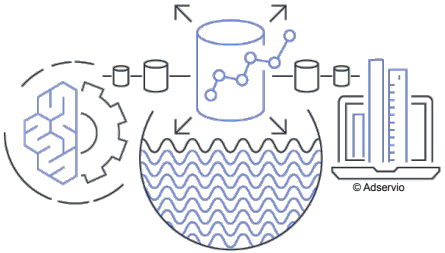


random
multiprobe
dx rendezvous
maglev
anchor ring

A Trace-driven Performance Evaluation of Hash-based Task Placement Algorithms for Cache-enabled Serverless Computing

Sacheendra Talluri[1], Nikolas Herbst[2], Cristina Abad[3], Animesh Trivedi[1], Alexandru Iosup[1]

[1]Vrije Universiteit Amsterdam, The Netherlands
[2]University of Würzburg, Germany
[3]ESPOL, Ecuador
Corresponding e-mail: s.talluri@vu.nl




VU @Large Research

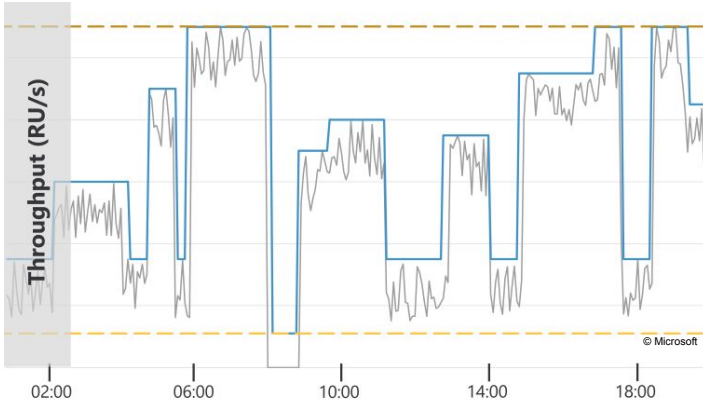
Data and code open-source on Zenodo:
https://zenodo.org/record/7812238

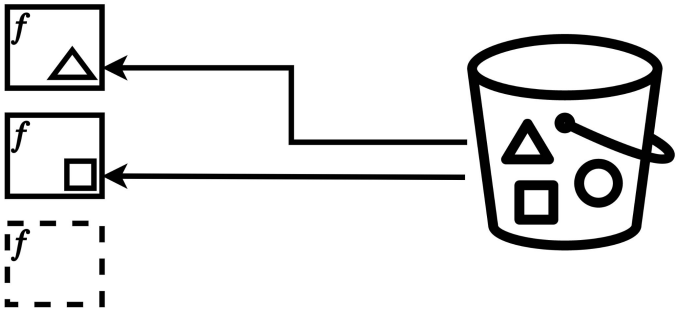
zenodo

Serverless Computing

Operational simplicity

Fine-grained resource usage





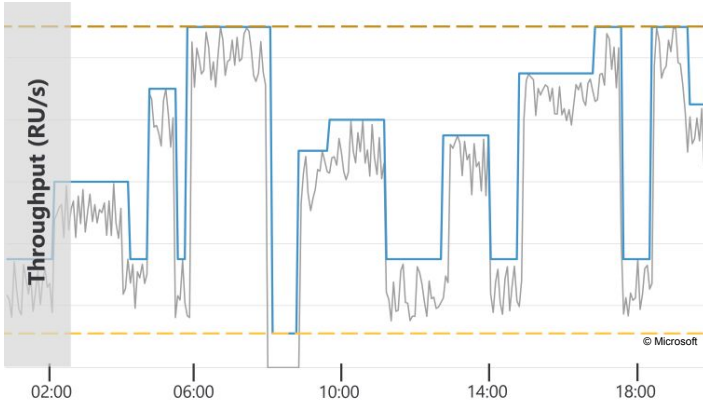
Serverless Computing

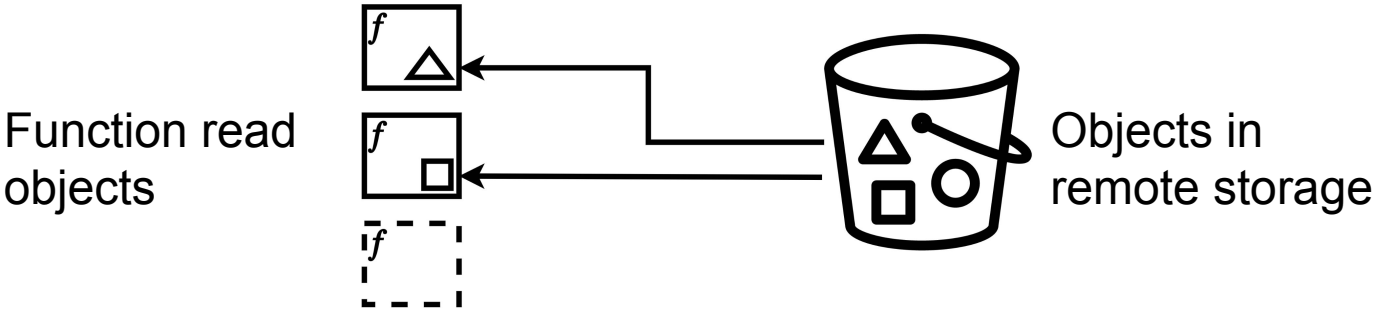
Operational simplicity

⟶ Independent resources

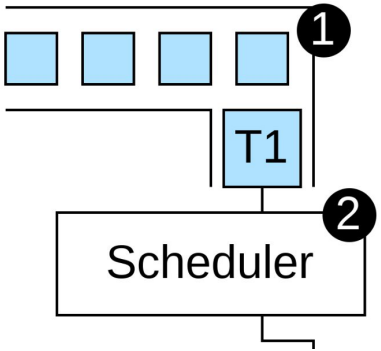
Fine-grained resource usage

⟶ Frequent cluster size changes

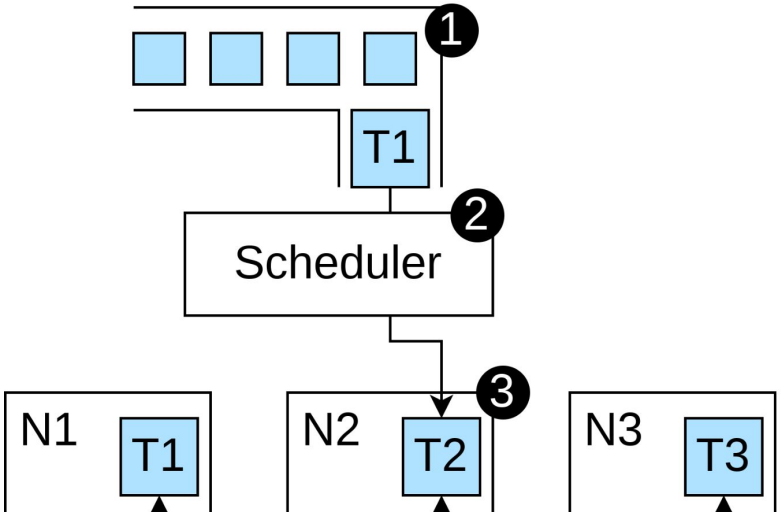




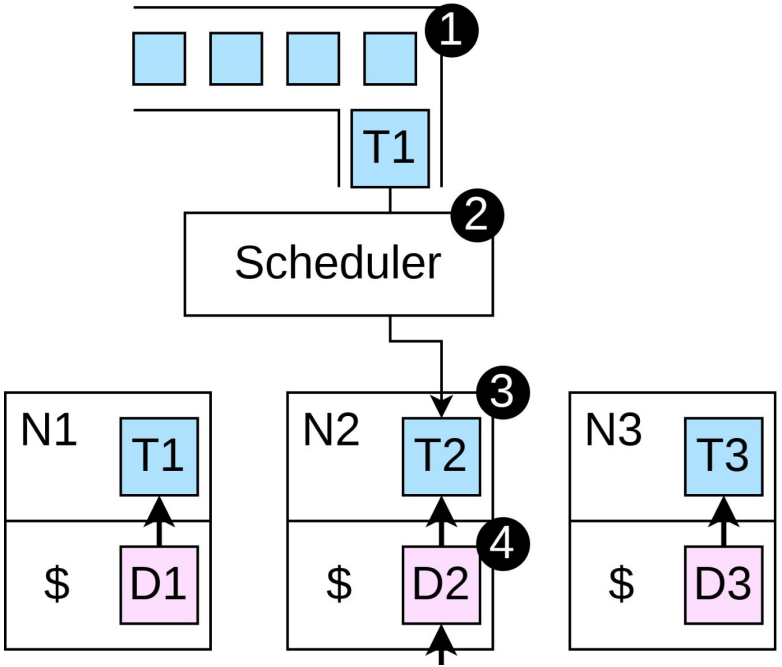
Serverless Data-processing



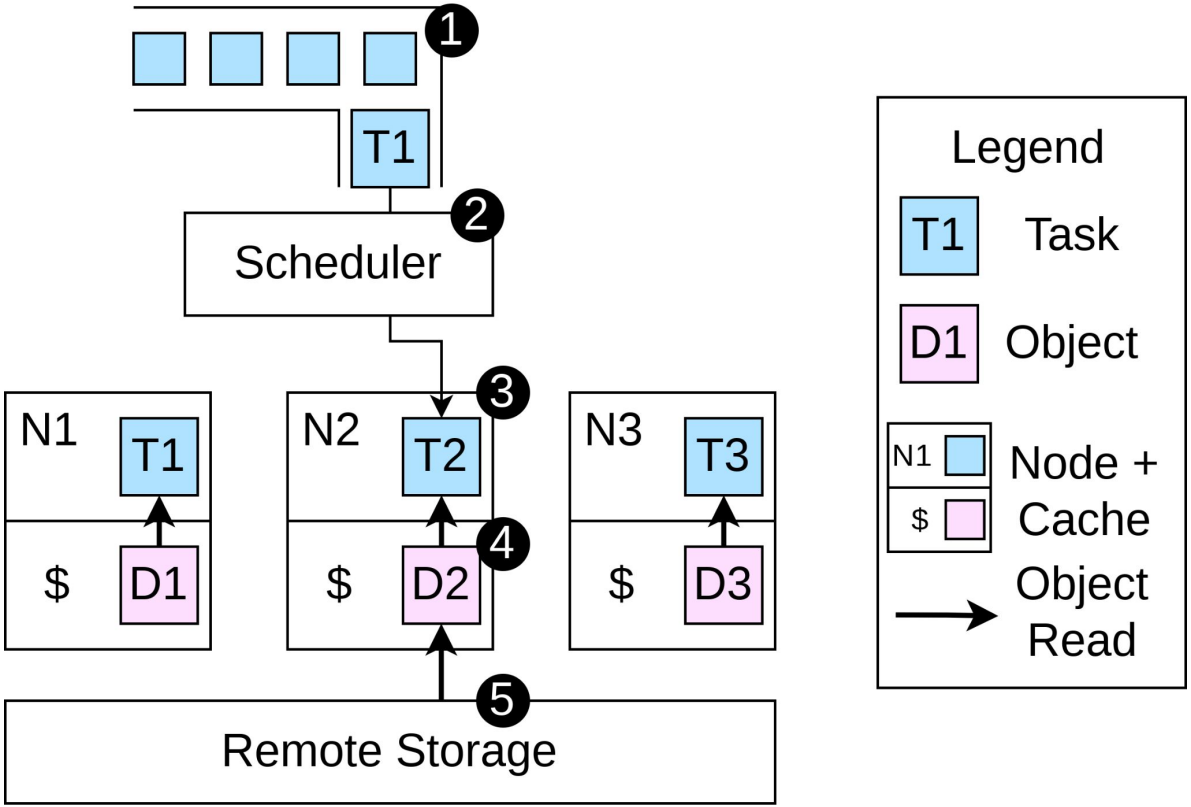
Serverless Data-processing



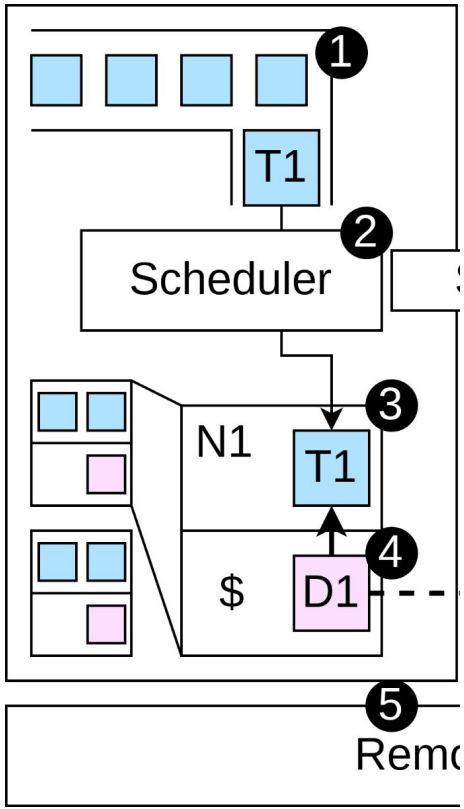
Serverless Data-processing



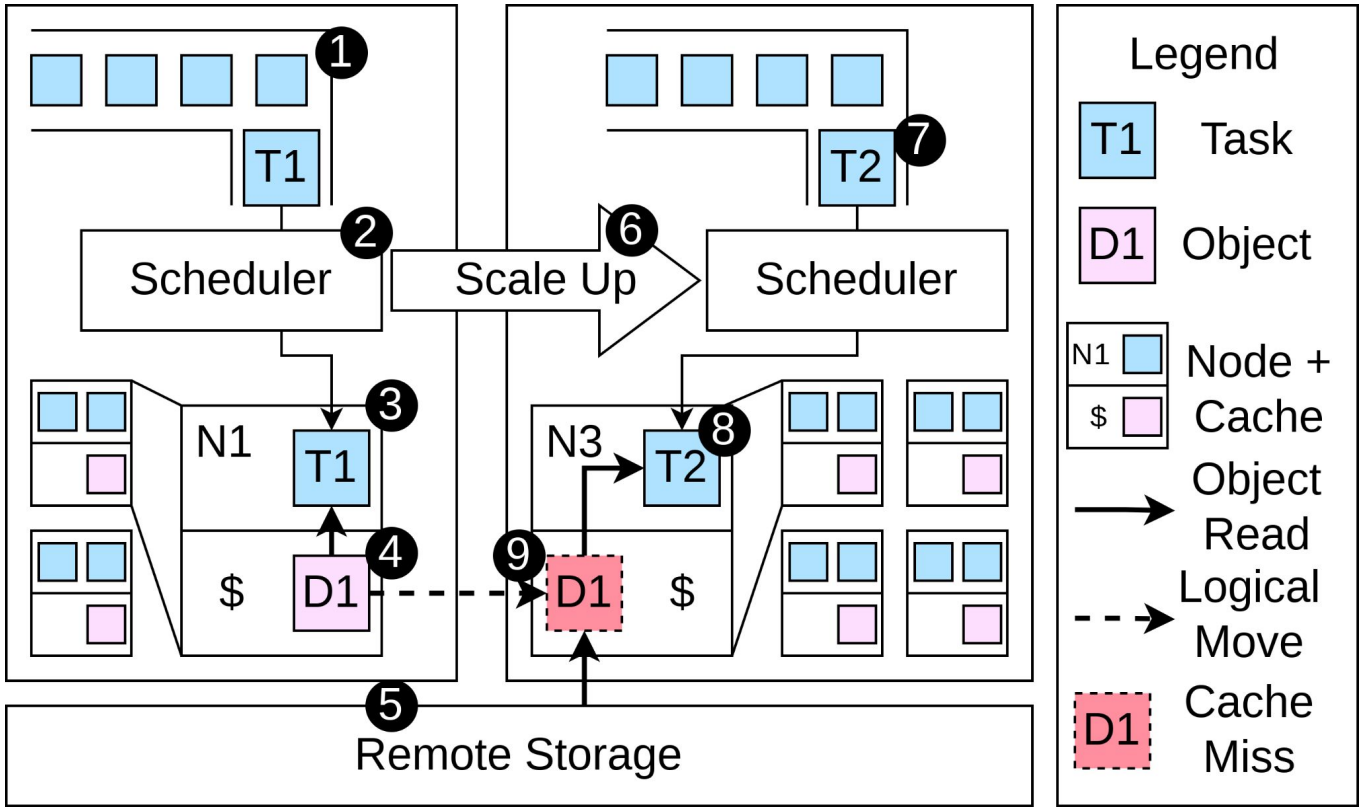
Serverless Data-processing



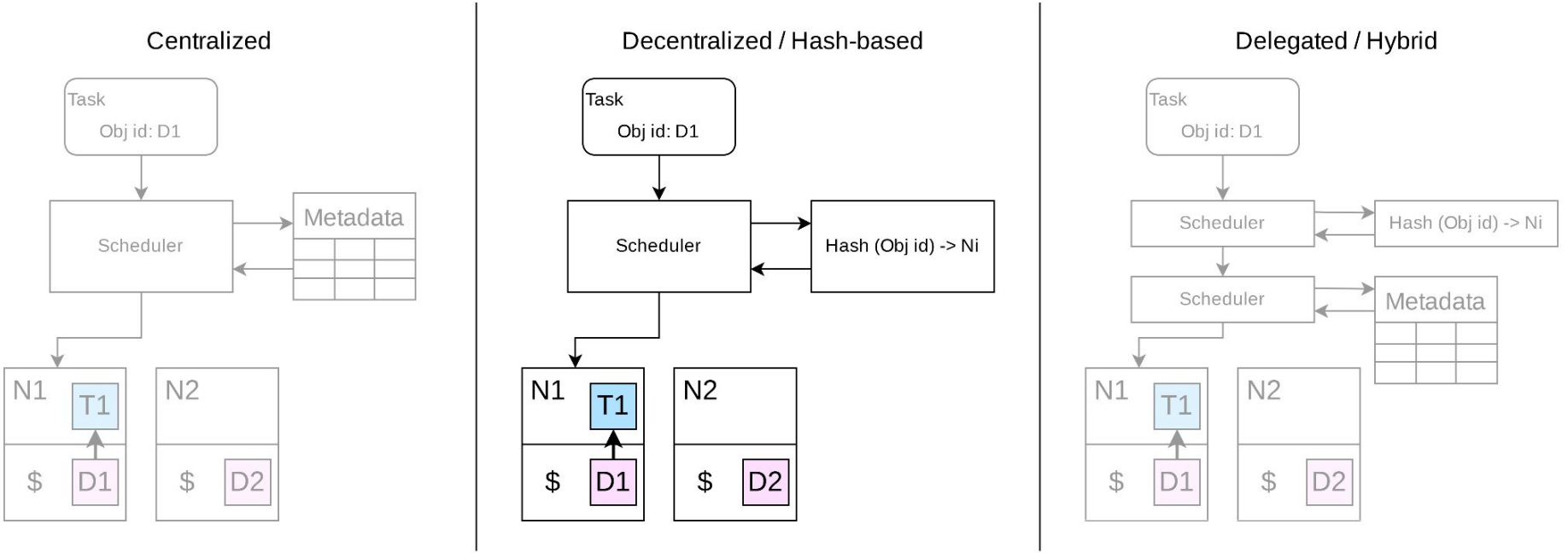
Serverless Data-processing (Autoscaling)



Serverless Data-processing (Autoscaling)

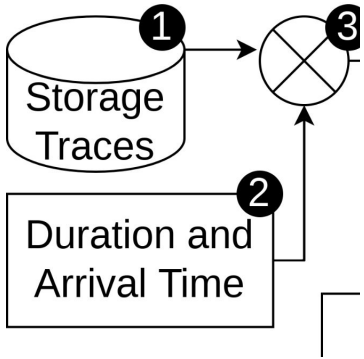


Task Placement in Serverless Data Processing



Method: Trace-driven Simulation

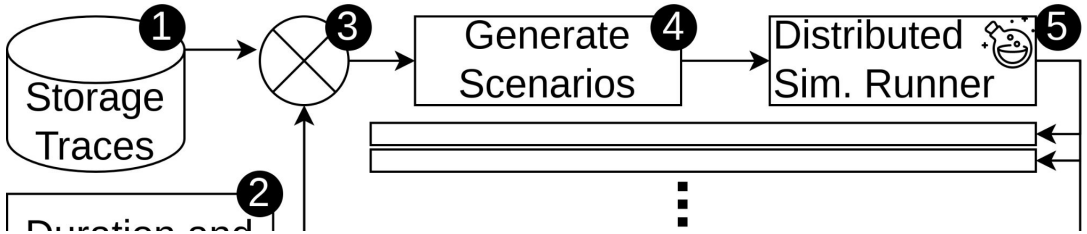
- OpenDC datacenter simulator
- 29 IBM Cloud Object Storage traces



https://github.com/atlarge-research/opendc

Method: Trace-driven Simulation

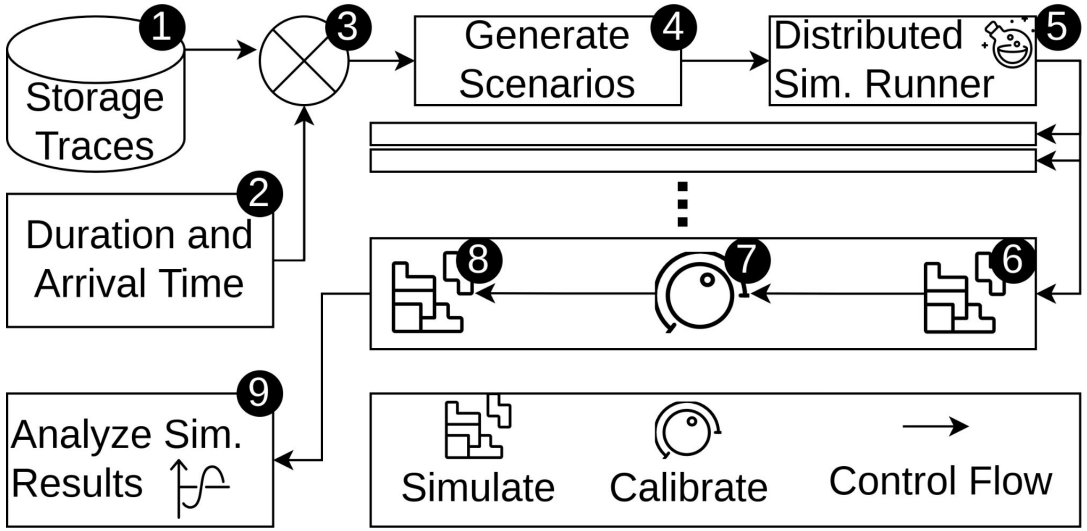
- OpenDC datacenter simulator
- 29 IBM Cloud Object Storage traces
- 8 task placement policies
- 500 simulations



https://github.com/atlarge-research/opendc

Method: Trace-driven Simulation

- OpenDC datacenter simulator
- 29 IBM Cloud Object Storage traces
- 8 task placement policies
- 500 simulations
- 3 million reads per simulation
- Real-world latency values

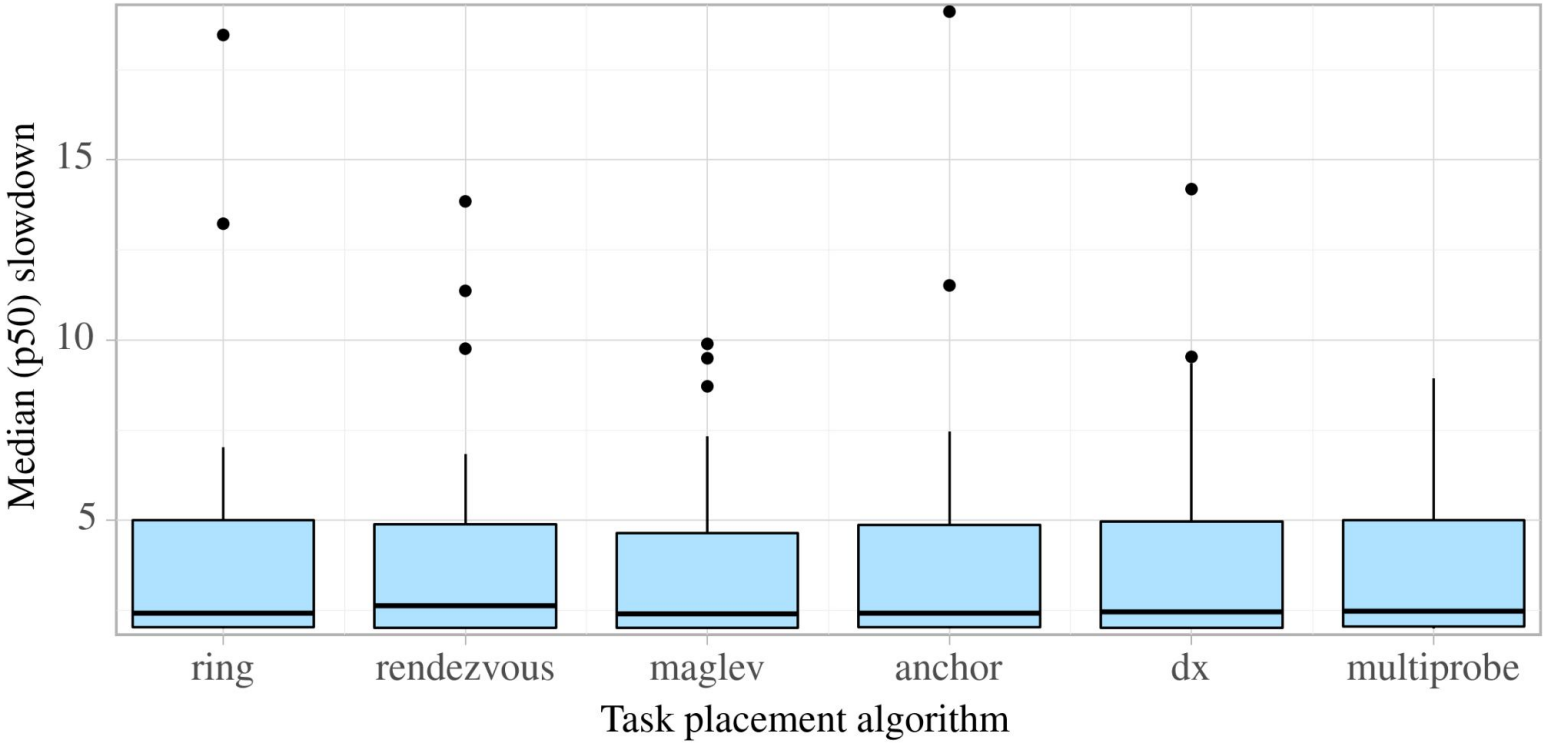


https://github.com/atlarge-research/opendc

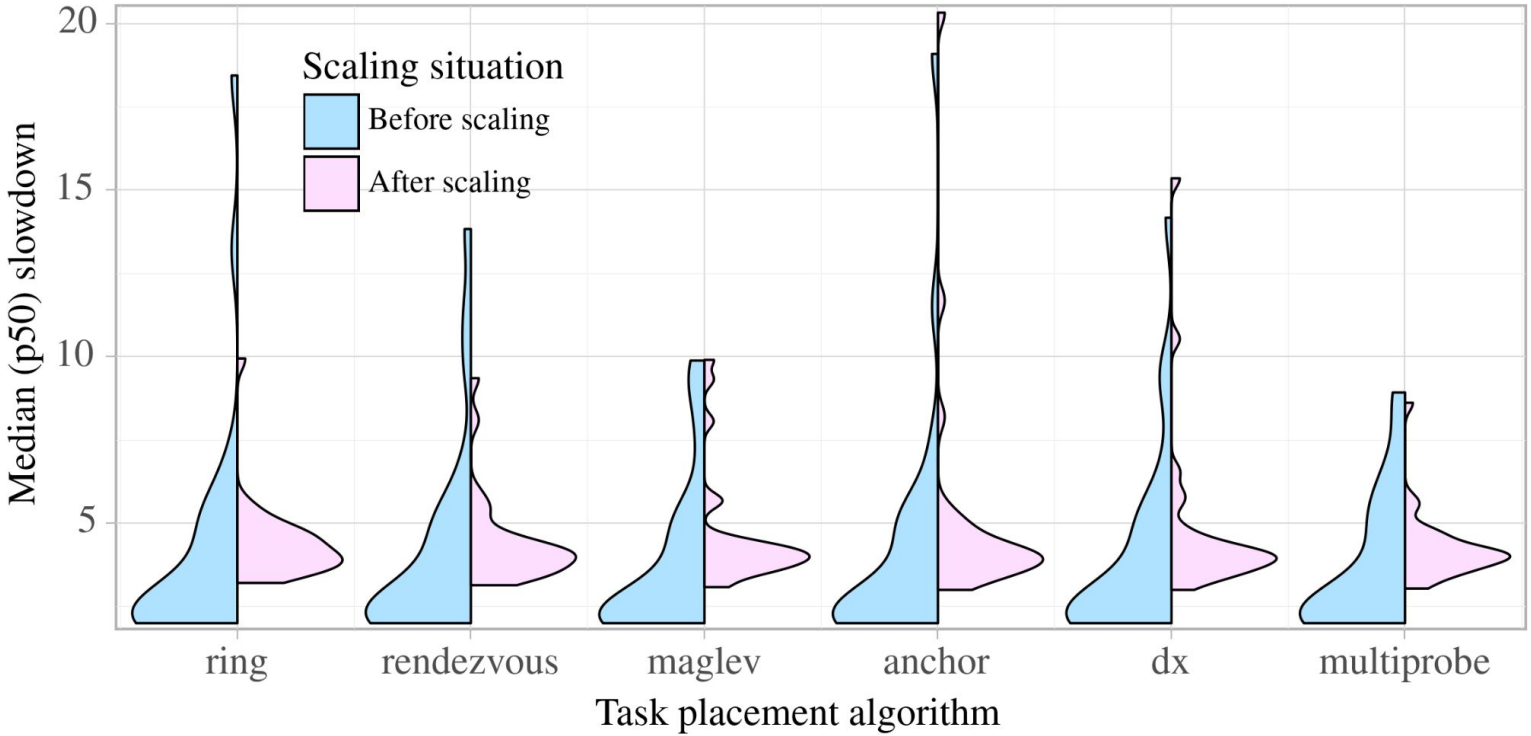
Metric: Slowdown

Slowdown = Real execution time ÷ Ideal execution time

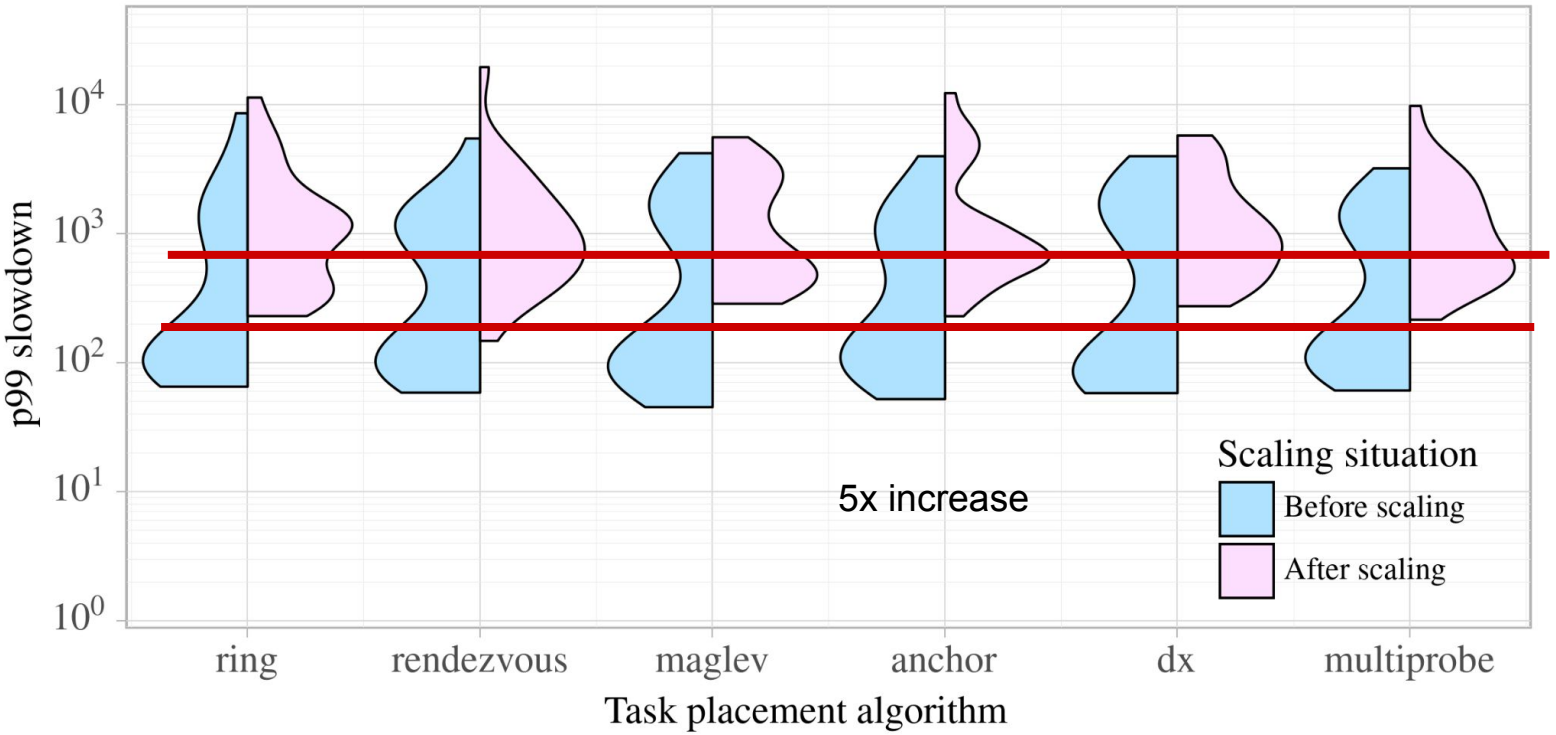
Median Slowdown



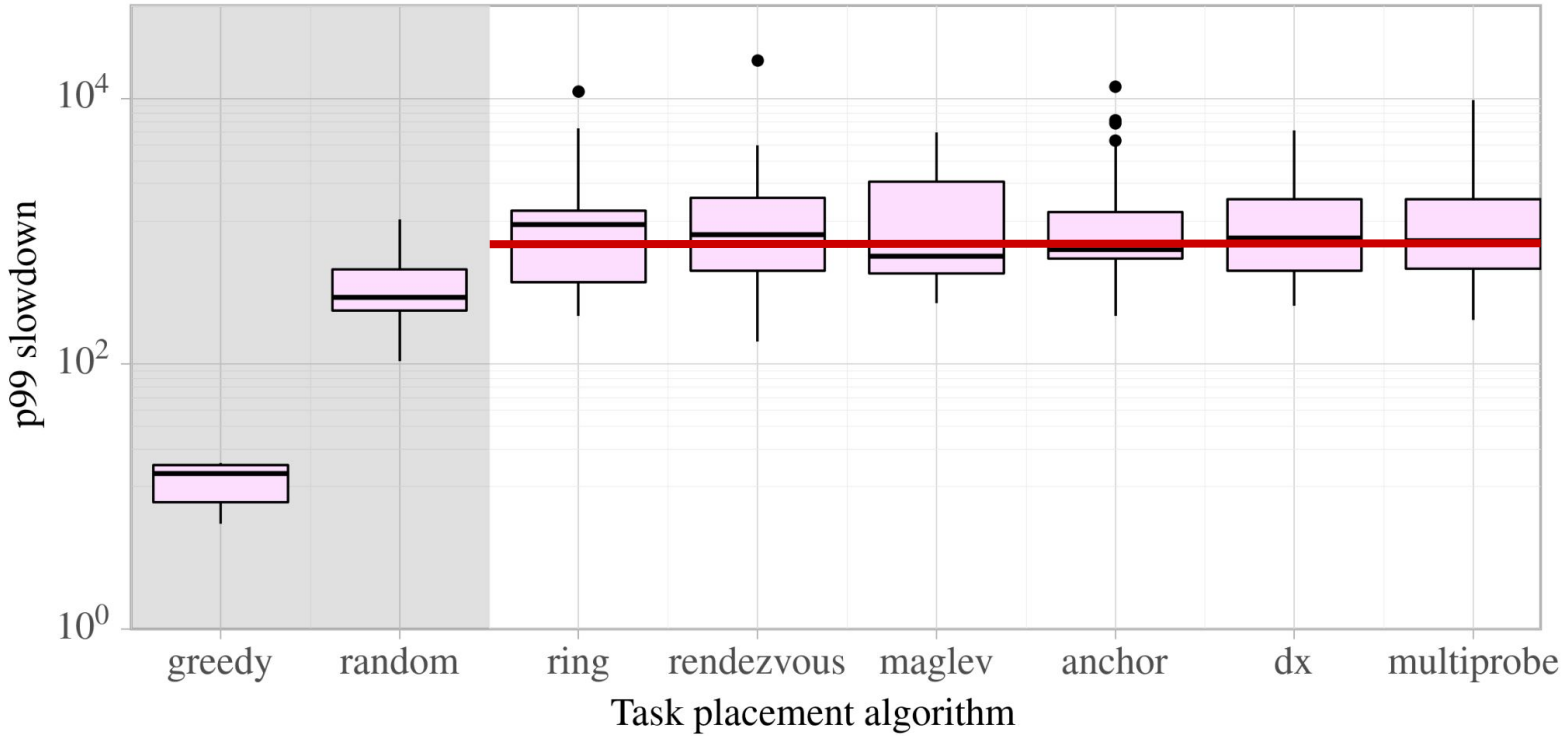
Median Slowdown



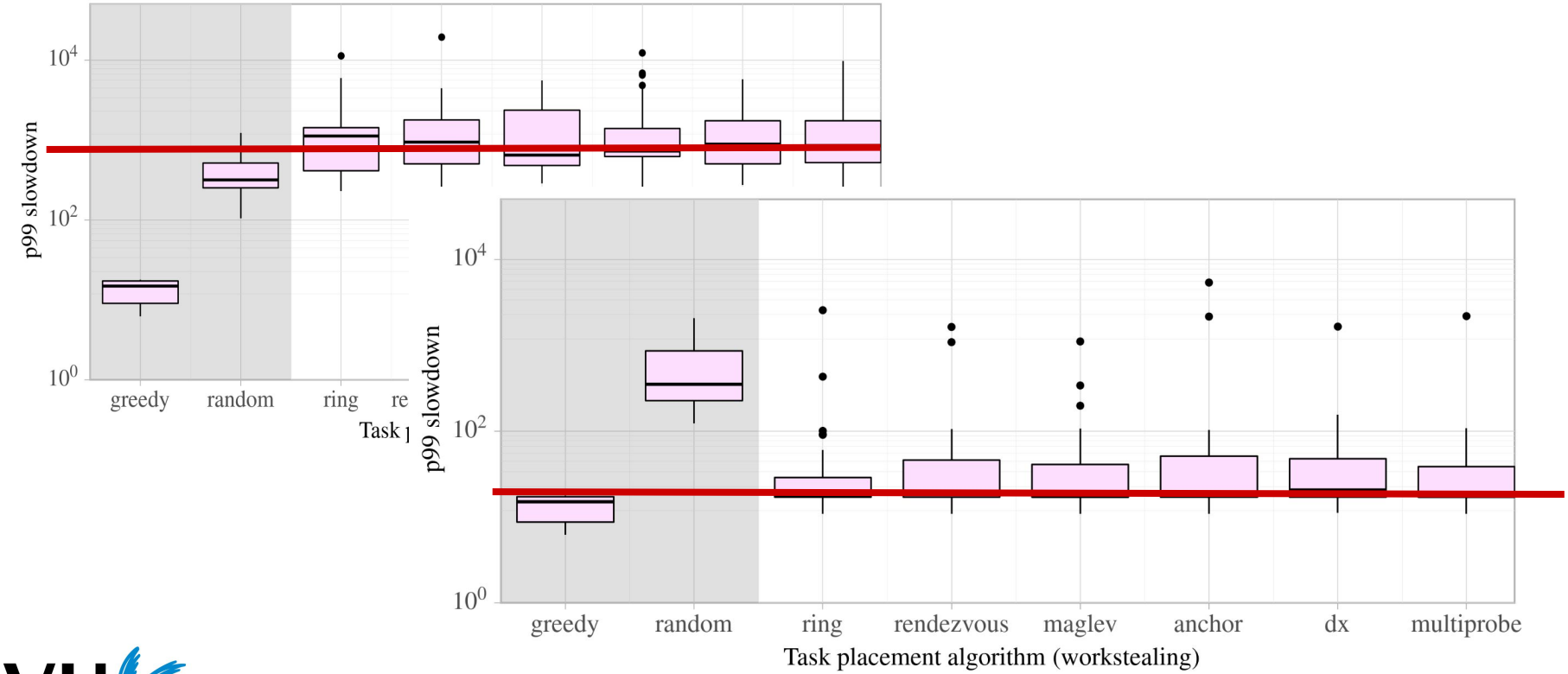
Tail Slowdown



Tail Slowdown



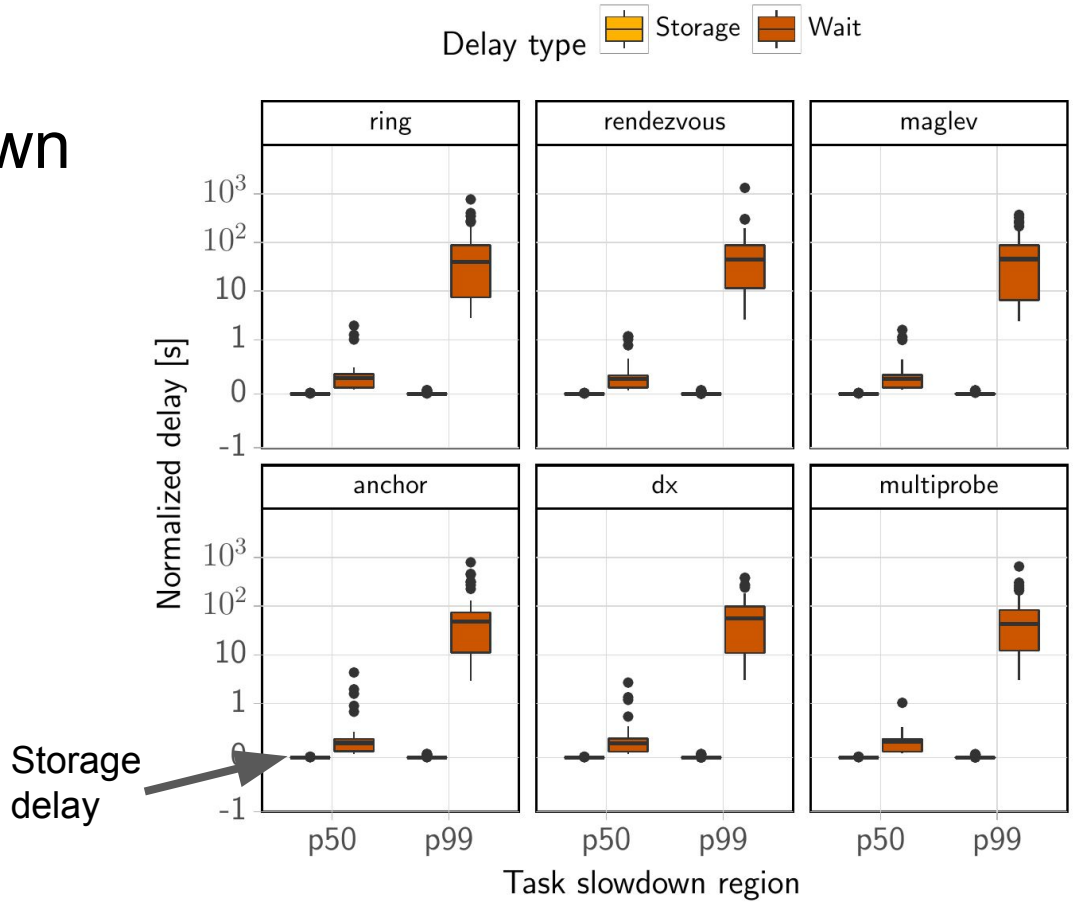
Workstealing



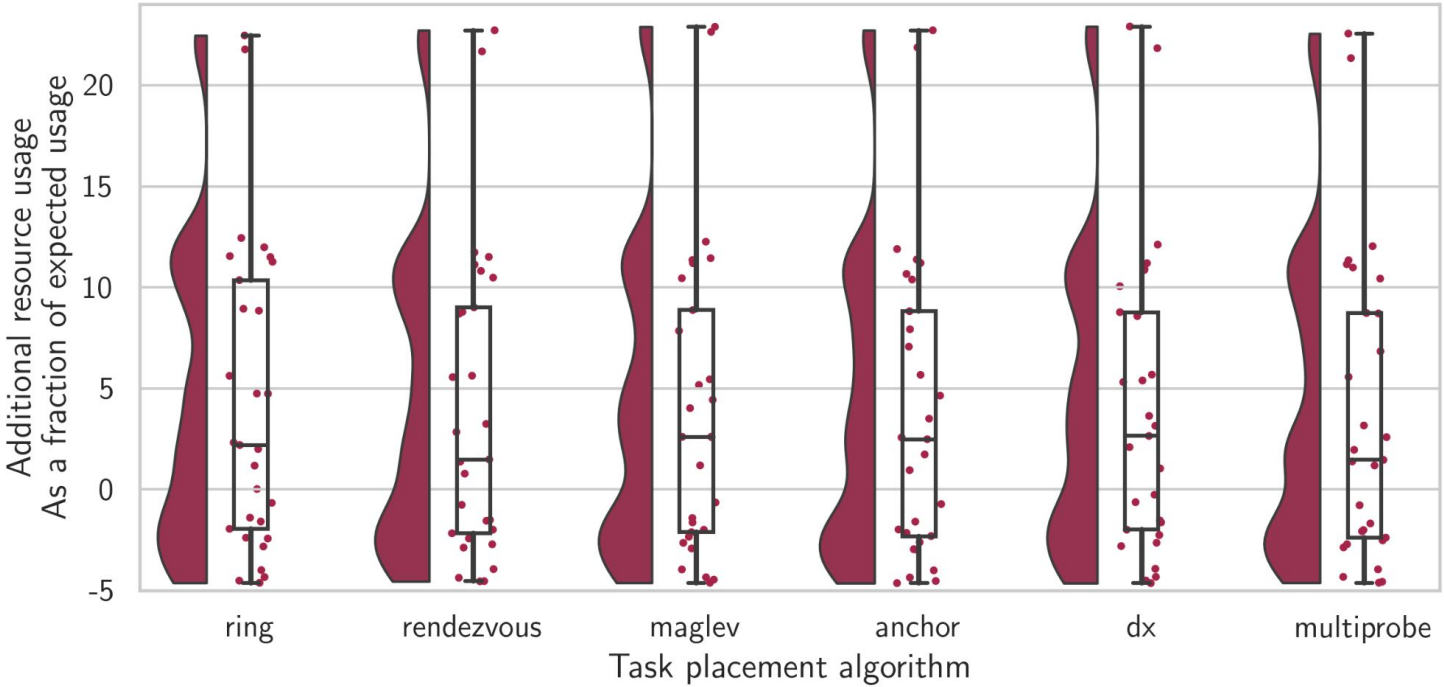
Cause of Slowdown

Storage delay: waiting for data from object storage

Wait: waiting in the scheduler for node resources to free up



Increased Resource Use After Scaling



Key Takeaways

- The task slowdown due to load imbalance can be much higher than the object imbalance.
- Hash-based task placement algorithms match data-oblivious greedy scheduling only when combined with workstealing.
- Cache-enabled serverless cluster can consume more resource after scaling (up to 22% more) due to cache misses.

Future Work

- Evaluating scheduler architectures
- Designing scheduler APIs
- … More tools to design and evaluate schedulers
- Join us!

Sacheendra Talluri

Website: https://atlarge-research.com/stalluri/

E-mail: s.talluri@vu.nl

VU @Large Research

Data and code open-source on Zenodo:
https://zenodo.org/record/7812238