Data-Driven Edge Resource Provisioning for Inter-Dependent Microservices with Dynamic Load

Ruozhou Yu, *North Carolina State University*

Szu-Yu Lo, Fangtong Zhou, *North Carolina State University*

Guoliang Xue, *Arizona State University*

NC STATE UNIVERSITY

Outlines

Background and Motivation

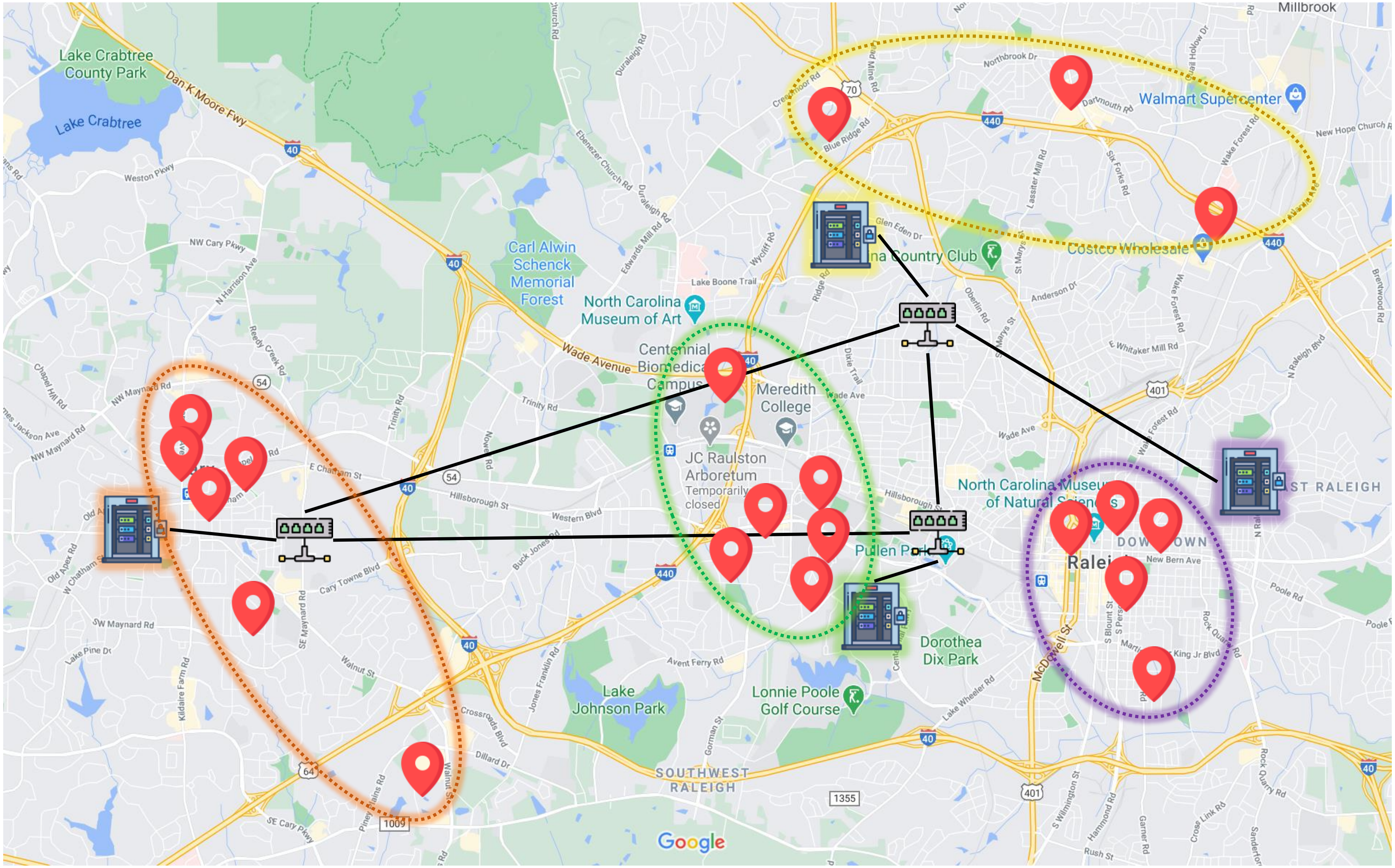
System Modeling

Solution Design

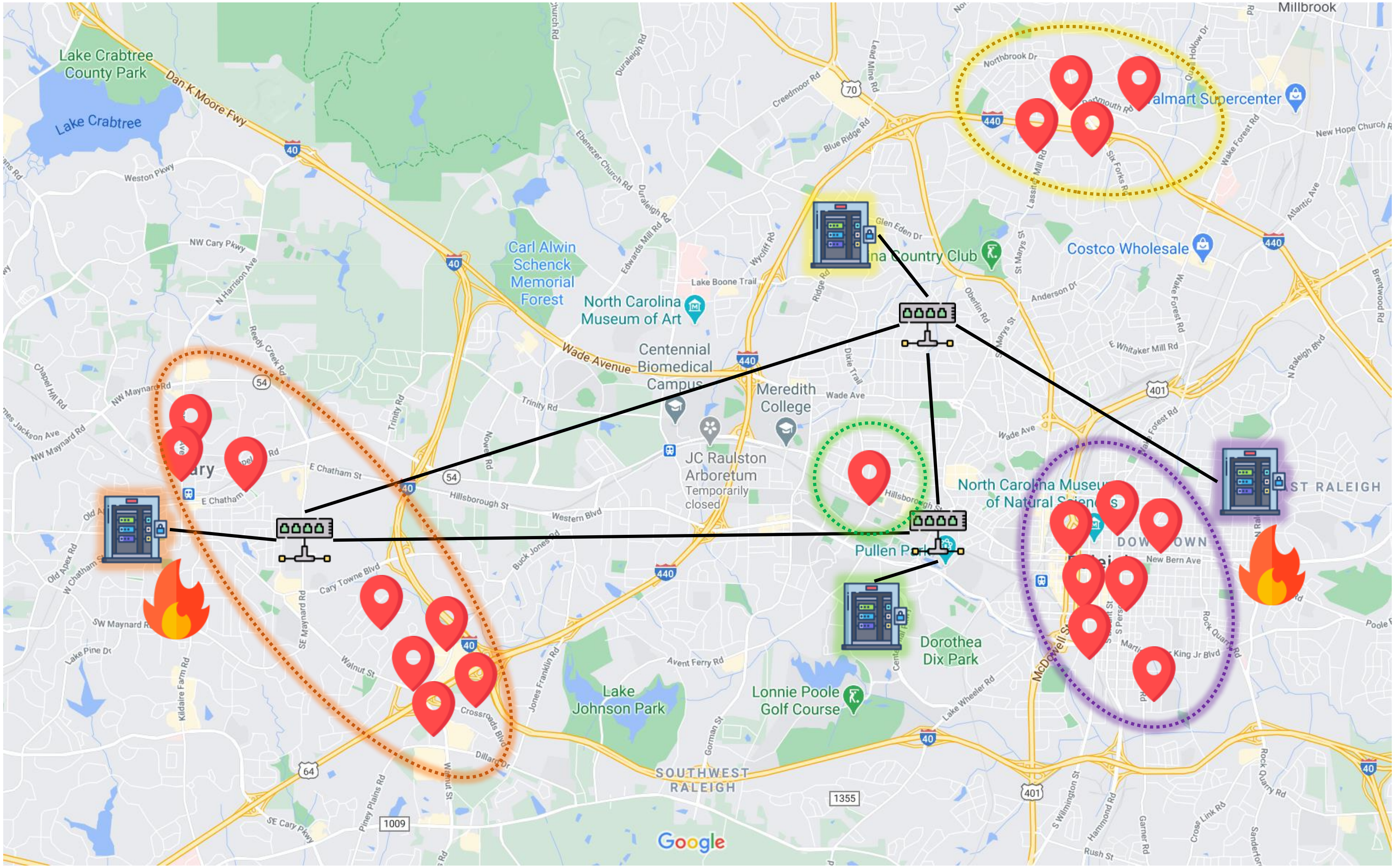
Performance Evaluation

Discussions, Future Work and Conclusions

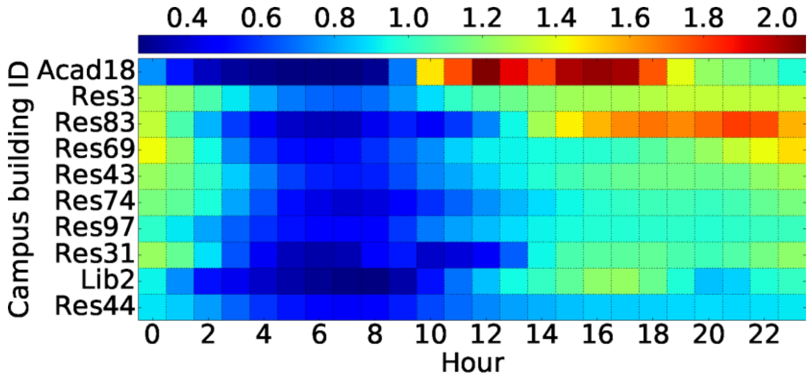
Geo-Distributed Services & Edge Computing



Time-Varying Demands in Geo-Distributed Apps

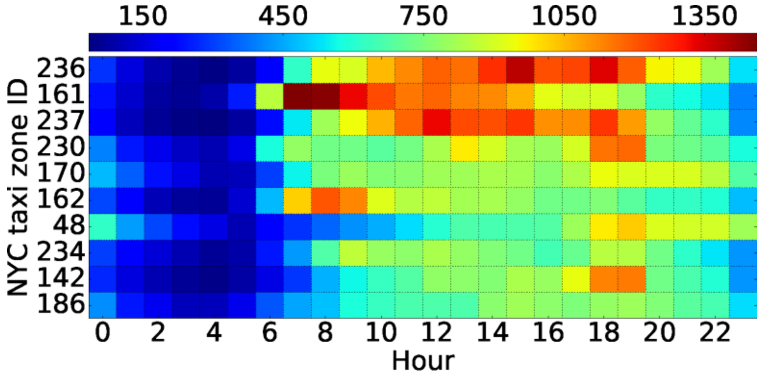


Patterns in Real-world Datasets [1]



Dartmouth College Wireless APs

- Top 10 APs with highest loads
- Load: avg. # devices / hour
- Averaged over a year (9/2002-9/2003)



NYC Yellow Taxi 2018

- Top 10 zones w/ most drop-offs
- Load: avg. # passenger drop-offs
- Averaged over a year

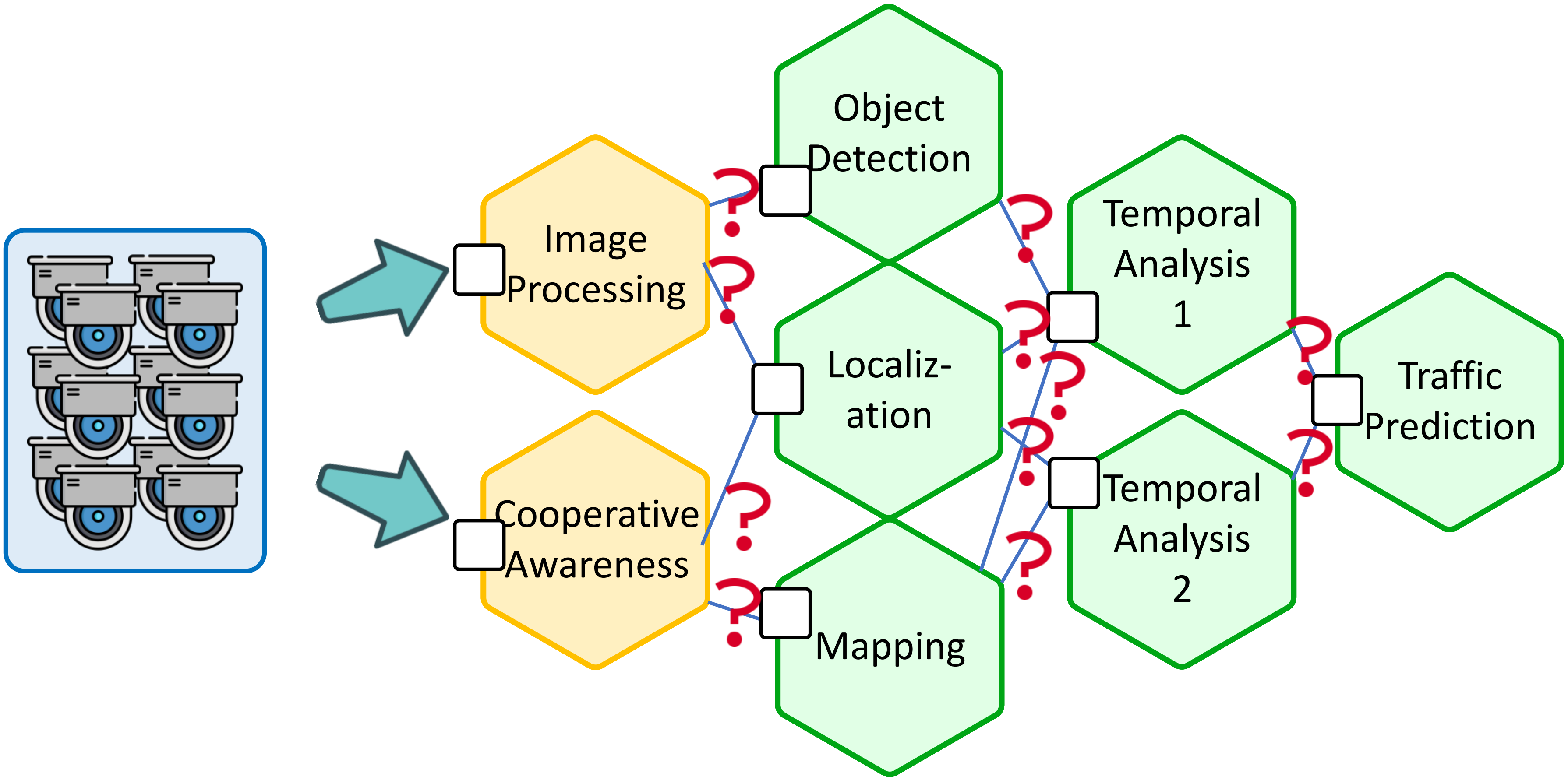
Observation 1: Non-i.i.d. demand distributions across time & locations.

Observation 2: Repeating / seasonal patterns in temporal domain.

[1] **Yu, Ruozhou**, Guoliang Xue, Yinxin Wan, Jian Tang, Dejun Yang, and Yusheng Ji. "Robust resource provisioning in time-varying edge networks." In *Proceedings of the Twenty-First International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, pp. 21-30. 2020.

The Microservice Load Balancing Problem [2]

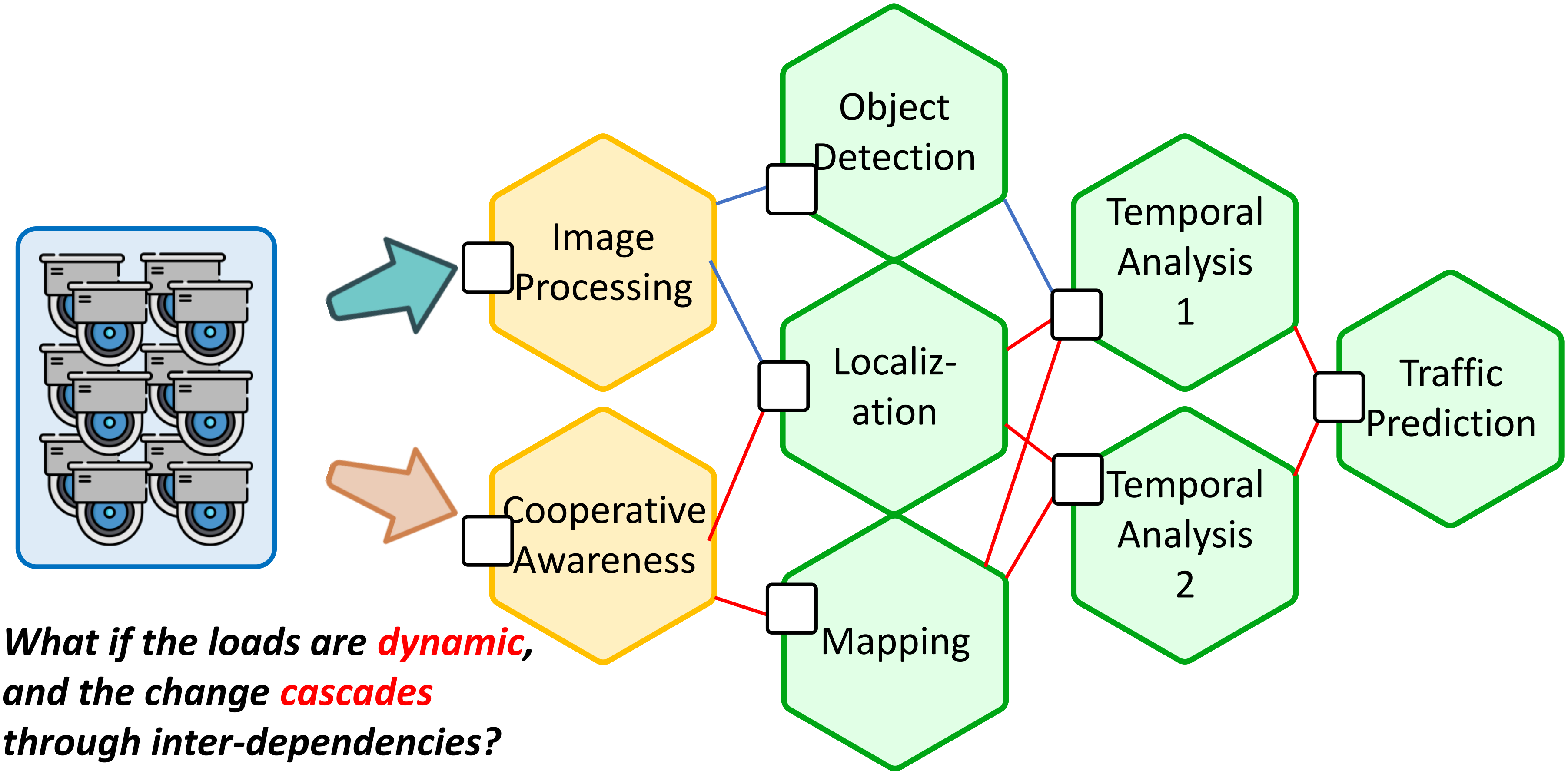
- Edge-based microservices can be easily **saturated**.
- **Challenge: interdependent** microservices.



[2] **Yu, Ruozhou**, Vishnu Teja Kilari, Guoliang Xue, and Dejun Yang. "Load balancing for interdependent IoT microservices." In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 298-306. IEEE, 2019.

The Microservice Load Balancing Problem [2]

- Edge-based microservices can be easily **saturated**.
- **Challenge: interdependent** microservices.



What if the loads are dynamic, and the change cascades through inter-dependencies?

[2] **Yu, Ruozhou**, Vishnu Teja Kilari, Guoliang Xue, and Dejun Yang. "Load balancing for interdependent IoT microservices." In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 298-306. IEEE, 2019.

Methodology Overview

Inputs:

App Structure & Demands
Inter-dependency, dynamics

Edge Network
Topology, geo-dist. services

System-wide Optimization:

Abstract System Model

- Basic provisioning formulation
- System risk model (VaR, CVaR)
- Stochastic constraints

Data-Driven Approach

- Sample average approximation
- Static/time-varying provisioning

Outputs:

Robust Provisioning Solution
Provisioning and load balancing under tail load distribution

Outlines

Background and Motivation

System Modeling

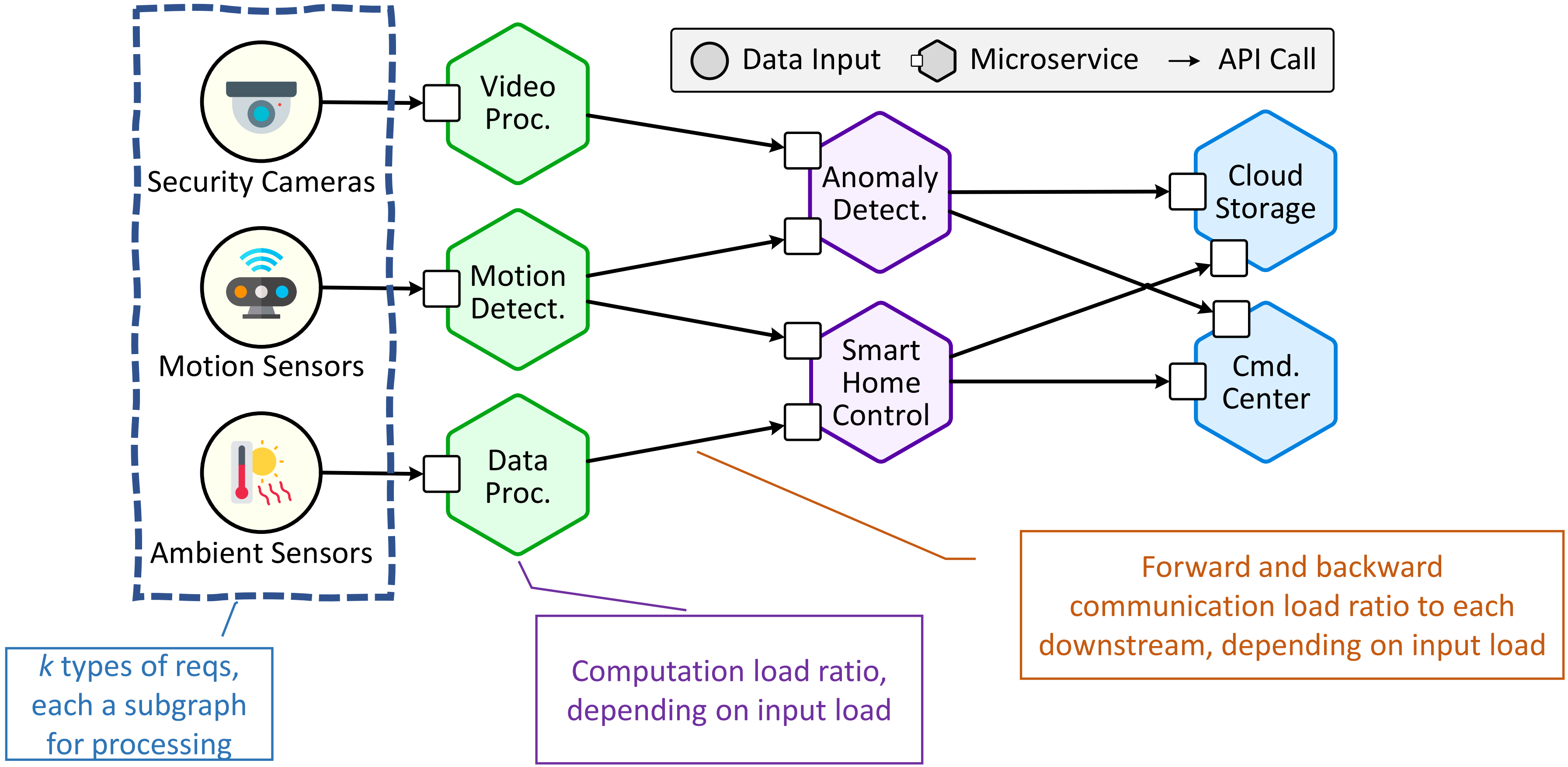
Solution Design

Performance Evaluation

Discussions, Future Work and Conclusions

Application with Interdependent Microservices

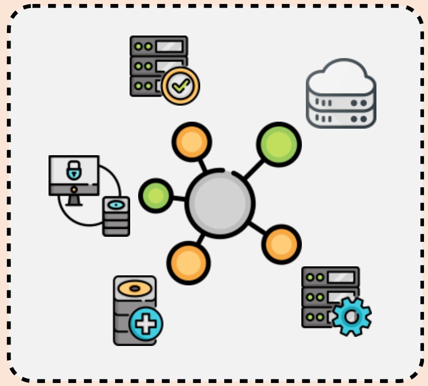
- General DAG-based application graph (App-Graph).
  - *Captures complex interdependencies, unlike existing line graph-based models.*



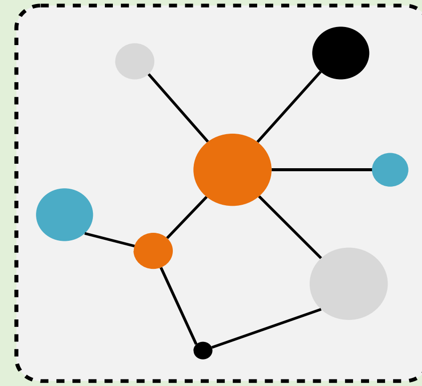
Edge Network: A General Model

- **Challenge:** heterogeneous network environments

Wireless RANs:
- Geo-distributed
- Limited capacity
- Interference



Edge Network:
- Complex topo
- Distributed
- Dynamic load



Backbones:
- Large-scale
- High latency
- ISP policies

- **Model:** general directed graph G=(N, L), with edge nodes H and APs A
  - Weights: link bandwidth, edge node capacity, *deployed microservices instances*

Objective and Overall Formulation

- **Goal:** routing & load allocation to minimize max node/link load

$$\min_{f,\psi} \quad \psi \tag{1}$$

***Obj:** minimize global load factor*

$$\sum_{k\in[K]}\sum_{a\in A}\sum_{e\in E_k}\left(\rho^k_{e,1}\cdot f^k_a(e,l)+\rho^k_{e,2}\cdot f^k_a(e,l'_l)\right)\le b_l\cdot\psi,\ \forall l\in L; \tag{2}$$

Link bw & load

$$\sum_{k\in[K]}\sum_{a\in A}\sum_{v\in V_k\cap V_h}\rho^k_v\cdot f^k_a(v,h)\le c_h\cdot\psi,\ \forall h\in H. \tag{3}$$

Node cap & load

$$\sum\nolimits_{l\in L_{\text{out}}(n)} f^k_a(e,l)-\sum\nolimits_{l\in L_{\text{in}}(n)} f^k_a(e,l) = \mathbf{1}_{\{n\in H_u\}}\cdot f^k_a(u,n)-\mathbf{1}_{\{n\in H_v\}}\cdot f^k_a(v,n), \quad \forall k\in[K], a\in A, n\in N, e=(u,v)\in E_k, \tag{4}$$

Inter-dependencies

$$f^k_a(v_k,n)\ge\delta^k_a,\quad \forall k\in[K], n=a\in A. \tag{5}$$

Geo-distributed demands of each type

- But $\{\delta^k_a\}$ are random and time-varying…

Outlines

Background and Motivation

System Modeling

Solution Design

Performance Evaluation

Discussions, Future Work and Conclusions

SO and CVaR

- **Stochastic Optimization (SO)**: optimize a function in presence of randomness (random objective and/or constraints)
  - Traditional approach: expectation optimization / constraints

    $\min_{\mathcal{X}\in\mathcal{F}} \mathbb{E}[\, R\,]$ or $A\mathcal{X} \geq \mathbb{E}[\, R\,]$
  - **Issue**: unbounded risk in rare but unfortunate scenarios
    - E.g., abnormal demands due to public events, rare large-scale failures, …
  - How to model these *unfortunate scenarios*?
  - Value-at-Risk (VaR) and Conditional-Value-at-Risk (CVaR):
    - Widely used in economics and finance
    - $\mathrm{VaR}_\alpha(\mathrm{R})$ = min { c ∈ $\mathbb{R}$ | R does not exceed c with at least α prob. }
    - $\mathrm{CVaR}_\alpha(\mathrm{R}) = \mathbb{E}[\, \mathrm{R} \mid \mathrm{R} \geq \mathrm{VaR}_\alpha(\mathrm{R})\,]$
      - Expectation of R in the worst (1-α) scenarios
  - **Our approach**: optimize with CVaR constraints

$$f_a^k(v_k, n) \geq \mathrm{CVaR}_\alpha(\delta_a^k),\ \forall k \in [K], n = a \in A.$$

Transformation and Data-Driven Approach

- **Challenge 1:** CVaR not written in closed-form
- **Technique:** LP transformation by Rockafella & Uryasev

$$\mathrm{CVaR}_\alpha(\mathbf{R}) = \min_r \left\{ r + \frac{1}{1-\alpha}\mathbb{E}[(\mathbf{R}-r)^+] \right\}, \qquad (9)$$

  - A convex optimization problem given α.

- **Challenge 2:** unknown distributions to random variables
- **Technique:** Sample Average Approximation (SAA)
  - *I.i.d. Samples:* observed demand data in historical periods.
  - $\{\delta_a^k\}$ expanded to $\{\delta_a^{k,i}\}$ for $i = 1 \ldots N$ samples.

$$\mathrm{CVaR}_\alpha(\delta_a^k) \approx \min_r \left\{ r + \frac{1}{1-\alpha}\frac{1}{N}\sum_{i=1}^{N}(\tilde{\delta}_a^{k,i} - r)^+ \right\}. \quad (10)$$

Outlines

Background and Motivation

System Modeling

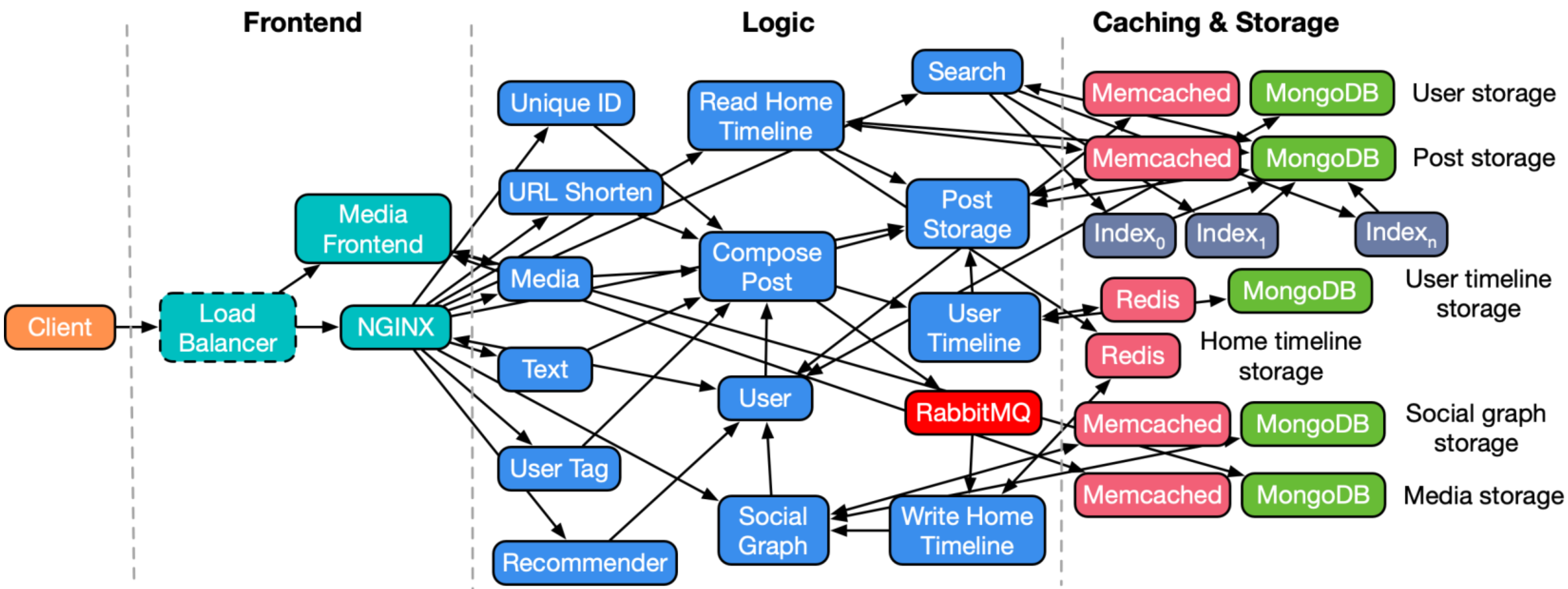
Solution Design

Performance Evaluation

Discussions, Future Work and Conclusions

Simulation Setting - Application

- Social Network (SN) from DeathStarBench [3]



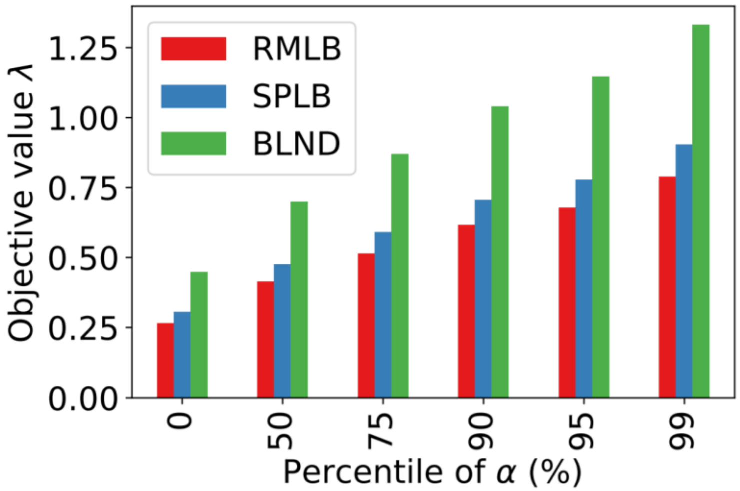
- 23 microservices, 3 types of workloads (compose-post, read-home-timeline, read-user-timeline) profiled.
- Implemented and profiled for actual communication load (# bytes); computation load/capacity synthesized based on communications.

[3] Gan, Yu, Yanqi Zhang, Dailun Cheng, Ankitha Shetty, Priyal Rathi, Nayan Katarki, Ariana Bruno et al. "An open-source benchmark suite for microservices and their hardware-software implications for cloud & edge systems." In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 3-18. 2019.

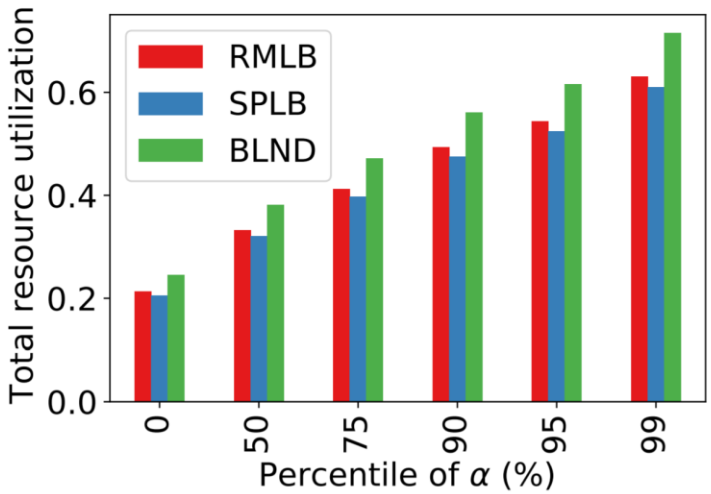
Simulation Settings – Demand & Network

- Settings
  - Dataset: NYC Yellow Taxi 2018
    - 12 months of Taxi pick-up/drop-off data (~112 million taxi trips)
    - Picked 20 most popular zones out of 262 (55% of all demands)
    - Mapped demands (drivers/passengers, pick-up/drop-off) to SN requests
    - 20% training (optimization) & 80% testing (deployment)
  - Synthetic Data
    - Random topologies: Watts-Strogatz with k = 4 and p = 0.3
    - Each microservice deployed on 20% random edge nodes
    - Network conditions: 1Gbps links, 2.5Gbps computation capacity (normalized)
  - $\alpha = 0.95$ (CVaR confidence)
  - Comparison: Shortest Path-based Heuristic, and Blind Load Balancing Heuristic

Selected Experiment Results



(a) Max load factor λ in training

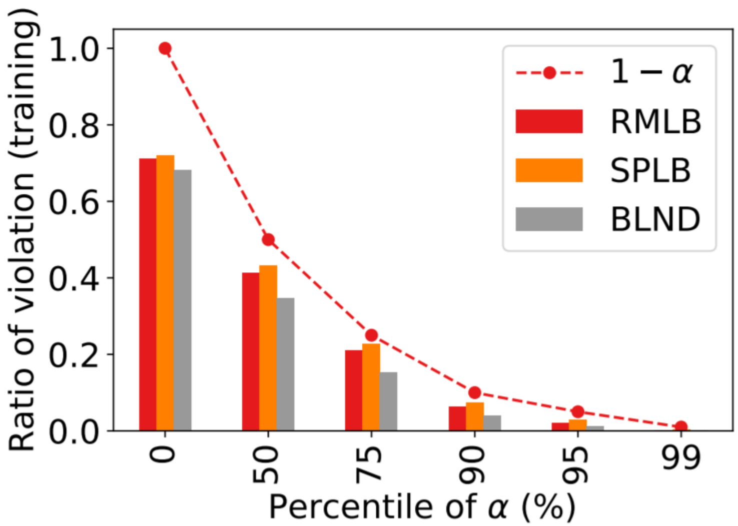


(b) Total edge resource consumption

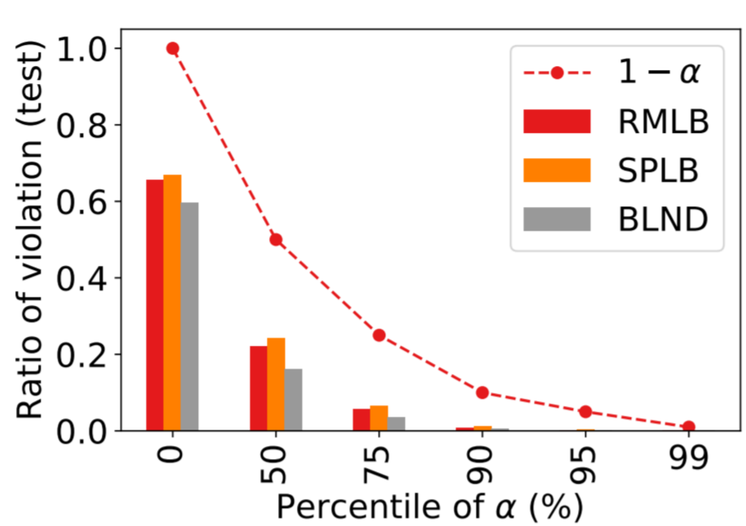
Maximum Load (Resource Provisioning)

- RMLB (our formulation) achieves best inter-dependency-aware provisioning. Other algorithms result in higher maximum load, and lower/higher total load.

Selected Experiment Results



(c) Ratio of violation (training)

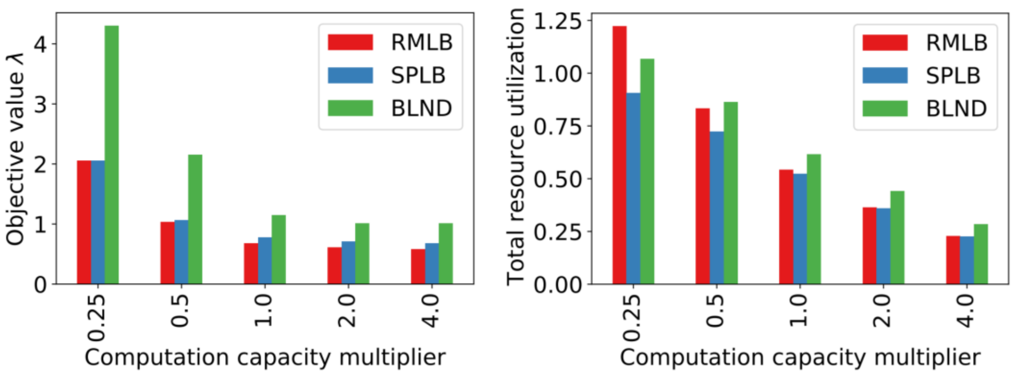


(d) Ratio of violation (testing)

Robustness (Actual Demands)

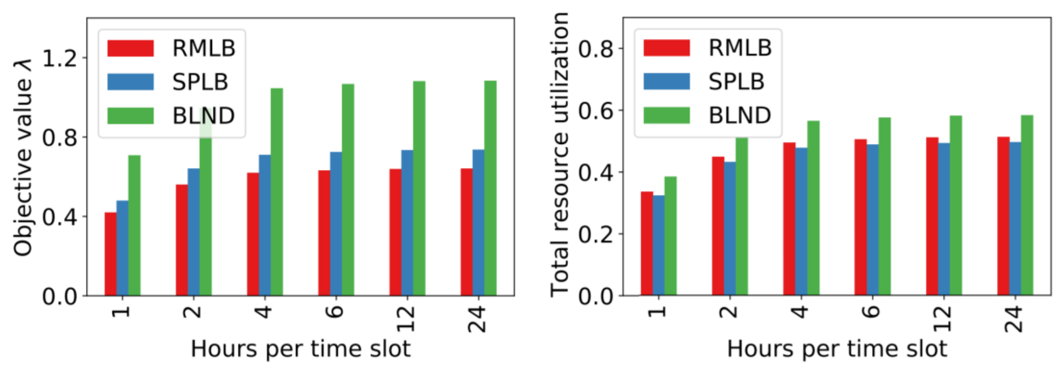
- We provisioned for the training set (left), with bounded ratio of load violation.
- In the test deployment, we observe similar (lower) ratio of load violation.
- The $(1-\alpha)$ percentile is never violated, depending on our setting of α.

More Results in Paper



(a) Max load factor λ in training (b) Total edge resource consumption

With computation vs. network bottlenecks



(a) Max load factor λ in training (b) Total edge resource consumption

With/without time-varying provisioning

Outlines

Background and Motivation

System Modeling

Solution Design

Performance Evaluation

Discussions, Future Work and Conclusions

Other Perspectives, Conclusions

- So far, we've talked about
  - Microservice inter-dependencies
    +
    dynamic demands

First-attempt modeling & solving

- What could be improved
  - QoS constraints: latency, throughput, reliability
  - Mixture spatial-temporal distributions
  - Distribution-aware formulations
  - Queueing-based risk analysis
  - Improved optimization methods
  - Improved statistical & learning-based methods

Modeling Perspective

Stochastic Perspective

Algorithmic Perspective

- **Conclusions:** app-aware, robust computing & networking.

Thank you very much!

Q&A?