PAPS: A first step into the implementation phase. Partitioning a network.

Fabio Losavio, Oscar Francesco Pindaro

Dipartimento di Elettronica, Informazione e Bioingengeria
Politecnico di Milano, Italy
{fabio.losavio,oscarfrancesco.pindaro}@mail.polimi.it

xx - May - 2020

Abstract

The emergence of latency-sensitive and data-intensive applications requires to move the computational power closer to users on nodes at the edge of the network (*edge computing*). This work starts from PAPS [1], a framework which aims to tackle the complexity of edge infrastructures by means of decentralized self-management and serverless computing. This paper shows how the first of the four PAPS phases, *partitioning*, is implemented and integrated with a common open-source system: Kubernetes [3], to easily deploy the system in an already existing network. The partitioning is mainly focussed on the division of the network in multiple communities in order to reduce the scope of the following parts. The division is performed using the SLPA [2] algorithm, which uses a label spreading technique to assign each node to a community. Once all the nodes are assigned, the Kubernetes node will be modified using the API labeling it in order to easily recognize and manage each community.

Keywords: Edge computing · Partition · Kubernetes · PAPS · Community division · SLPA

1   Introduction

PAPS is an edge computing framework which aims to improve network performances of some applications moving the computation closer to the final user instead of doing that on a centralized server. This will reduce the amount of traffic flowing into the network with the result of an improvement of the throughput allowing to better meet the SLAs of delay sensitive applications.

This framework focuses on MEC topologies, composed of geo-distributed nodes which access the system through cellular base stations. In a typical topology is possible to identify two different networks: the *fronthaul network*, which connects normal nodes to the MEC stations, and the *backhaul network*, which interconnects the MEC stations. PAPS aims to reduce the complexity of the problem working at three different levels: *system, community and node.*

System level

Since MEC topologies can be really big and complex, the first step is to *partition* the network in multiple delay-aware sub-networks called *communities*. Each community is composed by a set of nodes whose propagation delay from one another is below a given threshold. PAPS assumes the availability of a *supervisor* that has a global view of the MEC topology and uses SLPA [2] to create the communities. Each community will elect a *leader* which will be in charge of managing the community and the communication between the nodes.

Community level

Communities aims to minimize the likelihood of of SLA violations so that the MEC nodes can operate under feasible conditions.
Each community leader will manage the *allocation and placement* phases by looking at the aggregate demand to each service and the node capacities in order to decide how to distribute resources among the nodes in its community. This is done solving a *mixed integer programming (MIP)* problem whose goal is to minimize the overall community delay. The MIP is solved periodically by the community leader which will then contact the other members to communicate the new configuration.

Node level

The Node-Level Self-Management system aims to keep the response time for a given function fixed, by deploying and removing containers accordingly to the measured workload. In order to achieve this, every node has a controller for each function that is hosting; all the controllers run in parallel and independently. Each controller accepts as input the number of container allocated for its function and the measured arrival rate, and outputs the response time. The objective is to keep the response time at the control set point, which must be lower than the SLA.

2 References

References

[1] Baresi, L., Mendonça, D., Quattrocchi, G.: *PAPS: A Framework for Decentralized Self-management at the Edge.* In: Service-Oriented Computing, pp.508-522, (2019).

[2] Xie, J., Szymanski, B K, Liu, X.: *SLPA: Uncovering Overlapping Communities in Social Networks via A Speaker-listener Interaction Dynamic Process.* In Proc: Data Mining Technologies for Computational Collective Intelligence Workshop at ICDM, Vancouver, CA pp. 344-349, (2011)

[3] https://kubernetes.io/