HOW TO ASSESS AND OPTIMIZE THE ENERGY EFFICIENCY OF MICROSERVICES PLACEMENT

Imane Taleb

PhD Student

Supervisors: Jean-Loup Guillaume and Benjamin Duthil

L3i Laboratory (La Rochelle Université)

Profile and scope of the thesis

- Thesis framework
  - Scientific priority of the LUDI Institute: sustainability and digital technologies.
  - Objectives: produce tools to model and analyse the energy consumption of microservices-based applications and optimise their placement.
  - Funding: doctoral contract.

Cloud limits: Introduction to Fog/Edge

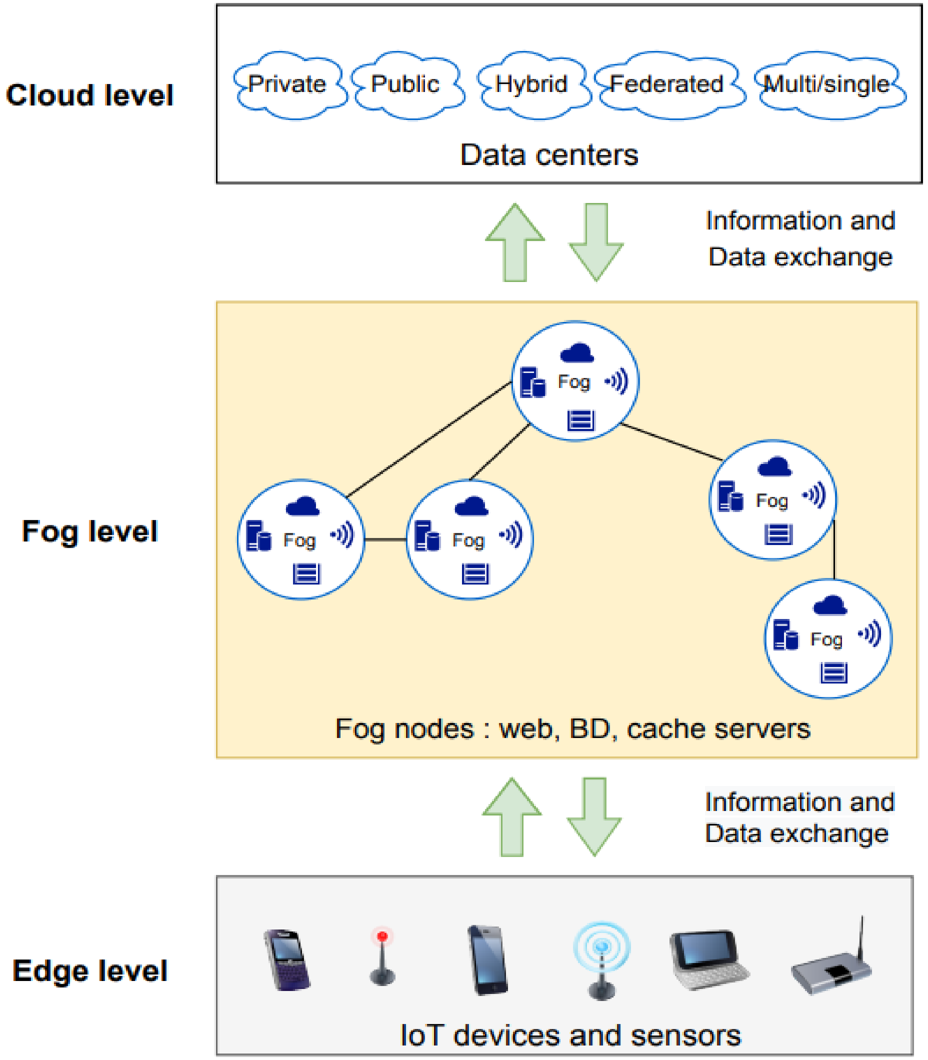
Cloud

- Centralized data processing and storage, away from users.
- Highlight limitations, such as latency, bandwidth, and privacy concerns.

Fog/Edge

- Fog Computing brings resources closer to the cloud, while Edge Computing brings them even closer to the data source.
- Bandwidth and latency Reduction.
- Local Availability
- The state of the Internet connection and its speed are less taken into account.

Cloud-Fog-Edge Continuum



Centralized
- analytics
- processing
- storage
- warehousing

Decentralized
- local data analytics
- monitoring
- data processing
- storage
- execution

Local
- monitoring
- real time processing
- micro storage
- execution

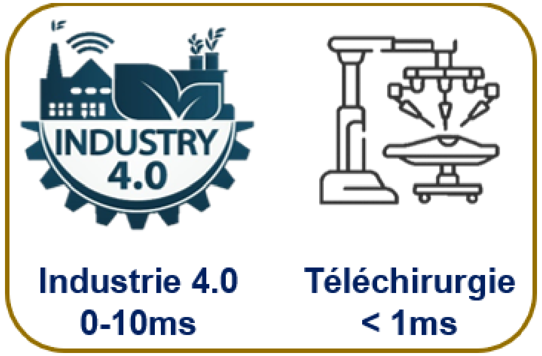
Imane Taleb

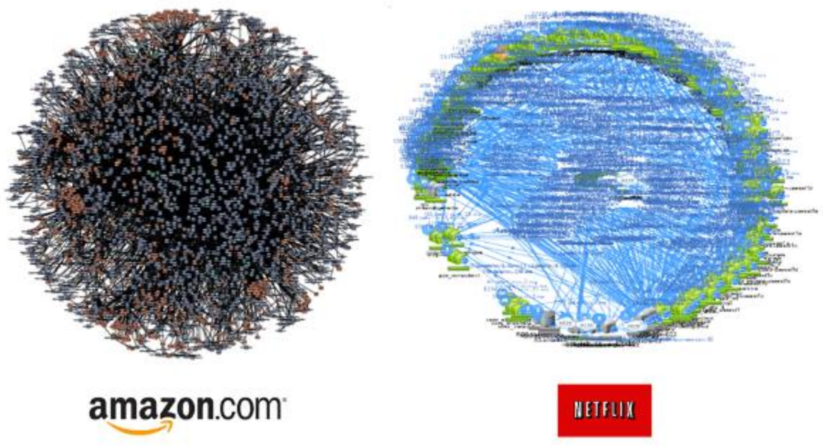
Microservices approach

- Fast, interchangeable components, easy to adapt and to scale.

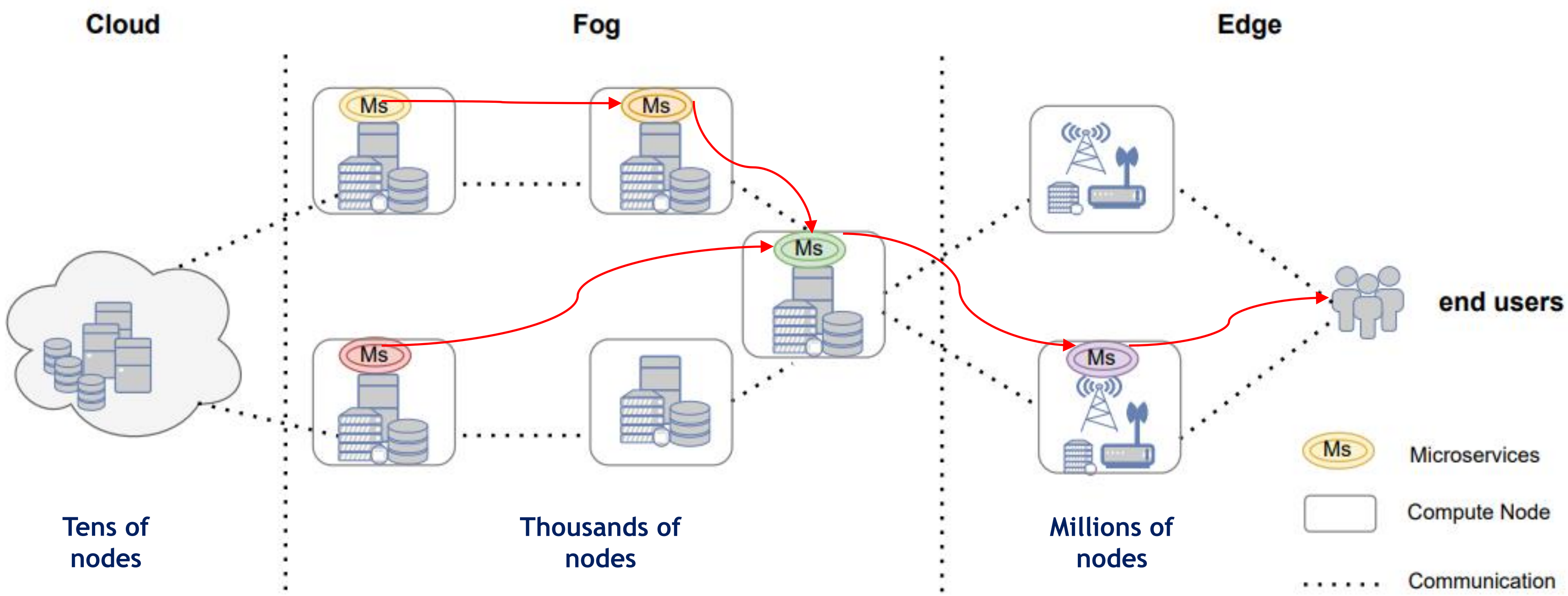
- An effective solution to provide services requiring low communication delay and high quality of service.

- Limited transit time in the network infrastructure.



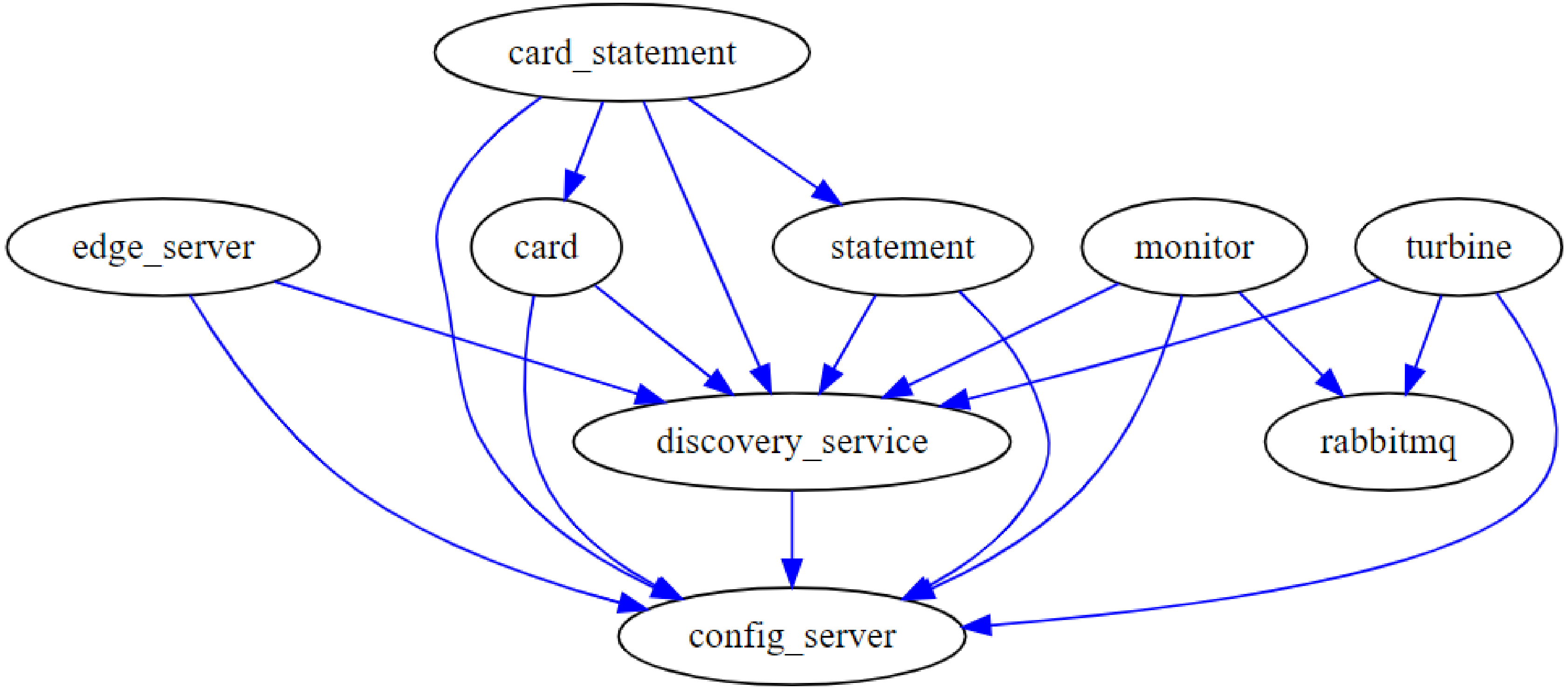


Microservices placement in Fog/Edge



Network infrastructure

Microservices Graph representation



A typical application based on MS architecture
(a DAG representation)

Characteristics and constraints

Infrastructure characteristics

- Geo-distributed infrastructure with a large number of nodes.
- Heterogeneous nodes in terms of location and resources (process, storage).
- Dynamic infrastructure (failure, mobility, network congestion).

Microservices characteristics

- Several dependencies and function calls between microservices.
- Heterogeneous microservices in terms of required resources.

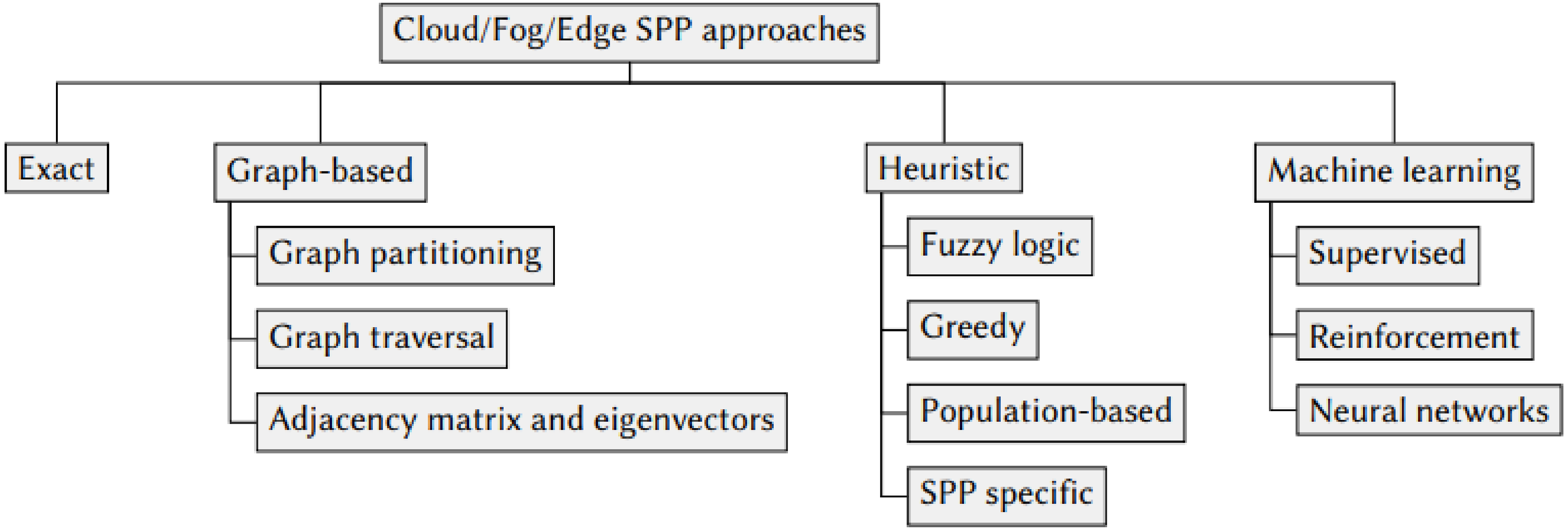
➔ Misplacing microservices network induces congestion and increase latency, energy consumption, cost and service availability (QoS).

Microservices placement in Fog/Edge

- Problem 1: How to place a microservices-based application to optimize latency and energy consumption?
  - A multi-criteria discrete optimization problem.
  - It is an **NP-hard** class problem.

- Problem 2: Which metrics are considered to evaluate the energy efficiency of microservices placement?
- Problem 3: How can we limit the exchange of data between microservices in the network?
- Problem 4: How to distribute load among microservices instances? (Long-term).

SPP placement and chaining approach



Categorization of approaches used to solve the SPP

SPP criteria

Criteria	# of articles
Latency	24
Cost	19
Quality of Service	19
Energy	14
Resources	10
Others (incl. performance)	18

- A multi-criteria discrete optimization problem (combinatorial optimisation)
- NP-hard class

Discussions

- A lack of public repositories providing microservices dependencies and workload datasets.
- Comparing microservices SFC's placement and chaining results with monolithic VNFs (in the telecommunication context).
- Rahman et al. (1) have introduced a small dataset of 20 microservices graphs dependencies to address this gap.
- Random graphs network dependencies: Barabasi-Albert networks, growing random networks (2).
- Mobility trajectories like the San Francisco Taxi dataset used to dynamically address the SPP.

Problem modelling

Network infrastructure

We model the physical topology of Cloud-Fog-Edge architecture as a connected graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ where vertices represent execution nodes and edges are network links.

Physical nodes

Each node $n_i \in \mathcal{N}$ has the following characteristics:

- A speed of processing capacity cpu_i in MIPS.
- A storage size $storage_i$ in TB.
- A memory size ram_i in GB.
- A power consumption characteristics of when the device in not used p_i^{idle} and when it used to the maximum p_i^{max}.

Network infrastructure

We model the physical topology of Cloud-Fog-Edge architecture as a connected graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ where vertices represent execution nodes and edges are network links.

Links

Network communication links can be physical or virtual. Each link is identified by the two nodes $\{n_i, n_j\}$ it connects and has the following characteristics:

- A physical link l is characterized by a bandwidth bw_l and a latency lc_l.
- A logical link l' represents the optimal path L composed of physical links between two nodes. $bw_{l'} = min_{l \in L}(bw_l)$ and $lc_{l'} = \sum_{l \in L}(lc_l)$, i.e. the minimum bandwidth on the path and the sum of the latencies on the path.
- A power consumption characteristics of the devices n_i and n_j: p_i^{idle}, p_j^{idle} p_i^{max}, p_j^{max} This can be considered to determine the unit cost of the link energy consumption.

Microservices applications

A micro-services application $\mathcal{A} = (\mathcal{S}, \mathcal{M})$ can be modeled as a directed acyclic graph (DAG) where the nodes represent the services $\mathcal{S} = (s_1, s_2, ..., s_t)$ and the edges represent dependencies requests between the services.

Nodes characteristics

Each service $s_i \in S$ requires some resources consumption:

- A requested CPU mi_i in million instructions.
- A requested RAM ram_i in GB.
- A requested storage size $storage_i$.

A micro-services application $\mathcal{A} = (\mathcal{S}, \mathcal{M})$ can be modeled as a directed acyclic graph (DAG) where the nodes represent the services $\mathcal{S} = (s_1, s_2, ..., s_t)$ and the edges represent dependencies requests between the services.

Links

Each directed edge $\mathcal{M}_{s_i,s_j}$ that connect s_i to s_j represent the request need and the data dependencies between this microservices. It is characterised by:

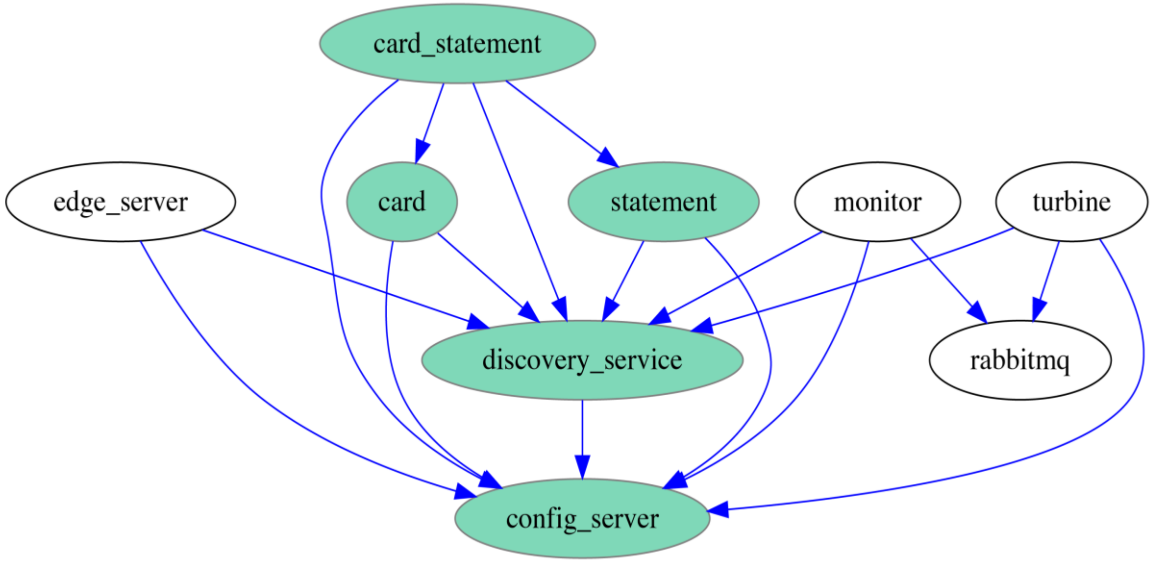
- A source and a destination
- A message size $Mess_{i,j}$

Problem statement

- Objective: MFP placement to reduce latency and energy consumption

$$\begin{cases} \text{Minimize } E_{tot} \\ \text{where} \quad E_{tot} = E_{nod} + E_{com} \end{cases} \quad \text{s.t.} \quad Lat_{dep} <= Lat_{req}$$

- E_tot: energy total
- E_nod: energy node treatment
- E_com: energy communication between microservices
- Lat_dep: deployement latency
- Lat_rep: required latency



- Microservices Function Paths (MFP) from DAG microservices dependencies

The total energy consumption is the sum of the energy from the node execution and the energy from network communication.

$$\begin{cases} \textit{Minimize } E_{tot} \\ \text{where} \quad E_{tot} = E_{nod} + E_{com} \end{cases} \text{s.t.} \quad Lat_{dep} <= Lat_{req}$$

$$\mathrm{E_{nod}} = \sum_{i=0}^{S-1} \sum_{j=0}^{N-1} \overbrace{\frac{\text{cpu(required by a service}_i)}{\text{cpu(max of a node}_j)}}^{\text{CPU consumption}} \underbrace{\left(p_j^{\text{max}} - p_j^{\text{idle}}\right) + p_j^{\text{idle}}}_{\text{Electric power of the execution node}}$$

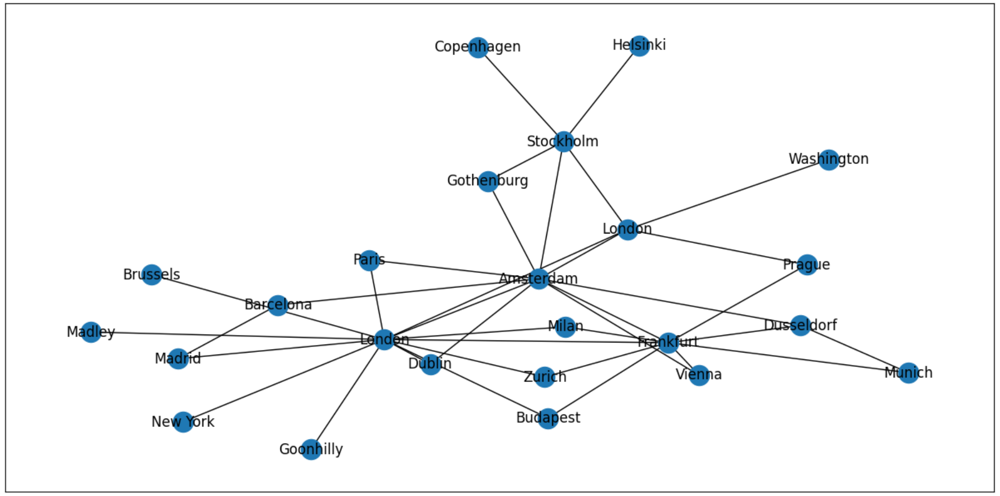
The total energy consumption is the sum of the energy from the node execution and the energy from network communication.

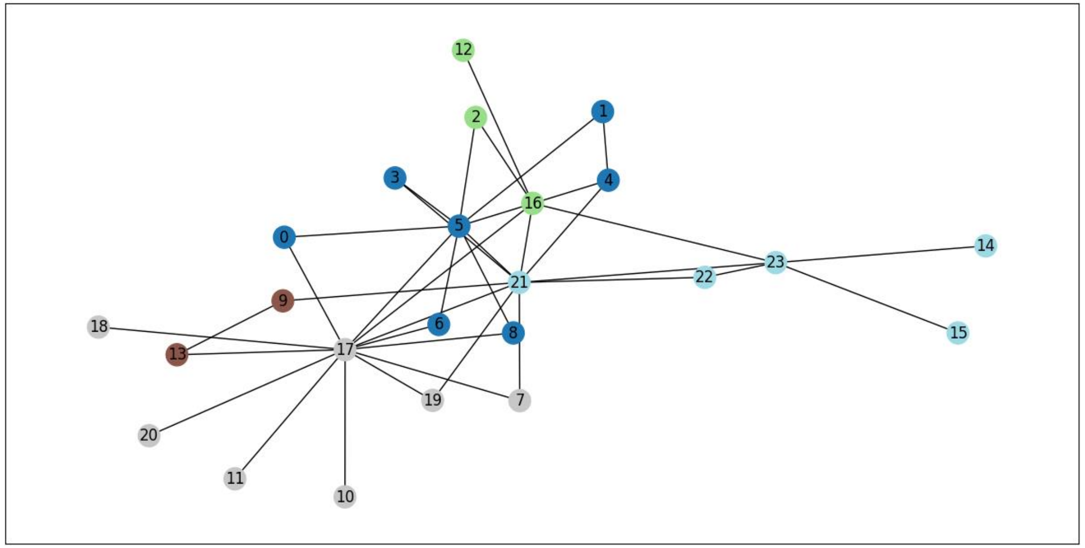
$$\begin{cases} \textit{Minimize } E_{tot} \\ \text{where} \quad E_{tot} = E_{nod} + E_{com} \end{cases} \text{s.t.} \quad Lat_{dep} <= Lat_{req}$$

$$\mathrm{E_{nod}} = \sum_{i=0}^{S-1} \sum_{j=0}^{N-1} \frac{\text{cpu(required by a service}_i)}{\text{cpu(max of a node}_j)} \left(p_j^{\text{max}} - p_j^{\text{idle}}\right) + p_j^{\text{idle}}$$

$$\mathrm{E_{com}} = \sum_{(i,l)\in S} \sum_{(j,p)\in N} \overbrace{\frac{data_{(i,l)}}{bw_{(j,p)}} + lc_{(j,p)}}^{\text{Data transmission capacity}} \underbrace{\left[p_j^{\text{max}} + p_p^{\text{max}} - p_j^{\text{idle}} - p_p^{\text{idle}}\right]}_{\text{Electrical power between two nodes}}$$

Community detection





Louvain Community detection

- Heuristic placement based on community detection
  - *A community of network nodes represent a cluster where the power electric cost is minimal compared to outside communities.*

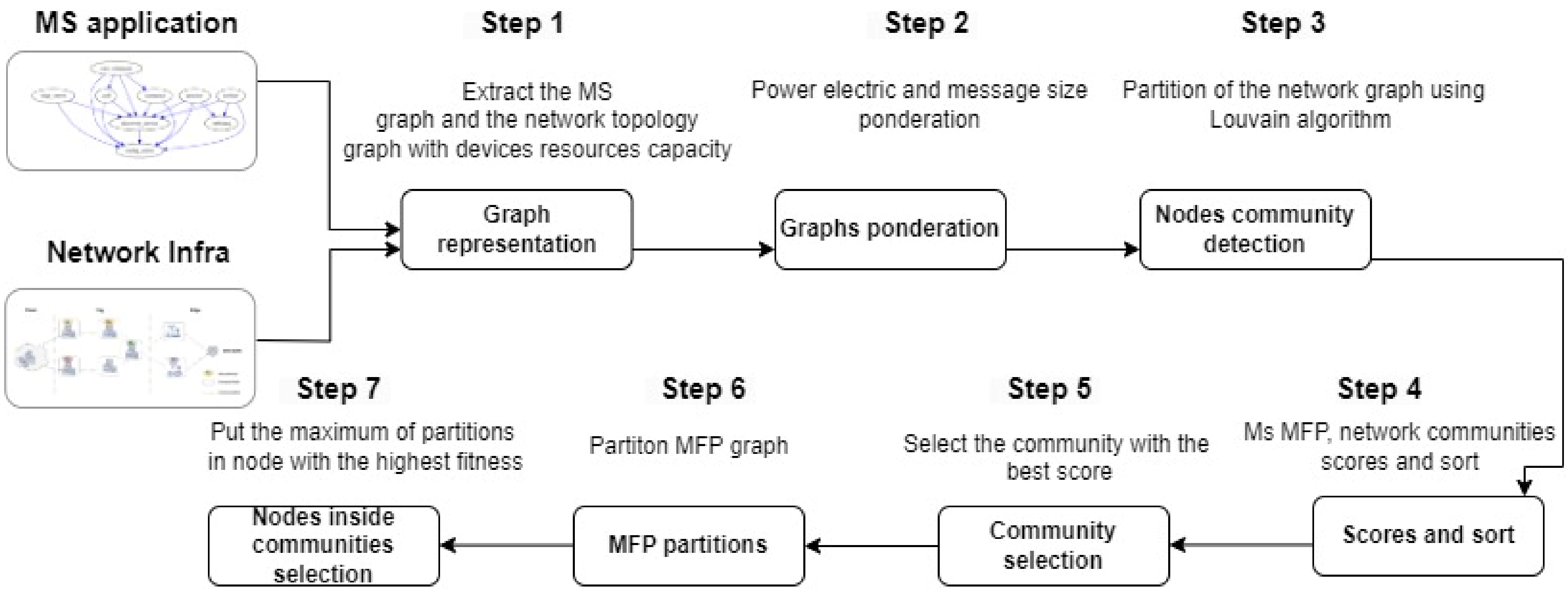
[1] The internet topology zoo

...ne Taleb

Informatique Image Interaction
Univ. La Rochelle

La Rochelle Université

Microservices heuristic placement



Microservices heuristic placement

- Allocate entire MFP microservices within a single community to minimize inter-community exchanges (Placing the largest MFP in the smallest community).

- Assumption is based on the notion that the most energy-intensive.

- Placing as many microservices as possible in the same node to reduce communication overhead.

Experiment scenario

Network topology

- Network topology Zoo Dataset / SDNlib
- > Topologies with [14,90] nodes > 1 Cloud (highest betweenness centrality), 50 % Fog, 50%Edge
- > Resource capacity: Cloud > Fog > Edge

Microservices MFPs

- > 20 Microservices applications
- > Service number [2-15] depending on MFPs YAFS

Simulator

- Python Fog simulator (4)

Future work

- Include users nodes in the network topology.
- Microservices instances management (microservices used by several users in different communities).
- Ensure a dynamic placement according to the user requests and resources changes.
- Include another metrics in the model: Global MS environmental footprint that includes : gray energy, resources, primary energy equivalent in Kwh, manufacturing equipment, energy consumption in DCs, networks(Switches, routers), water, greenhouse gas, etc.

Conclusion

- **Context:**
  - Fog-Edge computing provides a platform to deploy Microservice-based applications.
  - Microservices and data misplacement can result in an increase in latency, energy consumption.
- **Problems:**
  - How to manage Microservices placement in Cloud-Fog-Edge continuum infrastructure in order to optimize latency and energy consumption?
- **Contributions:**
  - An under construction method to manage microservices placement
- **Future work:**
  - Consider the availability and partition tolerance when placing microservices instances and data replicas.

Reference

1. M. I. Rahman, S. Panichella, and D. Taibi, “A curated dataset of microservices-based systems,” SSSME-2019, 20192. J.E. Ibarra-Esquer, F. González-Navarro, B. Flores-Rios, L. Burtseva, et M. Astorga-Vargas. Tracking the evolution of the internet of things concept across different application domains. Sensors, 17(6) :1379, 2017.

2. V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” Journal of statistical mechanics: theory and experiment, vol. 2008, no. 10, p. P10008, 2008.

3. T. Bui, C. Heigham, C. Jones, and T. Leighton, “Improving the performance of the kernighan-lin and simulated annealing graph bisection algorithms,” in Proceedings of the 26th ACM/IEEE Design Automation Conference, pp. 775–778, 1989.

4 I. Lera, C. Guerrero, and C. Juiz, “Yafs: A simulator for iot scenarios in fog computing,” IEEE Access, vol. 7, pp. 91745–91758, 2019

5. M.I. Naas, J. Boukhobza, P. Raipin, et L. Lemarchand. An extension to ifogsim to enable the design of data placement strategies. In 2nd IEEE International Conference on Fog and Edge Computing ICFEC 2018. Washington DC, USA. Pages 1–8. 2018.

6. M.I. Naas, L. Lemarchand, J. Boukhobza, et P. Raipin. A graph partitioning-based heuristic for runtime iot Page 12 sur 13data placement strategies in a fog infrastructure. In Proceedings of the 33rd Annual ACM Symposium on Applied Computing SAC 2018. Pau, France. Pages 767–774. 2018.

7. F. Sujuan et S. Eiko. Self-organizing networks (SON) in 3GPP long term evolution. Nomor Research GmbH. Munich. Germany. Vol. 20. Pages 1-15. 2008.

8. R. Mayer, H. Gupta, E. Saurez, U. Ramachandran. Fogstore: Toward a distributed data store for fog computing. 2017 IEEE Fog World Congress FWC. Santa Clara, California, USA. Pages 1-6, 2017.

9. M.I. Naas, L. Lemarchand, P. Raipin, et J. Boukhobza. IoT Data Replication and Consistency Management in Fog Computing. Jouranl of Grid Computing. Vol. 19. Pages 1-25. 2021.

10. M. I. Naas, J. Boukhobza, P. Raipin et et L. Lemarchand. iFogStorC: a Heuristic for Managing IoT Data Replication Storage and Consistency in a Fog Infrastructure. Poster in Per3S: Performance and Scalability Storage Systems Workshop. INRIA, Bordeaux, France. 2019.

11. L. Ait-Oucheggou, M. I. Naas, Y. Hadjadj-Aoul et J. Boukhobza. When IoT Data Meets Streaming in the Fog. In IEEE ICFEC: International Conference on Fog and Edge Computing. Taormina (Messina), Italy. 8 pages. 2022.

Informatique Image Interaction
Univ. La Rochelle

La Rochelle Université