UNIVERSITY OF CAMBRIDGE

iFogSim-Placement: A Simulation Framework for Edge Service Placement

Joseph Poon, Christian Cabrera Jojoa, and Neil D. Lawrence
Assistant Research Professor
chc.79@cam.ac.uk

Department of Computer Science and Technology

The Service Placement Problem in Edge Computing

The Service Placement Problem in Edge Computing

Optimal assignment of services to edge servers

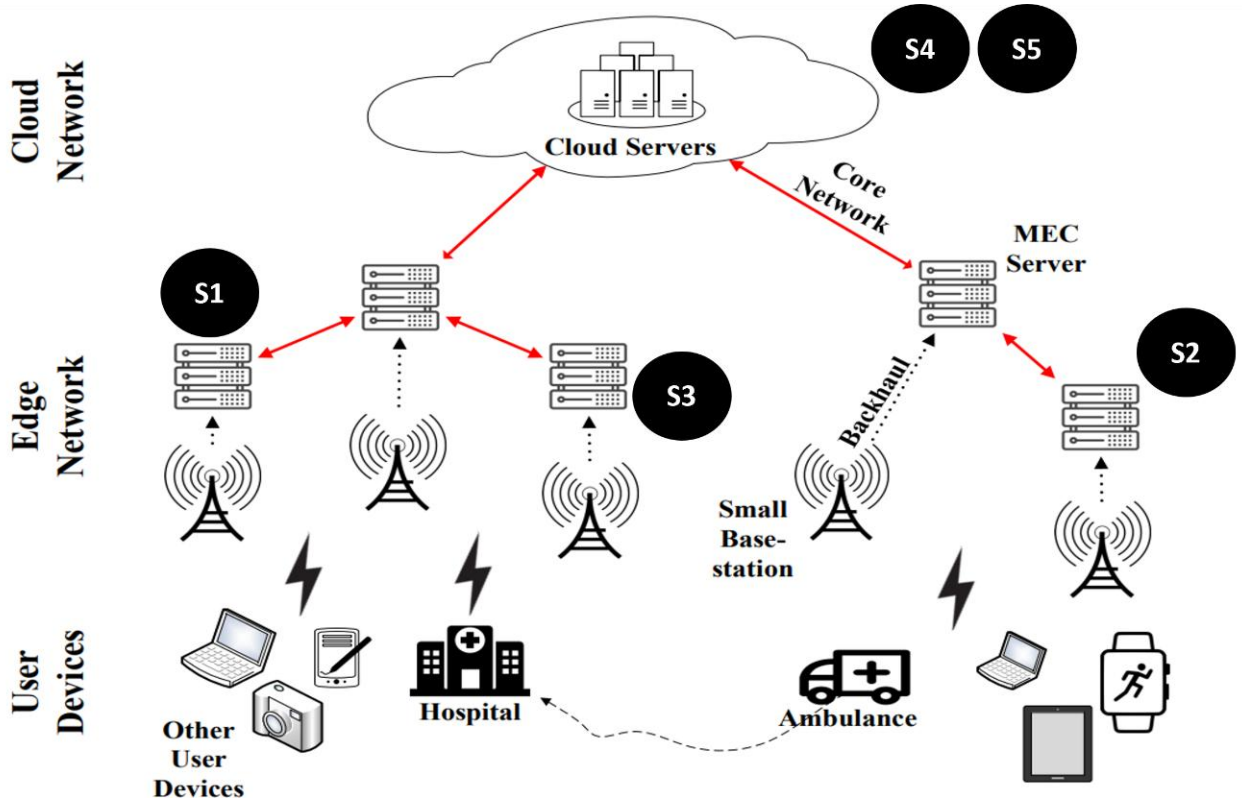
- Multi-objective optimisation problem
- Optimising conflicting goals (e.g., minimsing latency while maximising distribution of resource usage.

$$\min_j lat(D_j)$$

$$\min_i \sqrt{\omega_p \sigma(RR^{CPU})^2 + \omega_m \sigma(RR^{RAM})^2}$$

$$\sum_{k=1}^{n} rr_{ik}^{CPU} \leq r_i^{CPU}$$

$$\sum_{k=1}^{n} rr_{ik}^{RAM} \leq r_i^{RAM}$$



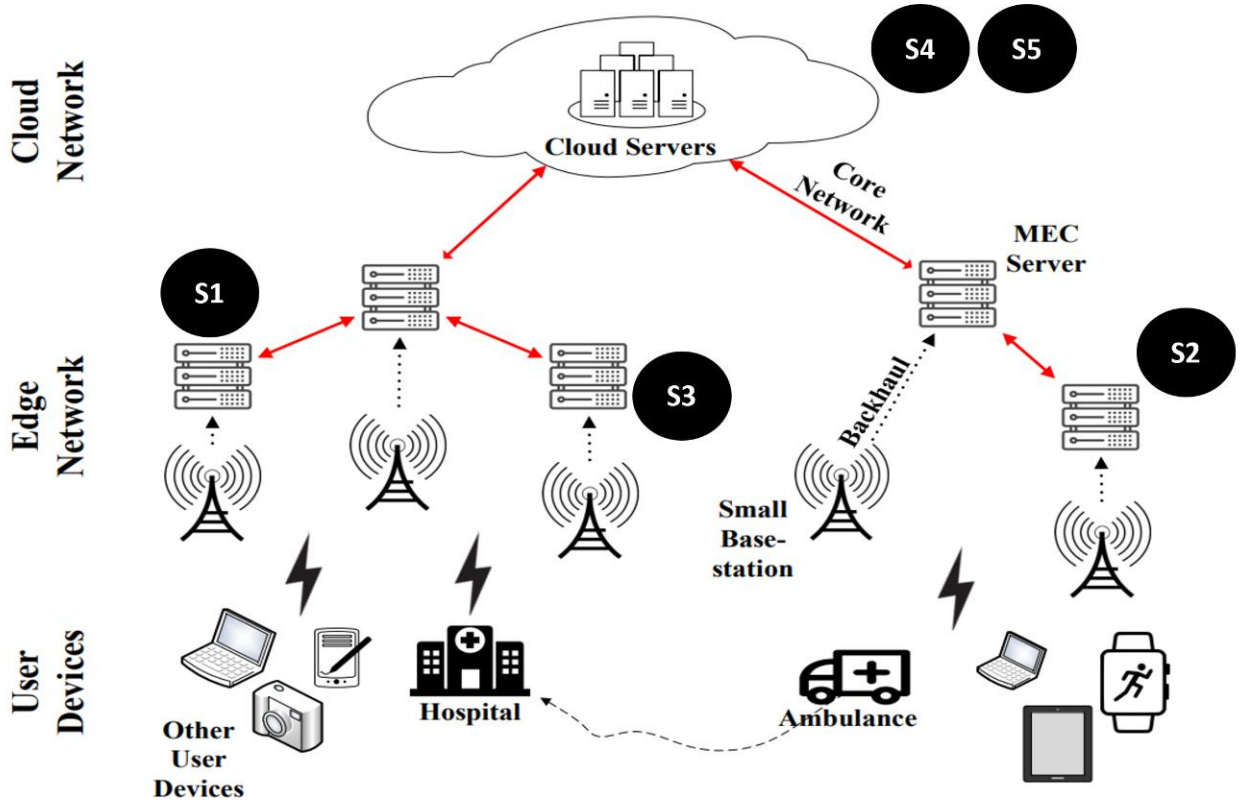
Previous Simulators

Network Simulators

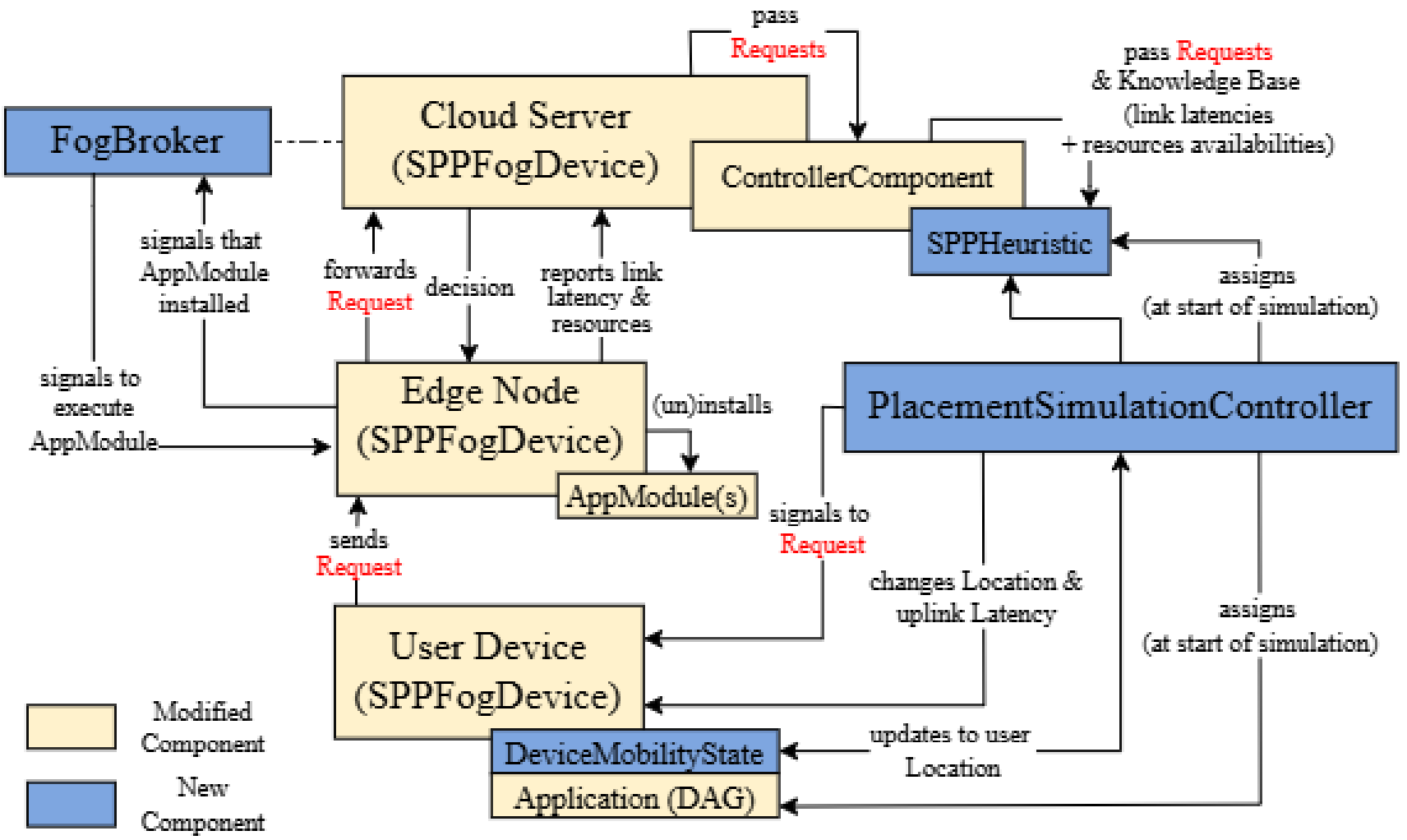
- Network links and latencies between devices and mobility support
- **Unrealistic resource allocation and application execution time**

iFogSim

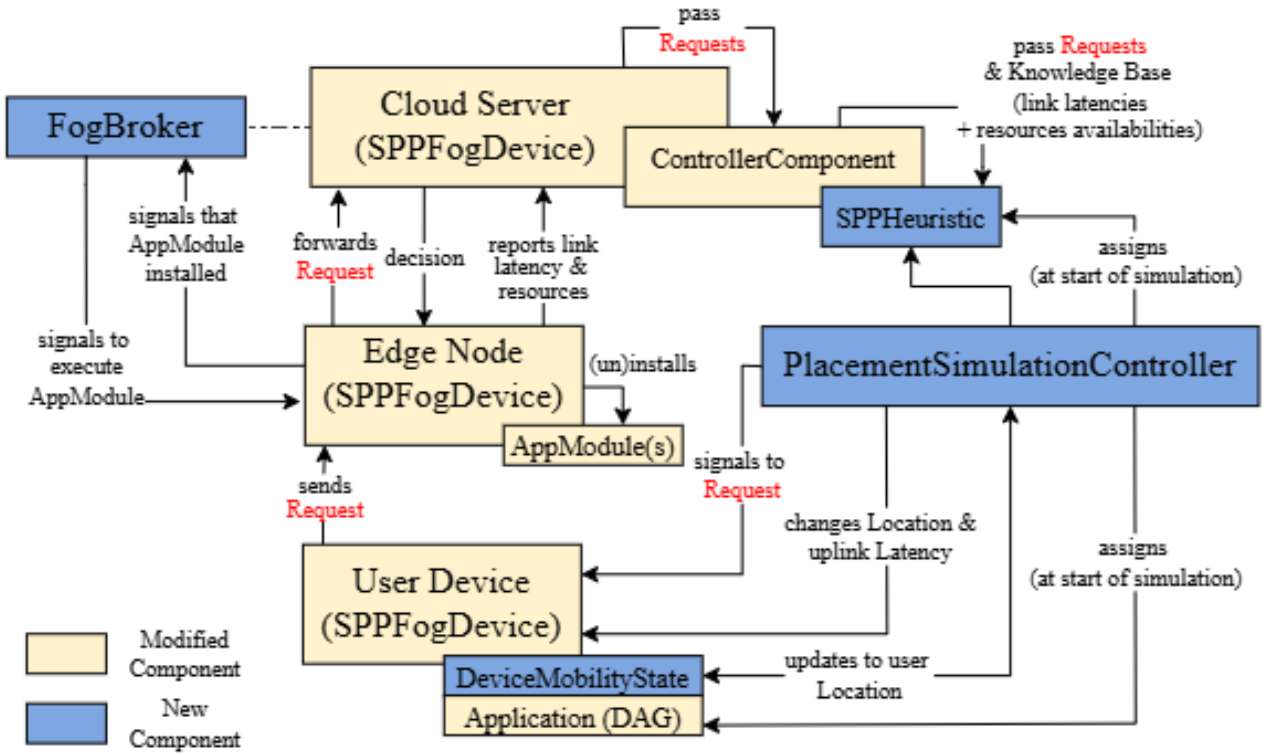
- More natural to simulate edge environments
- Heterogeneous resource availability and more realistic workload simulation
- **Offline service placement**



iFogSim-Placement



iFogSim-Placement



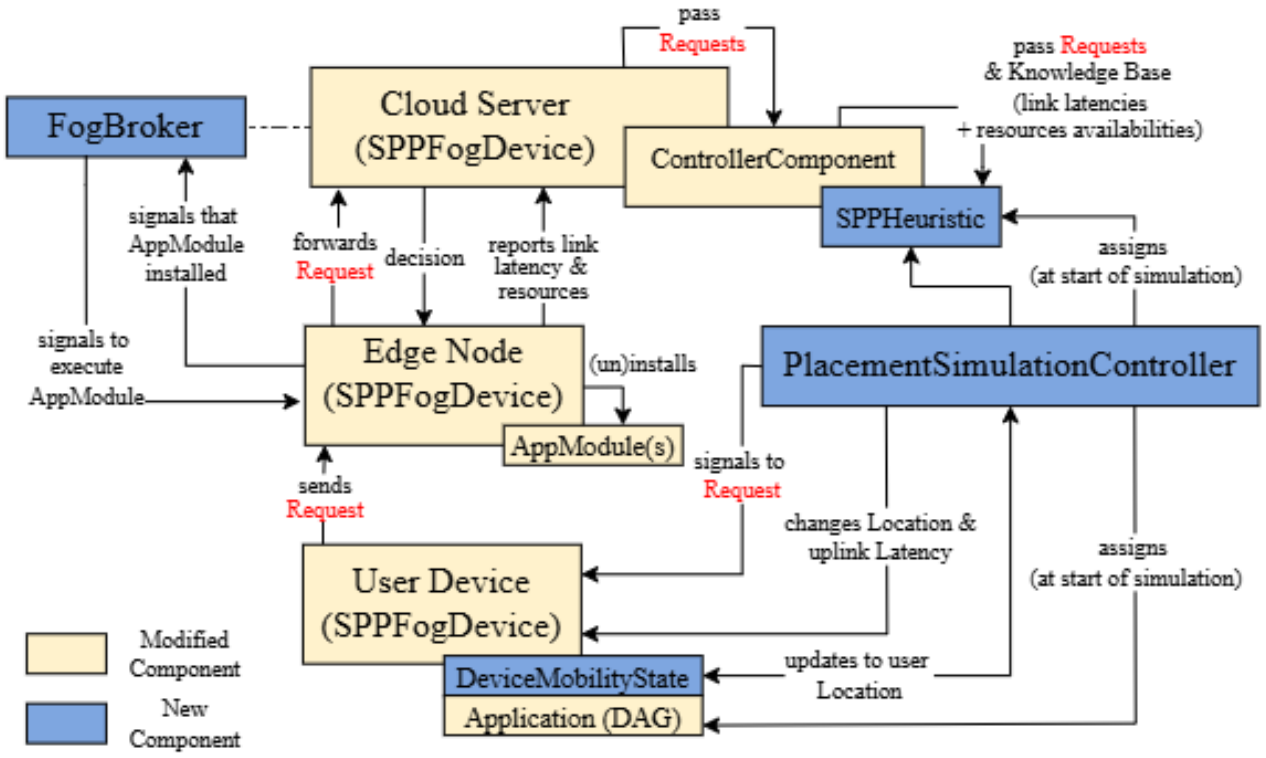
Online Placement Model

- Cloud and edge servers **monitor** requests, triggered by the placement controller.
- The SPPHeuristic **analyses and plans**
- The FogBroker **executes** the plan
- The **knowledge base** is updated with network, edge servers, and allocation information.

Architectural Separation

- **Simulation state** globally available but decision-makers do not have access to it. These only see a **perceived state**

iFogSim-Placement



Dynamic Environment

- Request generation utilising independent Poisson distributions (rate λ)
- Nodes install, execute, and uninstall services. *ExecTime = service size/allocated CPU*
- Mobility support using attractor points and finite state machines (e.g., GraphHopper)

Algorithmic Extensibility

- The SPPHeuristic API uses the **Template Method pattern**. Practitioners can extend the abstract class and implement the **processOnePr** method

iFogSim-Placement

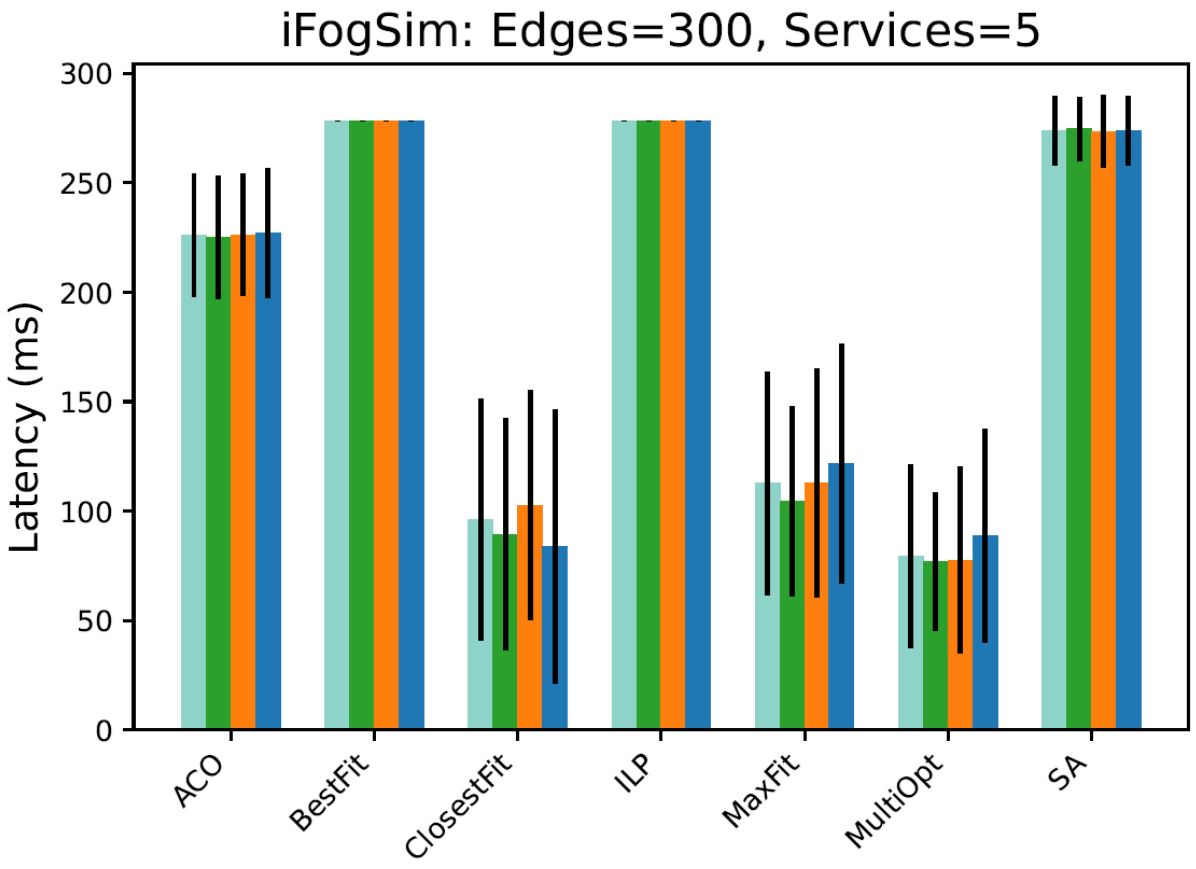
Parameter	Format	Description
Config Files		
`resourcesLocationPath`	"./path/to/csv/file"	Edge node coordinates
`usersLocationPath`	"./path/to/csv/file"	Initial user coordinates
`osmFilePath`	"./path/to/pbf/file"	OSM file (routing)
Boundary polygon	4 points (lat, lon)	Defines simulation area
`minLat, maxLat`	$[-37.823, -37.804]$	Latitude bounds
`minLon, maxLon`	$[144.947, 144.979]$	Longitude bounds
`pointsOfInterest`	json	Points of Interests
YAML Parameters		
N_{edge}	numeric (e.g., 100)	Number of edge nodes
$\lvert\mathcal{A}\rvert$	numeric (e.g., 1300)	Application pool size
`usersPerType`	key:val (e.g., g:100)	Users per type
`intervalValues`	key:val (e.g., g:300s)	Poisson λ
`placementProcessInterval`	numeric (e.g., 4.0s)	Decision-making interval
`experimentSeed`	numeric (e.g., 33)	Seed for
`locationSeed`	numeric (e.g., 42)	Seed for locations
`mobilityStrategySeed`	numeric (e.g., 123)	Seed for mobility
`heuristicSeed`	numeric (e.g., 456)	Seed for placement

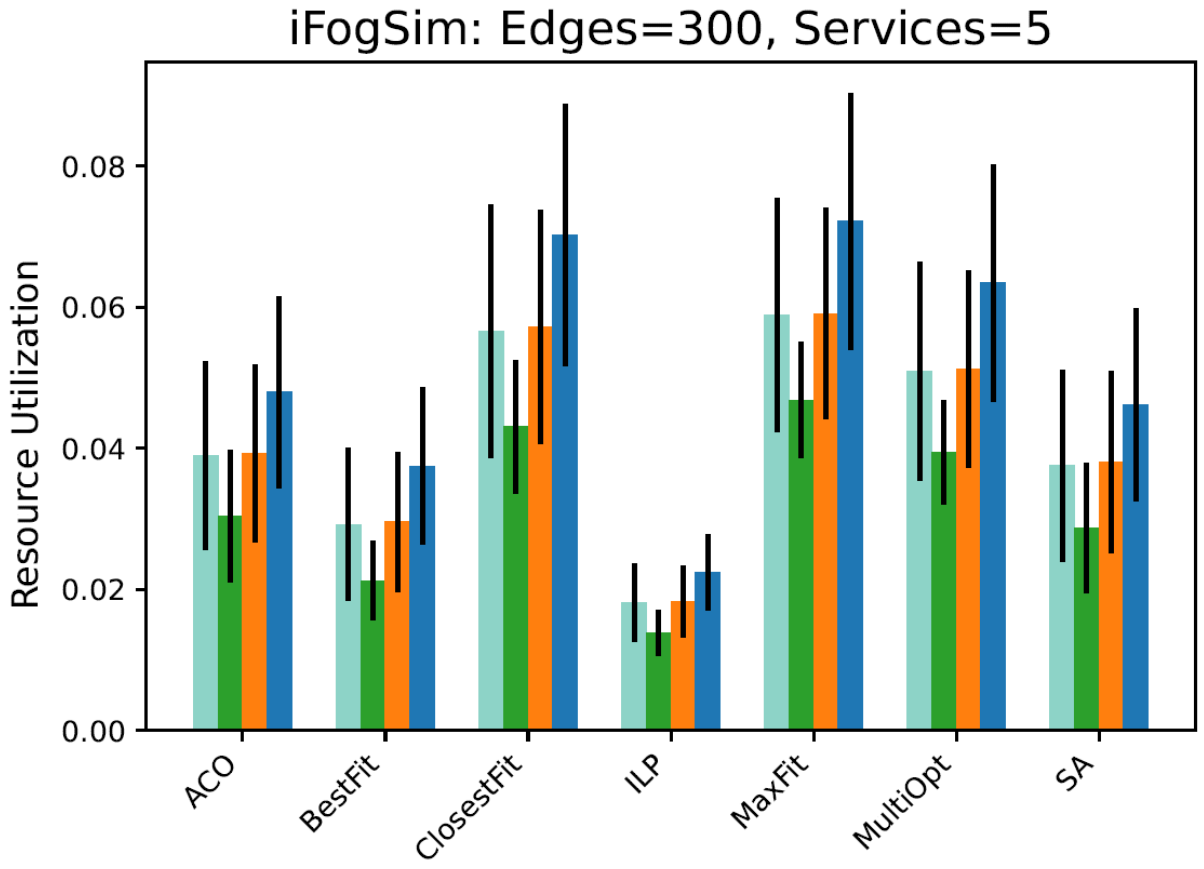
Parameter	Format	Description
Code parameters (in Experiment file)		
R_i^{CPU}	$\sim U[min, max]$	Edge node CPU
R_i^{RAM}	$\sim U[min, max]$	Edge node RAM
$(r_k^{\text{CPU}}, r_k^{\text{RAM}})$	$\sim U[min, max]$	Per-service resources
Cloud CPU / RAM	numeric / numeric	Resources in MB
User MIPS / RAM	numeric / numeric	User device specs in MB
$\delta(d)$	$30 + 0.01 \cdot d$ ms	Link latency (d in km)

Configuration parameters

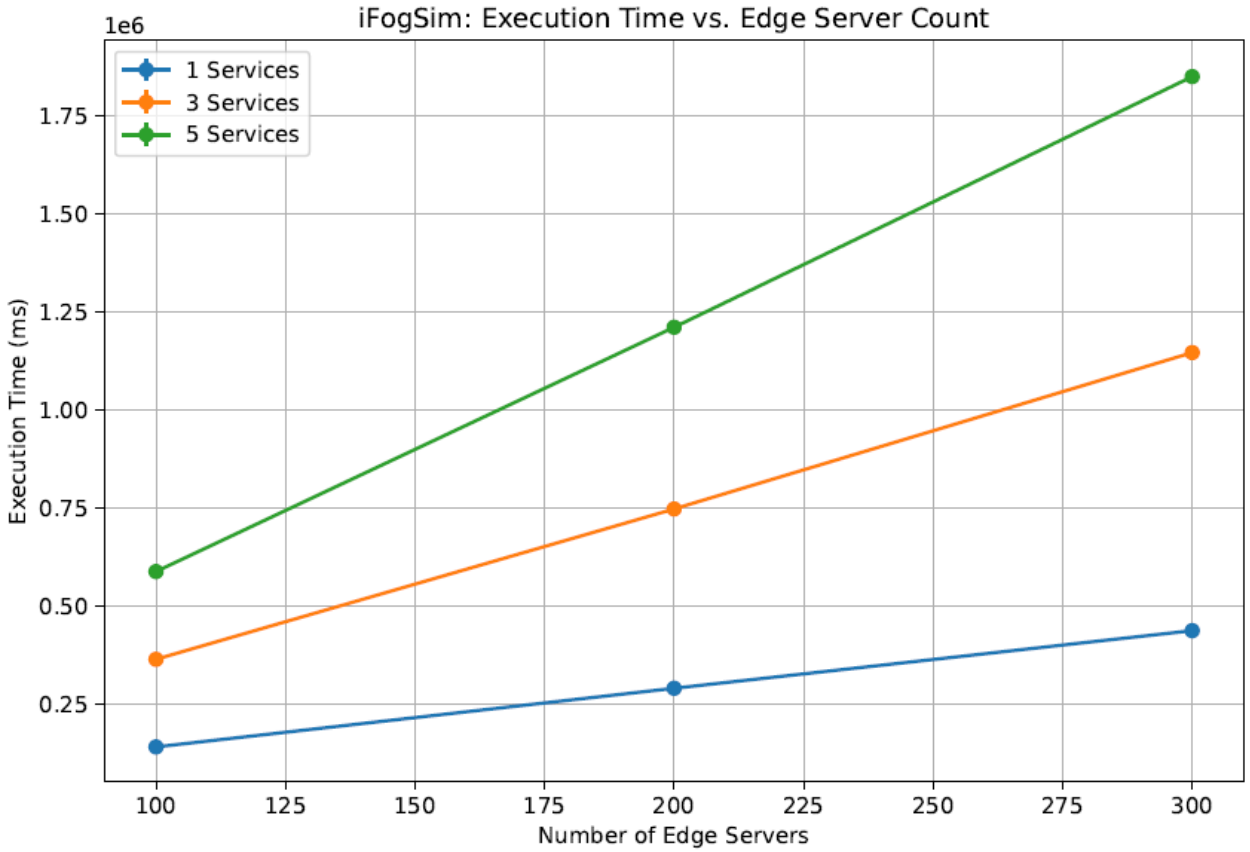
- Geo-localisation and mobility
- Network scale, application's size, requests' frequency, user types, and seeds
- Available resources, application requirements, and network latency

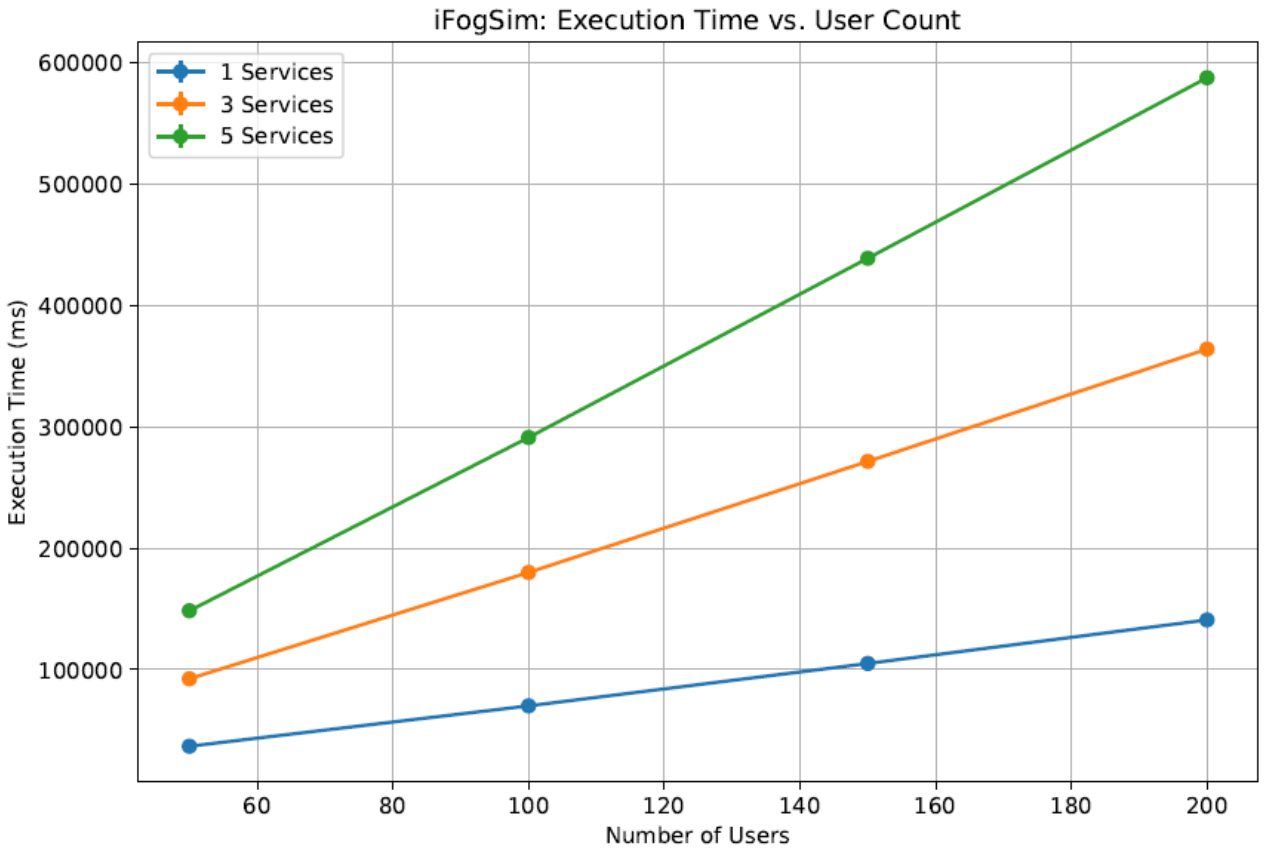
iFogSim-Placement



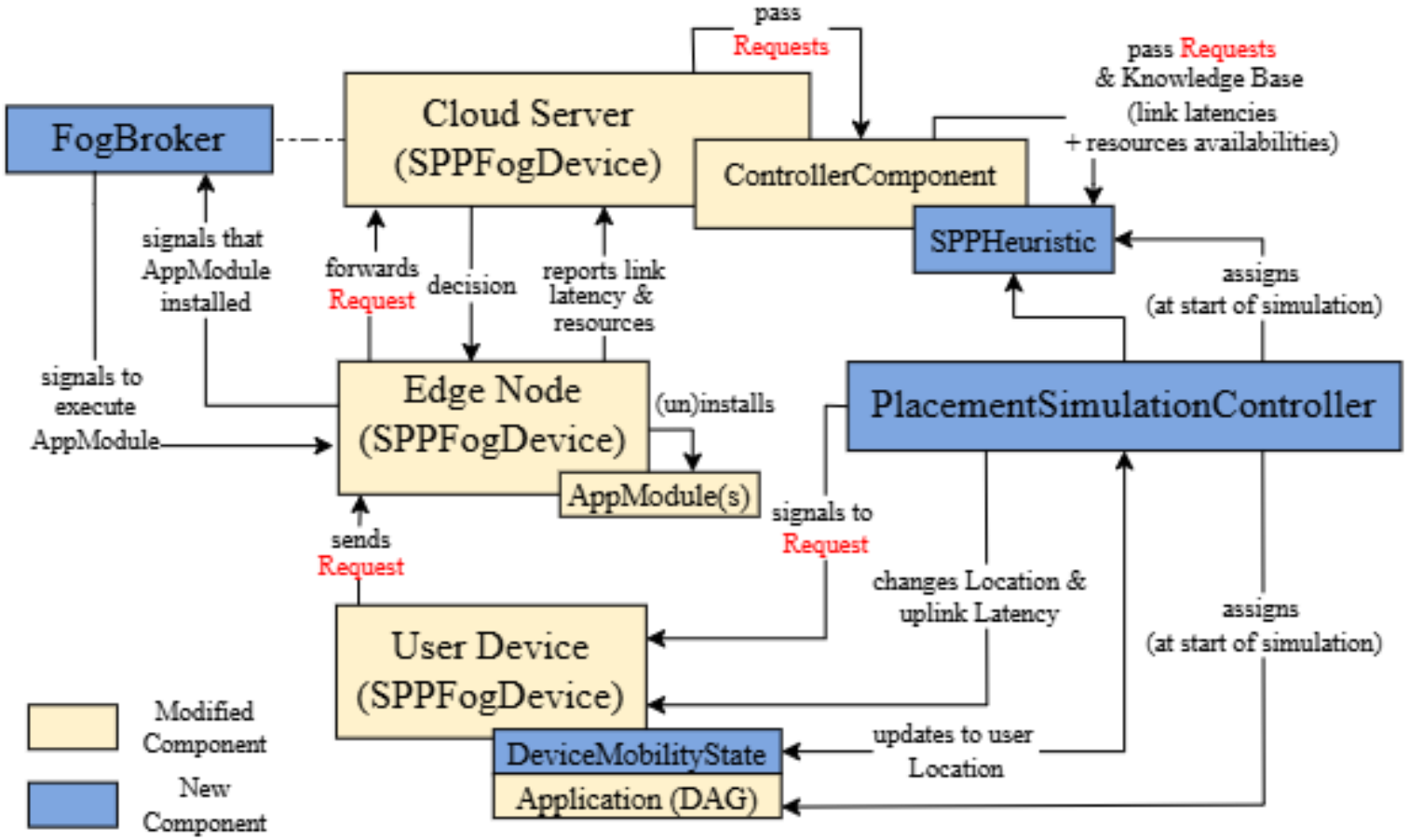


iFogSim-Placement





iFogSim-Placement



Thank you!

chc79@cam.ac.uk