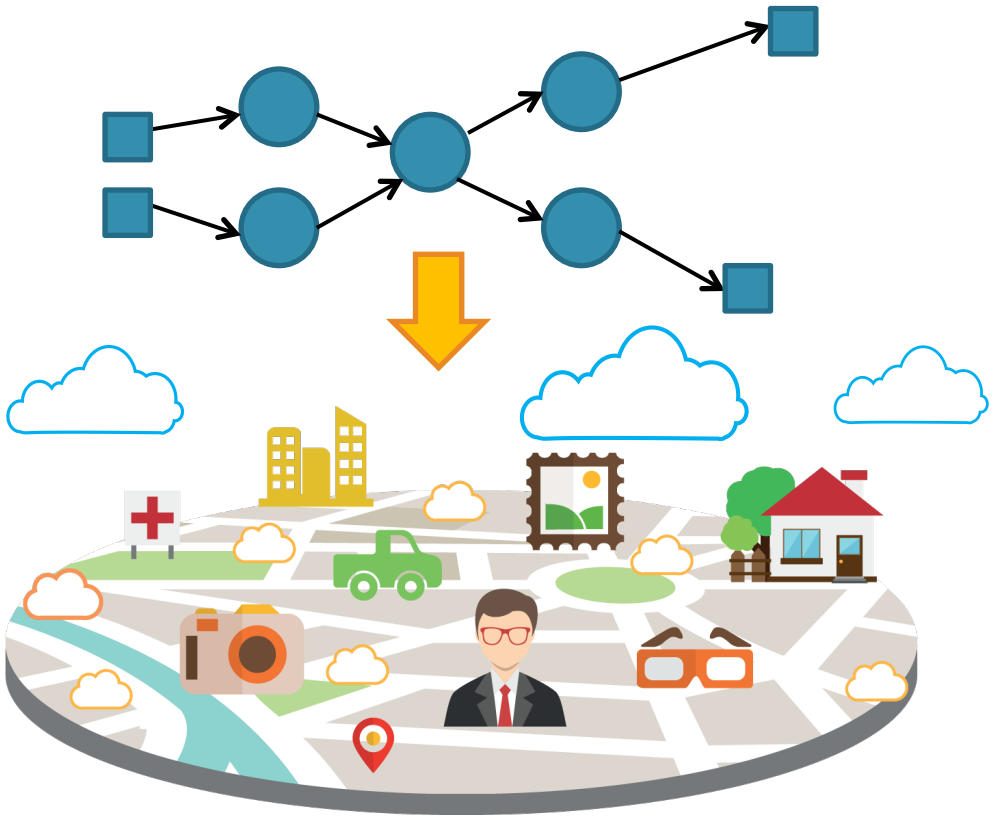


… that poses old and new challenges

- Network latencies are significant
- Computing and networking resources are heterogeneous (e.g., business constraints, capacity limits, …)
- Computing and network resources are not always available
- Data cannot be processed everywhere
- …

Goal of the talk

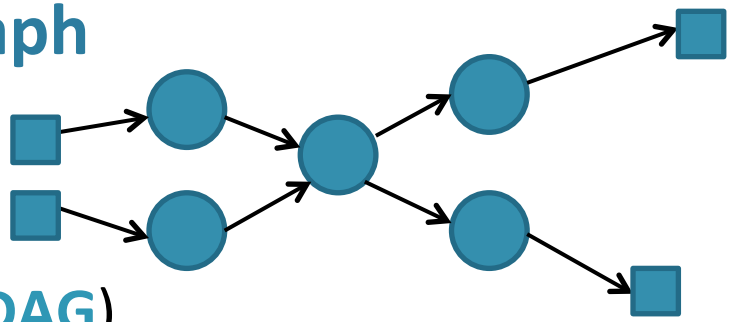
- Give a flavor of some challenges and their possible solutions that arise when deploying data stream processing applications in a fog/edge environment



DSP application basics

- A network of *operators* connected by *data streams*, at least one *data source* and one *data sink*

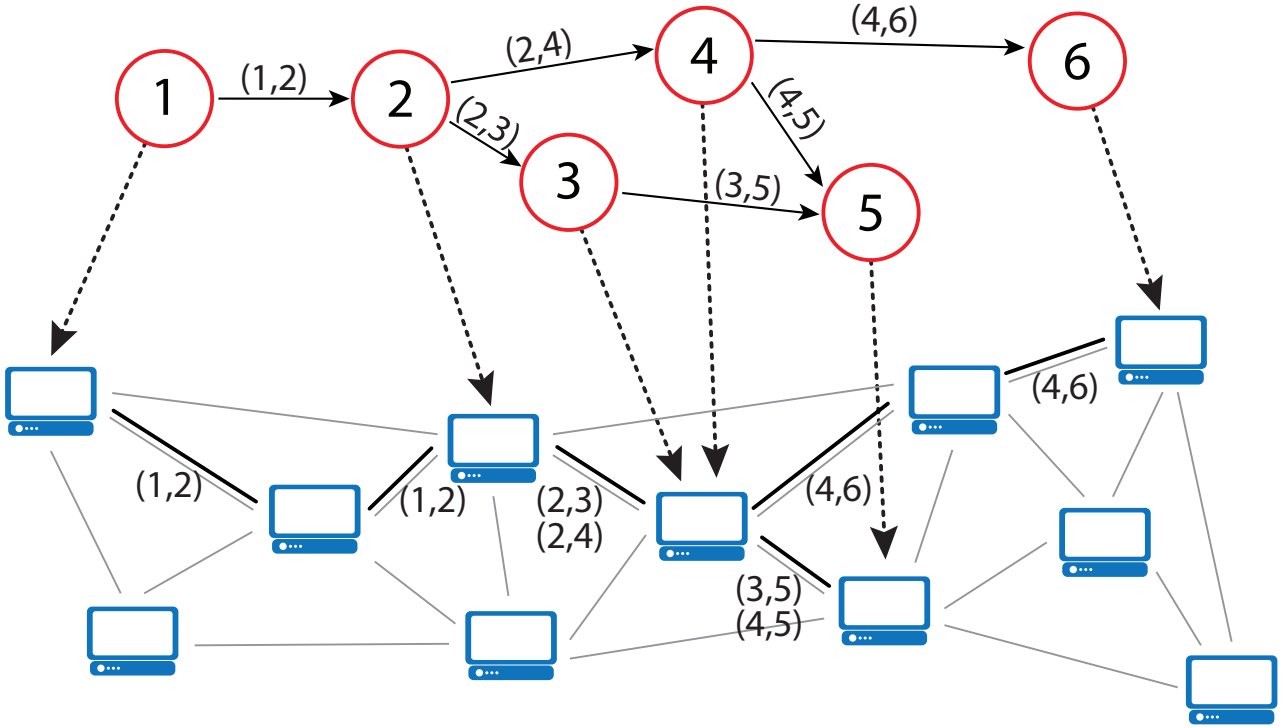
- Represented by a **directed graph**
  - Graph vertices: operators
  - Graph edges: data streams
  - Usually directed acyclic graph (**DAG**)



- Operator:
  - Processing element that transforms one or more input streams into another stream
  - Can be stateless or stateful

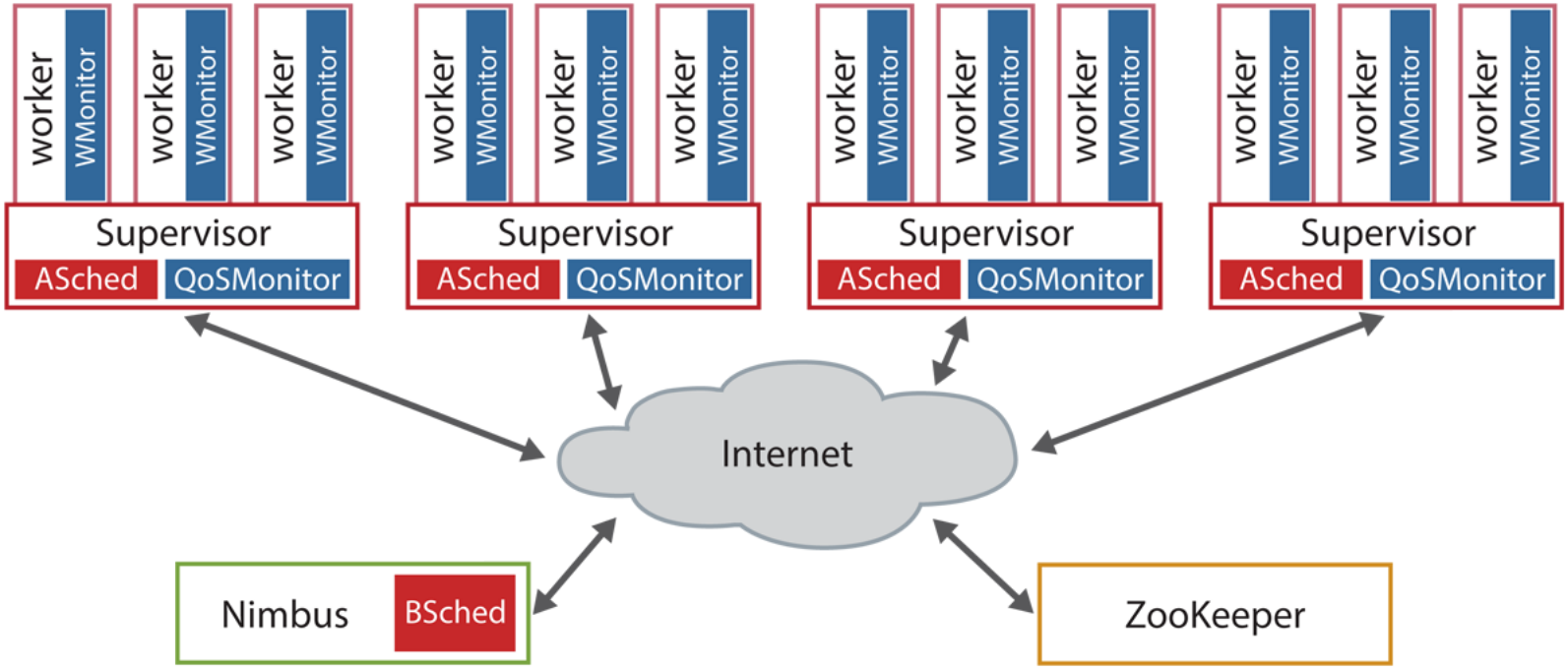
Challenge 1: Operator placement

- How to assign the DSP operators to computing nodes which are distributed in a Fog environment



The beginning: Distributed Storm

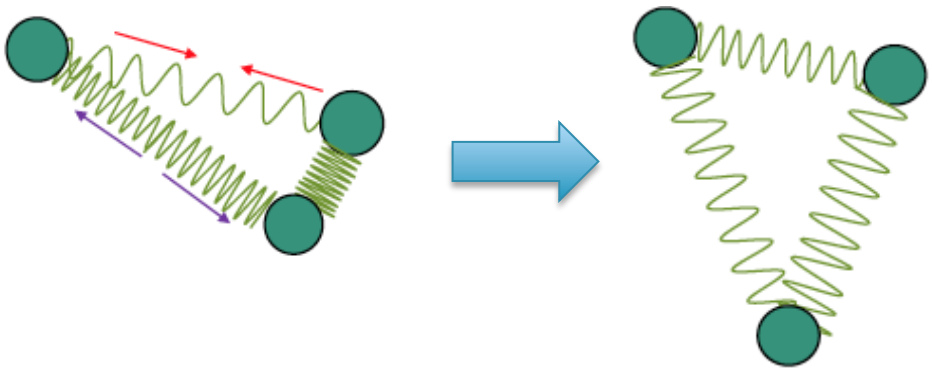
- Current DSP systems (e.g., Storm, Flink, Heron) are designed to run in single data centers
- Our initial goal: to extend Storm for a large-scale distributed and heterogeneous environment



V. Cardellini, V. Grassi, F. Lo Presti, M. Nardelli, Distributed QoS-aware scheduling in Storm. DEBS '15.

Network latency estimation

- How to provide an efficient estimation of the network delay between pairs of nodes?
- Use a network coordinates system
  - To predict latencies without performing direct measurements
    - E.g., Vivaldi network coordinates: decentralized and gossip-based scheme



Operator placement policies

- Operator placement: NP-hard problem
- Several placement policies in literature (mainly heuristics) that address such problem but
  - Different assumptions (system model, application topology, QoS attributes and metrics, …)
  - Different objectives
  - Not easily comparable

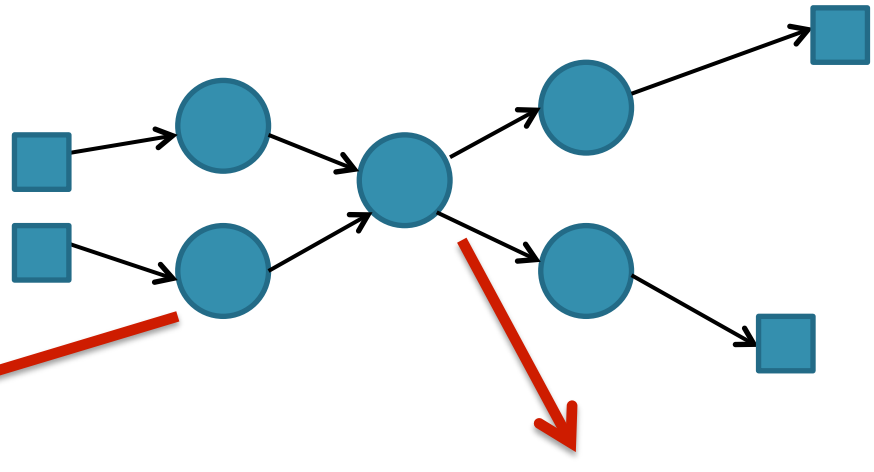
ODP: **O**ptimal **D**SP **P**lacement

- We propose ODP
  - Centralized policy for optimal placement of DSP applications
  - Formulated as Integer Linear Programming (ILP) problem
- Our goals:
  - To compute the **optimal placement** (of course!)
  - To provide a **unified general formulation** of the placement problem for DSP applications (but not only!)
  - To consider multiple **QoS attributes** of applications and resources
  - To provide a **benchmark** for heuristics

V. Cardellini, V. Grassi, F. Lo Presti, M. Nardelli, Optimal Operator Placement for Distributed Stream Processing Applications, DEBS '16.

ODP: model

DSP application



Operators

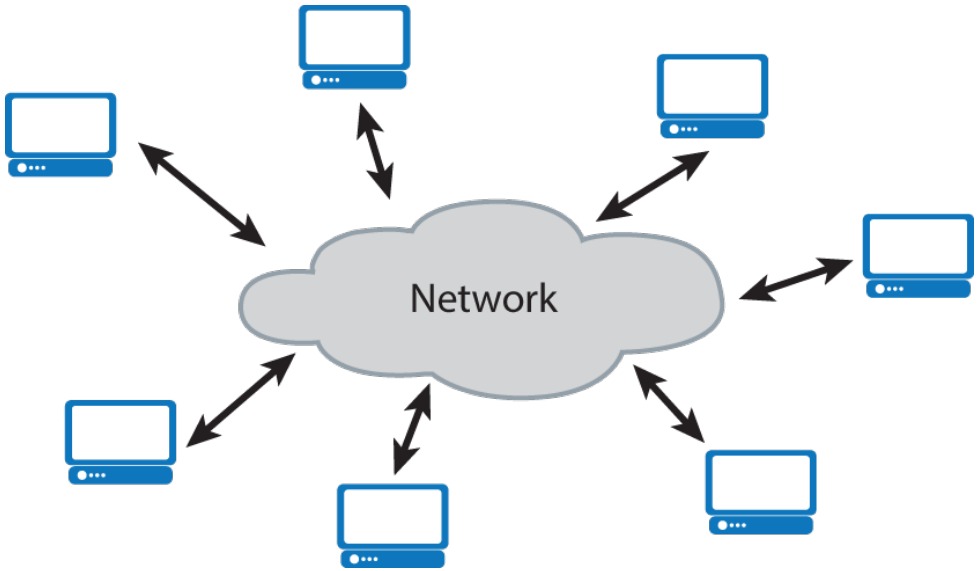
- C_i required computing resources
- R_i execution time per data unit

Data streams

- $\lambda_{i,j}$ data rate from operator *i* to *j*

ODP: model

Computing and network resources



Computing resources

- C_u amount of resources
- S_u processing speed
- A_u resource availability

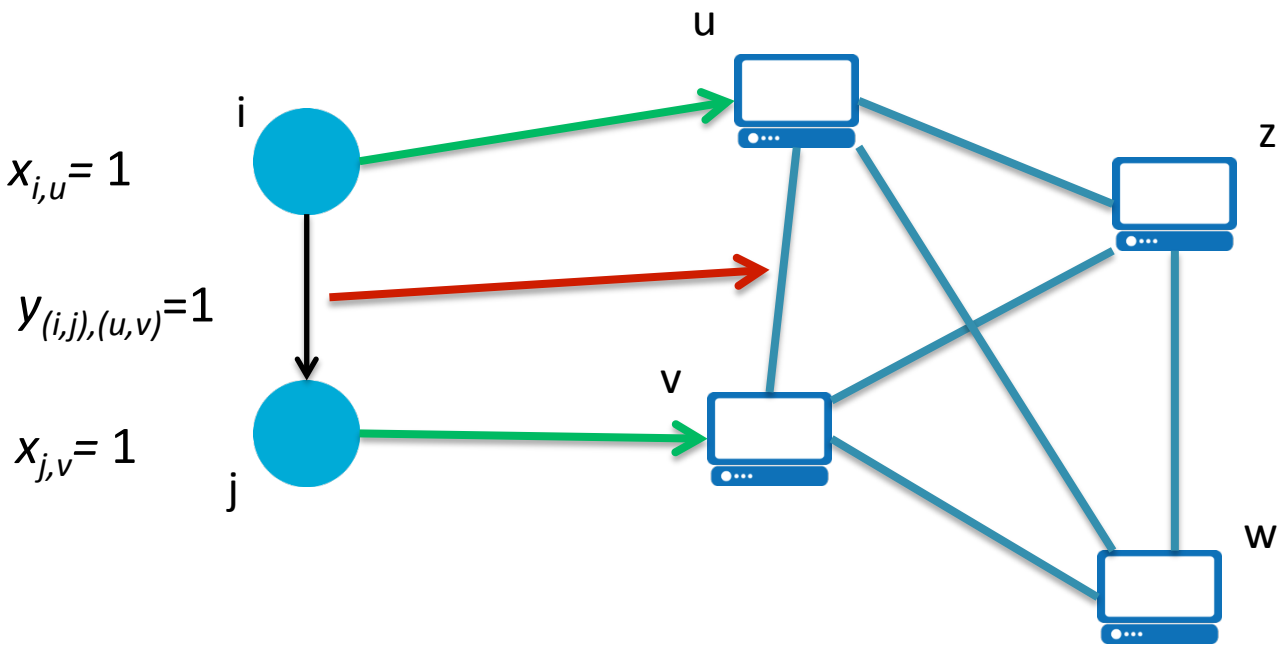
(Logical) Network links

- $d_{u,v}$ network delay from u to v
- $B_{u,v}$ bandwidth from u to v
- $A_{u,v}$ link availability

ODP: model

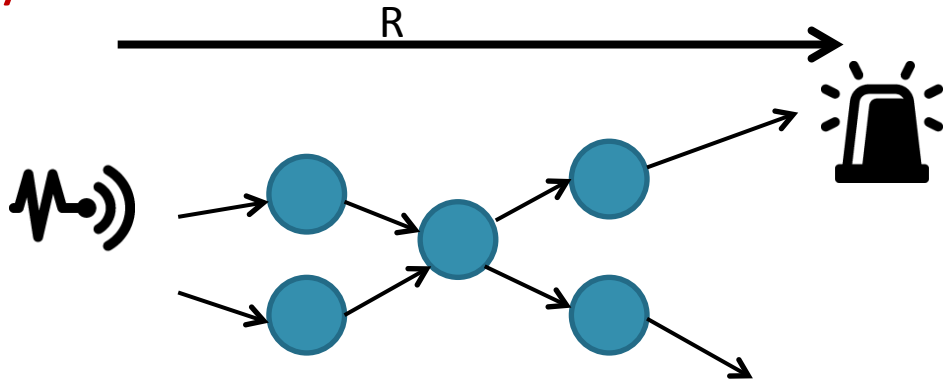
Decision variables

Where to map operators and data streams



ODP: some QoS metrics

- Latency
  - Max end-to-end delay between sources and destinations



- Availability
  - Prob. that all operators/links are up and running
- Latency and bandwidth
  - Inter-node traffic
  - Network usage
    - In flight bytes

$$\sum_{links \in l} rate(l)Lat(l)$$

ODP: ILP formulation

Tunable knobs to set the optimal placement goals

$$\max_{\boldsymbol{x},\boldsymbol{y},r} F(\boldsymbol{x},\boldsymbol{y},r)$$

subject to:

Latency

$$r \geq \sum_i \sum_u \frac{R_i}{S_u} x_{i,u} + \sum_{(i,j)} \sum_{(u,v)} d_{(u,v)} y_{(i,j),(u,v)} \qquad \forall p \in \pi_G$$

Availability

$$a(\boldsymbol{x},\boldsymbol{y}) = \sum_i \sum_u a_u x_{i,u} + \sum_{(i,j)} \sum_{(u,v)} a_{(u,v)} y_{(i,j),(u,v)}$$

Network bandwidth and node capacity constraints

$$B_{(u,v)} \geq \sum_{(i,j)} \lambda_{(i,j)} y_{(i,j),(u,v)} \qquad \forall u \in V_{res}, v \in V_{res}$$

$$\sum_i C_i x_{i,u} \leq C_u \qquad \forall u \in V_{res}$$

Assignment and integer constraints

$$\sum_u x_{i,u} = 1 \qquad \forall i \in V_{dsp}$$

$$x_{i,u} = \sum_v y_{(i,j),(u,v)} \qquad \forall (i,j) \in E_{dsp}, u \in V_{res}$$

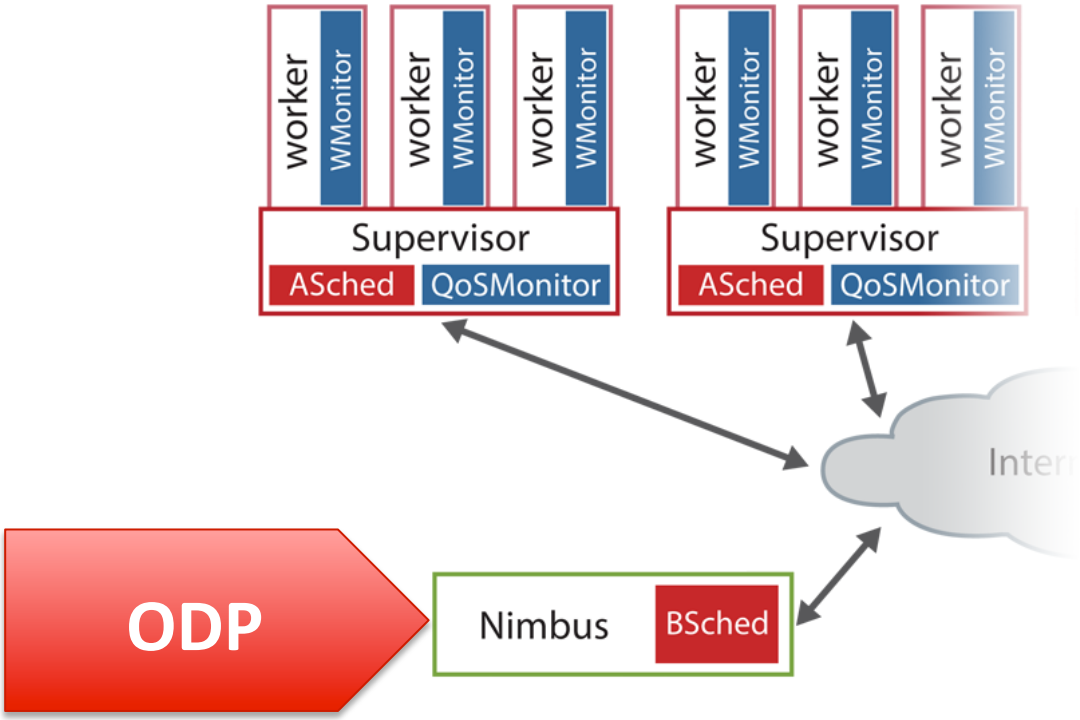
$$x_{j,v} = \sum_u y_{(i,j),(u,v)} \qquad \forall (i,j) \in E_{dsp}, v \in V_{res}$$

$$x_{i,u} \in \{0,1\} \qquad \forall i \in V_{dsp}, u \in V_{res}$$

$$y_{(i,j),(u,v)} \in \{0,1\} \qquad \forall (i,j) \in E_{dsp}, (u,v) \in E_{res}$$

ODP and Apache Storm

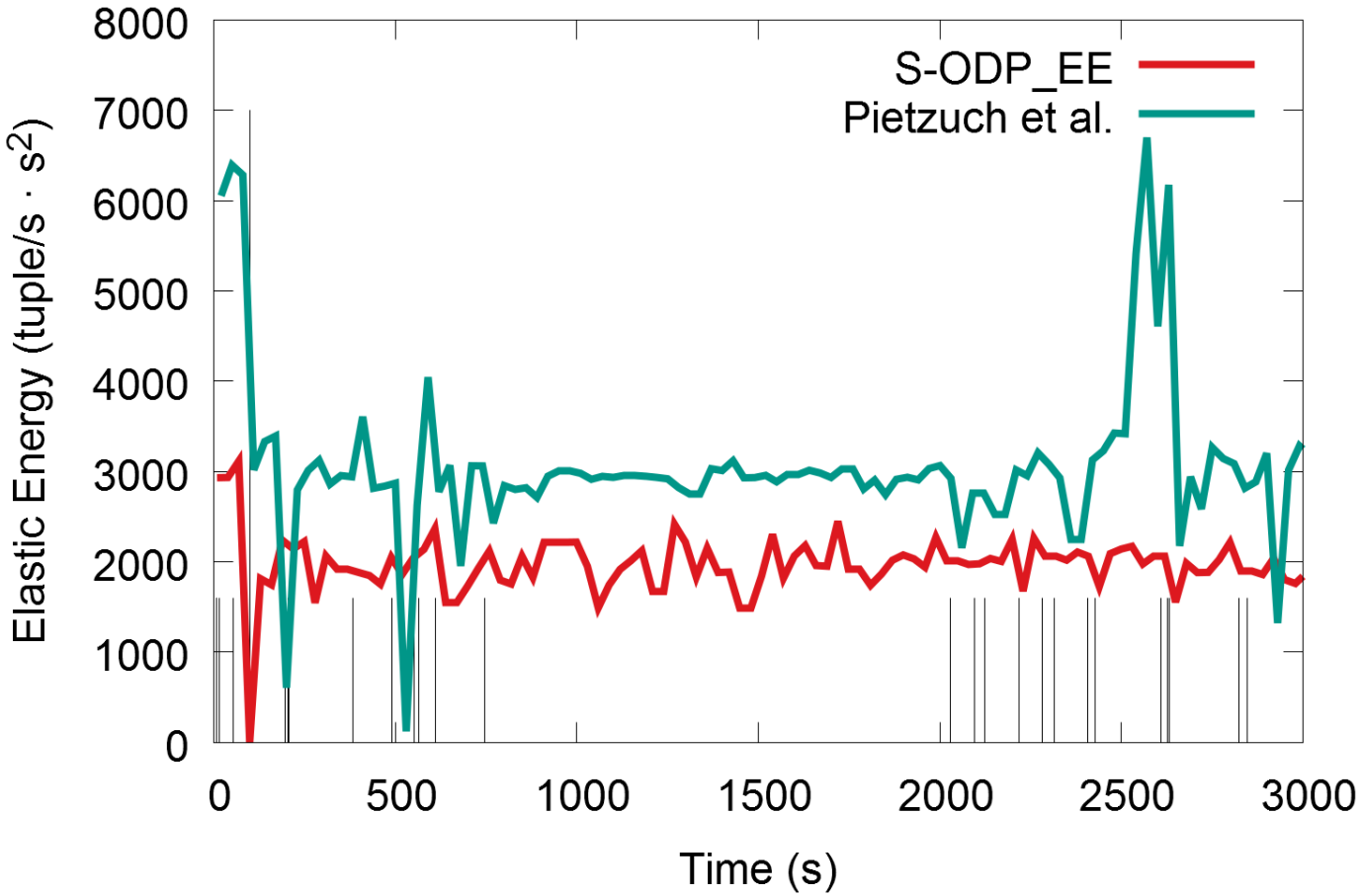
- We can use ODP
  - to determine the optimal placement
  - as benchmark to evaluate existing heuristics



ODP: Benchmark for placement heuristics

Distributed placement heuristic that minimizes network usage

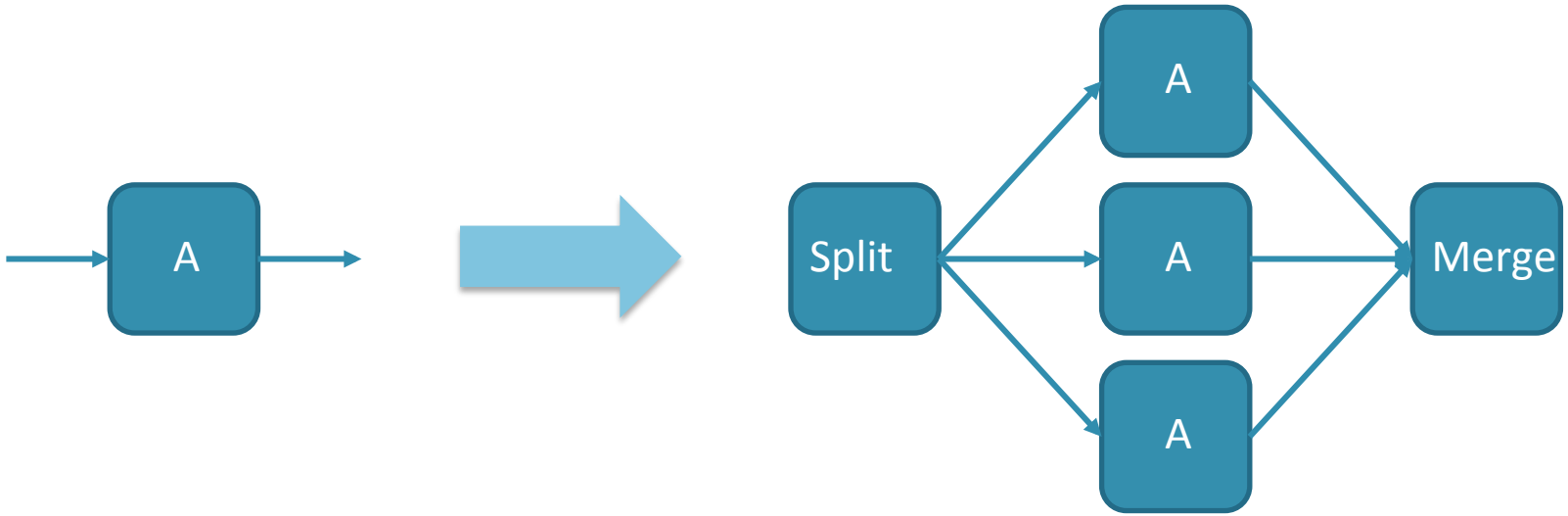
Pietzuch et al. : $\min EE(\boldsymbol{y}) = \min \sum_{(i,j)} \sum_{(u,v):u \neq v} \lambda_{(i,j)} d^2_{(u,v)} y_{(i,j),(u,v)}$



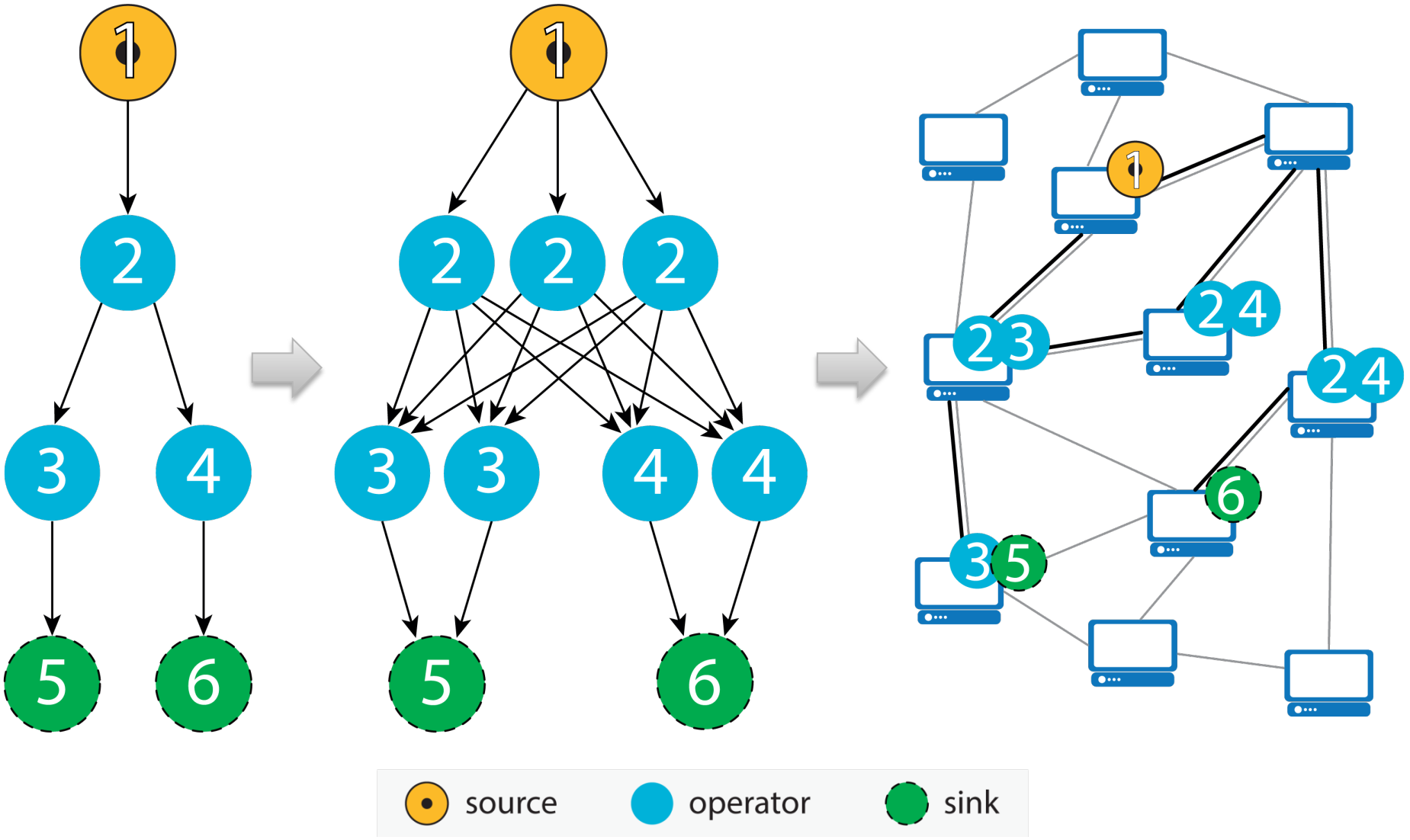
P. Pietzuch et al., Network-aware operator placement for stream-processing systems, ICDE '06.

Challenge 2: placement and replication

- Exploit application-level parallelism by replication operators



Operator placement and replication

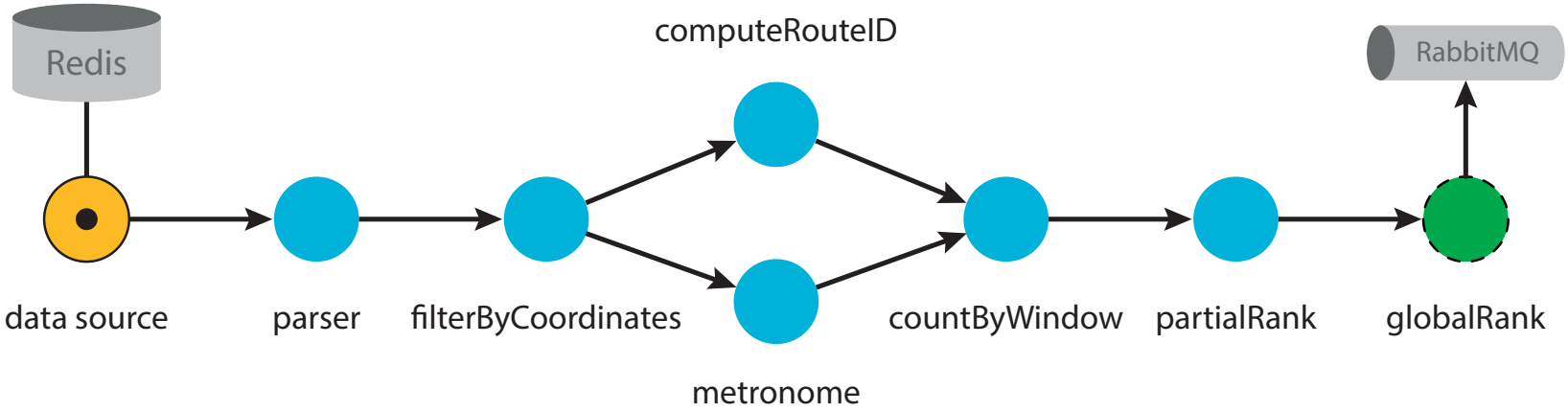


ODRP: **O**ptimal **D**SP **R**eplication and **P**lacement

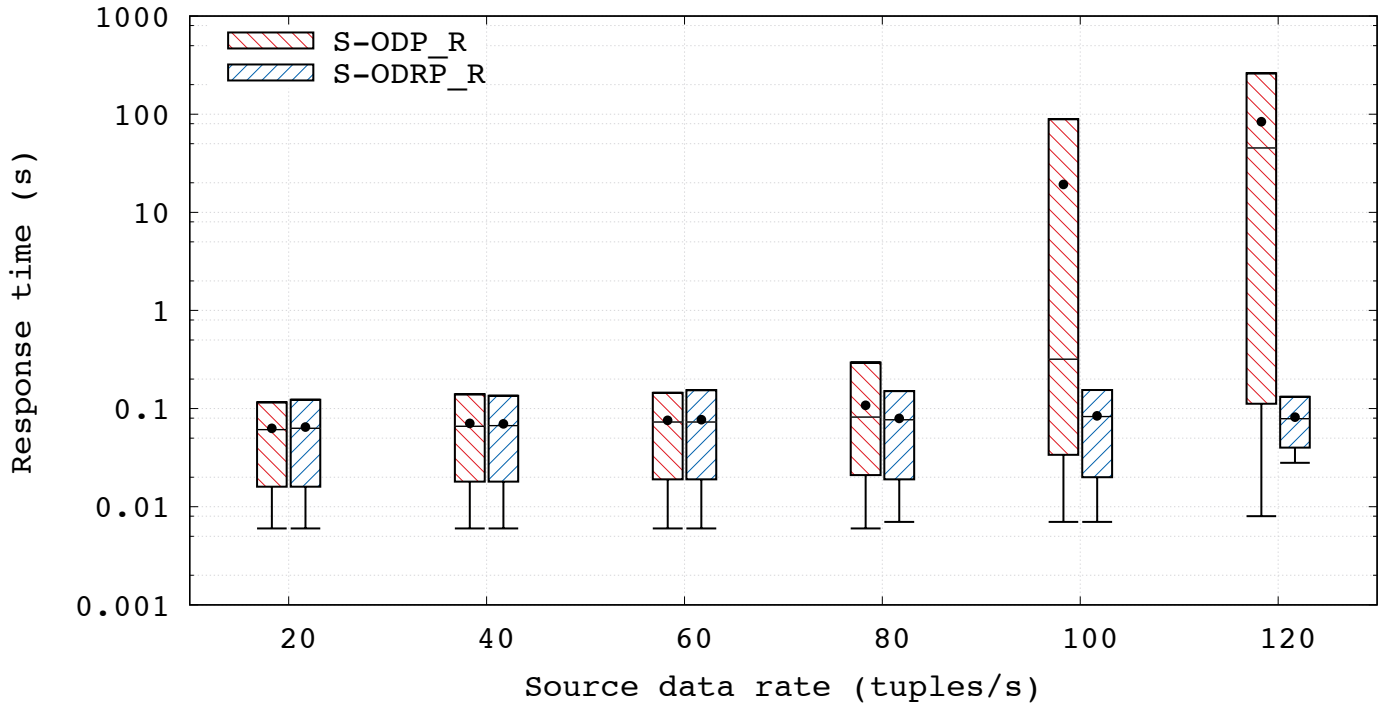
- We propose **ODRP**
  - Centralized policy for optimal replication and placement of DSP applications
  - Formulated as Integer Linear Programming (ILP) problem
- Our goals:
  - To jointly determine the optimal **number of replica** and their **placement**
  - To consider multiple **QoS attributes** of applications and resources
  - To provide a **unified general formulation**
  - To provide a **benchmark** for heuristics

V. Cardellini, V. Grassi, F. Lo Presti, M. Nardelli, Optimal operator replication and placement for distributed stream processing systems. ACM Perf. Eval. Rew., 2017.

ODRP performance



DSP application: DEBS 2015 Grand Challenge



Challenge 3: runtime deployment

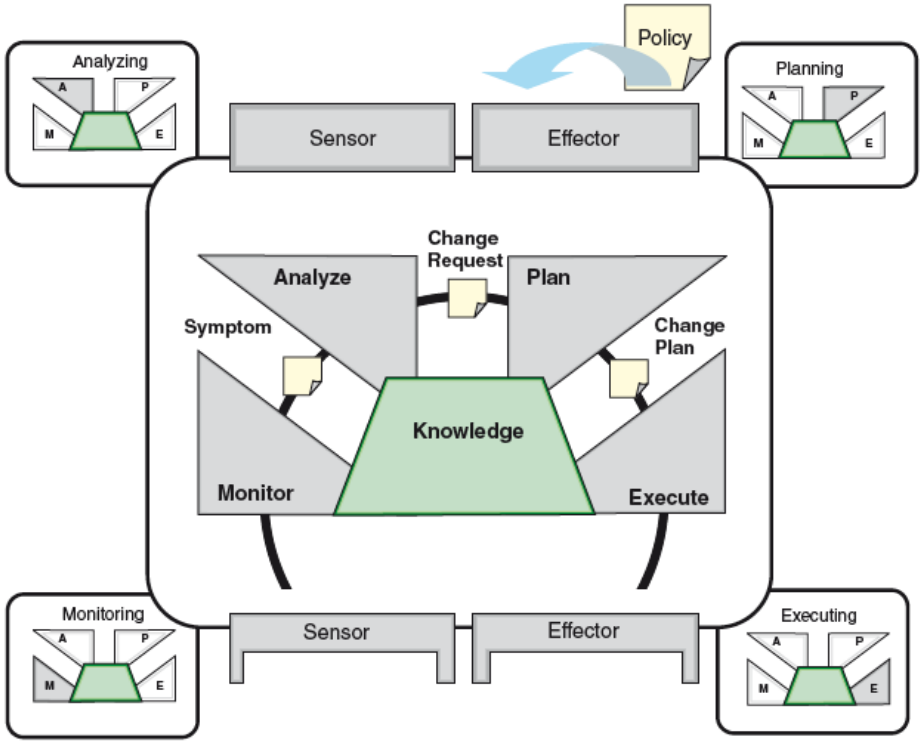
- Many factors may change at runtime, e.g.,
  - Load variations, QoS attributes of resources, cost of resources (e.g., due to dynamic pricing schemes), network characteristics, node mobility, …

- How to adapt the placement and replication when changes occur?

  Exploit self-adaptive deployment

Self-adaptive deployment

- **MAPE** (**M**onitor, **A**nalyze, **P**lan and **E**xecute)



- **Plan** phase: how to **reconfigure** the application deployment

Reconfiguration challenges

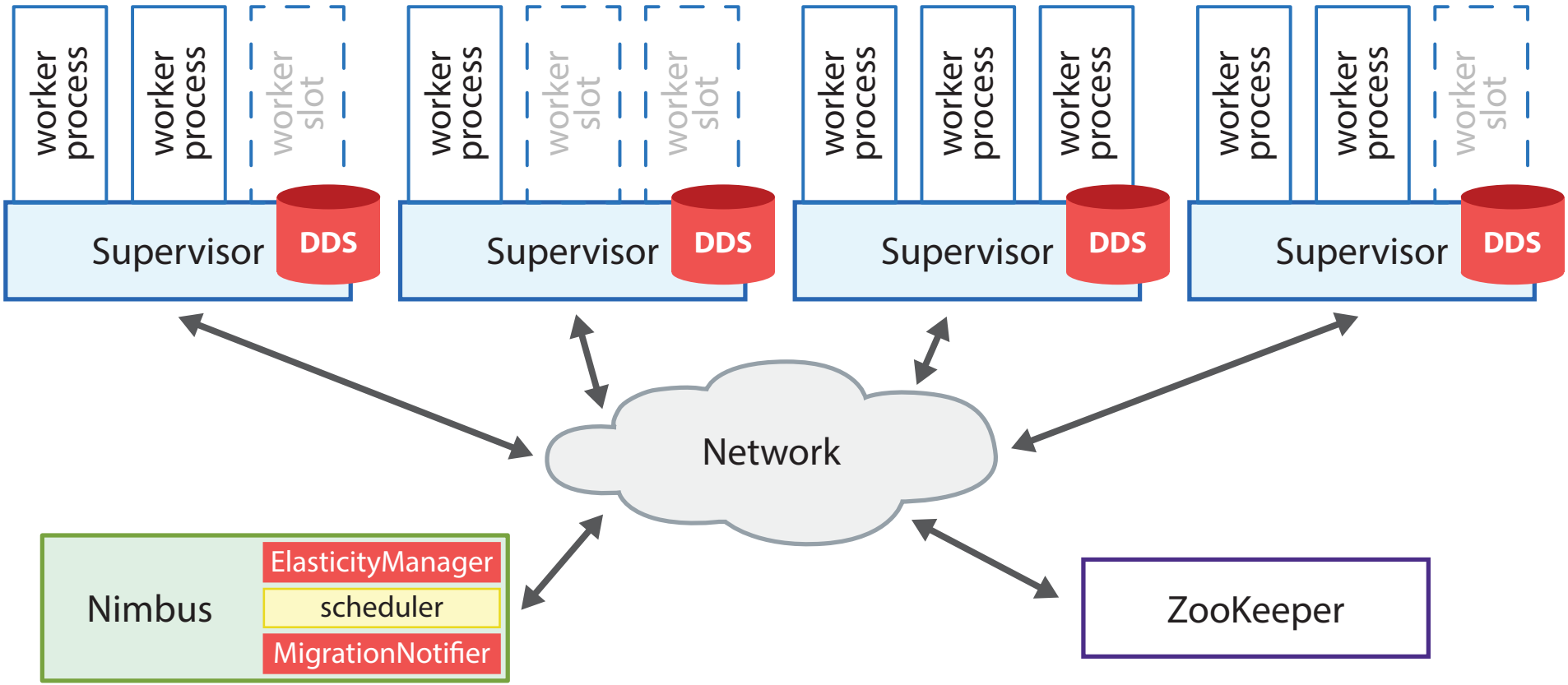
- Reconfiguring the deployment has a non negligible cost!
- Can affect negatively application performance in the short term
  - Application freezing times caused by operator migration and scaling, especially for stateful operators

Perform reconfiguration only when needed

Take into account the overhead for migrating and scaling the operators

Elastic stateful migration in Storm

- We develop mechanisms for elastic stateful migration in Apache Storm



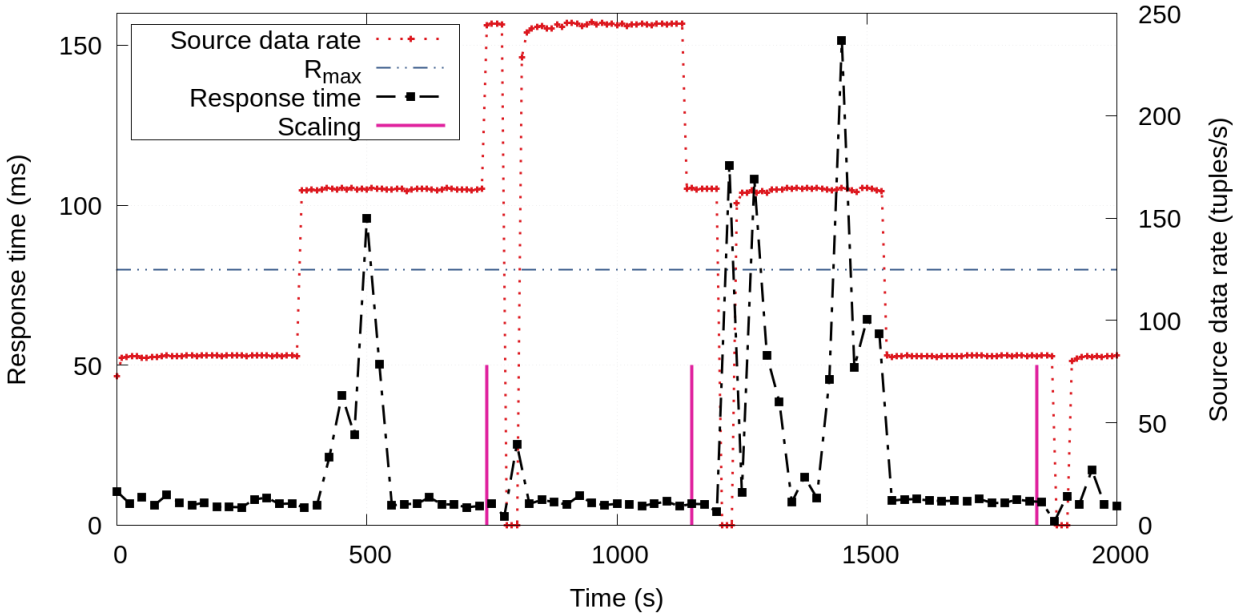
V. Cardellini, M. Nardelli, D. Luzi, Elastic stateful stream processing in Storm, HPCS '16.

EDRP: **E**lastic **D**SP **R**eplication and **P**lacement

- Unified framework for the QoS- aware initial deployment and runtime elasticity management of DSP applications
- We model **reconfiguration costs**
  - Related to migrating or scaling in/out the operators
- Centralized policy formulated as Integer Linear Programming (ILP) problem

V. Cardellini, F. Lo Presti, M. Nardelli, G. Russo Russo, Optimal operator deployment and replication for elastic distributed data stream processing, under review, 2017.

EDRP performance



With reconfiguration penalties:

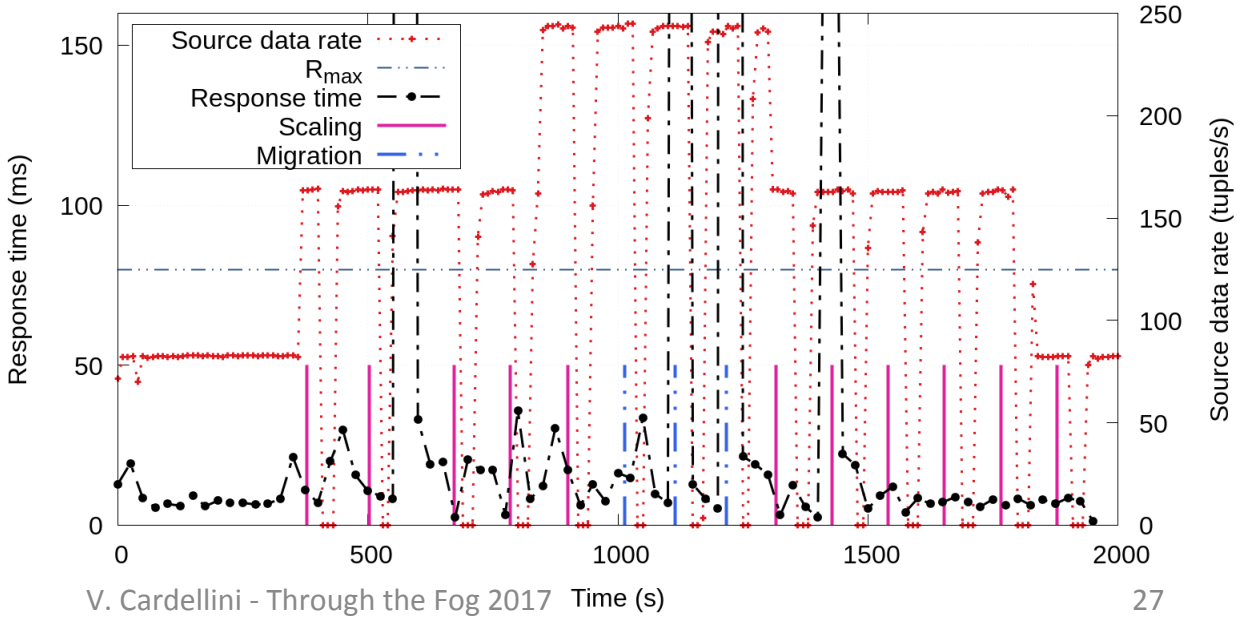
Availability: 95.5%

Downtime: < 90 s

Without reconfiguration penalties:

Availability: 81.5%

Downtime: 370 s



Future work

- Study efficient heuristics to deal with large problem instances
- Deal with uncertainty: take uncertainty of parameters into account and design robust placement algorithms
- Study how to deploy multiple competing applications in the Fog
- Integrate placement decision with SDN
  - With SDN, network into the control loop
- Study cross-layer strategies that involve multiple Big data frameworks in the Fog
  - E.g., Heron + Apache Aurora + Mesos

Acknowledgments – Co-authors

Vincenzo Grassi

Francesco Lo Presti

Matteo Nardelli

Thank you! Any questions?

Valeria Cardellini

cardellini@ing.uniroma2.it

http://www.ce.uniroma2.it/~valeria