A comparison of static and dynamic μ-service placement for edge computing

Davide Agostini[†], Thiago Alves de Queiroz[‡], Manuel iori[†], Riccardo Lancellotti[†],

[†] University of Modena and Reggio Emilia

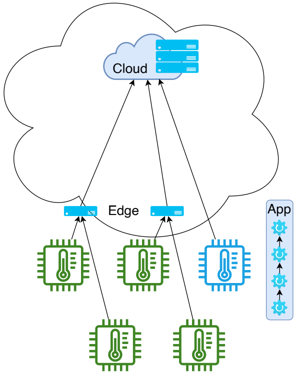
[‡]Federal University of Catalão

18/9/2025

Edge computing overview

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

- Characteristics of modern applications
  - μ-services for modularity and maintainability
  - Applications defined as chains of μ-services
  - Service Level Agreement (SLA)
  - On-demand resources and billing
- Cloud Computing may not be always viable
  - Potentially high cost of data center BW
  - Latency to/from data center
- Elaboration close to the data sources: edge computing
  - Geographically distributed edge nodes
  - Computationally constrained nodes



Reference sceanrio

- Edge computing
  - Must avoid overload
  - Network delays between nodes
  - SLA on response time
  - Must reduce power consumption (green+economic reasons)
  - Dynamic workload fluctuations (e.g., daily patterns)
- Classical approach: periodically compute new deployment
  - During re-deployment μ-service unavailable
  - Unneeded power-on/off cycles on edge nodes
  - Risk of service disruption
- Contributions
  - Model for re-deployment of micro-services
  - Validation in two scenarios
  - Quantify benefits of dynamic placement

Problem overview

- Time divided in time slots t
- Application $a \in \mathcal{A}$
  - SLA for application a: T_a^{SLA}
  - Request rate for a: $\lambda_a(t)$
- Composed by μ-services $m \in \mathcal{M}_a$
  - Service time of m: S_m
  - Variance of S_m: σ_m^2
- Infrastructure of edge nodes $e \in \mathcal{E}$
  - Heterogeneous nodes and network
  - Speedup: C_e
  - Network delay: $\delta_{e,f}(t)$

- Two problem models
  - Static problem (only info @ time t)
  - Dynamic problem (considers also previous allocation @ time $t-1$)
- Decision variables for static problem
  - Deploy m on e: $x_{m,e}(t)$
  - On/Off node e: $y_e(t)$
- Decision variable for dynamic problem
  - Migrate m to/from $x_{m,e}^{+}(t), x_{m,e}^{-}(t)$
  - Switch on/off e: $y_e^{+}(t), y_e^{-}(t)$

Performance model

- Queuing theory approach
- ≥ 1 μ-services on each edge node
  - Different service time $\forall$ μ-service
  - Mixture of distributions $\rightarrow$ M/G/1
  - Average service time:

$$S_e(t) = \frac{1}{C_e} \cdot \sum_{m \in \mathcal{M}} x_{m,e}(t) \frac{\lambda_m(t)}{\lambda_e(t)} S_m$$

  - Variance $\sigma_e^2(t)$ can be also computed
- Model validated through simulation

- Pollaczek-Khinchin formula for waiting time

$$W_e(t) = \frac{S_e^2(t) + \sigma_e^2(t)}{2} \cdot \frac{\lambda_e(t)}{1 - \lambda_e(t) S_e(t)}$$

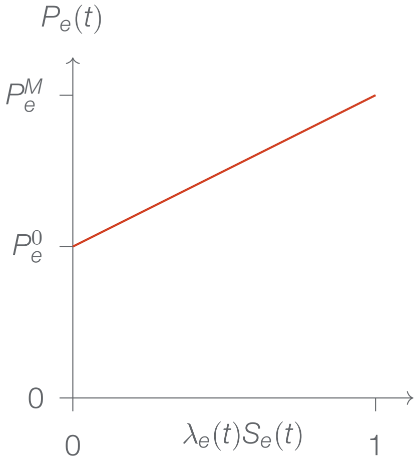
- Can compute application response time $R_a(t)$
  - Service time
  - Wait time
  - Network delays

Energy model

- Energy consumption of each Edge node
- Linear model (widely used in literature)
- Power consumption:

$$P_e(t) = y_e(t)P_e^0 + (P_e^M - P_e^0)\lambda_e(t)S_e(t)$$

  - Idle power contribution if edge node is on
  - Contribution proportional to node utilization
- Additional energy consumption for state changes
  - Powering on/off edge nodes
  - Migrating μ-services



Static placement

- Double objective
  - Minimize power consumption $\min P(t) = \sum_{e \in \mathcal{E}} P_e(t)$
  - Minimize response time $\min R(t) = \sum_{a \in \mathcal{A}} \frac{\lambda_a(t)}{\Lambda(t)} R_a(t)$
- Constraints
  - No overload $\lambda_e(t) S_e(t) \leq (1 - \epsilon) y_e(t)$
  - SLA $R_a(t) \leq T_a^{SLA}$
  - No duplicate placement $\sum_{e \in \mathcal{E}} x_{m,e}(t) = 1$

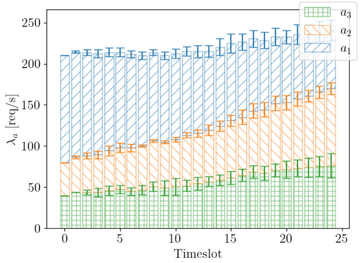
Dynamic placement

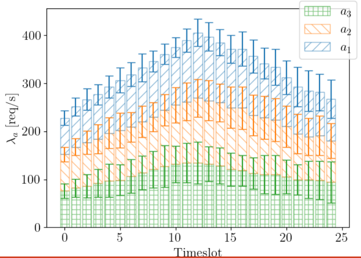
UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

- New decision variables
  - Powering on/off edge nodes $y_e^+(t), y_e^-(t)$
  - Migrating to/from edge nodes $x_{m,e}^+(t), x_{m,e}^-(t)$
- Knows previous deployment status $\bar{x}_{m,e}(t-1), \bar{y}_e(t-1)$
- Revised energy objective (includes energy and cost of service disruption)
  - Power consumption $W_p \sum_{e \in \mathcal{E}} P_e(t)$
  - Contribution for powering on nodes $W_y^+ \sum_{e \in \mathcal{E}} y_e^+(t)$
  - Contribution for powering off nodes $W_y^- \sum_{e \in \mathcal{E}} y_e^-(t)$
  - Contribution for migrating μ-services $W_x \sum_{m \in \mathcal{M}} \sum_{e \in \mathcal{E}} \left(x_{m,e}^-(t) + x_{m,e}^+(t)\right)$
- Additional constraints
  - Migrations have from and to $\sum_{e \in \mathcal{E}} x_{m,e}^+ = \sum_{e \in \mathcal{E}} x_{m,e}^-$
  - Migration from previous state $x_{m,e}^-(t) \leq \bar{x}_{m,e}(t-1)$
  - Migration to different node $x_{m,e}^+(t) + x_{m,e}^-(t) \leq 1$
  - Power on if previously off $y_e^+(t) \leq 1 - \bar{y}_e(t-1)$
  - Power off if previously on $y_e^-(t) \leq \bar{y}_e(t-1)$

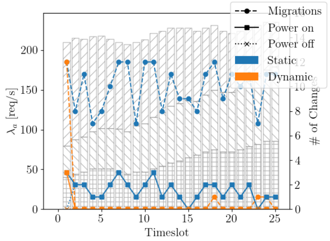
Experimental setup

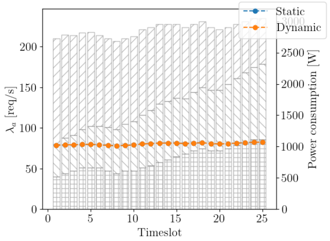
- Common setup
  - 3 application
  - Application with 4-5 μ-services
  - 24+1 time slots
  - Application service time: 10 ms $\rightarrow$ SLA=100 ms
  - Edge node power $\in [200 - 400]$ W
- Two scenarios
  - Constant-intensity workload
  - Variable-intensity workload

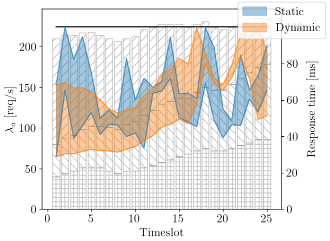




Constant-intensity workload

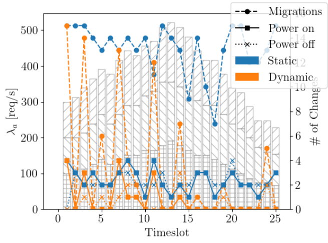


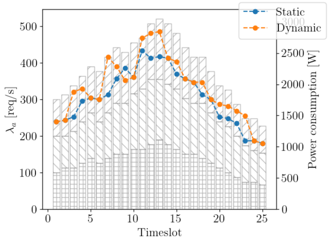


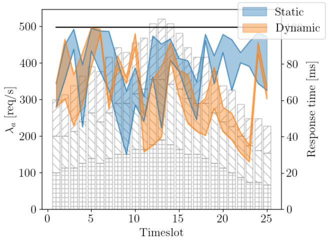


- Minimal changes when dynamic approach is used
- Identical Power consumption
- No SLA violations

Variable-intensity workload







- Many migrations, limited amount of power on/off transitions
  - Dynamic approach still reduces service disruption
- Static approach uses slightly less energy
  - accept sub-optimal deployment to reduce disruption
- No SLA violations

Summary of results

Dynamic approach clearly preferable

- Reduction of migrations $\in$[75%-94%]
- Reduction of Power-on $\in$[72%-92%]
- Reduction of Power-off $\in$[75%-100%]
- Power consumption increases by $\leq$7.5%
- Variable intensity workload is more challenging

	Static	Dynamic	Δ %
Constant-intensity			
Energy[Wh]	26076.0	26076.0	0%
Migrations	249	15	94.0%
On	39	3	92.3%
Off	36	0	100.0%
Variable-intensity			
Energy [Wh]	45100	48700	7.39%
Migrations	328	81	75.3%
On	59	17	72.0%
Off	56	14	75.0%

Conclusions

- Challenges of Edge computing
  - Distributed and heterogeneous infrastructure
  - Limited computational resources
- Proposal of new model for deployment of applications
- Validation of proposed model for two scenarios
- Significant savings in terms of service disruption
  - Savings $\geq 72\%$ for every type of change
- Future work
  - Heuristics (GA/VNS)
  - Receding Horizon and prediction
  - Multi-version μ-services