ABSTRACT

Stutzel, M.C. *A framework for orchestrating resources with a focus on scalability for applications on the Internet of Things.* 129 f. Dissertação (Mestrado em Engenharia Eletrônica) - Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro (UERJ), Rio de Janeiro, 2020.

Considering the growing number of devices connected to the Internet and the mechanisms and technologies developed in the context of the Internet of Things, this dissertation proposes a framework and presents a reference implementation capable of orchestrating the use of resources and promote scalability for IoT applications. Given a set of available, potentially distributed, services running on fog, edge and cloud resources, the framework orchestrates communication between clients, agents, and these services. This orchestration considers the use of CPU, memory and network latency, to select the most appropriate resources and reorganize the interaction topology between the elements, with the objective of minimizing these metrics, in average. With that, it is expected that IoT applications with a large number of elements can carry out their activities within an acceptable time. The proposed framework is transparent to IoT applications, which continue using middleware services. The applications do not need to know the framework, nor the effect of its operation. The reference implementation uses Docker microservices as the basis and SiteWhere middleware, widely used in distributed application development. In the implementation process, a limitation was identified in the used version of Docker, related to virtual networks, called swarm. This limitation prevented transparent communication between elements on different networks, which in the case of IoT is very common, given the distributed nature of the applications and the use of different clouds. It was therefore necessary to propose and implement a new overlay network that would allow transparent communication between the elements of different networks, even those behind a NAT server. The framework performs cycles where the metrics considered are monitored across the entire set of available resources and then triggers a resource allocation algorithm. This algorithm assesses the state of the elements of the application and the metrics monitored and then proposes a new topology of interaction between the elements to improve the performance of the application. Five data allocation algorithms resources selected in the literature have been integrated into the framework. The approach of each one of these algorithms tends to favor the optimization of different metrics in the allocation resources, which could result in greater scalability or better use of resources, which can affect applications differently. The reference implementation was evaluated in two groups of tests. The first test compared the performance of the selected resource allocation algorithms. With that it was possible to verify if the framework would be modular to accept different algorithms, and if the reference implementation would behave properly for any algorithm. We use latency metrics to select the algorithm that would most favor scalability, optimizing the time for the exchange of messages between the application elements and the middleware services. The ERA algorithm showed the best results. The second test evaluated the scalability of the framework with a scenario of up to 10,000 simultaneous clients using ERA as a resource allocation algorithm. The framework offered support for 10,000 customers, starting from an initial allocation topology, up to a topology that minimized the initial average latency by 33%.

Keywords: IoT; Microservices; Microservice Allocation; Framework.