UNIVERSITY OF TEXAS ARLINGTON

Towards Efficient Microservices Management Through Opportunistic Resource Reduction

Md Rajib Hossen, Mohammad A. Islam

Dept. of Computer Science and Engineering
The University of Texas at Arlington

Email: mdrajib.hossen@mavs.uta.edu

RiDL
Rigorous Design Lab

IARIA

RiDL
Rigorous Design Lab

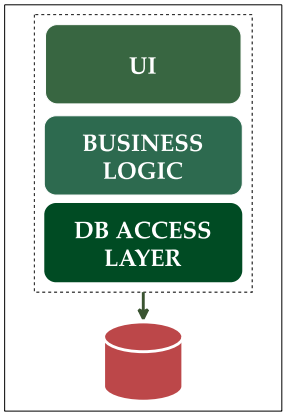
Md Rajib Hossen

I'm a PhD Candidate in Computer Science Department at the University of Texas at Arlington under the supervision of **Dr. Mohammad Islam**.

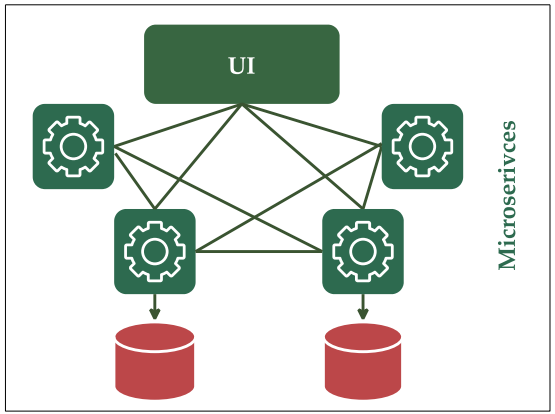
My primary research interests are Microservices, Distributed Systems, and Machine Learning for Systems. Currently, I'm working on finding efficient resource allocation for microservices, building, and managing large-scale microservice applications.

Microservices

- Set of loosely coupled services
- Deployed independently
- Communicate via API/RPC
- Easily deployable, highly scalable, easy to update components than monolithic



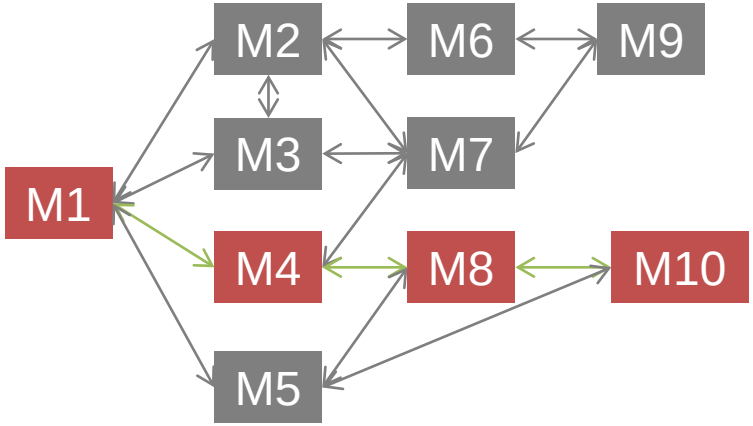
Monolithic



Microserivce

Challenges

- Complex Communications – high coupling
- **Resource managements introduces new challenges**
- Current cloud solutions fail to consider the complexities



Existing Works

- Threshold based approaches fail to capture the inter-dependencies[1, 2]
- ML based systems require offline training data and intentional SLO violations for boundary conditions[3,4,5]

1. A. Kwan, J. Wong, H.-A. Jacobsen, and V. Muthusamy, “Hyscale: Hybrid and network scaling of dockerized microservices in cloud data centres,” in ICDCS, 2019
2. Kubernetes Horizontal Pod Autoscaler, Vertical Pod Autoscaler
3. Y. Zhang, W. Hua, Z. Zhou, G. E. Suh, and C. Delimitrou, “Sinan: Ml-based and qos-aware resource management for cloud microservices,” in ASPLOS, 2021
4. H. Qiu, S. S. Banerjee, S. Jha, Z. T. Kalbarczyk, and R. K. Iyer, “FIRM: An intelligent fine-grained resource management framework for slo-oriented microservices,” in OSDI, 2020
5. G. Yu, P. Chen, and Z. Zheng, “Microscaler: Automatic scaling for microservices with an online learning approach,” ICWS 2019

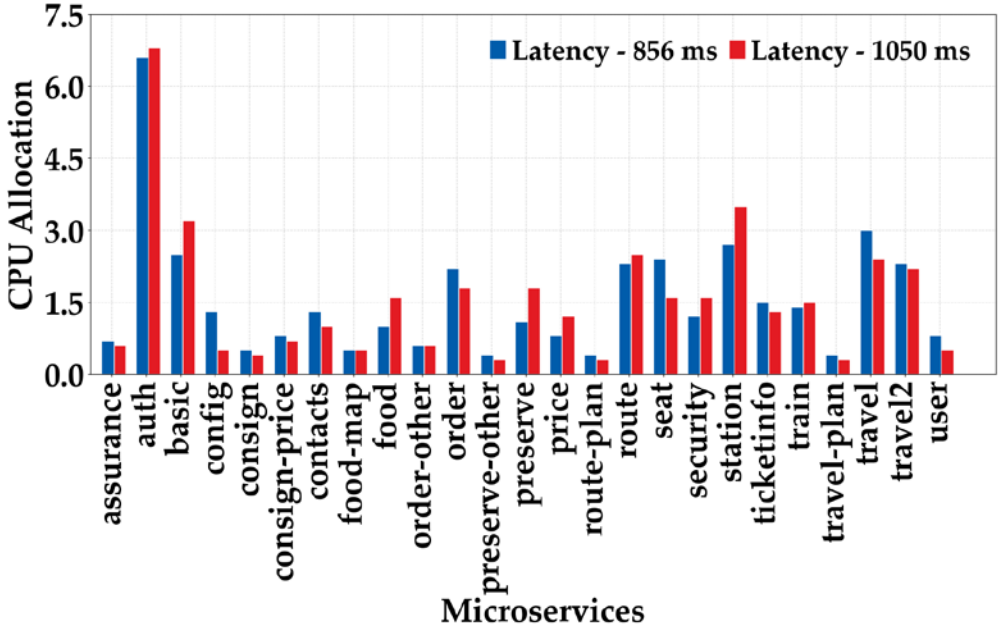
Proposed Solution

- Develop a light-weight interactive resource manager for microservices
- Our Goal is to minimize total resource allocation where we -
  - Do not require offline training data
  - Do not violate SLO during learning

Challenges

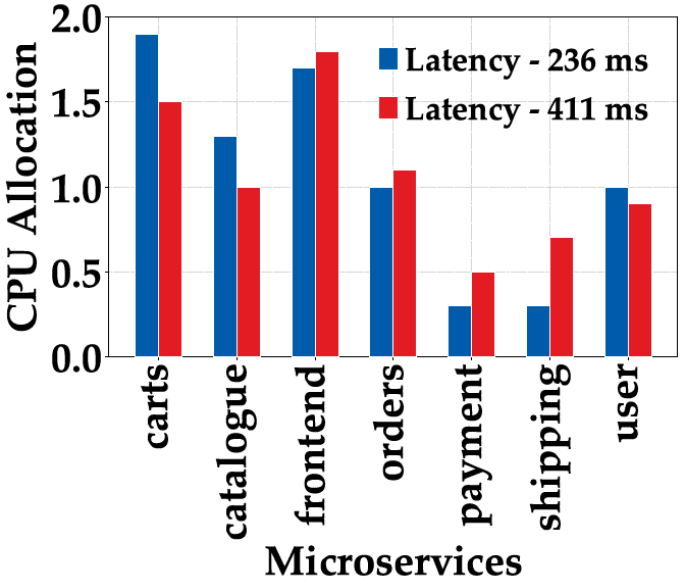
- Estimation of response time for microservices is hard in practice
- Microservices behaves dynamically based on workload and assigned resources

Challenges

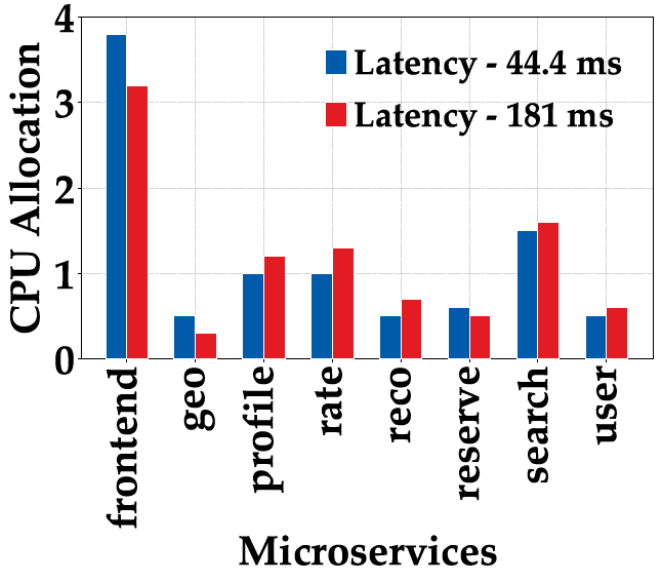


Train Ticket (SLO – 900ms)

Challenges



Sock Shop (SLO-250 ms)



Hotel Reservation (SLO-50 ms)

Opportunistic Resource Reduction

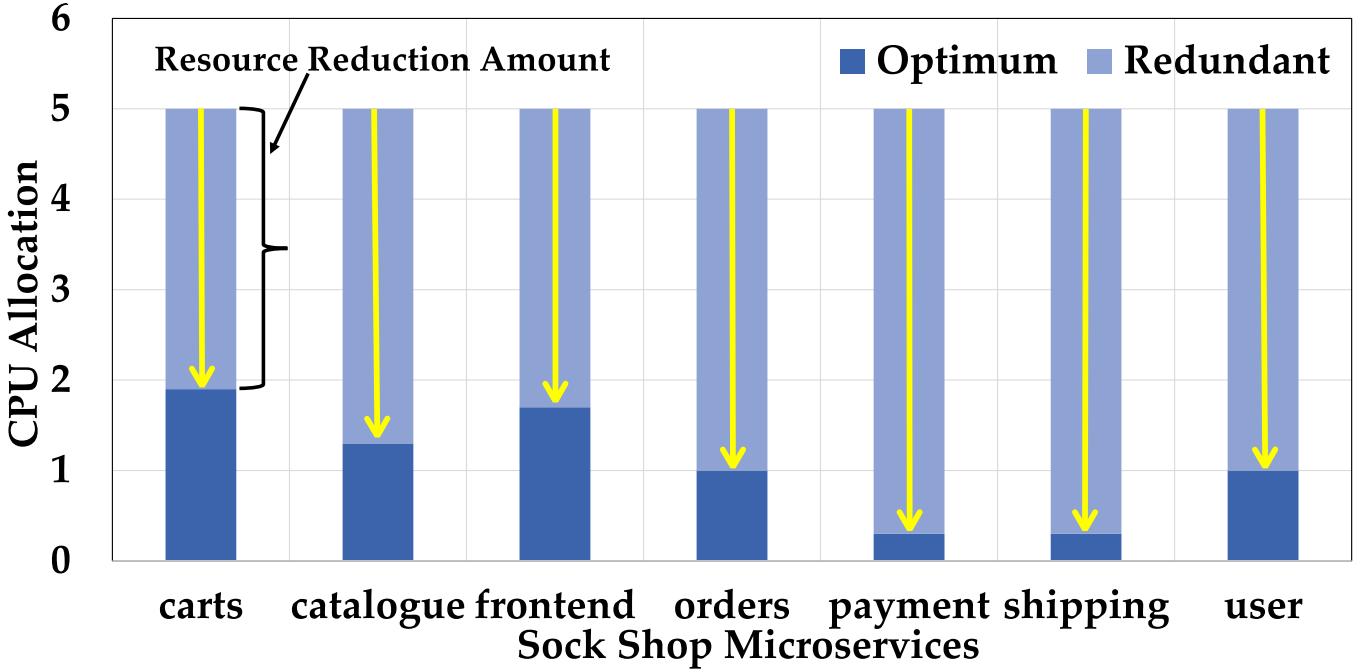
- Instead of estimating response time, we adopt a feedback-based approach
- Get system feedback to find resource reduction opportunity
- Reduce resources in the next time slot

- Need to be careful as resource reduction may violate SLO

Opportunistic Resource Reduction

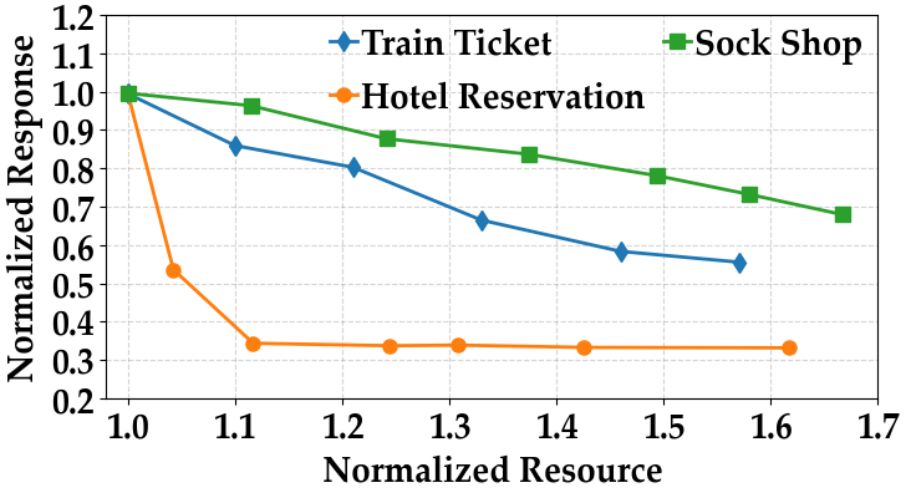
- Calculate the distance between response time and SLO
- Find out the resource reduction opportunity
- Divide the reduction in several small steps
- Execute and get feedback of resources in each time steps

Opportunistic Resource Reduction



Intuition Support

- Experiments on resource and response support our design intuition



Conclusion

- This paper is a work-in-progress towards a complete resource manager with features such as
  - No human intervention
  - No degradation of Quality of Service (QoS)
- To guarantee these features
  - Maintain per microservices utilization upper limits
  - Reduce resources only if current metrics will not violate the dynamic limits
  - Conservative in resource reduction
  - Dynamic workload group to account for workload changes

RiDL
Rigorous Design Lab

Thank You!

For questions, please reach out to
mdrajib.hossen@mavs.uta.edu or conference threads