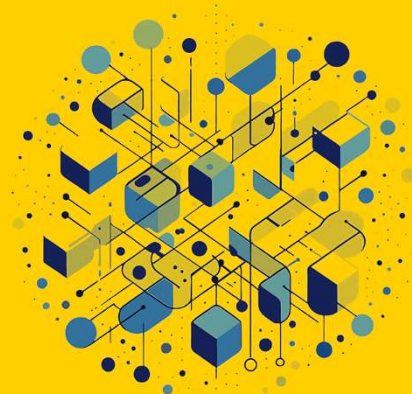




swarmchestrator



Edinburgh Napier
UNIVERSITY

School of Computing,
Engineering & the
Built Environment

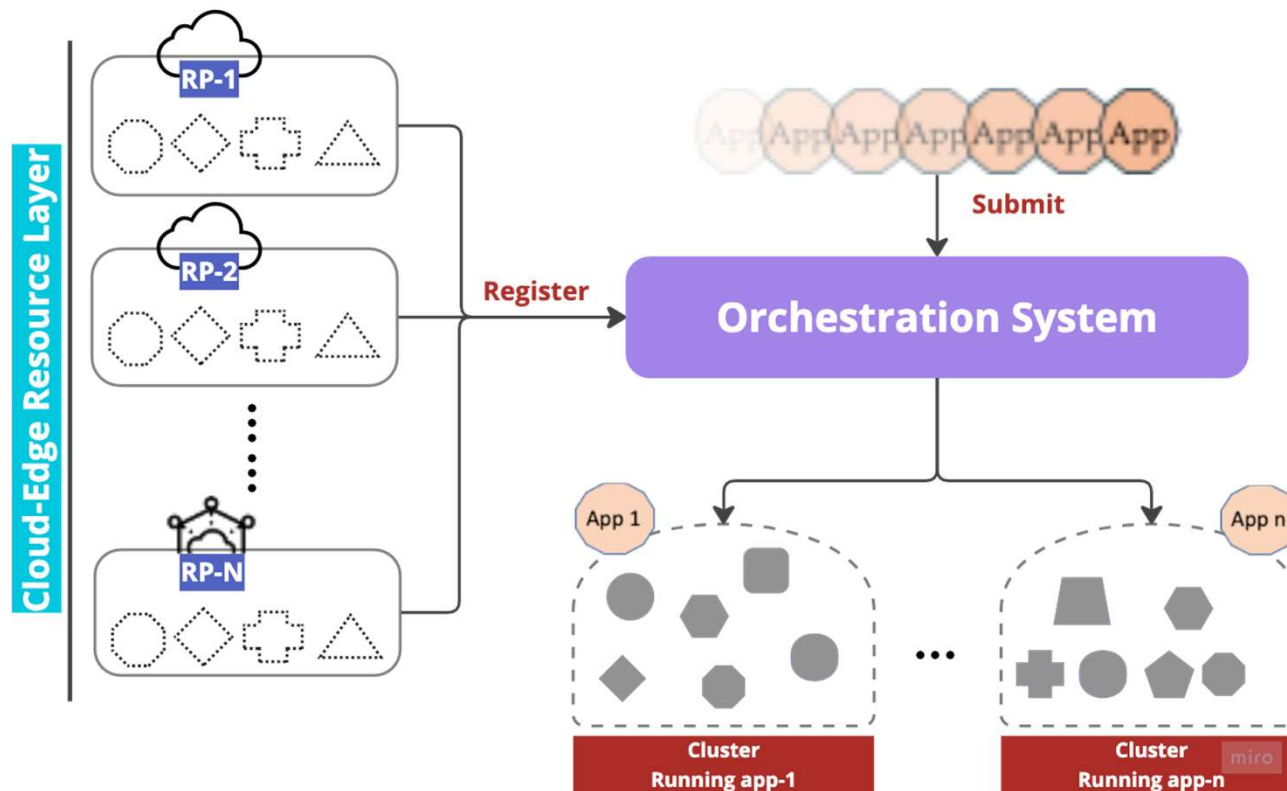
Distributed Resource Selection for Self-Organising Cloud-Edge Systems

Quentin Renau, **Amjad Ullah**, Emma Hart

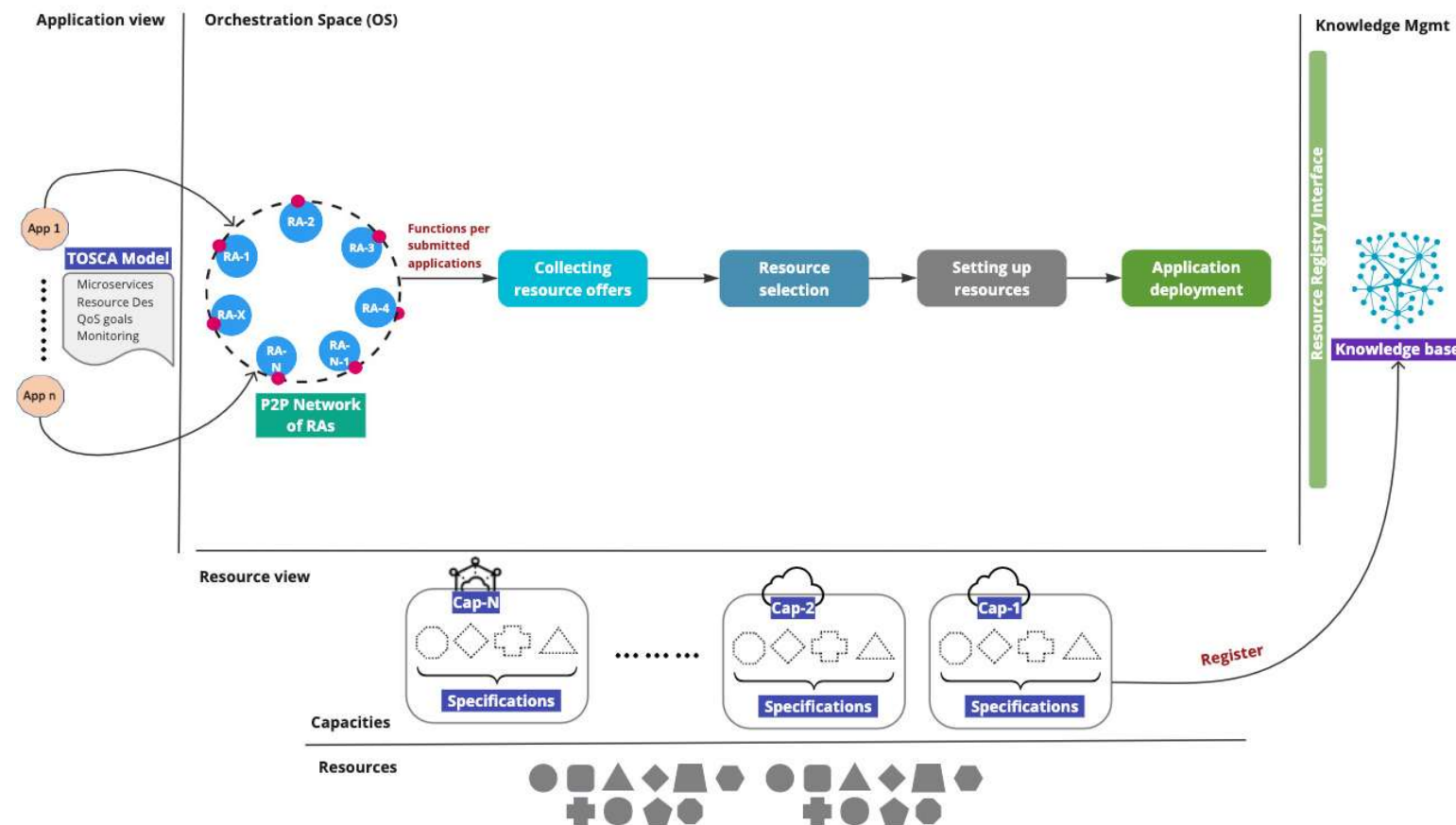
Outline

- Background and context
- Proposed method: CBBA-based resource selection
- Evaluation
- Conclusion

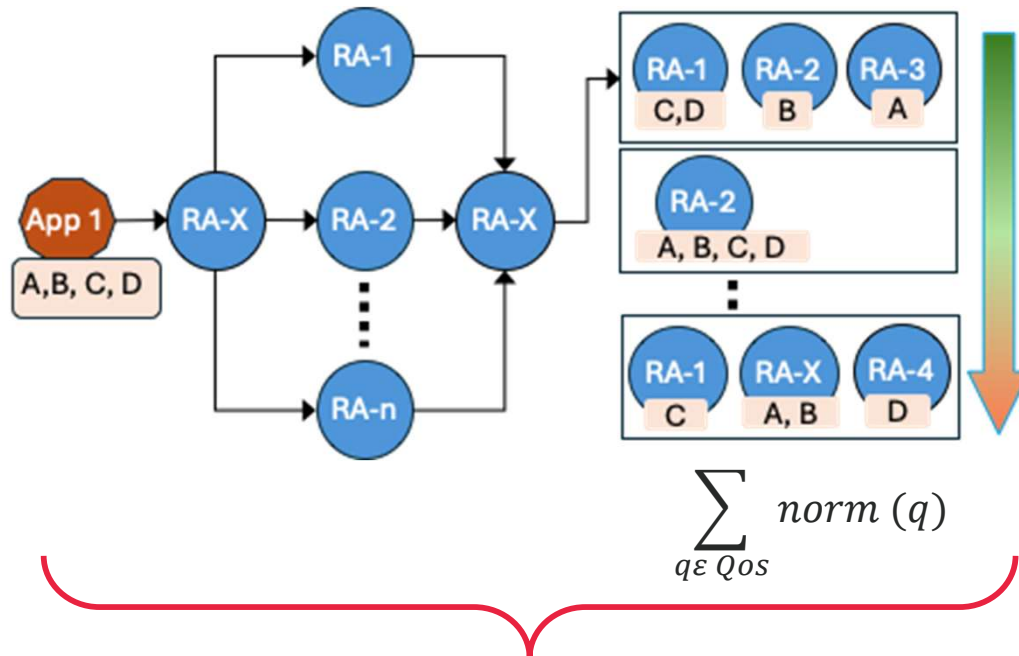
Orchestration system



Swarmchestrator – EU project



Resource offers → selection

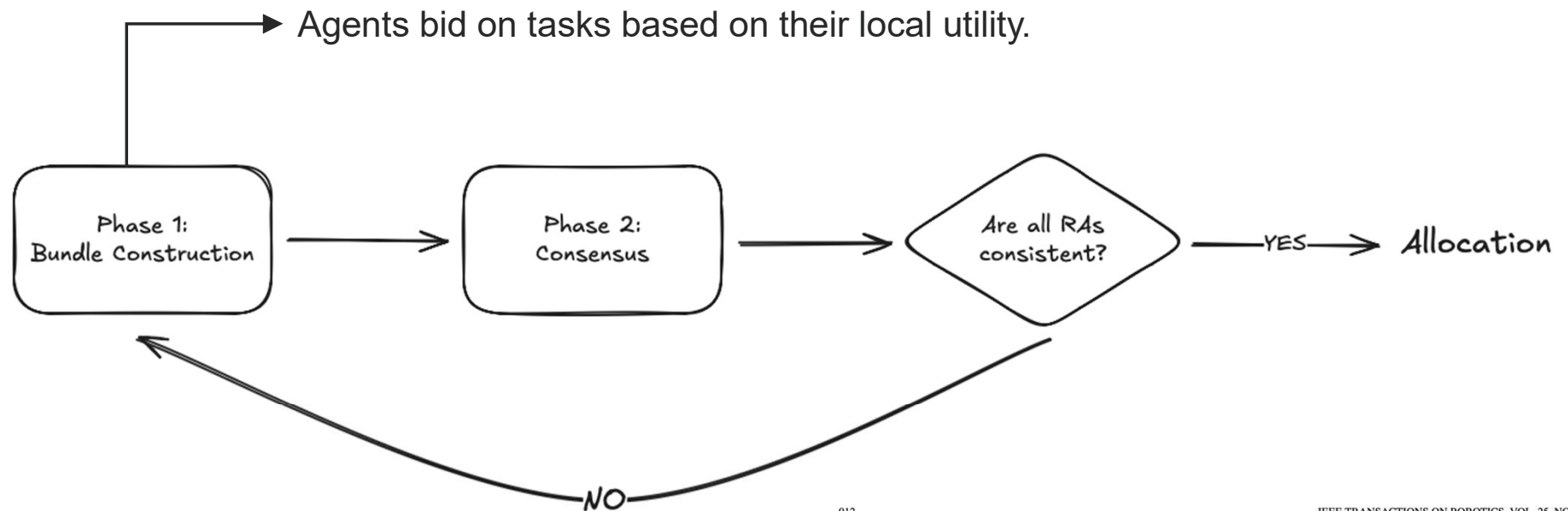


- Example: app1 (4 microservices)
 - A, B, C, D
 - A: VM: CPU:8, mem: 16GB,
 - B: EDGE: CPU:2, mem:4GB, disk space:20GB
Location = Edinburgh

Objective:

- Remove the centralised behaviour
- RAs collaborate to identify a suitable set of resources across for the submitted application while considering different characteristics

Consensus-based decentralized auction (CBBA)



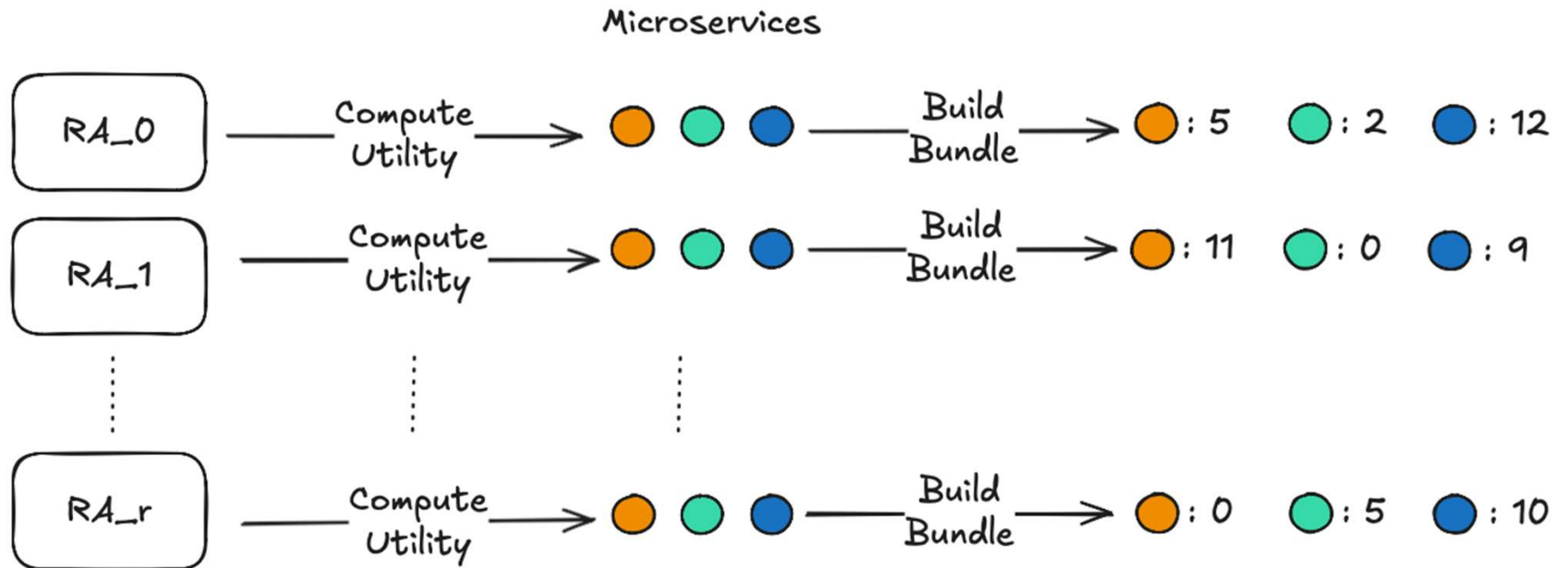
912

IEEE TRANSACTIONS ON ROBOTICS, VOL. 25, NO. 4, AUGUST 2009

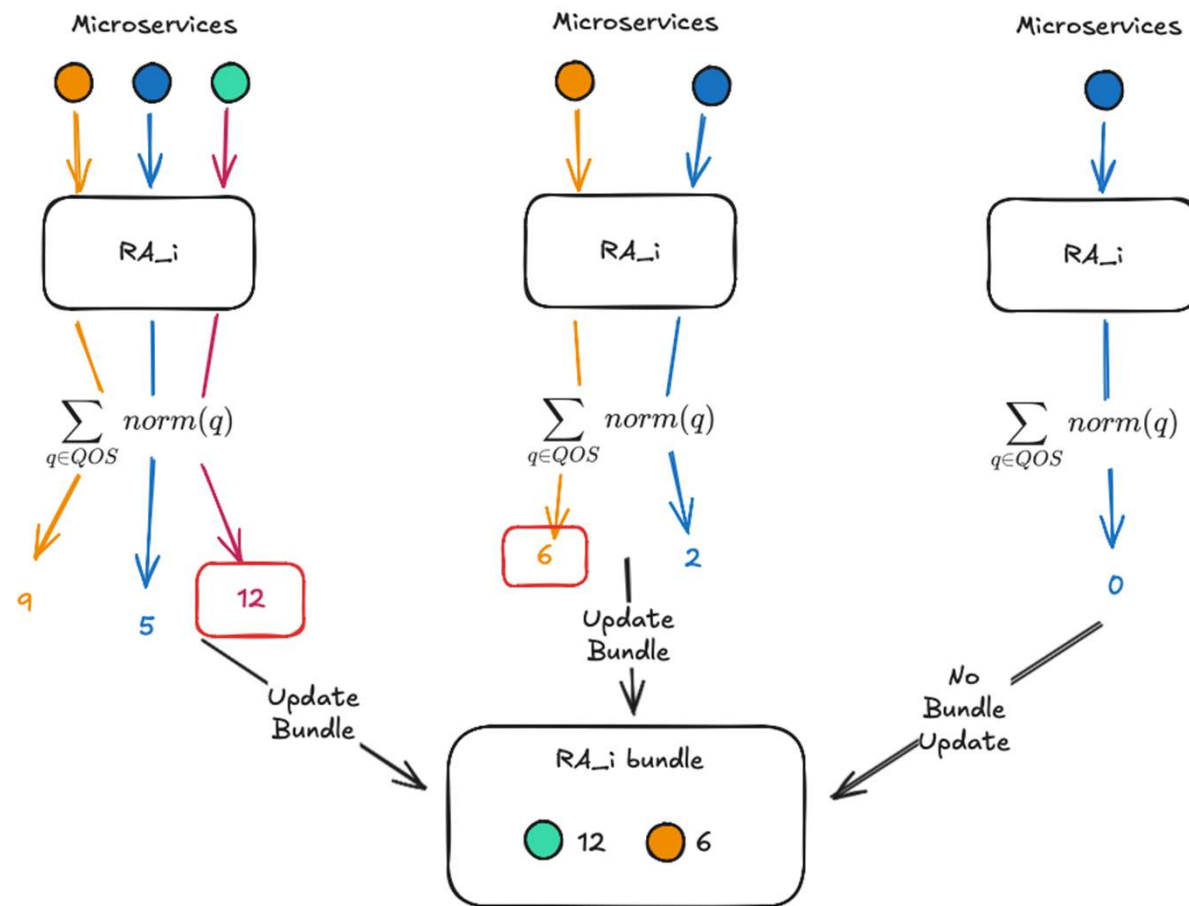
Consensus-Based Decentralized Auctions for Robust Task Allocation

Han-Lim Choi, *Member, IEEE*, Luc Brunet, and Jonathan P. How, *Senior Member, IEEE*

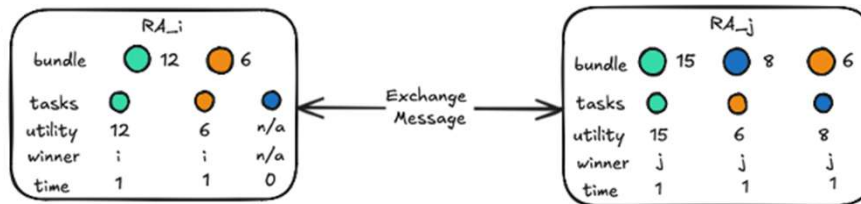
Phase 1: Bundle construction



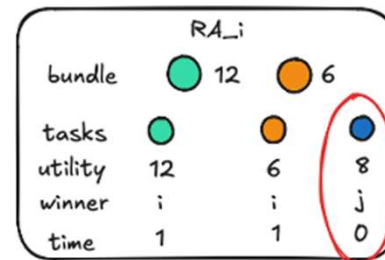
Phase 1: Build Bundle



Phase 2: Consensus



Step 1: Resolve timestamps → RA_i adds info on 



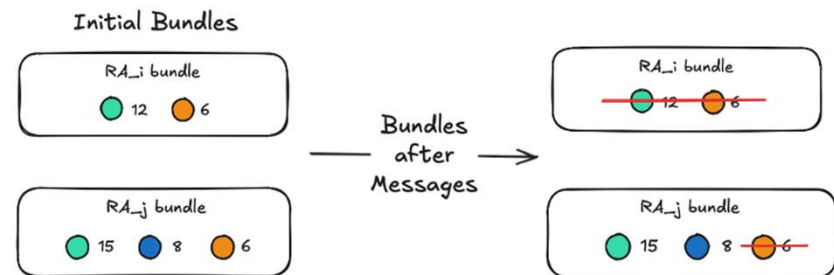
Step 2: Compare utility → RA_i modifies info on 



Step 3: Break ties with ID → RA_j modifies info on 



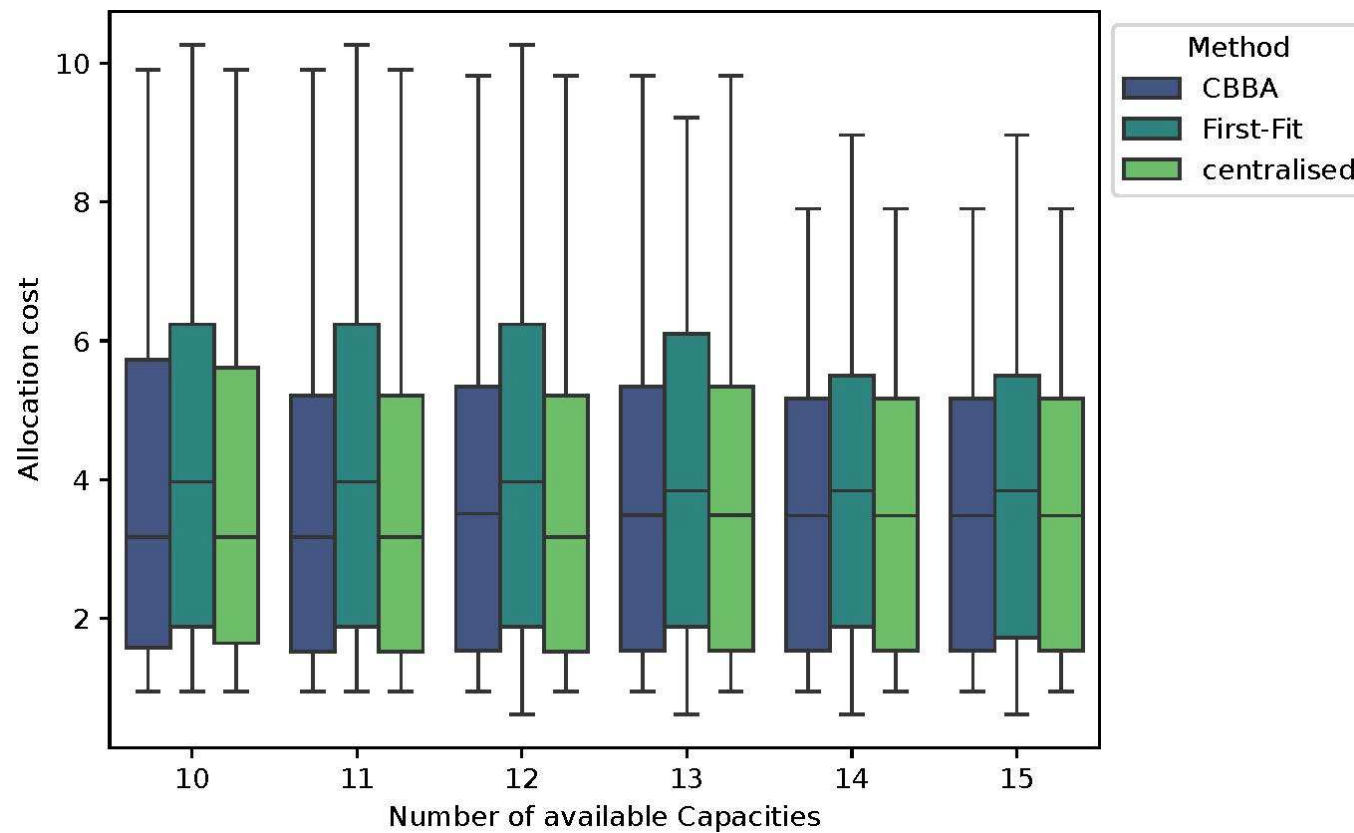
Step 4: Prune bundles of tasks allocated to other agents (and subsequent tasks)



Evaluation

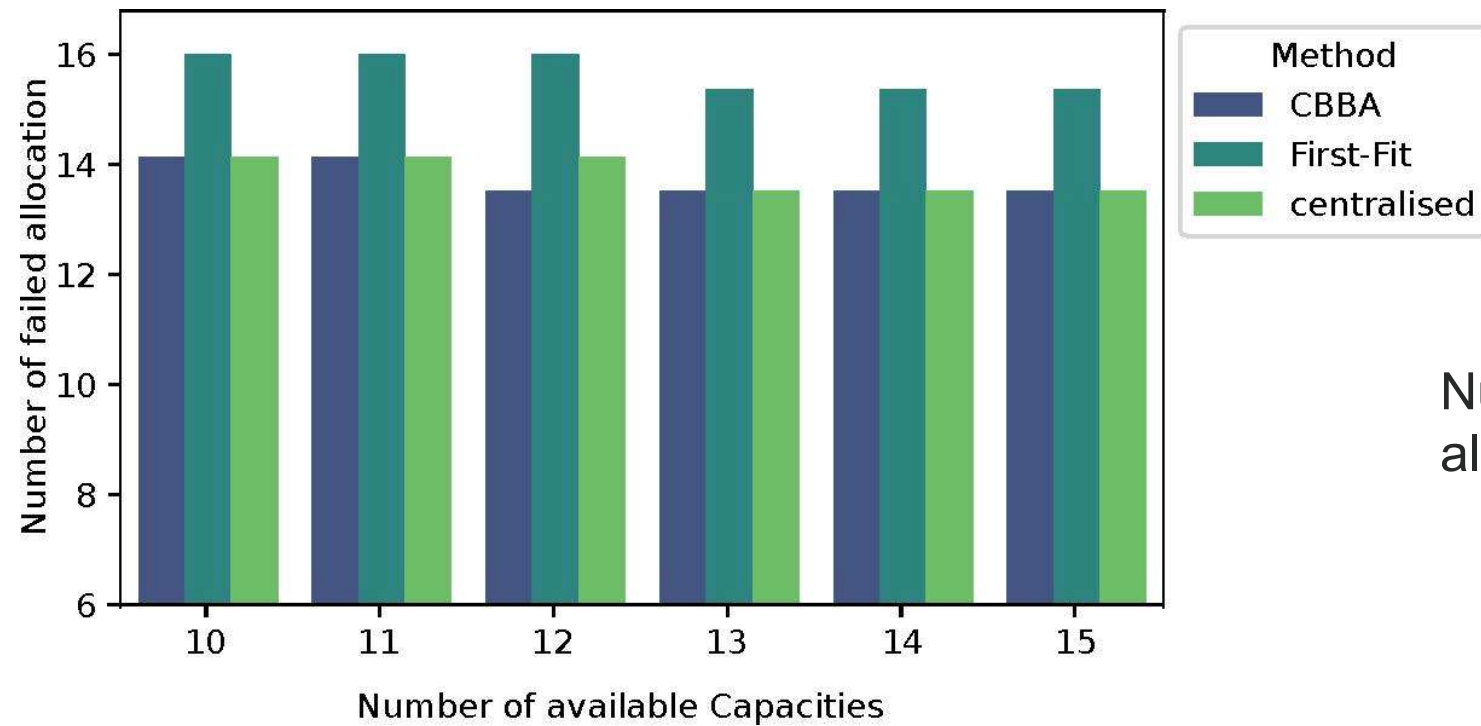
- RAs facilitate access to a capacity
 - Associated Attributes: Latency, Price, Bandwidth, and Energy
- Applications (comprises m microservices)
- Evaluation objective:
 - Performance
 - Efficiency
 - Scalability
- Baselines:
 - Centralised – all enumerations
 - Centralised – First fit

Allocation cost



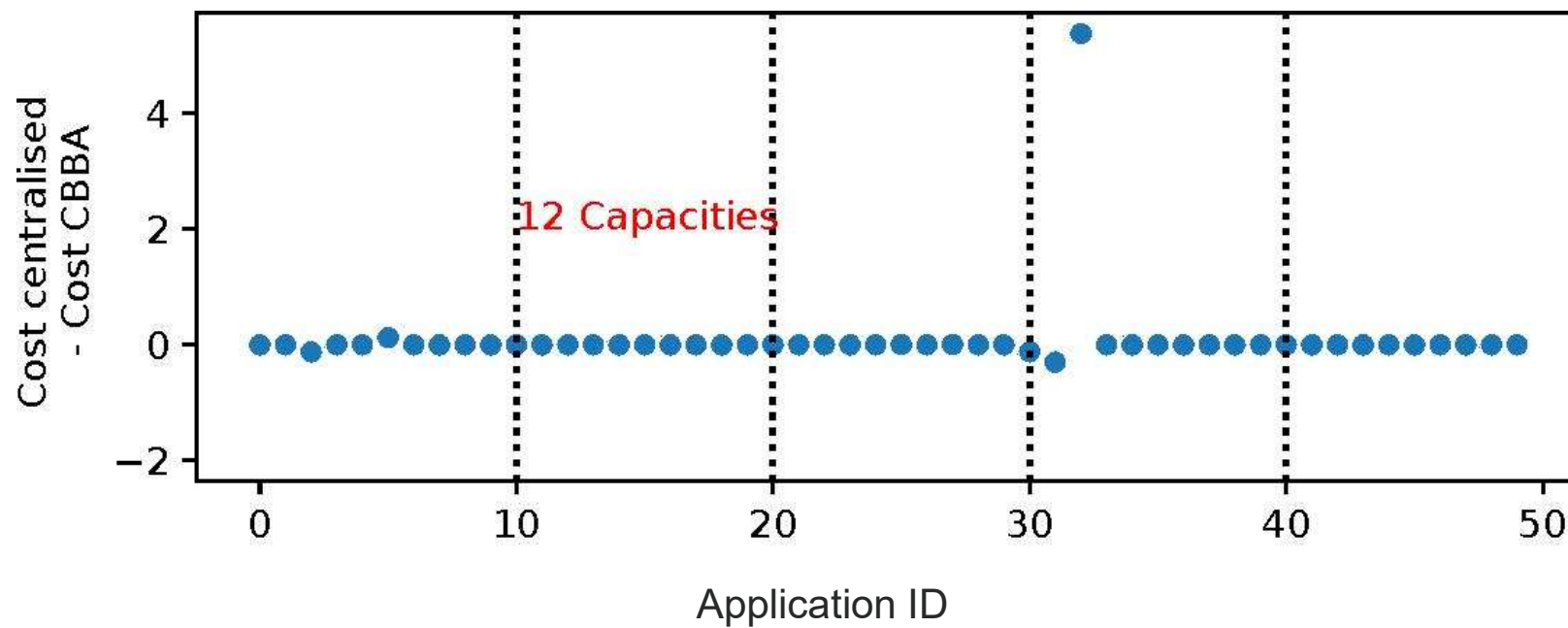
Allocation cost for 50
apps with a varying
number of capacities

Failed allocations

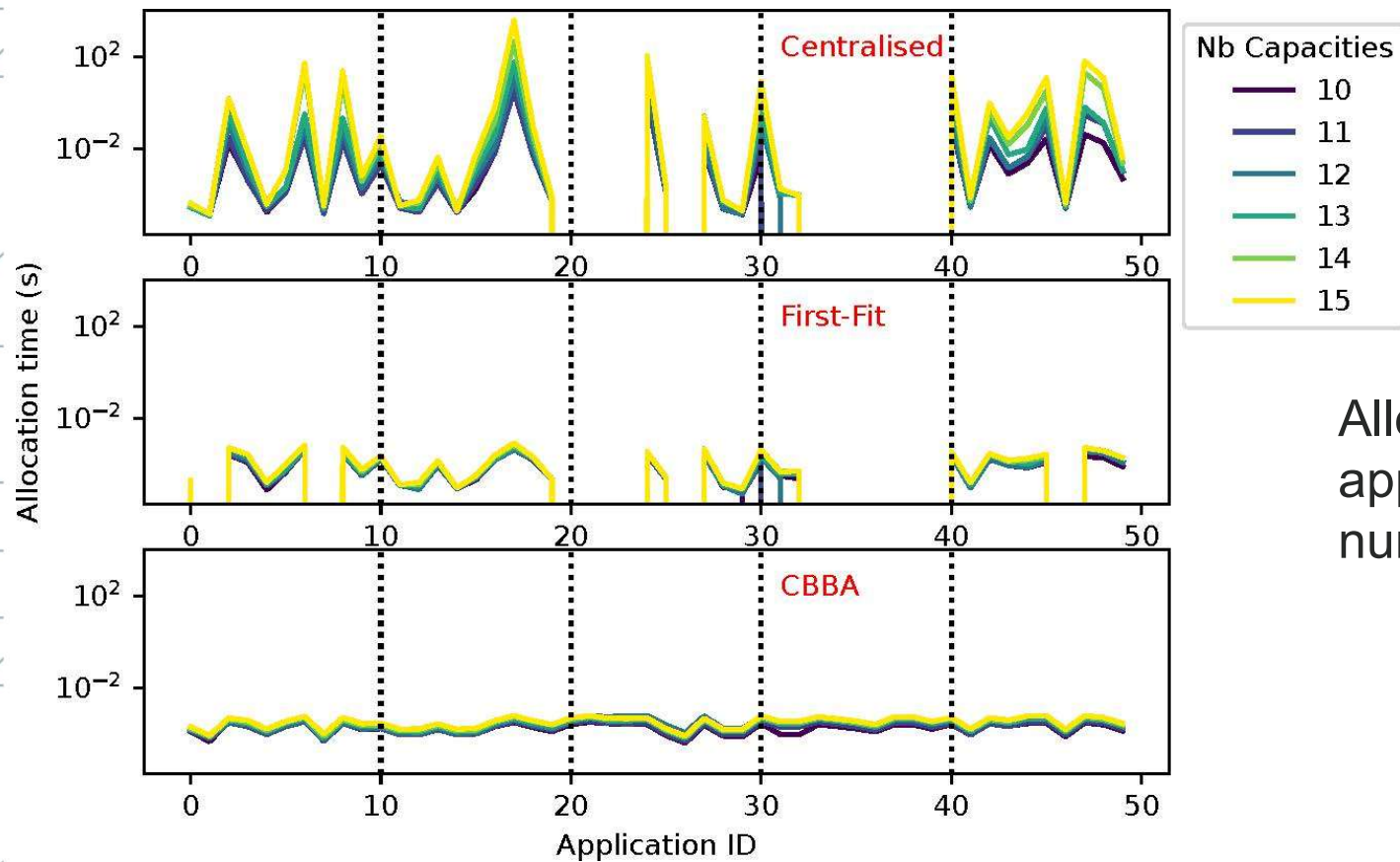


Number of failed
allocations in 50 apps

Cost analysis

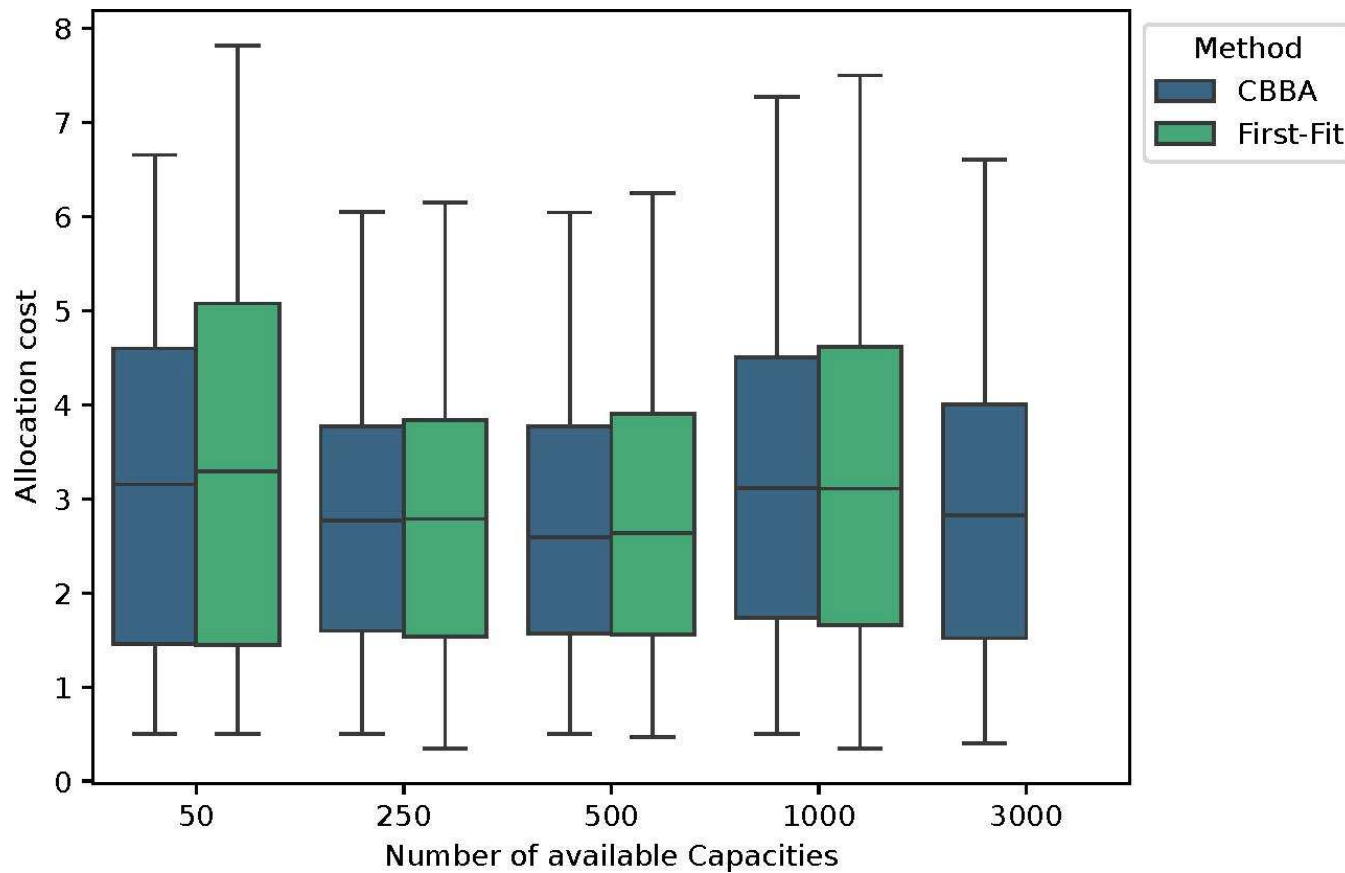


Allocation time



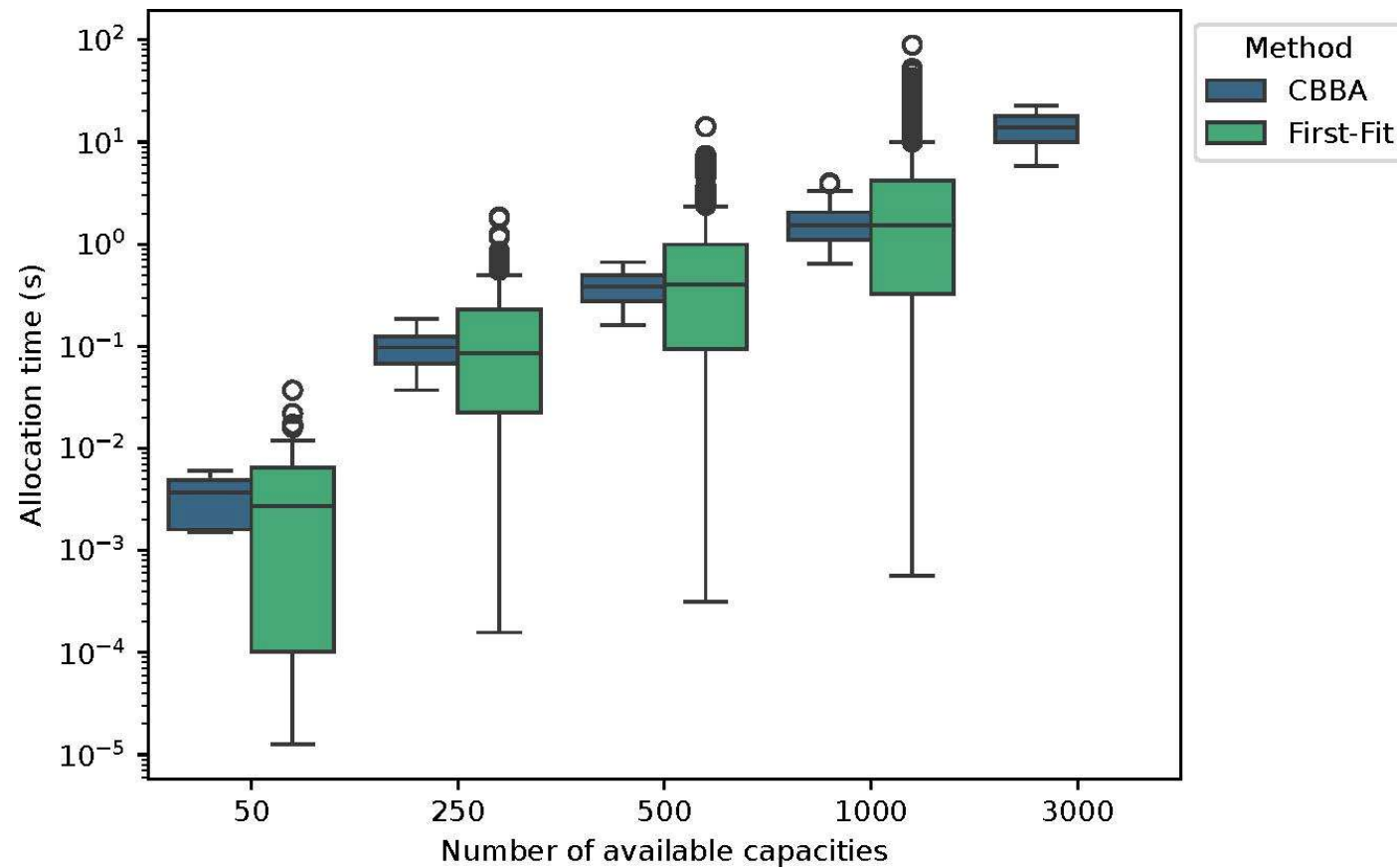
Allocation time by
application with a varying
number of capacities

Large-scale: allocation cost



Allocation cost: for up
to 3,000 capacities

Large-scale: allocation time



Allocation time: for up
to 3,000 capacities

Conclusion

- Match the optimality of exhaustive search, and the efficiency of heuristic methods.
- In large-scale experiments,
 - Compared to a centralised heuristic,
 - 30x times faster
 - Same level of performance
 - The exhaustive search was infeasible



Edinburgh Napier
UNIVERSITY

School of Computing,
Engineering & the
Built Environment

Thank you!



a.Ullah@napier.ac.uk