Optimizing resource allocation for micro-services within Cloud and hybrid Cloud/Edge environments

Alessandro Di Stefano

DIEEI | University of Catania

Research context

The research activity is focused on cloud management in a multi-tenant, cooperative and competitive, hybrid environments. The aim is to manage the QoS and investigate how adaptive strategies can be considered in this context:

- Analyzing the issues related with the deployment of workflows in Cloud/Edge environments;
- Modeling micro-services applications and deployment clusters;
- Optimizing placement of container-based micro-services applications in Cloud, Edge and hybrid Cloud-Edge environments;

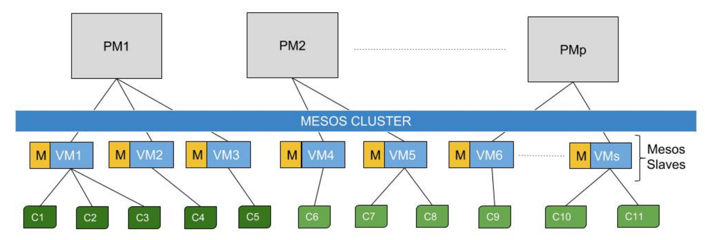
Communication-intensive micro-services placement: Coope4M

In recent years, the widespread use of container technologies has significantly altered the interactions between cloud service providers and their customers when developing and offering services. The shift away from Virtual Private Server scenarios in Infrastructure-as-a-Service environments requires drastically changing the deployment strategies adopted by service providers. This also opens many questions on what information must be supplied by customers and how to improve the performance of user applications, especially in the case of communication-intensive applications. Coope4M is a new framework for Mesos Clusters that aims to improve the deployment strategies of communication intensive applications. Coope4M is based on the partitioning of the user application graph via the Isolation Index parameter obtained through user-knowledge on the degree of the communication between its components.

$$IsoIndex = 1 - \sum_{h=0}^{k} \frac{\sum_{(i,j)\in S_h} w_{ij}}{\sum_{i,j} w_{ij}}$$

where:

- w_{ij} is the weight of communication between the components c_i and c_j of the graph G;
- $S_h = \{(i,j) \mid c_i \in Partition_h \wedge c_j \notin Partition_h\}$;



- In the formula the numerator represents the load of the communication of the h-th slave with all the other slaves while the denominator represents the total communication load of the application.

The optimal placement solution maximizes the formula above for each application.

Coope4M: results

We compared Coope4M with RoundRobin setting a minimum threshold for the isolation index in scenarios for which each node have 4 or 6 slots available for containers. 100 applications were executed with different topologies and weigths. The figures report the execution time and its standard deviation for those simulations. Coope4M results show how this solution produces lower standard deviation on execution time which stands for a better guarantee on QoS and SLAs.

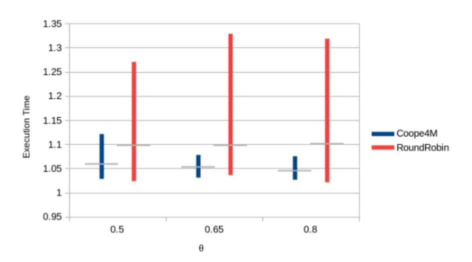


Figure 4 Execution time for a 4 slots per slave scenario

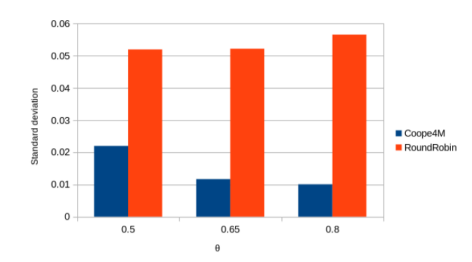


Figure 5 Standard deviation for a 4 slots per slave scenario

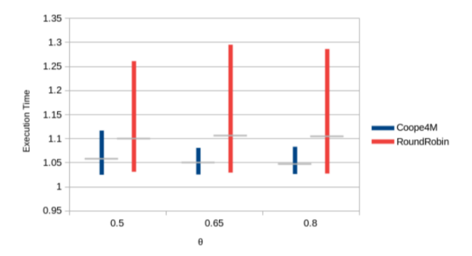


Figure 6 Execution time for a 6 slots per slave scenario

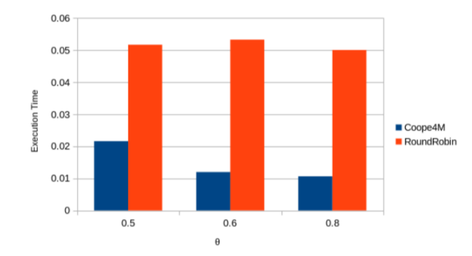


Figure 7 Standard deviation for a 6 slots per slave scenario

Cloud-Edge Resource Allocation: EdgeMORE

Under the paradigm of Edge Computing, a Network Operator deploys computational resources at the network edge and let third-party services run on top of them. Besides the clear advantages for Service Providers and final users thanks to the vicinity of computation nodes, a NO aims to allocate edge resources in order to increase its own utility, including bandwidth saving, operational cost reduction, QoE for its user, etc. However, while the number of third-party services competing for edge resources is expected to dramatically grow, the resources deployed cannot increase accordingly, due to physical limitations. Therefore, smart strategies are needed to fully exploit the potential of EC, despite its constrains.

To this aim, we propose to leverage *service adaptability*: each service can adapt to the amount of resources that the NO has allocated to it, balancing the fraction of service computation performed at the edge and relying on remote servers, e.g., in the Cloud, for the rest. Service availability is exploited in *EdgeMORE*, a resource allocation strategy in which SPs declare the different configurations to which they are able to adapt, specifying the resources needed and the utility providded to the NO. The NO then chooses the most convenient option per each SP, in order to maximize the total utility. We formalize EdgeMORE as a Integer Linear Program and we propose a polynomial time heuristic. We show via simulation that, by letting SPs declare their ability to adapt to different constrained configurations and letting the NO choose between them, EdgeMORE greatly improves EC utility and resource utilization.

EdgeMORE: ILP

$$\max \sum_{i=1}^{N} \sum_{j=1}^{N_i} u^{i,j} \cdot x^{i,j}$$

subject to:

$$\sum_{m=1}^{M} y^{i,j}_{z,m} = x^{i,j} \qquad i = 1 \dots N, \; j = 1 \dots J^i, \; z = 1 \dots Z^{i,j}$$

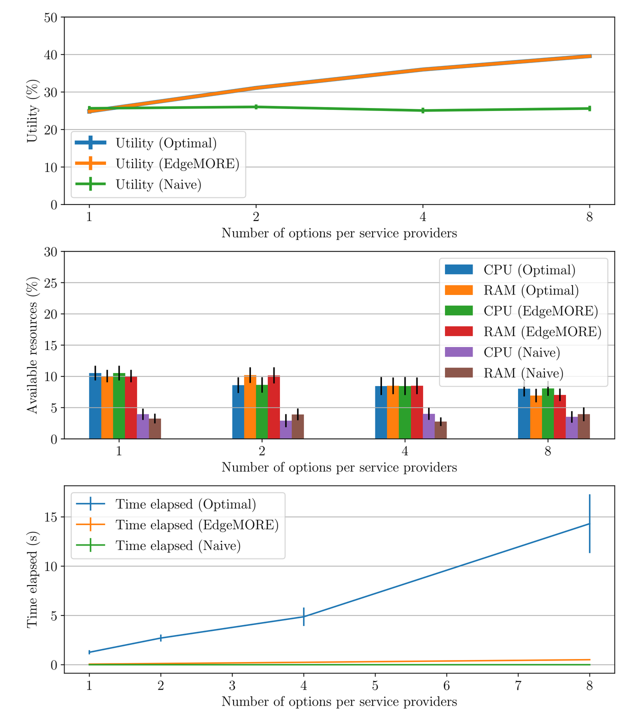
$$\sum_{i=1}^{N} \sum_{j=1}^{J^i} \sum_{z=1}^{Z^{i,j}} y^{i,j}_{z,m} \cdot w^{i,j}_{l,z} \le c_{l,m} \qquad l = 1 \dots 2, \; m = 1 \dots M$$

$$\sum_{j=1}^{J^i} x^{i,j} \le 1 \qquad i = 1 \dots N$$

Here, N is the number of Service Providers (SP), J^i the number of options of the i-th SP, $Z^{i,j}$ the number of contaienrs of the j-th option of the i-th SP. M is the number of the available nodes at the Edge and l is the resource considered (e.g. CPU, RAM). The decision variables are $x^{i,j}$ which is 1 if the j-th option of the i-th SP is deployed and $y^{i,j}_{z,m}$ which is 1 if the z-th container of the j-th option of the i-th SP is placed in the m-th node. A heuristic is also available but not provided here for lack of space.

EdgeMore: Results

We compared the ILP model and our heuristic with a naive strategy (Random-based first-fit placement of a random option) on a synthetic scenario but also in real traces (Alibaba and Google traces). Here we report the utility, the resource usage after placement and the time to computation with these three strategies varying the number of options for 50 SPs providing 8 containers options in a cluster of 8 nodes. Note that the classic assumption corresponds to the first point of the plot, SP=1. While varying the number of options from 1 to 8 the utility has a gain almost equal to 60%, which would be lost with classic approaches and which instead we can grasp by exploiting service elasticity. This is equivalent to virtually increase the available resources, by just using them better. Observe also that EdgeMORE uses resources as the optimum, while Naive, despite providing poor utility, uses ~3.3 times more resources than optimal/EdgeMORE).



Published and Submitted Conference/Journal Papers

- Al. Di Stefano, An. Di Stefano, G. Morana & D. Zito. "Coope4M: A Deployment Framework for Communication-Intensive Applications on Mesos." 2018 IEEE 27th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE) (2018): 36-41.
- Al. Di Stefano, An. Di Stefano & G. Morana (In press). "Scheduling communication-intensive applications on Mesos" International Journal of Grid and Utility Computing
- A. Araldo, Al. Di Stefano & An. Di Stefano (Accepted) "EdgeMORE: Improving Resource Allocation with Multiple Options from Tenants" 2020 17th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, 2020
- A. Araldo, Al. Di Stefano & An. Di Stefano (Submitted) "Resource Allocation for Edge Computing with Multiple Tenant Configurations" Proceedings of the 35th ACM/SIGAPP Symposium on Applied Computing
- B. Steer, A. Di Stefano, R. Clegg & F. Cuadrado "Building distributed temporal graphs from event streams" 2018 Second Workshop on Advances in Mining Large-Scale Time Dependent Graphs (TD-LSG)

Attended classes or events

- **Deeplearn2019**: 3rd International Summer School on Deep Learning - Warsaw, Poland - July 22-26, 2019
- **MSCX2019**: Mediterranean School of Complex Networks - Salina, Italy - 31 Aug - 6 Sep, 2019

Work in progress

- Applying EdgeMORE to multi-edge scenarios;
- Heuristics for Edge-Cloud computing container allocation;;
- Extension of Coope4M to multi-cloud scenarios;

Alessandro Di Stefano
alessandro.distefano@phd.unict.it