On the Impact of Clustering for IoT Analytics and Message Broker Placement across Cloud and Edge

Daniel Happ (TU Berlin, happ@tkn.tu-berlin.de)
Suzan Bayhan (University of Twente)

EdgeSys 2020
April 27, 2020

Motivation

Internet of Things (IoT) becoming sharing economy

- Multiple Applications use same sensor data
- Sensor data processing **needed** for meaningful insights

Storage, distribution, and **processing** usually in cloud

- Plentiful resources
- Flexible (on-demand, pay-as-you-go)

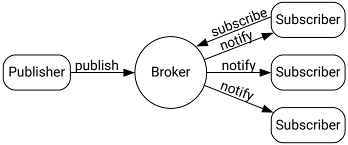
Cloud has latency and privacy issues, preventing certain use-cases

- Moving more processing to the edge

The case for Publish/Subscribe in IoT

- ”Sense once, notify many” translates well to publish/subscribe pattern

- Decoupling properties:
  - Time
  - Synchronization
  - Space



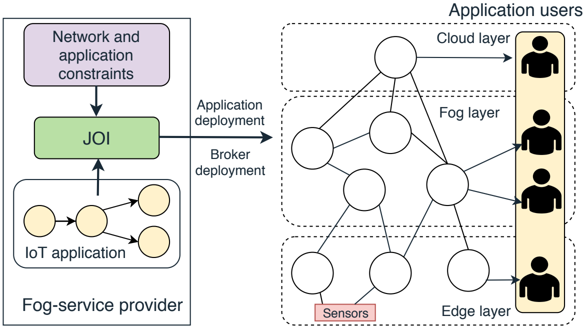
- Enable seamless processing operator offloading scheme[1]

[1] D. Happ and A. Wolisz, ”Towards gateway to cloud offloading in IoT publish/subscribe systems,” in 2017 Second International Conference on Fog and Mobile Edge Computing (FMEC), May 2017, pp. 101–106.

Research questions

1. How to place operators and brokers jointly across a cloud/fog/edge topology?
2. What is the impact of clustering of publishers and subscribers on the placement?
3. What is the impact of the network size?

JOI deploys Operators & Message Brokers[2]



Input: network and application constraints & application graph

Output: where to deploy operators and brokers

[2] Daniel Happ, Suzan Bayhan, and Vlado Handziski. 2020. JOI: Joint Placementof IoT Analytics Operators and Pub/Sub Message Brokers in Fog-centric IoT Platforms. Future Generation Comp. Sys. (2020). under review

JOI: Optimal Solution Sketch

Variables:

- $x_{i,j}$: Operator i placed on node j
- $y_{i,j}$: Operator i publishes to broker on node j

Constraints:

- Node resources (CPU, memory)
- Node-to-node bandwidth
- Node access network bandwidth

Objective:

- Minimize sum of subscriber delays

Greedy Heuristic for JOI

Preparation:

- Sort operators by hops to sink (stratum)

Depth-first search:

- Place ops with low stratum first
- Place op optimally (with current knowledge)
- Place its broker optimally (with current knowledge)

Finally:

- Join brokers until ”maximum number of brokers” constraint met

Tabu Heuristic for JOI

Starting-Point:

- Best solution of cloud and greedy

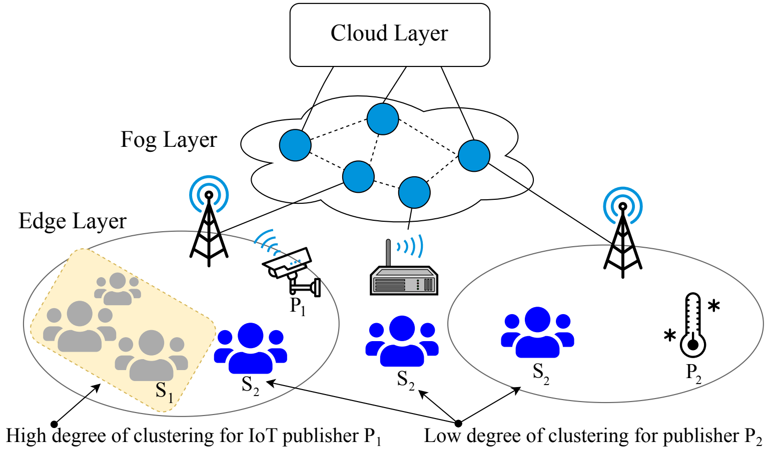
In each step, try to improve:

1. Placement of one operator
2. Placement of one broker
3. Operator to broker association

Tabu:

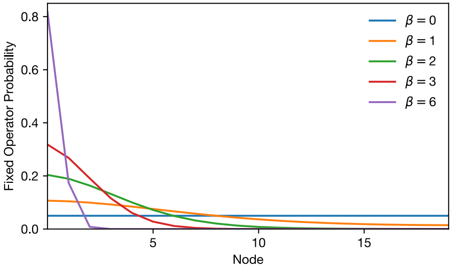
- Short term memory

Degrees of Clustering in IoT



Von Mises Distribution models Clustering

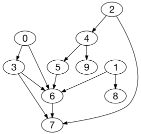
- Properties: uniform for $\beta = 0$, approaches the positive half of a normal distribution
- Nodes numbered according to delay to pivot node
- Fixed operators follow modified von Mises function



System-level Simulations

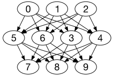
Application graphs:

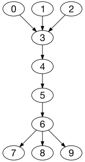
- Random, fanout, sequence



Fixed, but random, network topology:

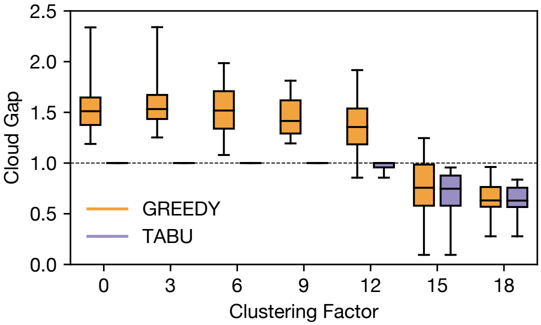
- Edge, fog, cloud
- Realistic delays and bandwidth (public route servers)





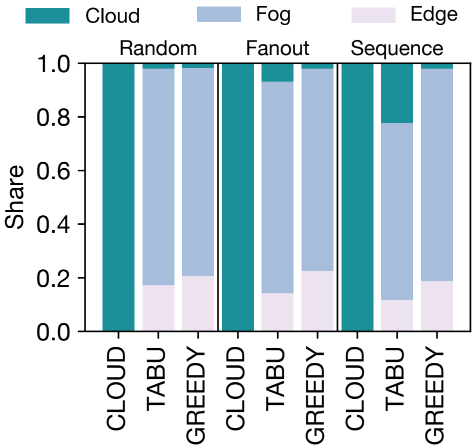
Metric: cloud gap

Impact of Clustering on Cloud Gap (Random)



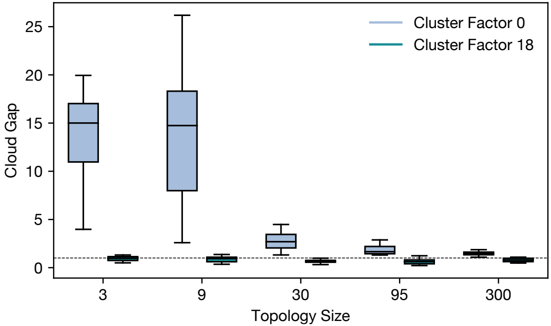
- clustering factor 0–9: cloud deployment best
- clustering factor 12: transition
- clustering factor 15+: improvement by greedy/tabu
- enables joint heuristic based on clustering factor

Placement of Operators (Clustering Factor 18)



- Greedy places most (approx. 80%) operators in the fog
- Tabu improves by putting less operator on edge & mostly more in cloud
- hard to give easy ”rules of thumb”

Impact of Network Size (Greedy)



- Difference between greedy and cloud more pronounced in smaller topologies
- Observations hold true for all the sizes considered

Conclusion

Contributions:

- We proposed two heuristics for joint operator & broker placement
- We proposed modified von Mises distributions for modelling clustering
- We conducted simulations to evaluate impact of clustering

Main Result:

- Increasing clustering leads to placement towards the edge

Future Work:

- Cluster-aware placement heuristic
- Adaptive/dynamic scheme