



HAL
open science

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-01875999v1>

Submitted on 18 Sep 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Latency-Aware Strategies for Placing Data Stream Analytics onto Edge Computing

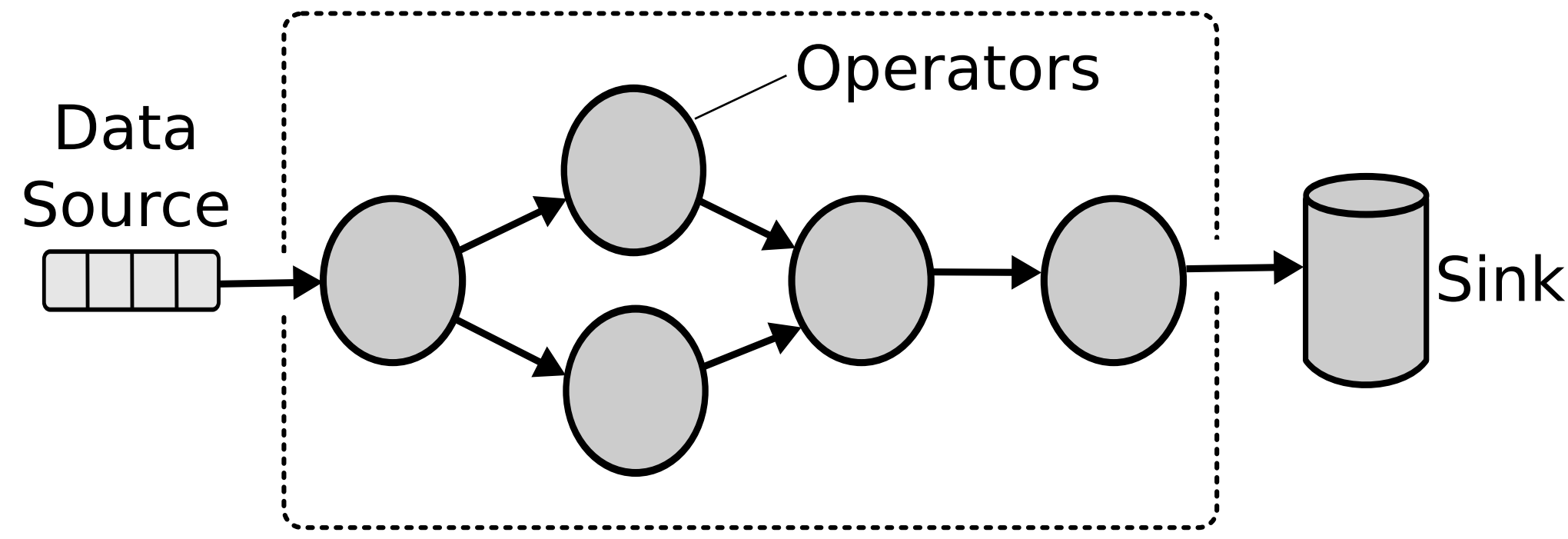
Alexandre da Silva Veith, Marcos Dias de Assunção, Laurent Lefèvre
Avalon, LIP, ENS Lyon, University of Lyon - Lyon, France
alexandre.veith@ens-lyon.fr

Scenarios



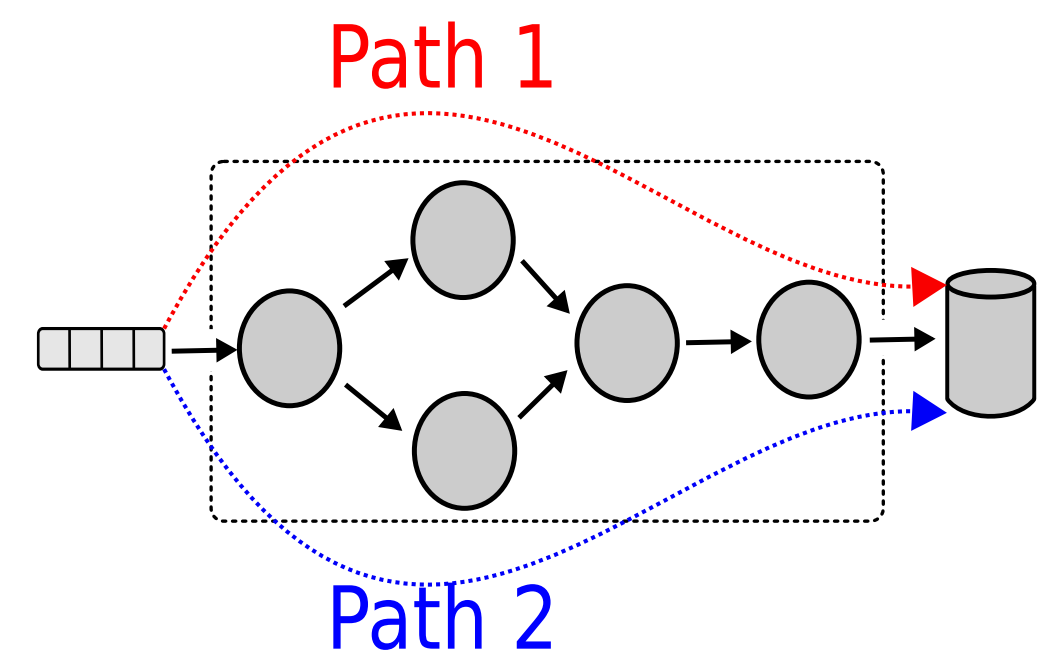
Monitoring of operational infrastructure
Anomaly detection, fraud detection
Social networks
Internet of Things (IoT)

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

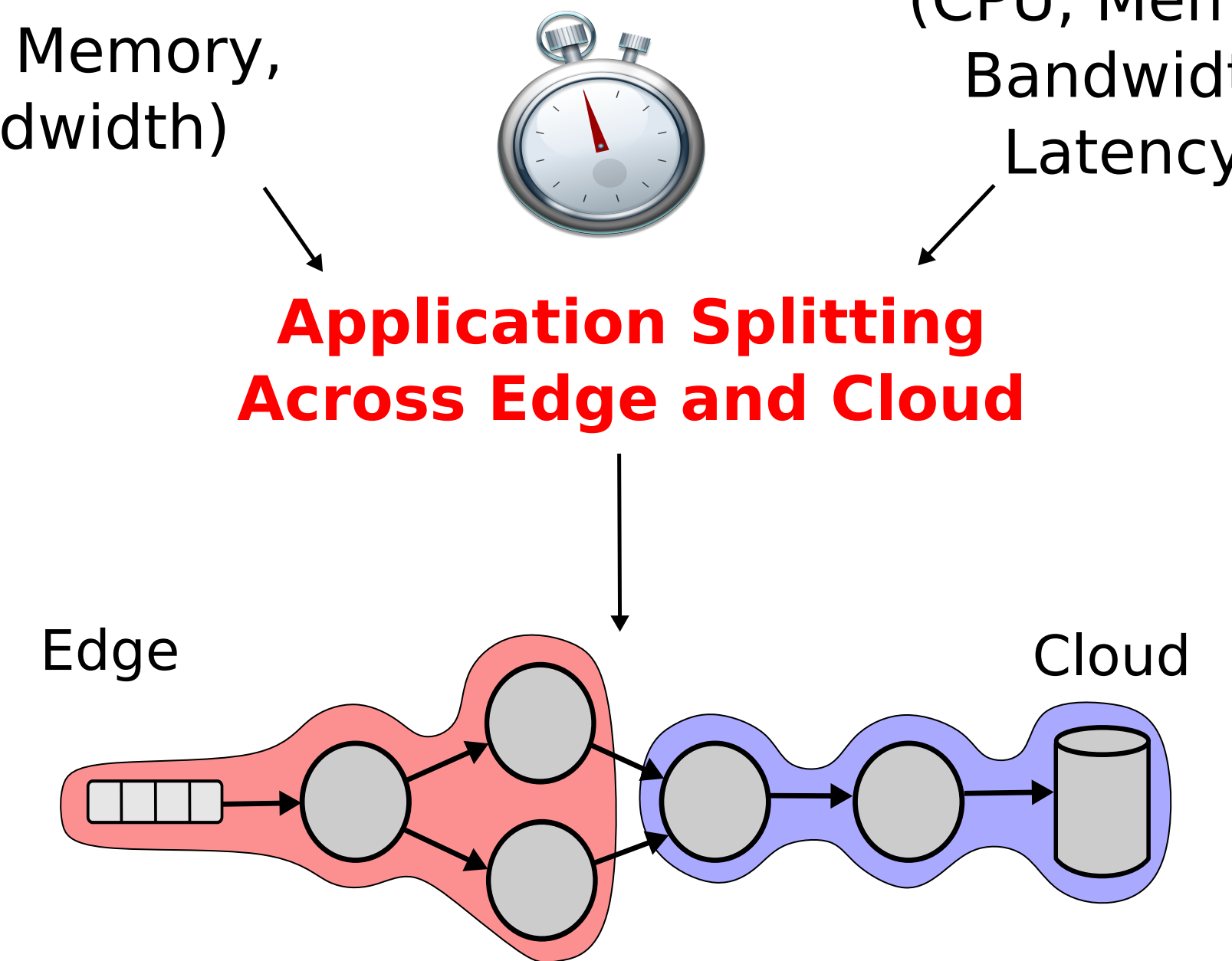
Selectivity: The ratio of number of input messages to output messages

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

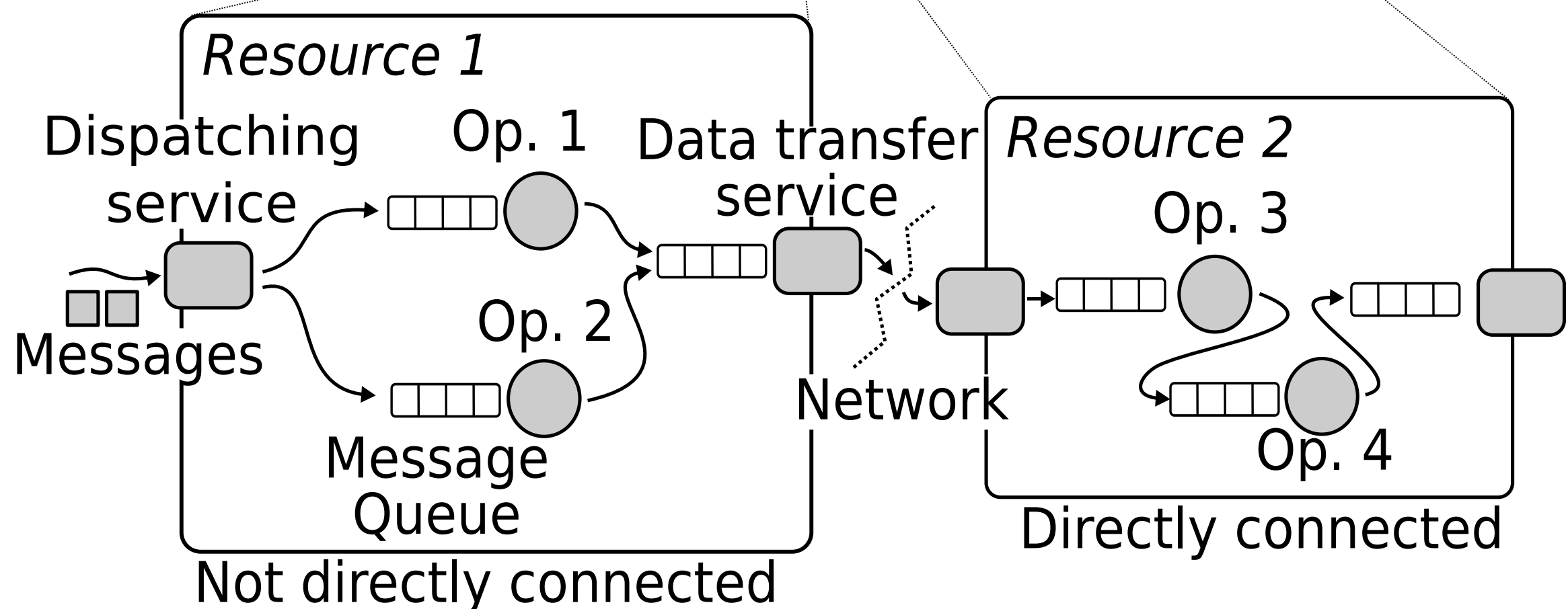
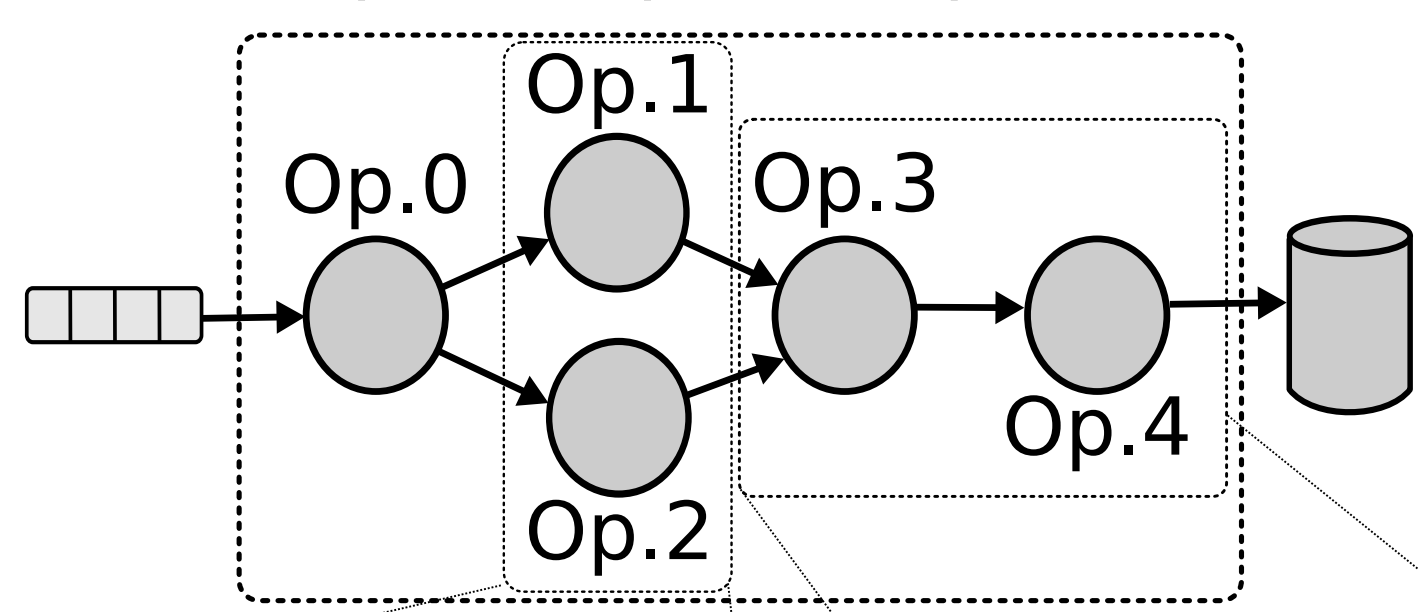
Latency-Aware Decisions

Operator Characteristics and Requirements (CPU, Memory, Bandwidth) → **Minimize the response times in all paths** → Edge and Cloud Capabilities (CPU, Memory, Bandwidth, Latency)



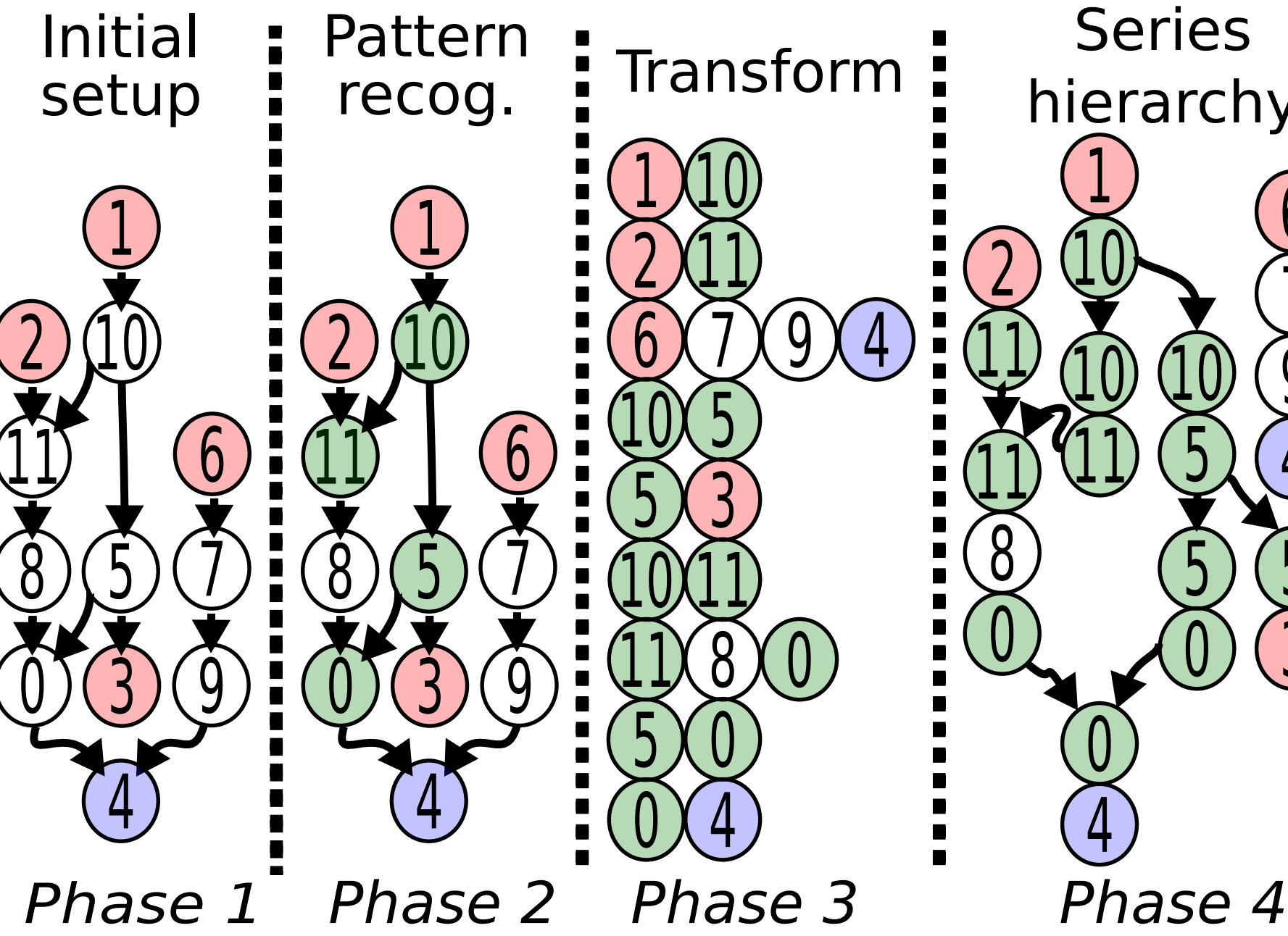
Resource and Application Models

Example of operator placement



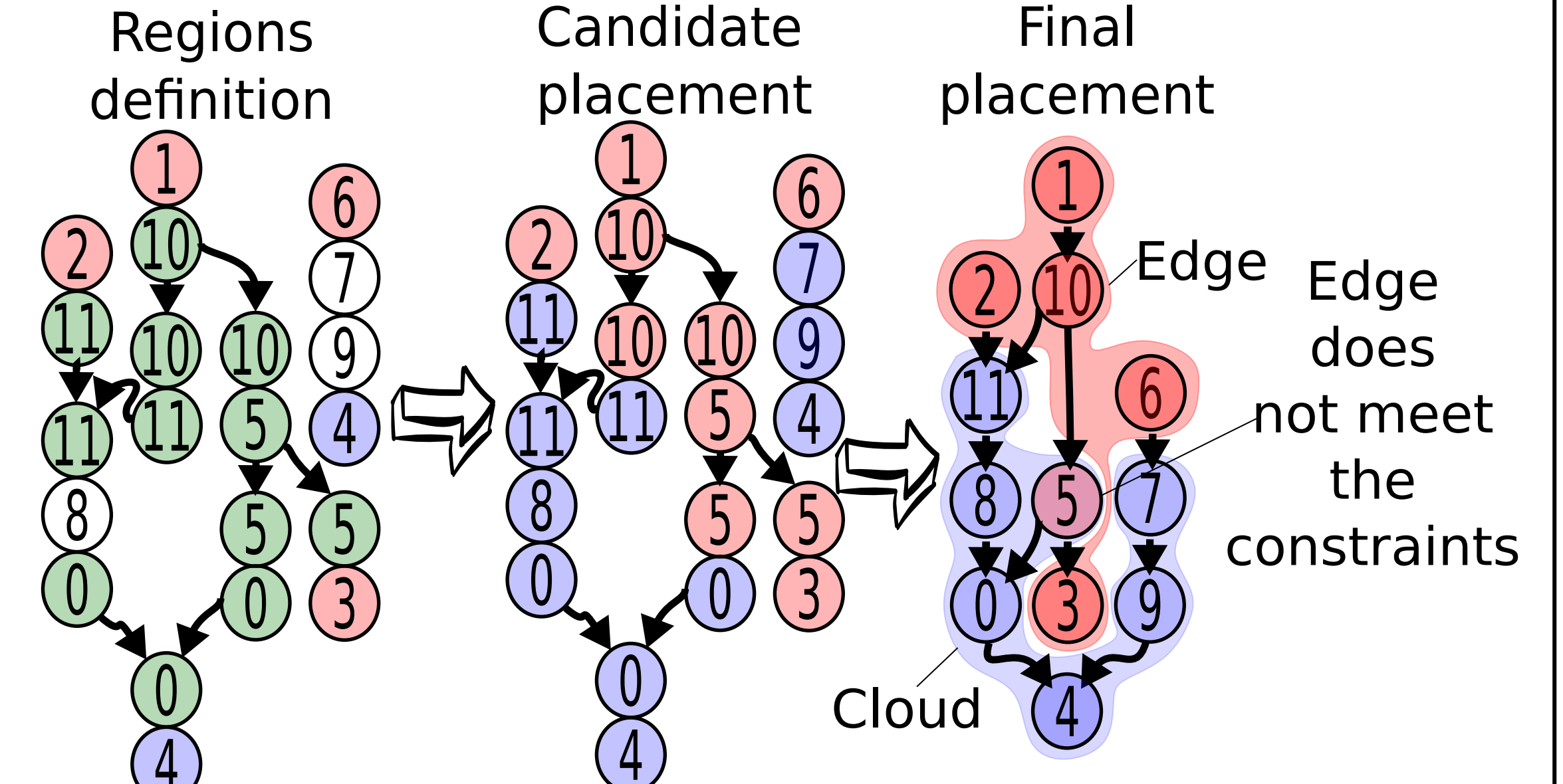
Operators - communication and computation services
Services - M/M/1 queues

Finding Application Patterns



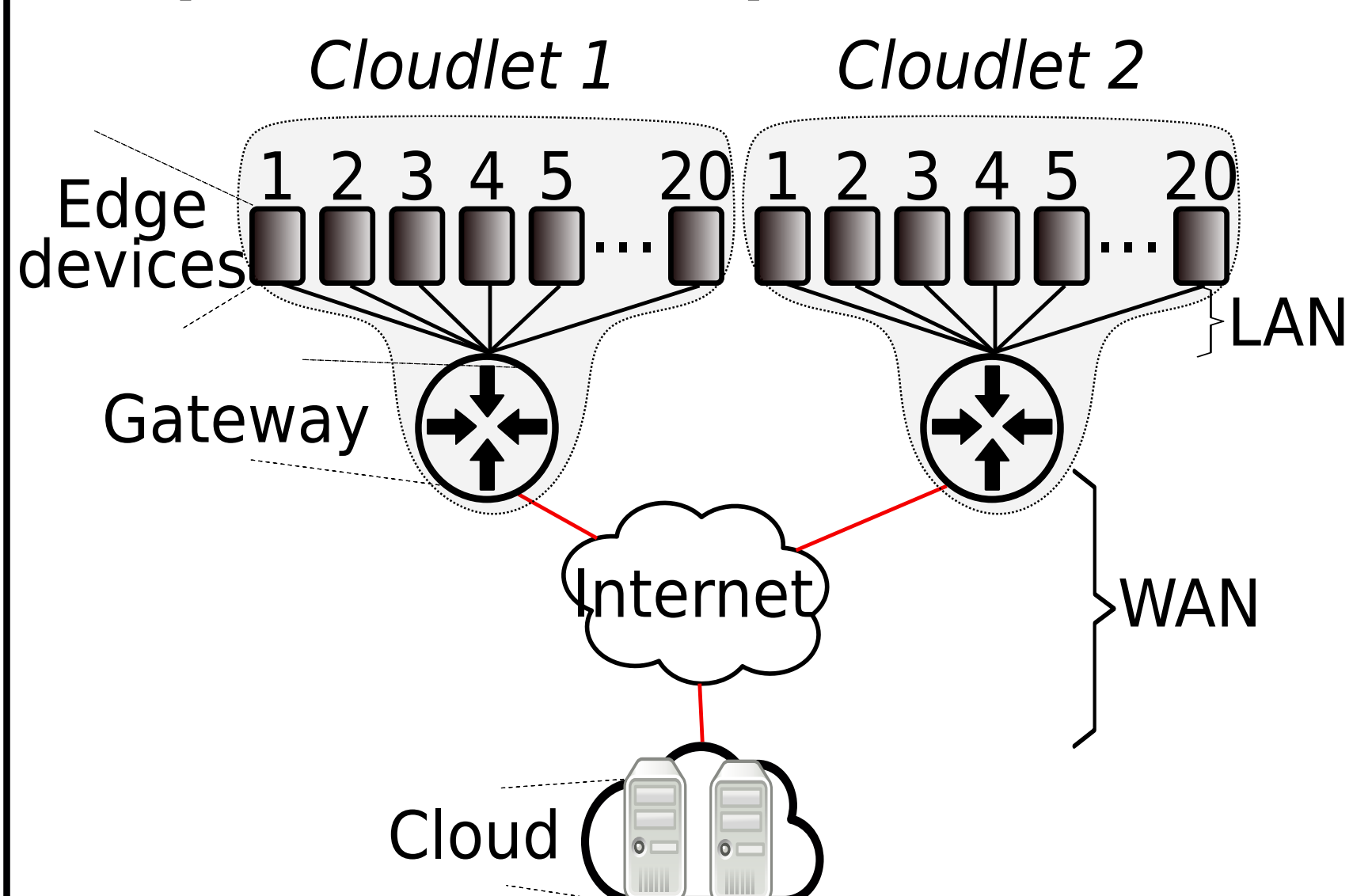
Response Time Rate (RTR) Strategy
1) BFS-Traversal algorithm = operator sequence
2) For each operator in the sequence
- Response time estimation for all resources
- Host with shortest response time is elected to host the operator

RTR with Region Patterns (RTR+RP) Strategy

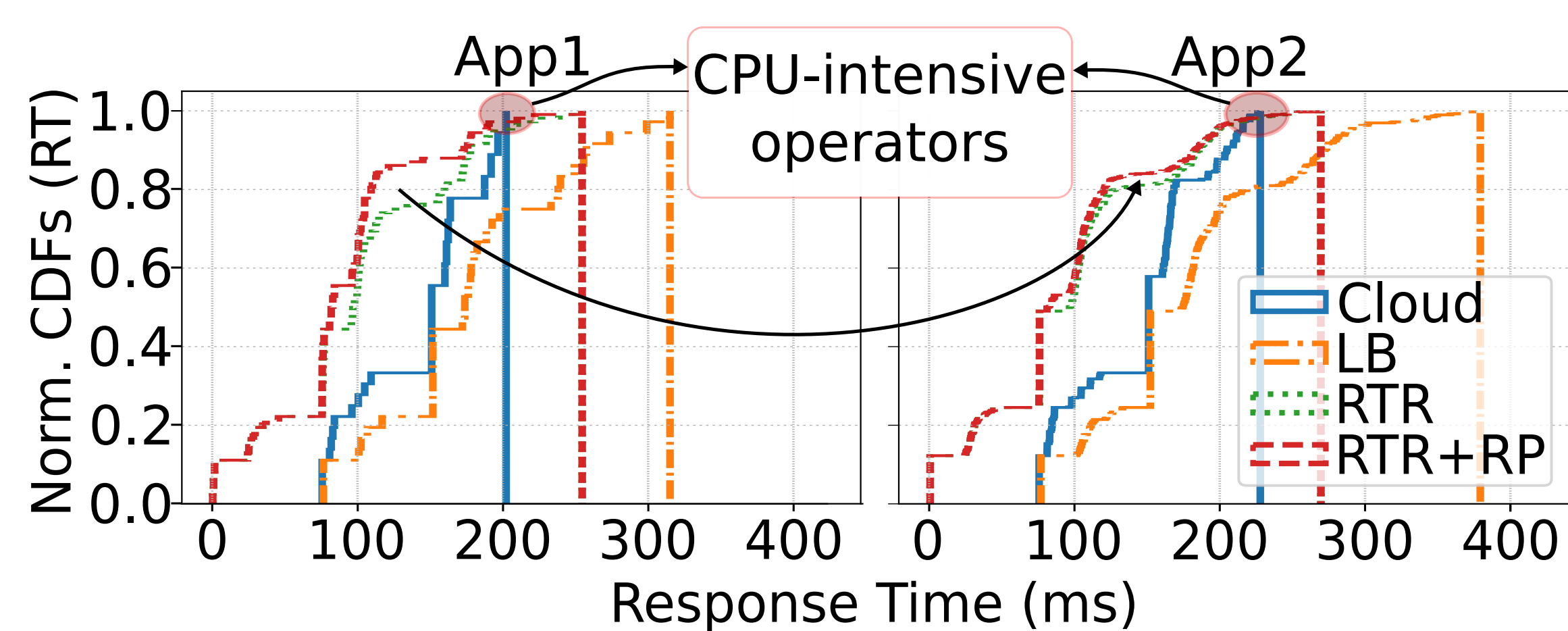
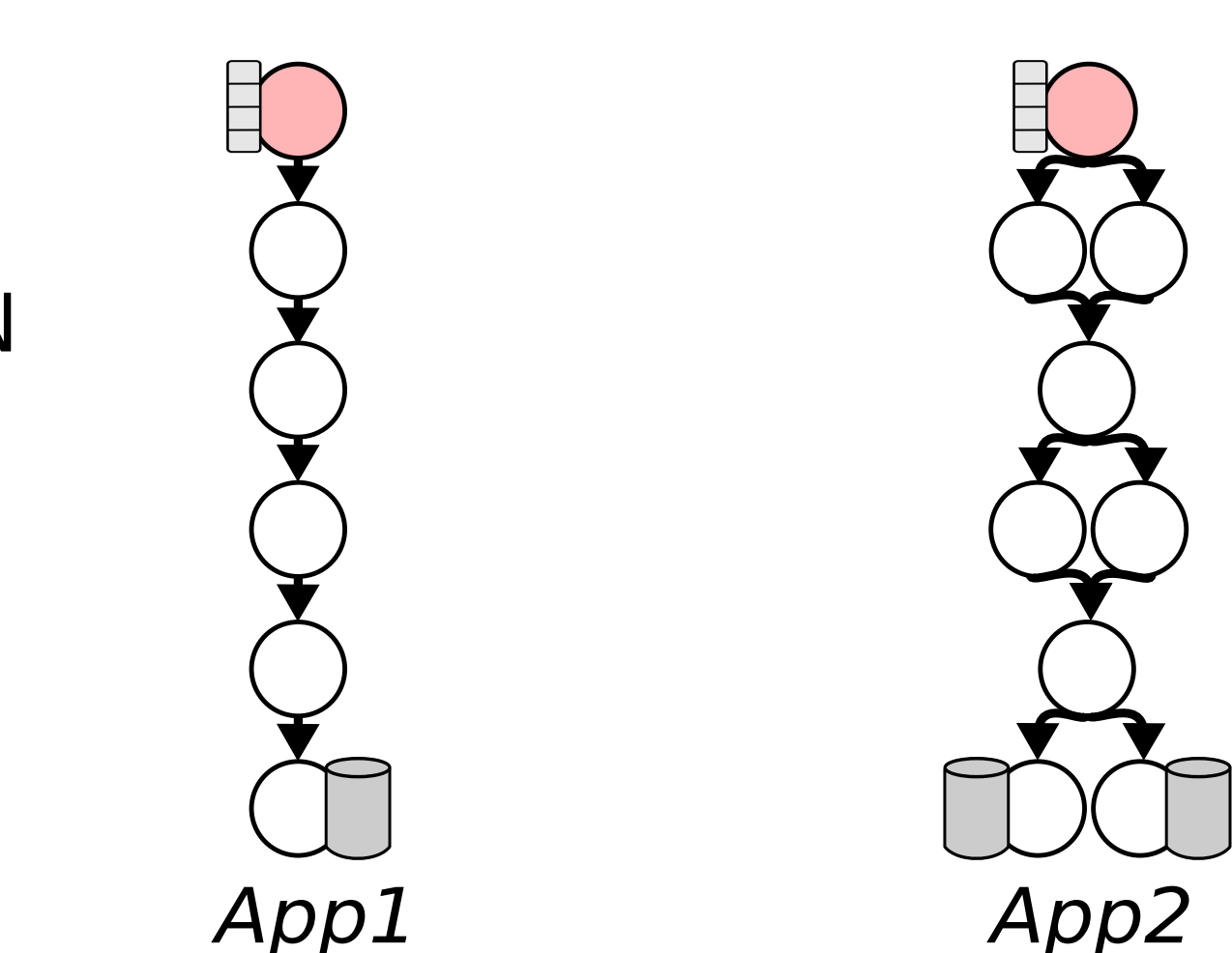


Operator sequence = RTR
Candidate placement = destination infrastructure of the message
- Only cloud = cloud
- Edge and/or cloud = edge
Response time estimation - only for edge candidates

Experimental setup



Microbenchmarks - Evaluate the impact of fork/join operators



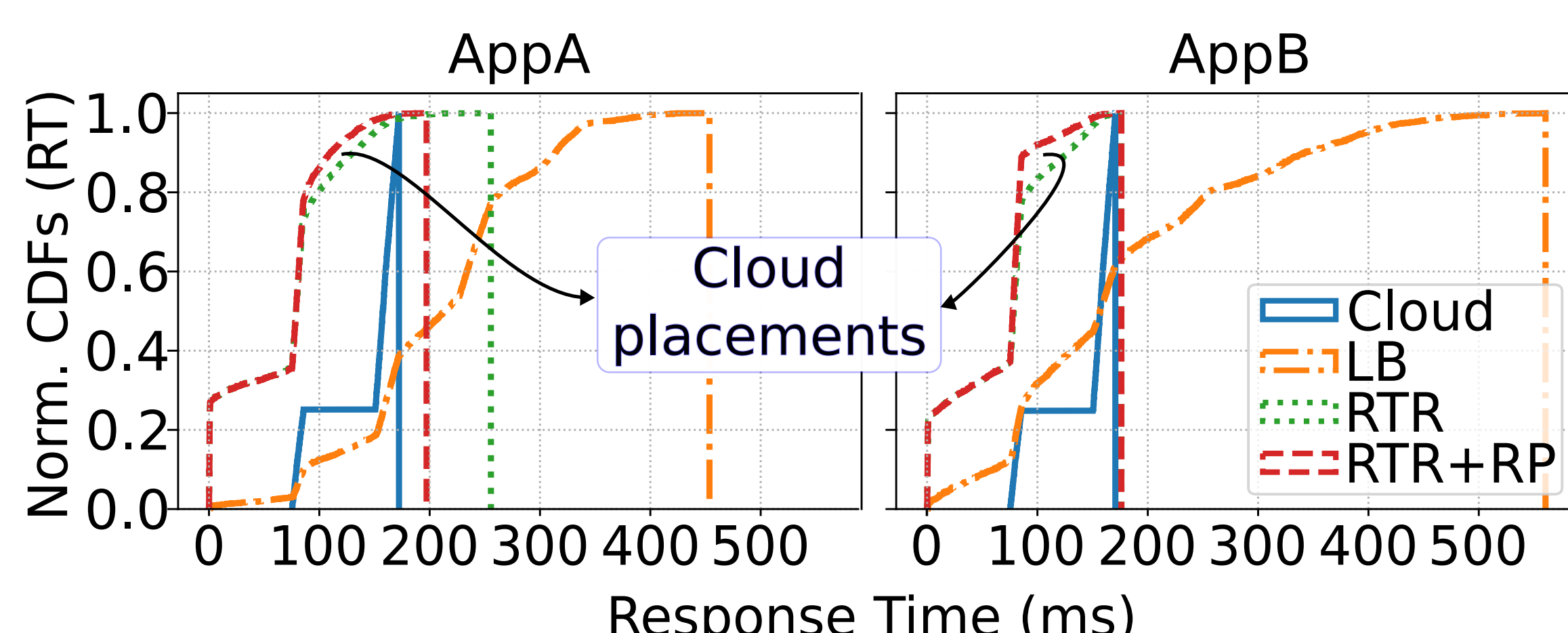
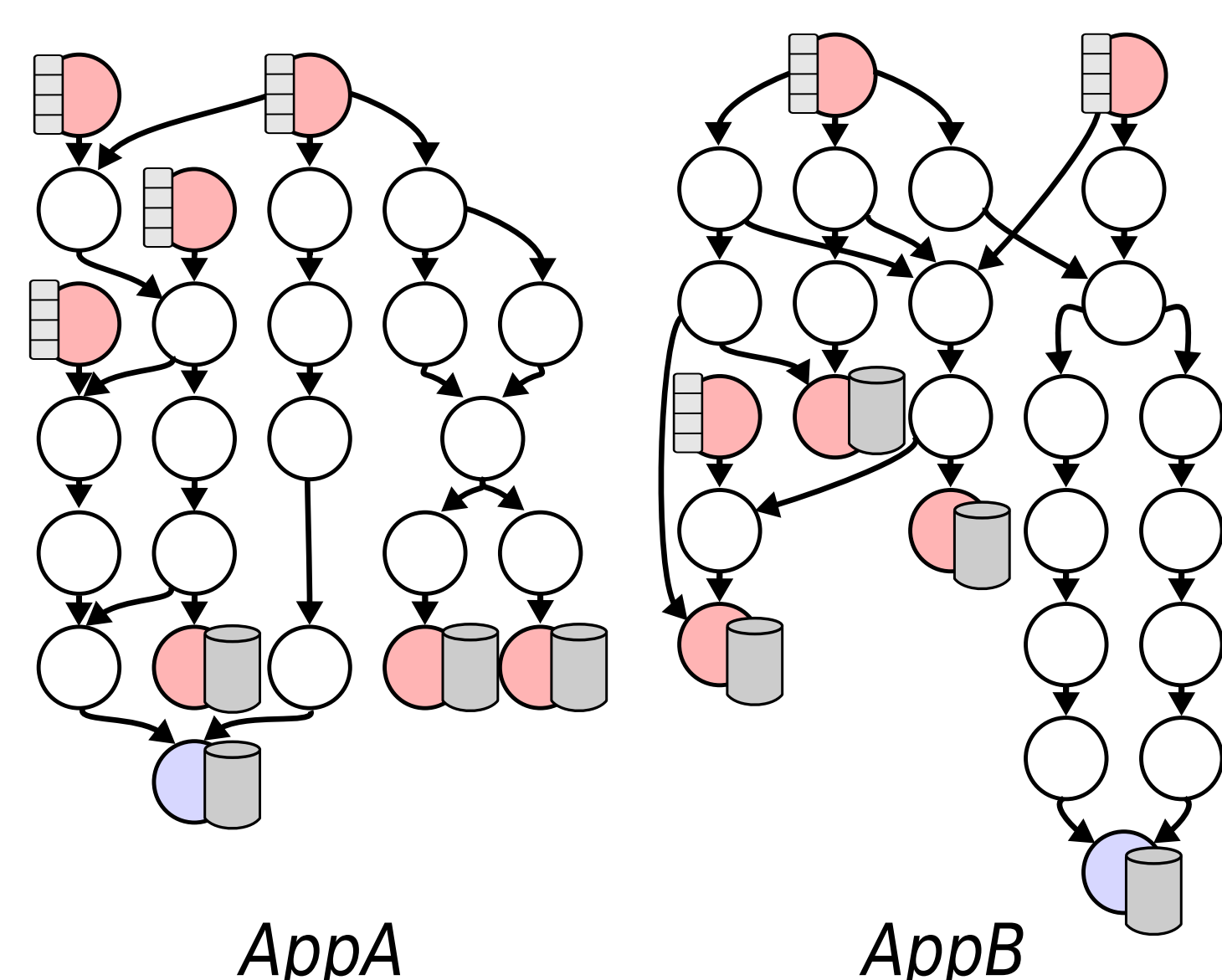
Analysis

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

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



Reference

[1] Taneja, M., Davy, A.: Resource aware placement of IoT application modules in fog-cloud computing paradigm. 2017