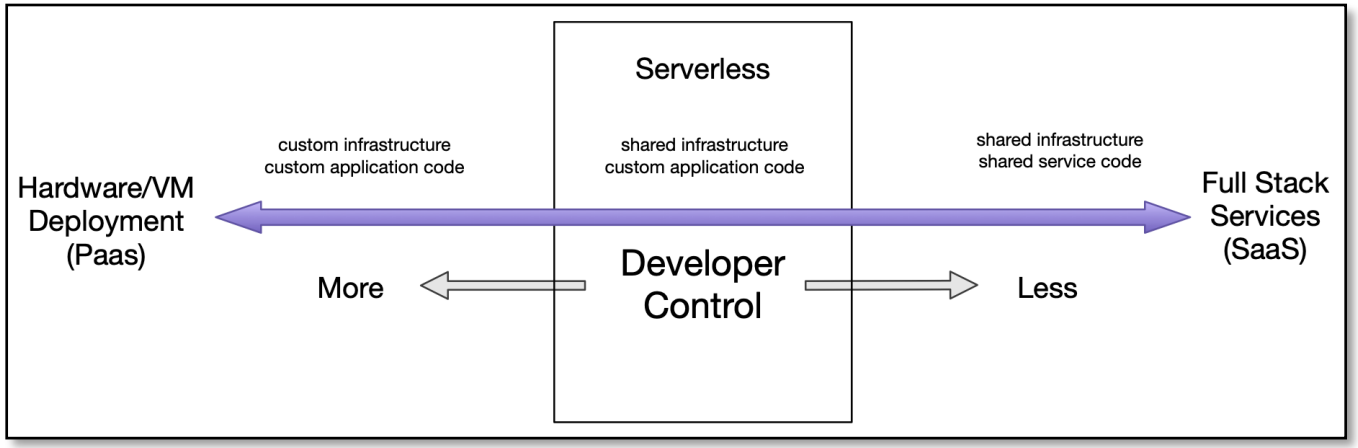
A Double Auction Mechanism for FaaS Scheduling in the Cloud Continuum

José Simão, Manuel Domingues, Luís Veiga

GECON 2026 - 21th International Conference on the Economics of Grids, Clouds, Systems, and Services
Pisa, Italy, 2026/09/10

Motivation

- The Functions-as-a-Service movement
  - Event-based functions executing mostly stateless operations

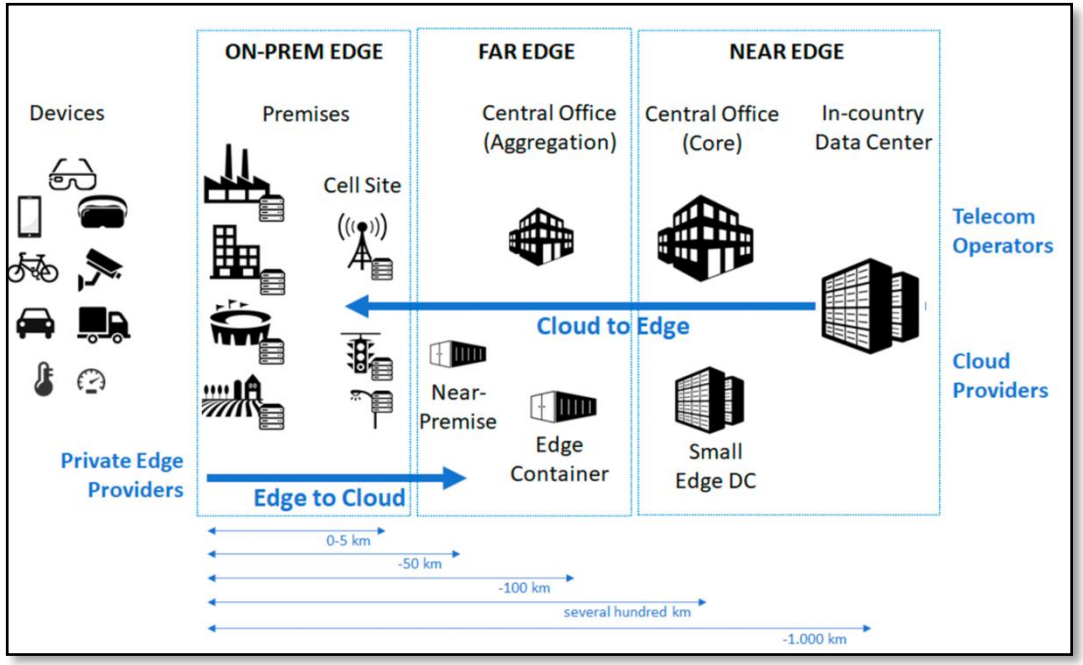


***Serverless Computing: Current Trends and Open Problems**, Research Advances in Cloud Computing. Springer (2017)*

  - Available as fully managed Cloud Services and Open-source Middlewares
  - Usually short lived computations that are more sensitive to memory usage

Motivation

- The Edge Computing movement
  - Heterogeneous set of devices available for computation, particularly on-prem
  - Reduces data transmitted to the cloud, saves bandwidth, enhances privacy



Study on the Economic Potential of Far Edge Computing in the Future Smart Internet of Things, European Commission (2021)

Challenges

Fixed posted pricing

- FaaS prices are set by the provider and do not react to demand

Scarce edge capacity under contention

- Under peaks the platform must decide which requests are best for the edge

Latency-dominated, short-lived functions

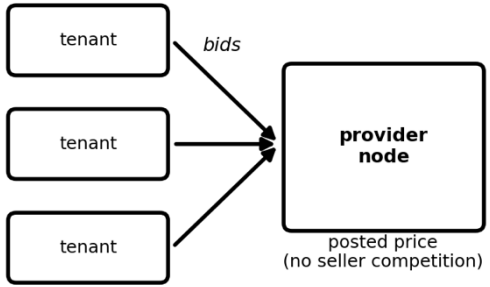
- Cold starts and transit dominate; batching an auction adds half a window of latency

Heterogeneous multi-tier, multi-provider continuum

- Edge, fog and cloud trade proximity against abundance: one posted price does not fit all

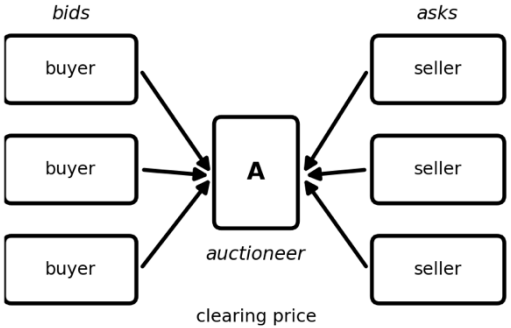
Related work

- **Single-sided** vs **two-sided** market mechanisms for FaaS placement



Single-sided auctions

- Tenants compete, providers do not: AuctionWhisk [9] runs one auction per request on OpenWhisk, with static bids fixed at deployment
- FaaSBid [2] discounts idle fog capacity through heuristics, but pricing remains one-sided



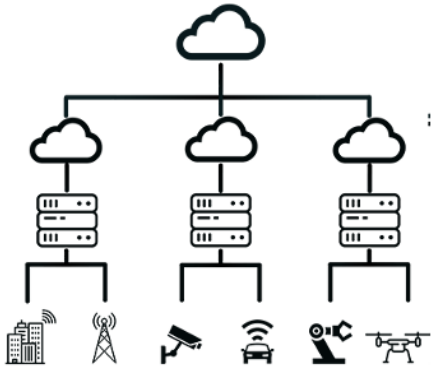
Double auctions

- Both sides bid: k-double / average, VCG, Trade Reduction, no mechanism is simultaneously IR, truthful, budget-balanced and efficient (every rule gives one up [17])

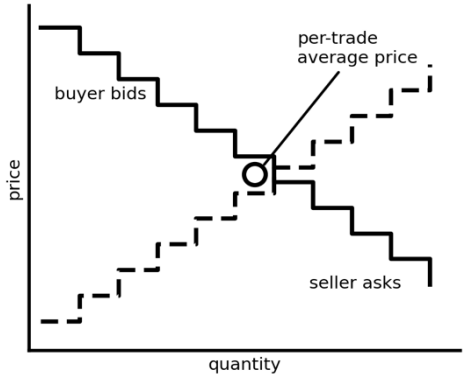
Contributions

- A double-auction **auctioneer middleware** (clearing rule, matching algorithm and adaptive bid protocol) that prices and places FaaS invocations across the edge-fog-cloud continuum

Function-as-a-Service

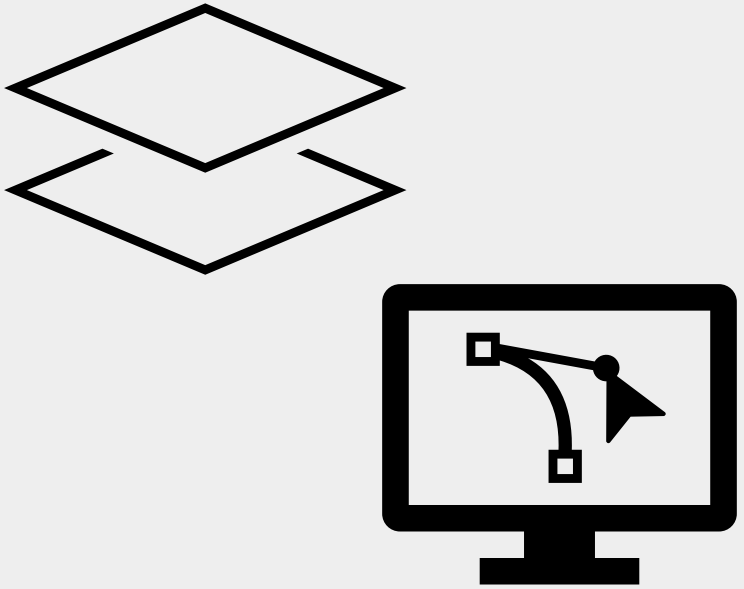


Cloud Continuum (Edge – Fog – Cloud)



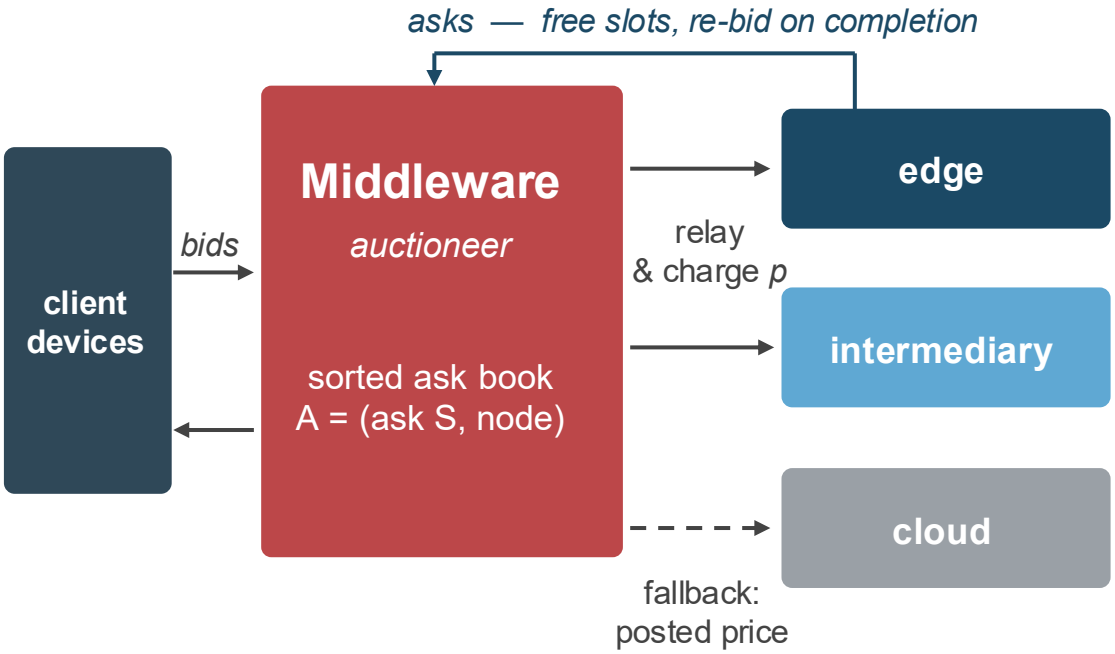
Double-Auction Pricing and Placement

Architecture and Algorithms



Architecture

- An **auctioneer middleware** between client devices and the multi-tier fog



Every request and every free slot passes through it
a global view of the capacity in its local group

Providers advertise free slots as asks
one message = an ask value + a count of identical slots

Bidding is fully automated
tenants set pricing policies, providers set capacity policies, i.e. nobody bids per invocation

Online clearing algorithm

- **No buyer-side batching:** asks are a standing, sorted book; each arriving bid is cleared immediately against it

Algorithm 1: per request

```
State: sorted ask book A of (S, node) pairs

on buyer bid B:
  release asks of completed jobs into A
  j ← rightmost ask S_j ≤ B            ▷ O(log N)
  if j exists:
      remove (S_j, node) from A         ▷ slot busy
      p ← (B + S_j) / 2
      relay request to node; charge p
      schedule completion at t + exec
  else:
      send to cloud at posted price

on job completion at node:
  node re-bids: ordered insert into A   ▷ O(log N)
```

Price: p = (B + S) / 2, buyer pays less than it bid, seller gets more than it asked, auctioneer keeps nothing

Match: closest ask below the bid: leaves cheaper nodes for cheaper requests

Cost: O(log N) per request, binary search on the sorted book

Re-bidding: the ask leaves on match and returns on completion, so the book stays current

Evaluation

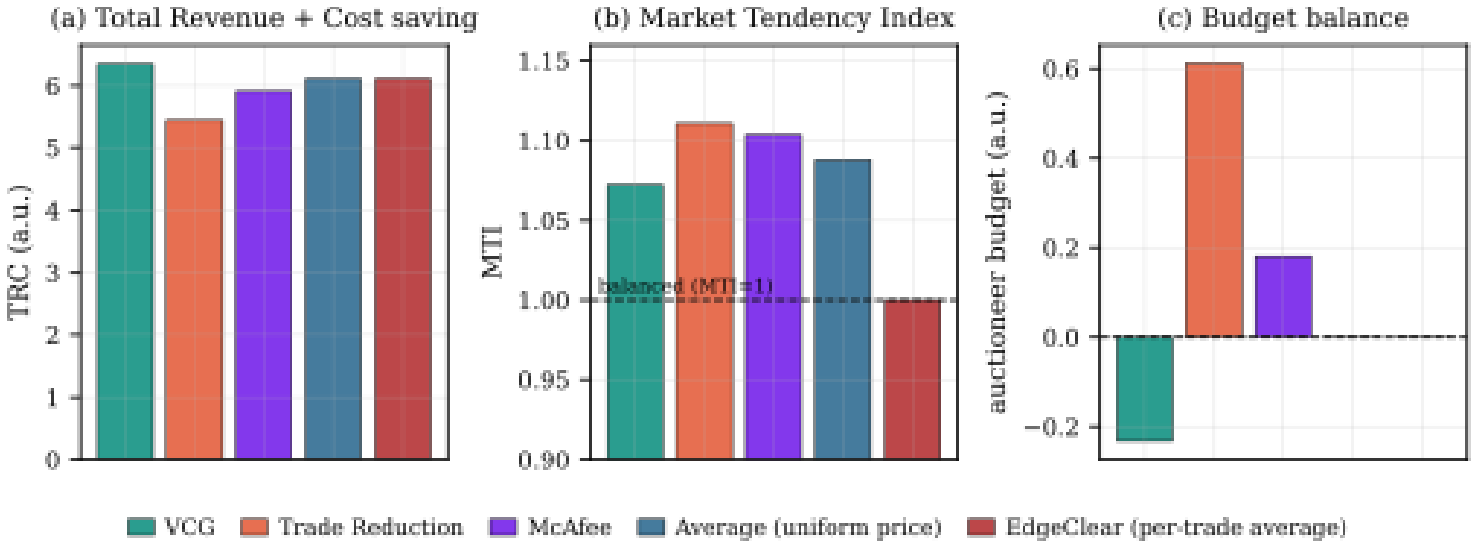
- Workloads characterization
- Bidding Mechanism
- Latency
  - The time taken to select a node
- Load, utilisation and pricing
- Marked segmentation

Workload Catalogue

Six profiles, calibrated from the SeBS [1] benchmark suite, with different properties of CPU and memory usage, warm and cold execution times

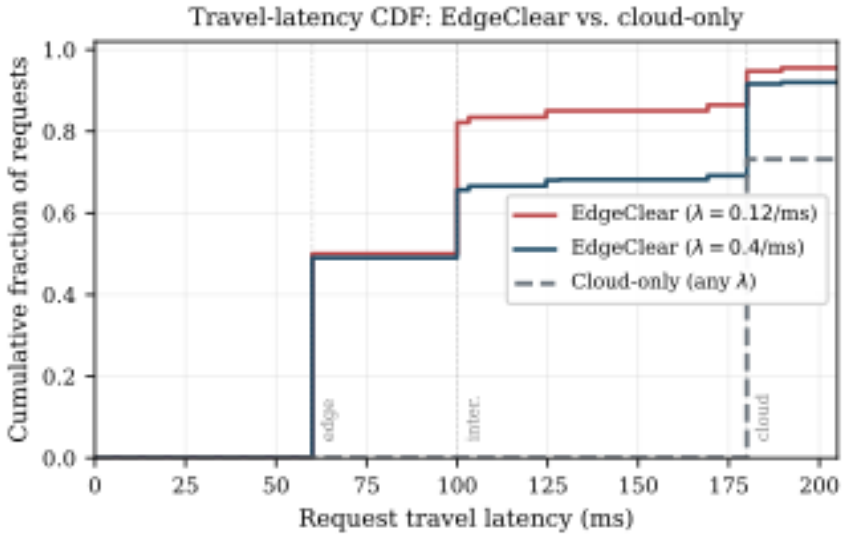
Workload	SeBS kernel	CPU	Mem (MB)	Warm (ms)	Cold (ms)	Description
sleep(t)	---	0.02	16	1.0	—	synthetic baseline
REST / HTTP	dynamic-html	0.99	32	1.2	**130**	short, large cold/warm gap
File hashing	compression	0.88	270	470	**607**	long, CPU-heavy
Image classification	image-recognition	0.99	179	125	**1268**	ML inference
Video transform	video-processing	0.90	500	1484	**1596**	long, CPU + mem intensive
Graph BFS	graph-bfs	0.99	128	37	**123**	short, CPU-bound

Clearing rules compared



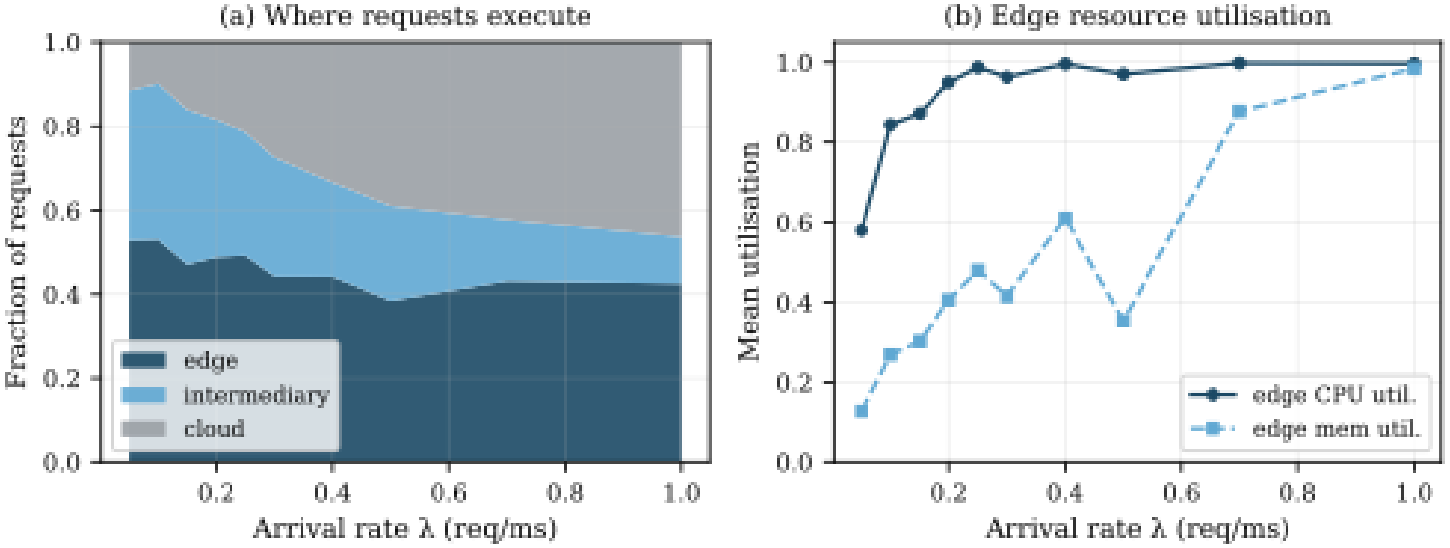
- Across 400 markets of 25 buyers and 25 sellers: (a) EdgeClear matches VCG on traded surplus, (b) alone lands on MTI = 1.00, and (c) needs neither a subsidy nor a margin

Latency



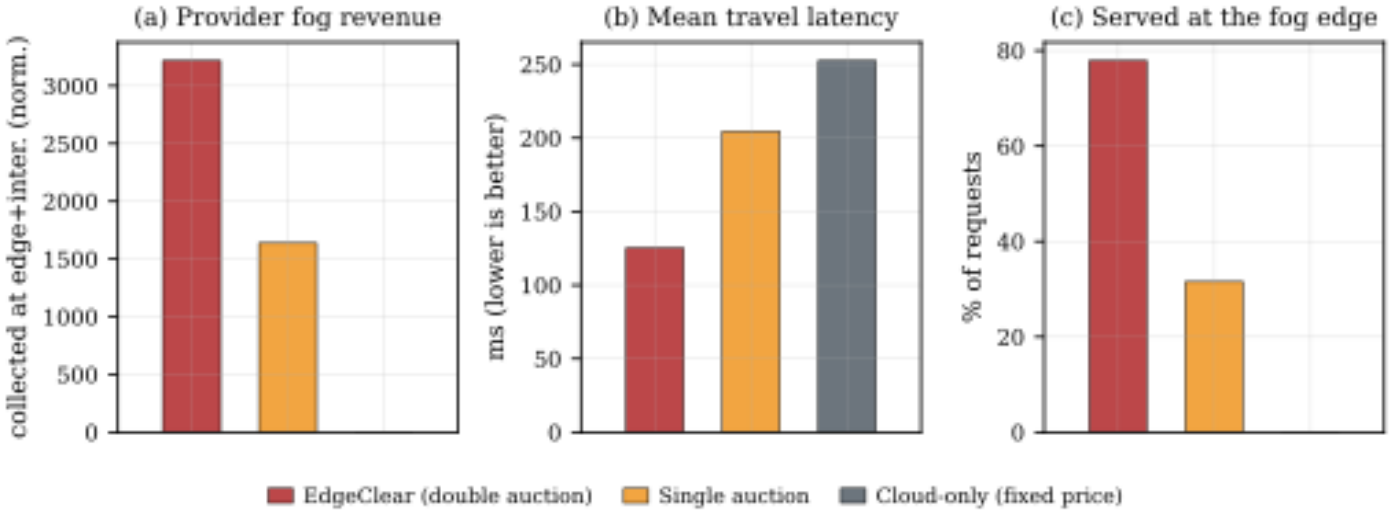
- Half the requests complete in ~60 ms at the edge and ~83 % within ~100 ms at the intermediary, against a flat ~180 ms round-trip for every cloud-only request.

Load, resource utilisation



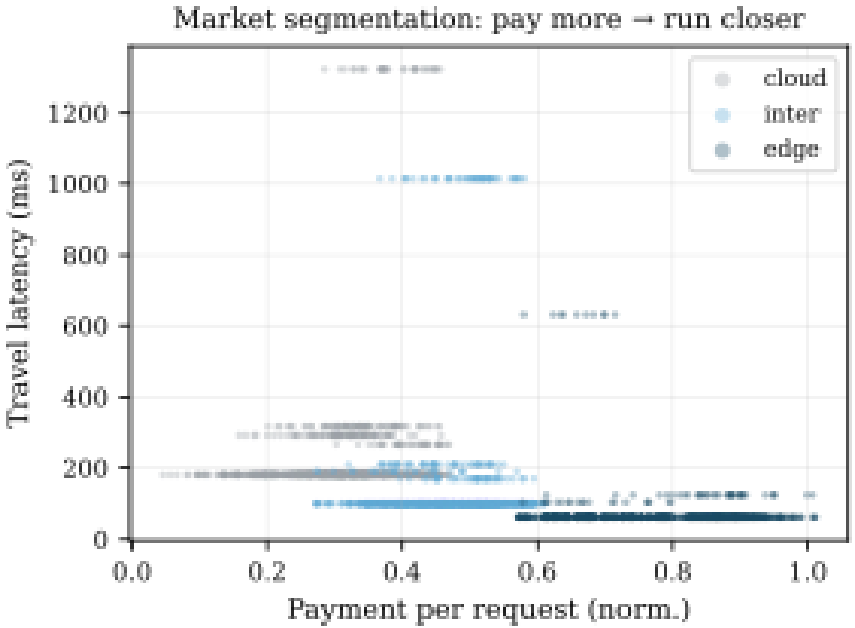
- The edge share shrinks and the cloud absorbs the overflow scarcity at the edge pushes the marginal request outward — no admission control needed
- Edge CPU saturates near 1.0 once λ is moderate, the mechanism keeps scarce resources busy rather than idle

Revenue



- Seller-side competition pays off on all three counts: EdgeClear roughly doubles provider fog revenue, cuts mean travel latency by ~40 % against cloud-only, and keeps ~78 % of requests in the fog (against ~32 % for the single auction)

Market segmentation



- The market sorts itself into three price vs. latency bands: pay more and run closer.

Conclusions

- EdgeClear clears FaaS requests online against a standing book of provider asks, serving more requests at the edge, at lower latency, while monetising idle fog capacity
- Tenants and providers still have an incentive to misreport their valuations; re-insertion is modelled as instantaneous, hiding a small efficiency leak from re-bidding delay; and profiles calibrated from SeBS on cloud hardware make real edge nodes slower than simulated.
- Next steps include a prototype over an open-source FaaS platform such as OpenWhisk, to validate the simulation on a real testbed, including the auctioneer hop and re-bidding delay, and to extend the auction to the cloud tier instead of treating it as a fixed-price fallback.

Thank you!

References

[1] Copik, M., Kwaśniewski, G., Besta, M., Podstawski, M., Hoefler, T.: SeBS: a serverless benchmark suite for Function-as-a-Service computing. Middleware '21, pp. 64–78. ACM (2021)

[2] Al-Qadhi, A.K., Athauda, R., Latip, R., Hussin, M.: FaaSBid: an auction-based model for Function as a Service in edge–fog environments using unallocated resources. Computer Communications 248, 108413 (2026)

[3] Abedrabboh, K., Karaki, A., Al-Fagih, L.: A combinatorial double auction for community shared distributed energy resources. IEEE Access 11, 28355–28369 (2023)

[4] Raith, P., Nastic, S., Dustdar, S.: Serverless edge computing — where we are and what lies ahead. IEEE Internet Computing 27(3), 50–64 (2023)

[5] Schmid, L., Copik, M., Calotoiu, A., Brandner, L., Koziolek, A., Hoefler, T.: SeBS-Flow: benchmarking serverless cloud function workflows. EuroSys '25, pp. 902–920. ACM (2025)

[6] The Apache Software Foundation: OpenWhisk documentation. openwhisk.apache.org/documentation.html (accessed 2026-08-10)

[7] Joosen, A., Hassan, A., Asenov, M., Singh, R., Darlow, L., Wang, J., Deng, Q., Barker, A.: Serverless cold starts and where to find them. EuroSys '25, pp. 938–953. ACM (2025)

[8] Babaioff, M., Nisan, N.: Concurrent auctions across the supply chain. Journal of Artificial Intelligence Research 21, 595–629 (2004)

[9] Bermbach, D., Bader, J., Hasenburg, J., Pfandzelter, T., Thamsen, L.: AuctionWhisk: using an auction-inspired approach for function placement in serverless fog platforms. Software: Practice and Experience 52(5), 1143–1169 (2021)

[10] Bermbach, D., Pallas, F., Pérez, D.G., Plebani, P., Anderson, M., Kat, R., Tai, S.: A research perspective on fog computing. ICSOC 2017 Workshops, pp. 198–210. Springer (2018)

[11] Du, D., Yu, T., Xia, Y., Zang, B., Yan, G., Qin, C., Wu, Q., Chen, H.: Catalyzer: sub-millisecond startup for serverless computing with initialization-less booting. ASPLOS '20, pp. 467–481. ACM (2020)

References

[12] Dukic, V., Bruno, R., Singla, A., Alonso, G.: Photons: lambdas on a diet. ACM SoCC '20, pp. 45–59 (2020)

[13] Eismann, S., Scheuner, J., van Eyk, E., Schwinger, M., Grohmann, J., Herbst, N., Abad, C.L., Iosup, A.: The state of serverless applications: collection, characterization, and community consensus. IEEE Transactions on Software Engineering 48(10), 4152–4166 (2022)

[14] Kagel, J.H.: Auctions: a survey of experimental research. In: The Handbook of Experimental Economics, pp. 501–586. Princeton University Press

[15] Khorasany, M., Mishra, Y., Ledwich, G.: Design of auction-based approach for market clearing in peer-to-peer market platform. The Journal of Engineering 2019 (2019)

[16] McAfee, R.P.: A dominant strategy double auction. Journal of Economic Theory 56(2), 434–450 (1992)

[17] Myerson, R.B., Satterthwaite, M.A.: Efficient mechanisms for bilateral trading. Journal of Economic Theory 29(2), 265–281 (1983)

[18] Rothkopf, M.H.: Thirteen reasons why the Vickrey–Clarke–Groves process is not practical. Operations Research 55(2), 191–197 (2007)

[19] Rustichini, A., Satterthwaite, M.A., Williams, S.R.: Convergence to efficiency in a simple market with incomplete information. Econometrica 62(5), 1041–1063 (1994)

[20] Stafford, A., Toosi, F.G., Mjeda, A.: Cost-aware migration to functions-as-a-service architecture. European Conference on Software Architecture, Workshops and Tutorials (2021)

[21] Varshney, P., Simmhan, Y.: Demystifying fog computing: characterizing architectures, applications and abstractions. IEEE ICFEC 2017, pp. 115–124 (2017)