

Serverless Edge Computing: Current Trends, Challenges and Future Directions

Shuo Li, Shayan Jalilian

November, 2024

Cloud Computing and
Applications (ENSE-885BD)



University
of Regina

Go far, together.

Agenda

- Introduction to Serverless Edge Computing
- Research Methodology
- Taxonomy & SOTA
- 5 Papers
- Conclusion & Future Research

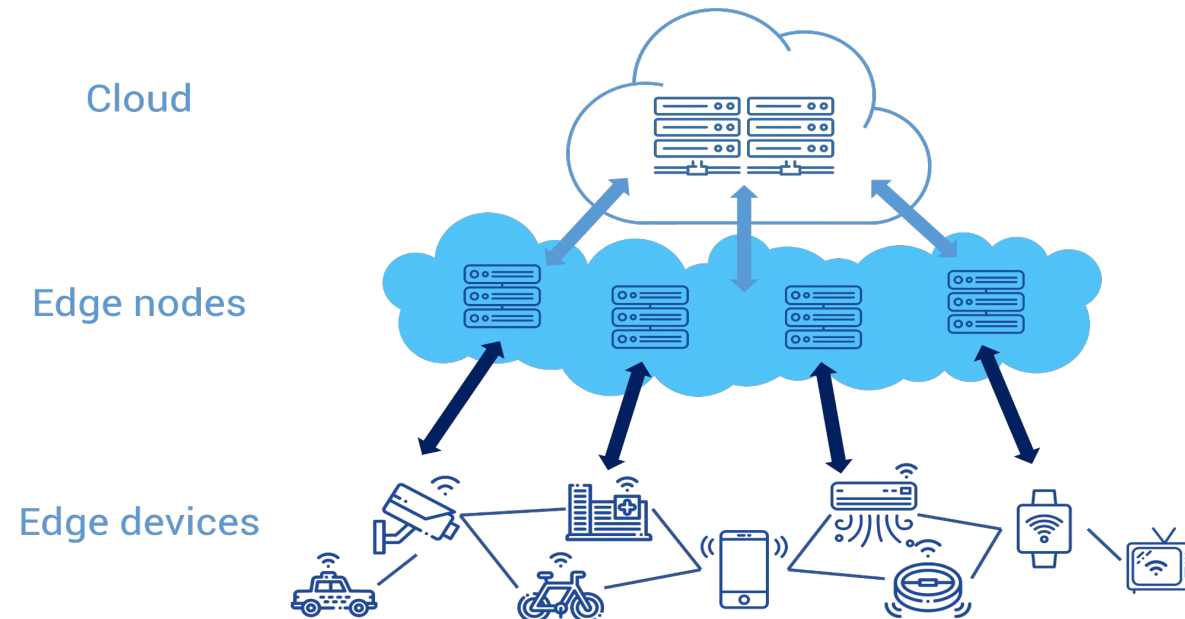
Introduction

Some Questions that this presentation will answer:

1. What is Serverless Edge Computing?
2. Why Serverless Edge Computing?
3. What are the current challenges and limitations?
4. What are the future research directions?

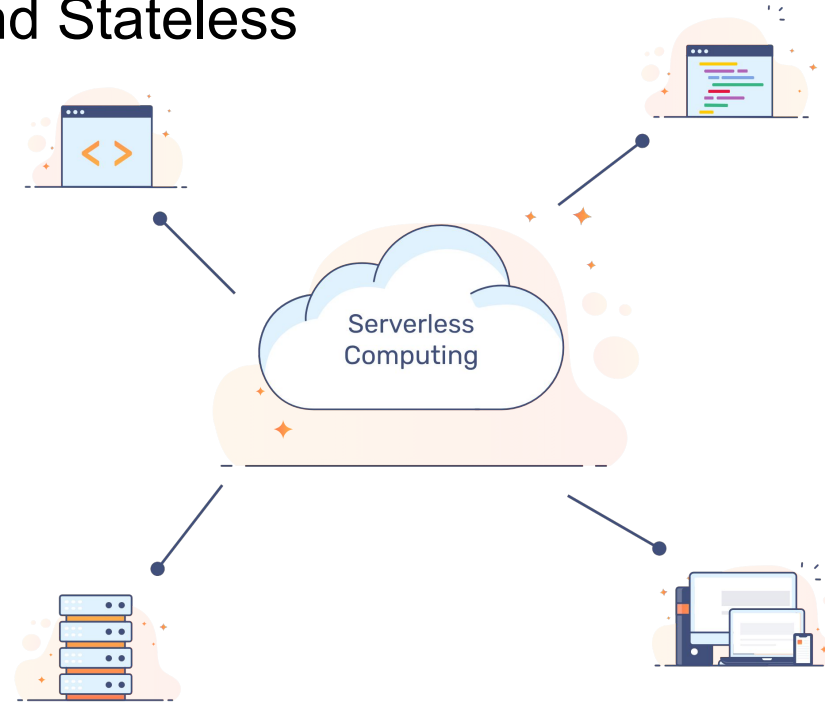
What is Edge Computing?

- Decentralized computing model
- Data processing happens near the source
- Reduces latency and improves performance for time-sensitive applications
- Main challenges are Scalability and Cost



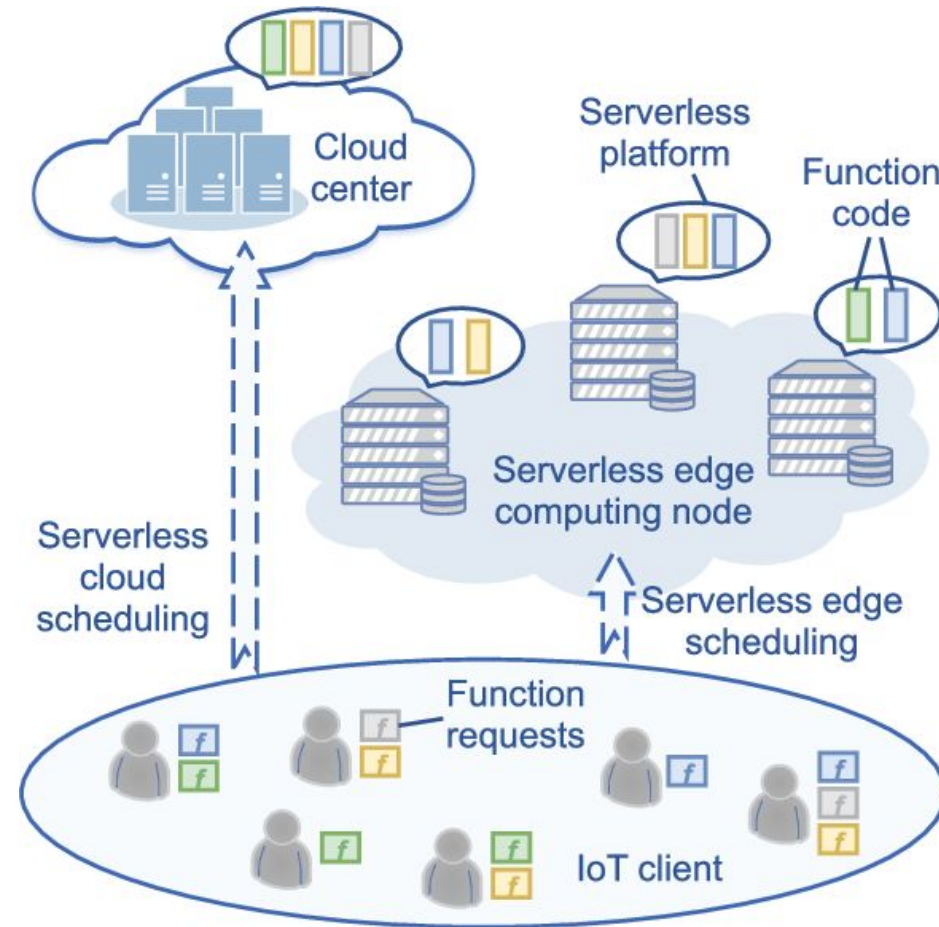
What is Serverless Computing?

- Cloud computing model (FaaS)
- Developers build and run functions without managing the infrastructure
- Cloud provider allocates resources and scales based on demand
- Main characteristics: Event-Driven and Stateless
- Main challenge: Increased Latency



Serverless Edge Computing: A Marriage of Two Worlds

- Combination of Serverless & Edge Computing
- Enables developers to deploy serverless functions at the edge and closer to the users.
- Highly Scalable (serverless), Low Latency (Edge)
- Minimal to no infrastructure management.
- Applications: IoT & Real-Time Analytics

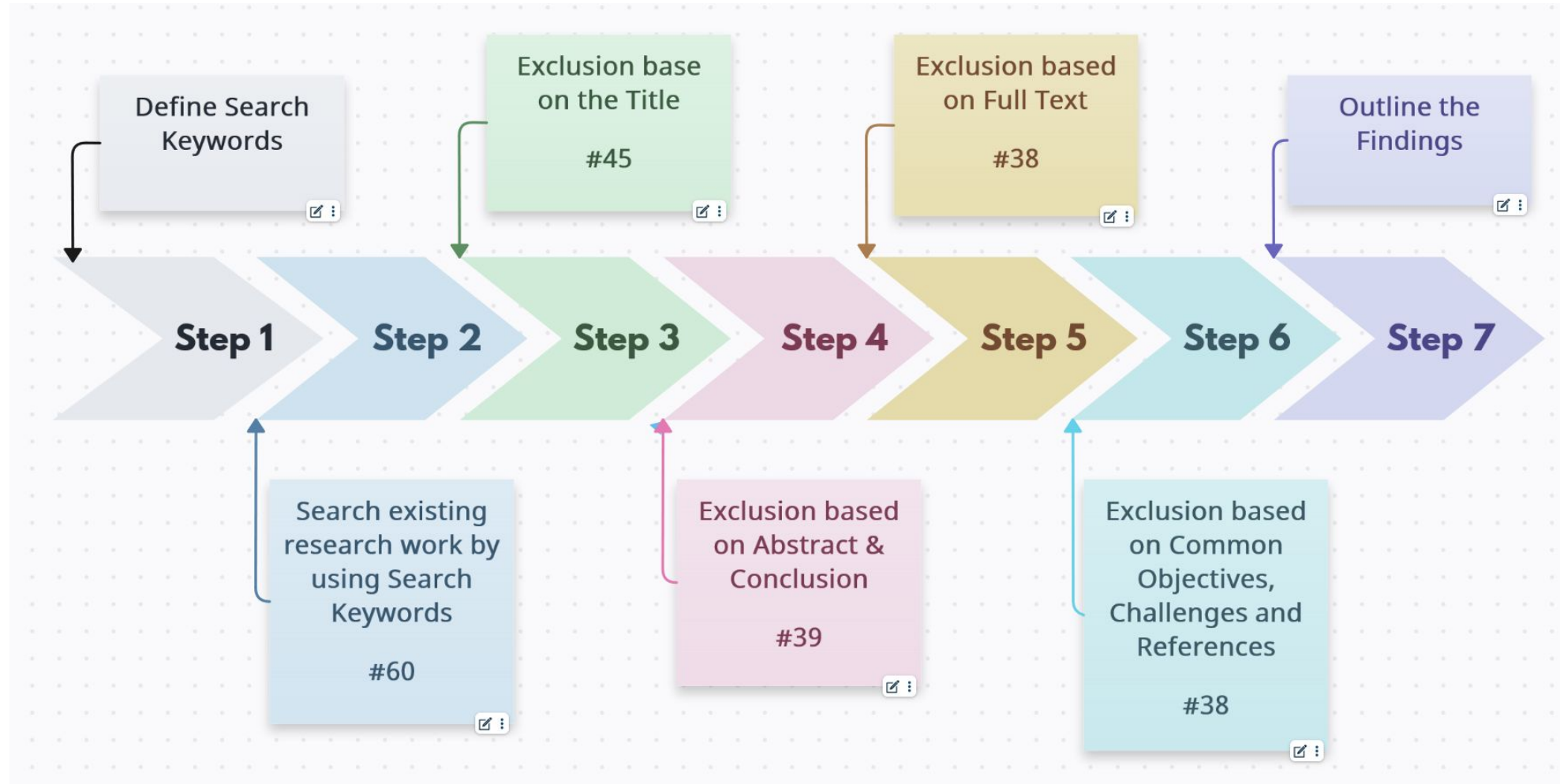


Goal of the Study

We are interested in answering the following questions:

- What are the usage and benefits of serverless edge computing (why serverless edge?)
- What are the challenges faced in adopting serverless edge?
- What are the current SOTA solutions offered for the common challenges and problems?

Review Methodology



Search Strings

Search Keywords	Period	Venue Type
Serverless Edge Computing Serverless Edge State Serverless Edge Resource Serverless Edge Security Serverless Edge Auto-Scaling Serverless Edge Load Serverless Edge Optimization Serverless Edge Application Serverless Edge Scheduling Serverless Edge Cold Start Serverless Edge Management Serverless Edge Fault	2024	Conference Journal Master & Ph.D. Thesis Pre-prints Books



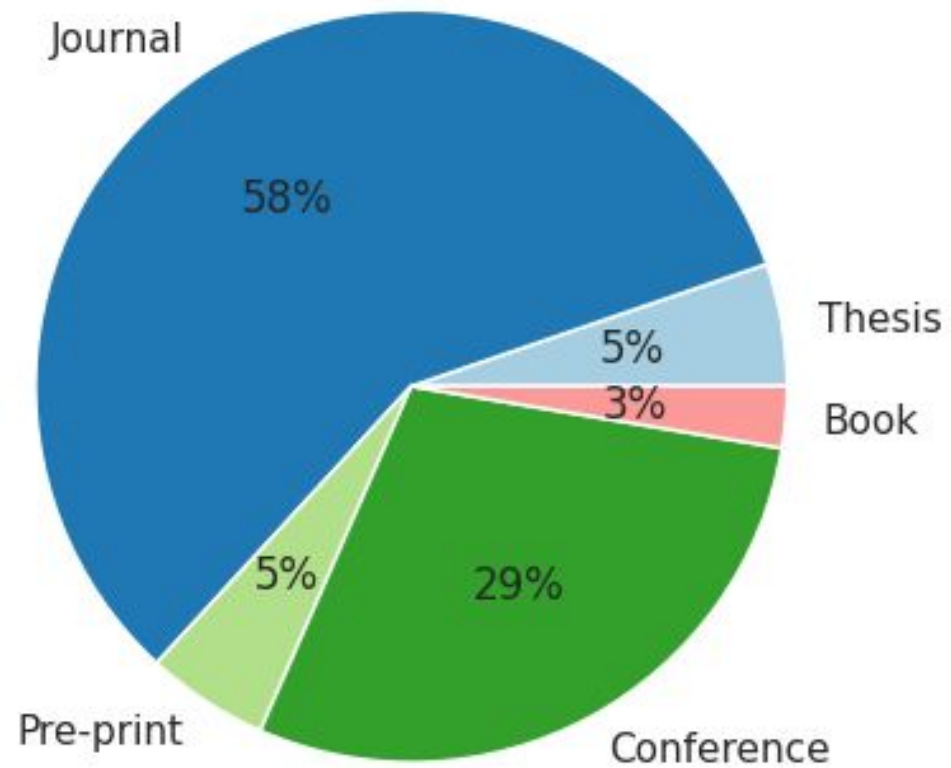
Research Questions

Category	Research Questions
Resource Management & Optimization	<ol style="list-style-type: none">1. What are the different ways to manage resources in Serverless Edge Environment?2. What are the trade-offs between energy efficiency and computational latency in serverless edge environments?3. What approaches can dynamically allocate resources in serverless edge computing to meet varying workload demands?
Security & Fault Tolerance	<ol style="list-style-type: none">1. What strategies can protect serverless edge systems from common security threats?2. What techniques can help serverless edge systems recover quickly from failures?3. What are the best ways to handle hardware or network failures in edge environments?
Applications	<ol style="list-style-type: none">1. What are some of the applications for serverless edge computing?2. What are the main challenges in designing and deploying applications for serverless edge environments?

Research Questions

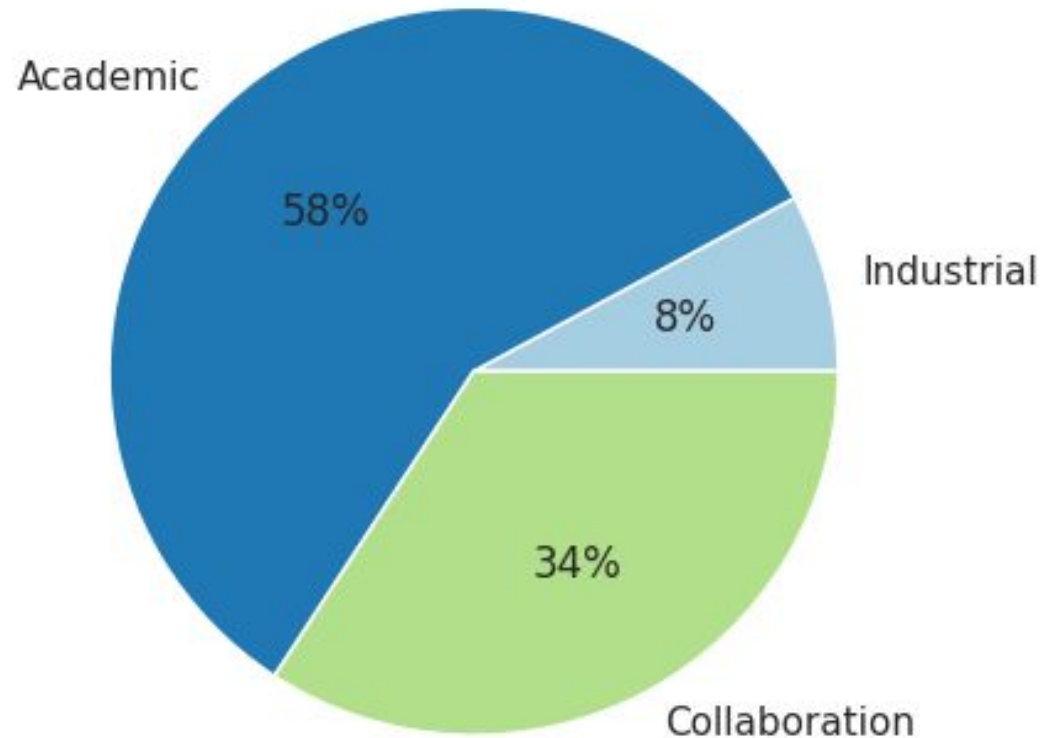
Category	Research Questions
Scheduling	<ol style="list-style-type: none">1. What techniques can minimize the impact of cold starts on serverless edge applications?2. What can be done to ensure serverless edge systems meet performance goals like SLOs and SLAs?3. What scheduling methods can reduce latency while still using resources efficiently?4. What techniques can dynamically balance the trade-offs between latency and resource efficiency in edge scheduling?5. What load balancing techniques are most effective in serverless edge environments?6. What scheduling algorithms effectively handle priority-based execution in serverless edge systems?
Stateful Management	<ol style="list-style-type: none">1. What are the best ways to manage state efficiently for functions in serverless edge environments?2. What approaches can maintain consistency in state across highly distributed edge systems?

Quantitative Results

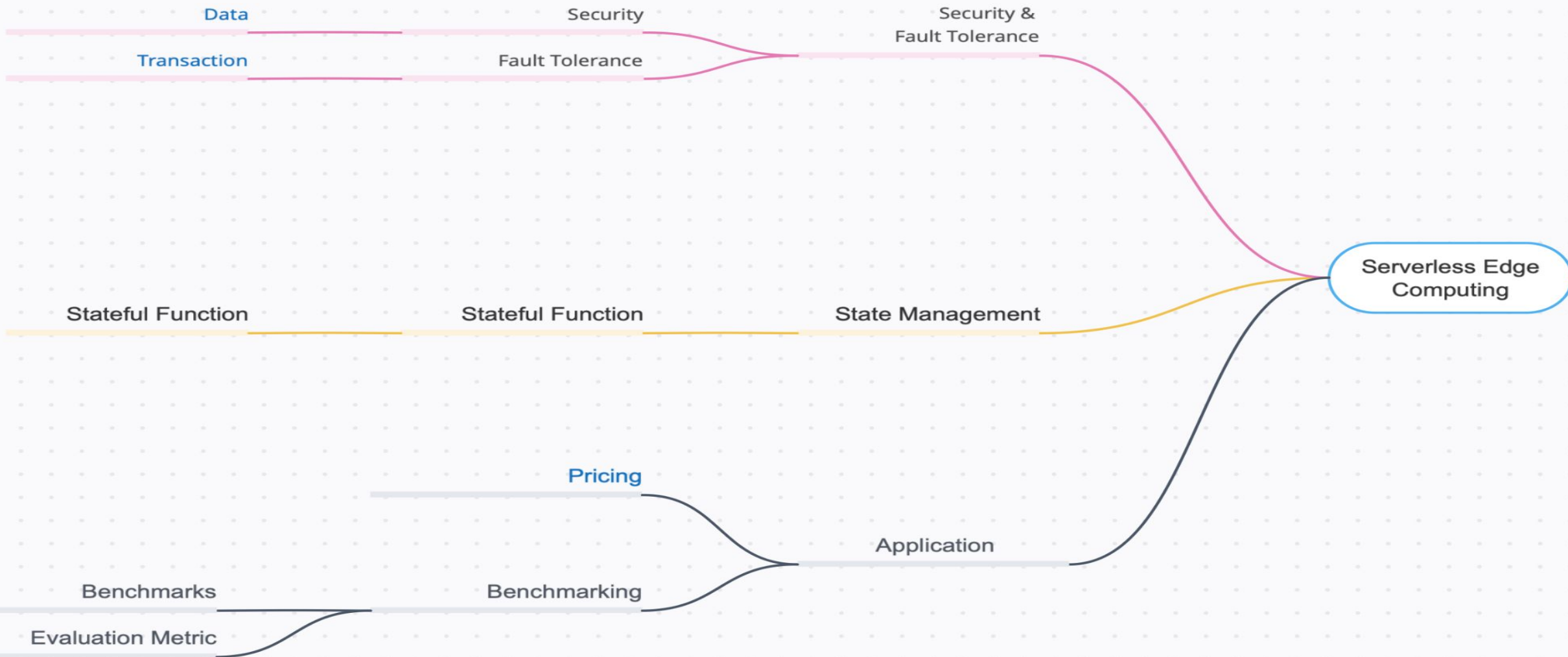


Quantitative Results

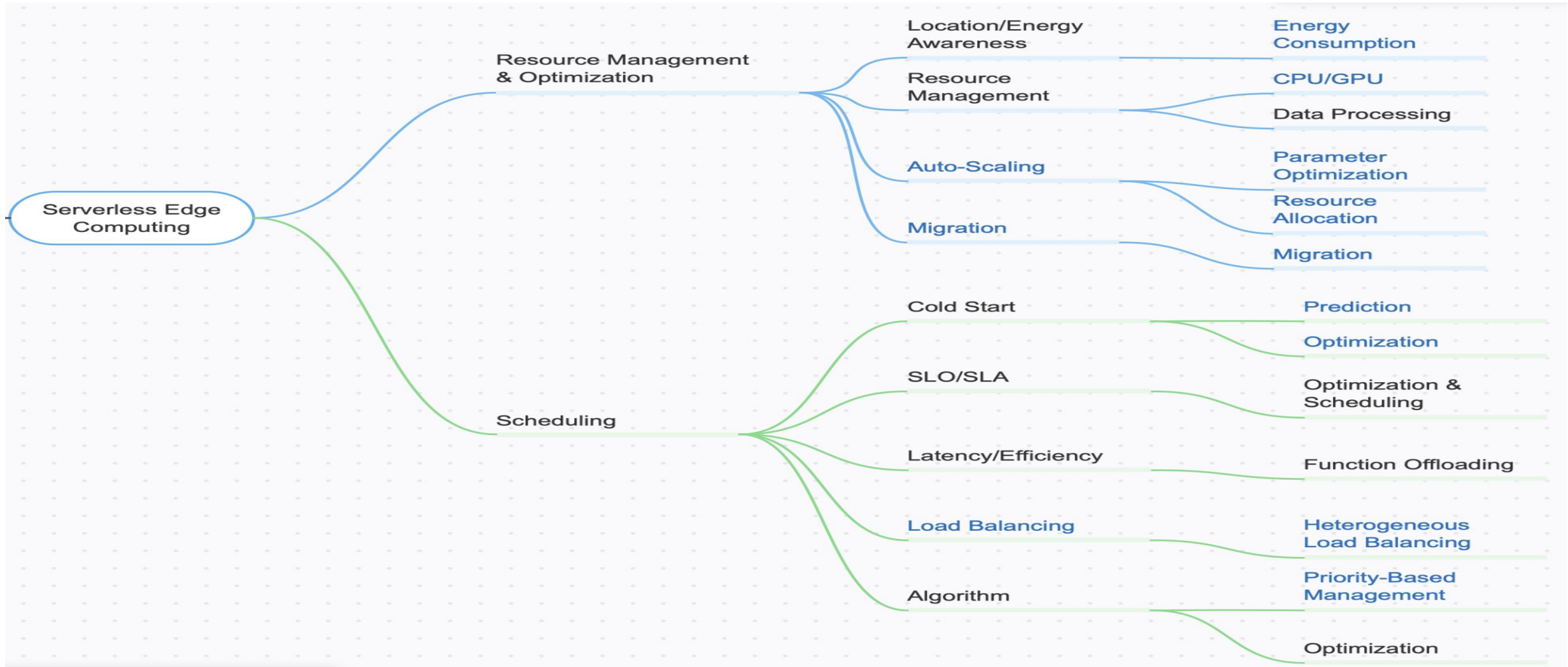
Papers by Author Affiliation



Taxonomy



Taxonomy



Load Balancing for Heterogeneous Serverless Edge Computing: A Performance-Driven and Empirical Approach [1]

Aslanpour et al.

Problem Definition

The main problem that the proposed method:

- Load Balancing of Heterogeneous Edge Nodes
- Multi-Objective Optimization: throughput, energy per task, response time, AI precision, and function cost.

This paper proposes a weighted sum method to optimize load balancing by addressing multiple objectives for heterogeneous edge nodes.

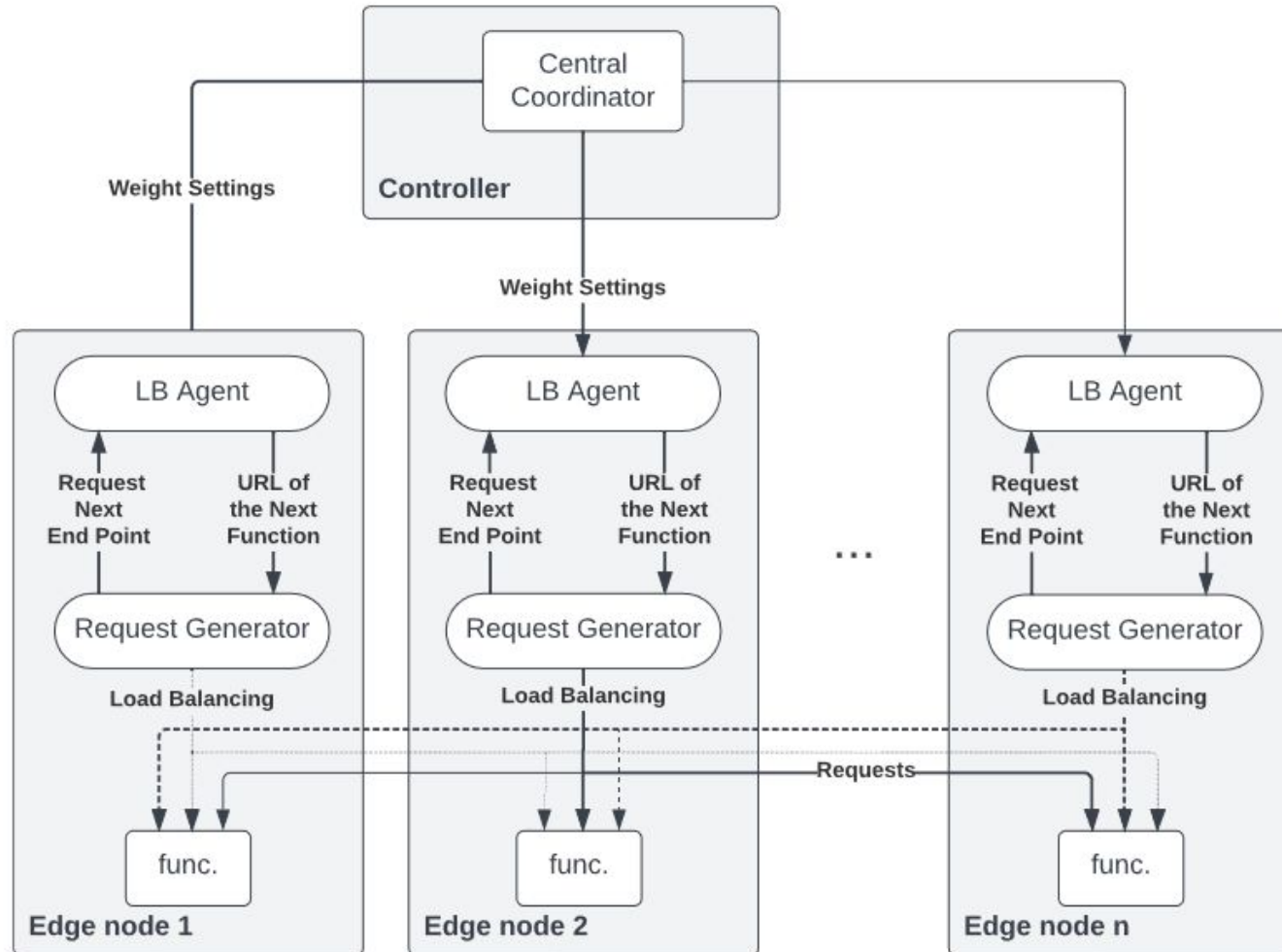


Fig. 4. Coordinated distributed load balancing.

Coordinated Distributed Load Balancing

- No centralized load balancer
- Central coordinator handles policy propagation to edge nodes
- Each node processes a load balancer agent
- Load balancer agents redirects request to edge nodes in the cluster according to policy of the central coordinator

Limitations & Future Directions

Limitations:

- Assuming unlimited energy and constant requests

Future Directions:

- Conduct research using variable energy inputs and fluctuating request arrivals

Efficient Serverless Function Scheduling in Edge Computing [2]

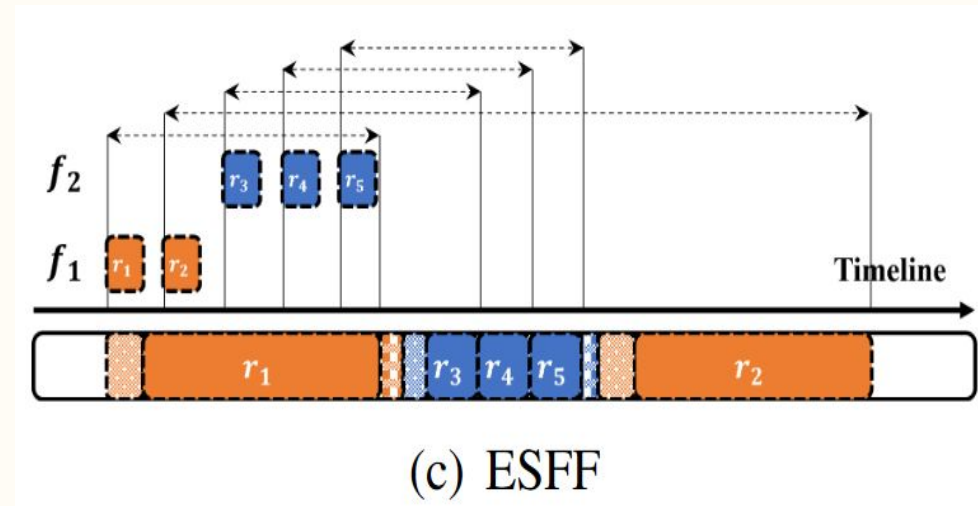
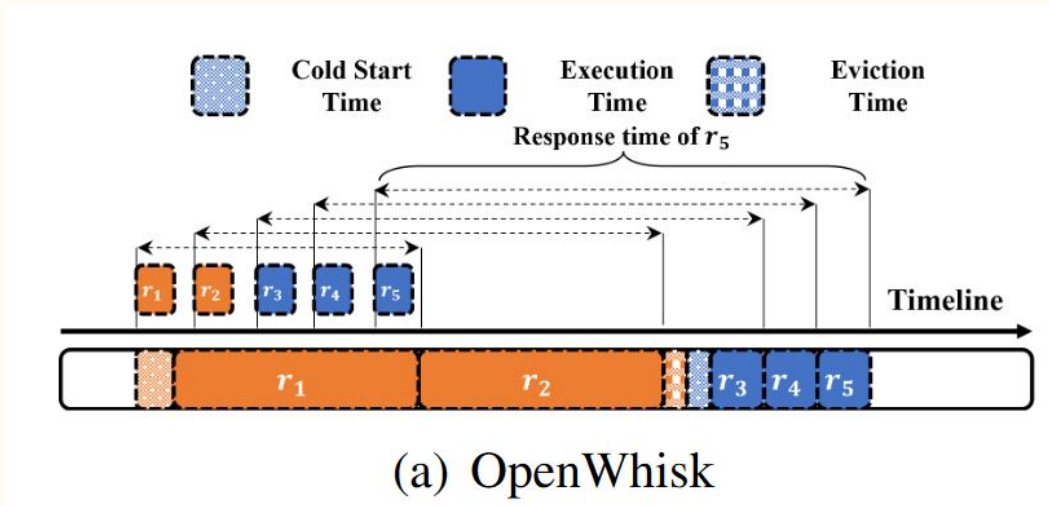
Lou et al

Problem Definition

The main problem that the proposed method:

- Cold Start Latency: Delays due to function initialization
- Request Blocking: Short requests blocked by long-running ones

This paper proposes a Enhanced Shortest Function First (ESFF) algorithm to minimize average response time for request.



Proposed Method

Uses Function Creation Policy (FCP) and Function Replacement Policy (FRP).

- Function Creation Policy (FCP): Decides whether to initialize a new function instance when a request arrives.
- Function Replacement Policy (FRP): Determines whether to replace an idle function instance when one finishes execution.

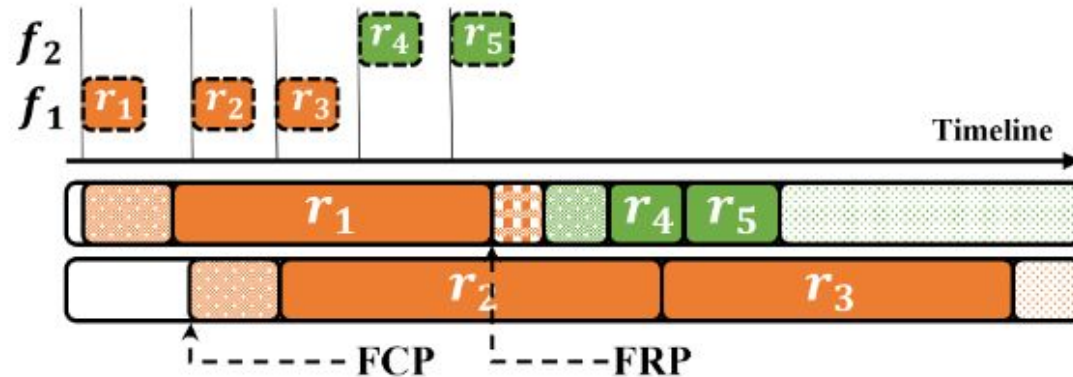


Fig. 3. The scheduling example of ESFF. FCP initializes a new function instance for the request r_2 , and replaces the function instance for request r_4 .

Limitations & Future Directions

Limitations:

- Precision in execution time measurements was affected by some inaccuracies, with some values defaulting to 1ms.

Future Directions:

- Investigate offloading long requests to powerful cloud



Toward an Edge-Friendly Distributed Object Store for Serverless Functions [3]

Chen et al



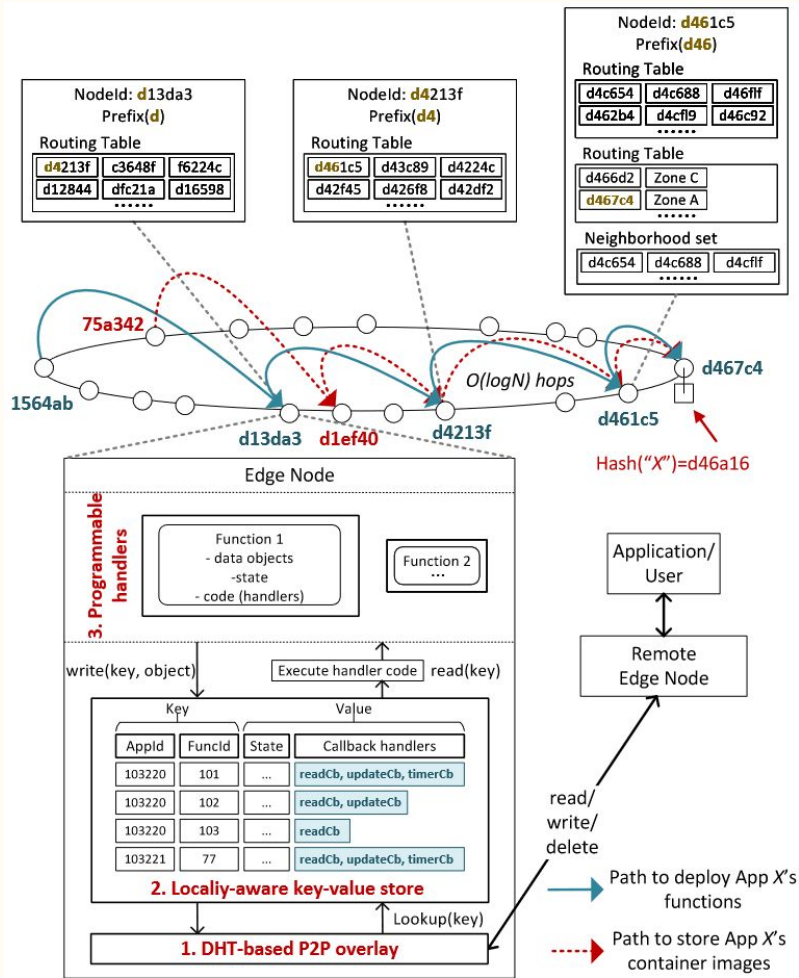
Problem Definition

The main problem that the proposed method:

- State Management
- Scalability

This paper presents **Capybara**, a distributed object store for managing state in serverless edge environments.

Proposed Method



Cappybara consists of three components:

1. Distributed Hash Table (DHT)-based P2P overlay
 - Implements the object-to-node mapping.
2. Locality-aware key-value store
 - Persistent storage of function data objects (state)
3. Programmable handler abstraction
 - Stores Operational code along with Key, Value pair
 - Operational code is structured as a set of programmable handlers which manages the states

Figure 3: The Cappybara system architecture.


Limitations & Future Directions

Limitations:

- Author's preliminary results did not showcase scalability of the proposed method extensively.

Future Directions:

- Implement more diverse state management policies



MASTER: Machine Learning-Based Cold Start Latency Prediction Framework in Serverless Edge Computing Environments for Industry 4.0 [4]

Golec et al

Problem Definition

- Main problem: Cold start latency in serverless edge computing for Industry 4.0.
- Challenges:
 - Delayed response times
 - Impacting scalability and increased costs.
 - Not resource-aware
- Motivation:
 - Need for scalable, resource-aware methods to address cold start latency.

Proposed Method

- The authors propose MASTER, a machine learning (ML)-based framework to predict cold start latency
- Key aspects of the framework:
 - Uses several AI models to predict latency
 - Employs a novel cold start dataset
 - Compares performance against existing frameworks
 - Analyzes energy consumption and carbon emissions for resource-awareness.

Experimental Results

Compared to baseline frameworks:

- Improved cold start prediction accuracy
- Faster than baselines
- More energy efficient
- Less Carbon emissions

Limitations & Future Directions

Limitations:

- Small Dataset
- Generalizability: broader applications needed

Future Directions:

- Better ML/DL methods
- Larger datasets



Faashouse: Sustainable Serverless Edge Computing Through Energy-Aware Resource Scheduling [5]

Aslanpour et al



Problem Definition

- Variable energy resources create operational problems for nodes and serverless edge systems
- Energy is wasted on powerful nodes
- Low-powered nodes have high failures
- Current serverless frameworks like Kubernetes are energy-agnostic, lacking mechanisms to handle these challenges effectively.

Proposed Method

- The authors propose Faashouse, an energy-aware resource scheduling framework designed for serverless edge computing.
- Key contributions:
 - An energy-aware scheduler:
 - Performs monitoring and offloading to balance energy usage
 - Implementation on real-world devices

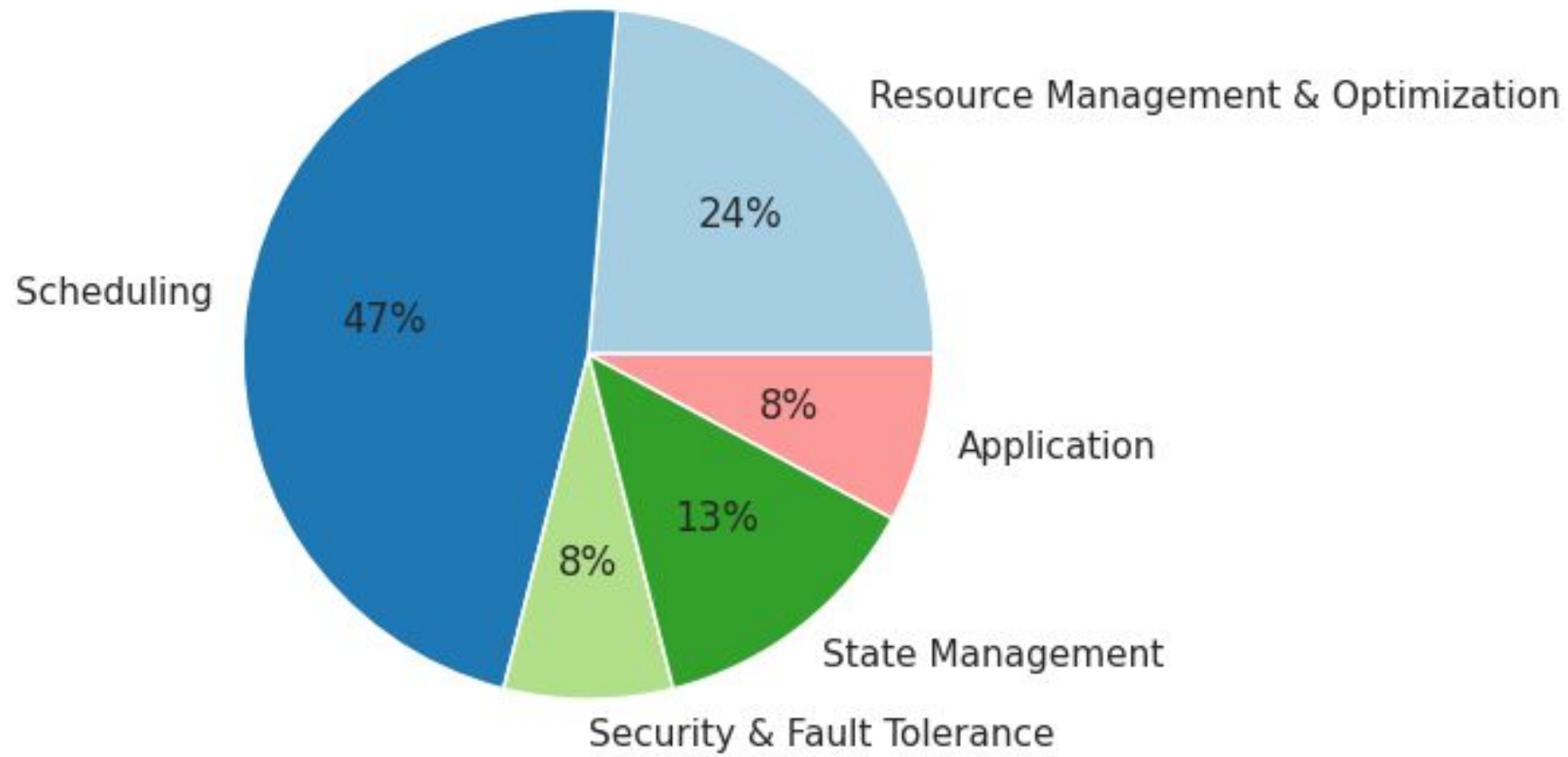
Experimental Results

- Improved node availability (36%) and less variation (better energy balance)
- Increased throughput (33%)
- Reduced energy wastage (76%)
- Decreased node failures (19%)

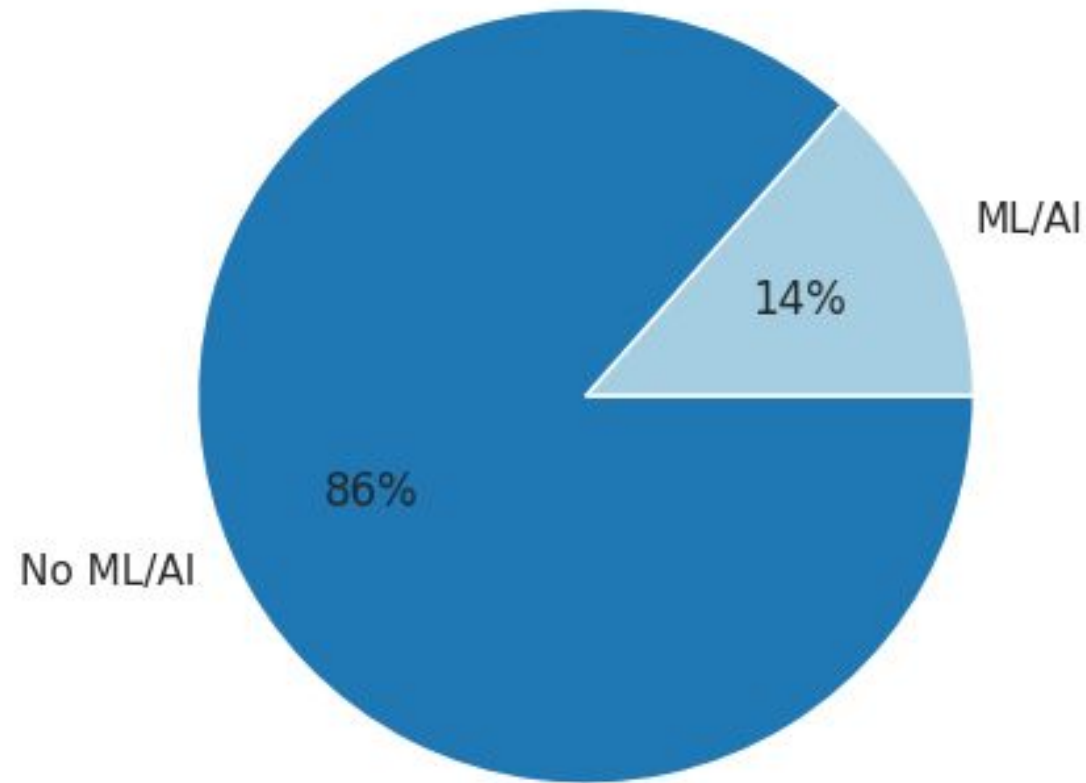
Limitations & Future Directions

- Tested on small-scale clusters. Larger-scale experiments required
- Using decentralized scheduling instead of centralized
- Using heterogeneous devices, including GPUs for intensive tasks

Analysis Results



Analysis Results



Conclusion and Future Research Directions

Current SOTA largely focuses on improving Serverless Edge Computing in two areas:

- Scheduling
- Resource Management & Optimization

The following challenges in Serverless Edge Computing needs to be addressed in future research:

- Exception and Failure Recovery for Stateless and Stateful Functions
- Security in Serverless Edge Computing
- Decentralized Scheduling
- Edge AI & GPU support
- Energy & Location Awareness
- Continuous and Long-Running Workloads

References

1. Aslanpour, M. S., Toosi, A. N., Cheema, M.A., Chhetri, M. B., & Salehi, M.A. (2024). Load balancing for heterogeneous serverless edge computing: A performance-driven and empirical approach. *Future Generation Computer Systems*, 154, 266-280.
2. Lou, J., Tang, Z., Lu, X., Yuan, S., Li, J., Jia, W., & Wu, C. (2024, June). Efficient Serverless Function Scheduling in Edge Computing. In *ICC 2024-IEEE International Conference on Communications* (pp. 1029-1034). IEEE.
3. Chen, X., Paidiparthi, M. P., & Hu, L. (2024, September). Toward an Edge-Friendly Distributed Object Store for Serverless Functions. In *Proceedings of the 15th ACM SIGOPS Asia-Pacific Workshop on Systems* (pp. 108-114).
4. Golec, M., Gill, S. S., Wu, H., Can, T. C., Golec, M., Cetinkaya, O., ... & Uhlig, S. (2024). Master: Machine learning-based cold start latency prediction framework in serverless edge computing environments for industry 4.0. *IEEE Journal of Selected Areas in Sensors*.
5. Aslanpour, M. S., Toosi, A. N., Cheema, M.A., & Chhetri, M. B. (2024). faasHouse: Sustainable Serverless Edge Computing through Energy-aware Resource Scheduling. *IEEE Transactions on Services Computing*.

Thank you!



University
of Regina

Go far, *together.*