Task Scheduling Model for Fog paradigm

Celestino Barros, Victor Rocio, André Sousa, Hugo Paredes

Task Scheduling Model for Fog paradigm

Celestino Barros[1,*], Victor Rocio[2], André Sousa[3] and Hugo Paredes[4]
[1]University of Cabo Verde; celestino.barros@docente.unicv.edu.cv
[2]INESC TEC and Open University of Portugal; vitor.rocio@uab.pt
[3]Critical TechWorks; asousa@roundstone.pt
[4]INESCT TEC and University of Trás-os-Montes and Alto Douro; hparedes@utad.pt

Abstract

Task scheduling in fog paradigm is highly complex and in the literature, there are still few studies. In the cloud architecture, it is widely studied and in many researches, it is approached from the perspective of service providers. Trying to bring innovative contributions in these areas, in this paper, we propose a model to the context-aware task-scheduling problem for fog paradigm. In our proposal, different context parameters are normalized through Min-Max normalization; requisition priorities are defined through the application of the Multiple Linear Regression (MLR) technique and scheduling is performed using Multi-Objective Non-Linear Programming Optimization (MONLIP) technique.

1 Introduction

The growth of mobile devices and the evolution of the Internet of Things (IoT) have stimulated the growth of devices connected to the Internet. This growth tends to increase significantly. On the other hand, several of these devices run applications that requires a part of the processing to run in large, centralized datacenters known as cloud. However, due to centralization and physical distance from end user's devices, it causes an increase in communication latencies and harms applications that require real-time responses. To minimize cloud processing by adopting local processing strategies and allow solving cloud limitations, different techniques have been proposed. One of such technique is the use of the fog computing paradigm [1].

According to [2], many of the task scheduling algorithms in the cloud architecture and fog paradigm found in the literature do not describe how the priority is defined, do not explain the method used to prioritize tasks, nor do they define the prioritization of tasks based on context information, and many defend the perspective of service providers. Others are applied in grouped tasks to decrease execution time. Some optimize only QoS. Others explore only some contexts. The author also claims that they allow solving many problems.

The main objective of this paper is proposing a model of context-aware task scheduling algorithm for the fog-computing paradigm. To achieve its main objective, some specific objectives were defined to contextualize concepts such as fog computing; task scheduling and context-aware; standardize the different context parameters using Min-Max normalization; define the priorities of the requests through the application of the MLR technique and optimize the scheduling using the MONLIP technique.

This paper is organized in four sections: In the first section, we introduce the paper, in the second section we deal with the contextualization of the subject. In the third section, we describe the contexts envisaged, the model and the proposed architecture. In fourth section, we addresses the conclusion of the paper.

2 Background

Mobile computing provides users with several utilities, allows portability, supports applications of various interests, and has several limitations such as scarcity of resources, reduced battery life, among others [2]. In recent years, several architectures have been proposed to solve these limitations, being cloud computing one of them. Despite the advantages, cloud is centralized and for optimization of energy and communications costs, the processing is done in concentrated data centers. To solve these inconveniences, several paradigms have been presented. Among them is fog computing, which aims to make the services offered by the cloud available at the edge of the network [3]. In this section, we contextualize and discuss concepts such as fog computing, context-aware and design of task scheduling algorithms.

2.1 Fog computing

According to [3], fog computing is a new paradigm that aims to overcome the limitations of cloud by providing services at the edge of the network. In [2], it is broadly defined and emphasis is given to some characteristics such as geographical distribution, predominance of wireless access, heterogeneity, distributed environment, among others. As reported in [1], it is: “A horizontal, system-level architecture that distributes computing, storage, control and networking functions closer to the users along a cloud-to-thing continuum.”

In the opinion of [4], fog computing from the perspective of mobile computing, aims to provide a cloud-like facility. However, lighter, closer to the users of mobile devices, it can serve these users through direct connection, shorter, compared to the cloud connection.

2.2 Context-aware and fog computing

According to [2], in mobile computing, the context of a user is very dynamic. In [5], a definition, the context categories and context sensitive applications are made available. Information and services, information marking with context and automatic service execution methodology are still presented as well as the survey of the state of the art regarding context-aware computing.

Bazire and Brézillon in [6], define the context as a set of constraints that influence the behavior relating to a given task. The context definition most used today, even in other fields, as in the operationalization was given by [7]:

“Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between user and an application, including the user and applications themselves” [7] (p. 45).

In mobile computing, context refers to the processing environment, user environment, physical environment, relevant for the interaction between a user and an application, including the user and the applications themselves [8].

When mobile devices communicate with cloud, they face high network latency and high transmission power consumption [2]. They also state that in fog paradigm, mobile devices send tasks to fog nodes in order to be processed and returned the result. This process reduces the power consumption of the mobile device, transmission delay, among others. Due to the lower capacity of the fog nodes when compared to cloud, the tasks, which cannot be executed in the fog, are sent to be executed in cloud.

2.3 Design of scheduling algorithms

According to [2], scheduling is the allocation of resources needed to execute a task. In its design, we must consider some constraints such as dependencies between tasks, cost of tasks and the location. It also guarantees that scheduling decisions can be Static - where decisions about scheduling are made during the compilation. On the other hand, it can be dynamic - where information about the state of the task flow is used at a given time during execution for the scheduling decisions. It is the best approach.

However, these problems are computationally demanding, require a strategy of parallelization and dynamic load balancing [2].

3 Proposed model and architecture

We assume that an appropriate code offloading technique (e.g. MAUI defined in [9], COMET presented in [10], among others) is being run on mobile devices in order to make the best decision as to whether or not to offload codes and which fog nodes [11].

We consider that a request includes battery level, QoS information and network signal values. We also assume, as in [4] that fog provides greater computing capabilities than mobile devices and can extract the contexts associated with the requests and make the scheduling decision accordingly.

Musumba and Nyongesa in [8], define the main contexts that can be explored in any mobile computing environment as: network connection; available processors; battery level; location; network bandwidth; network traffic; leased of Virtual Machines (VM) and application QoS requirements.

In our domain of the problem, the contexts of the service providers because we do not know them were ignored. In addition, after offloading the tasks in the fog nodes, it becomes unnecessary to consider the processors of the mobile device. The location of the device also does not affect the scheduling, as well as network traffic and bandwidth that are the same for all users. Based on these criteria we considered three context parameters: battery level, signal-to-noise network interference ratio (SIN) and application QoS.

In the following subsections, we illustrate and discuss the model and architecture of the proposed solution.

3.1 Proposed model

The fog nodes, with our proposal activated, consists of three units: *Context Information Retrieval Unit*, comprises an architecture, as defined in [12]. It retrieves context information (C_i) from each request ($r \in R$). The recovered context information is forwarded to the *Context-Aware Task Prioritization Unit*, which estimates the value of the context priority (P_r) for each individual request $r \in R$ and routes it to the *QoE and Context-Aware Scheduler Unit*, which schedules tasks to be executed in VMs so that QoE is optimized. Figure 1 shows the different units of the proposed model.

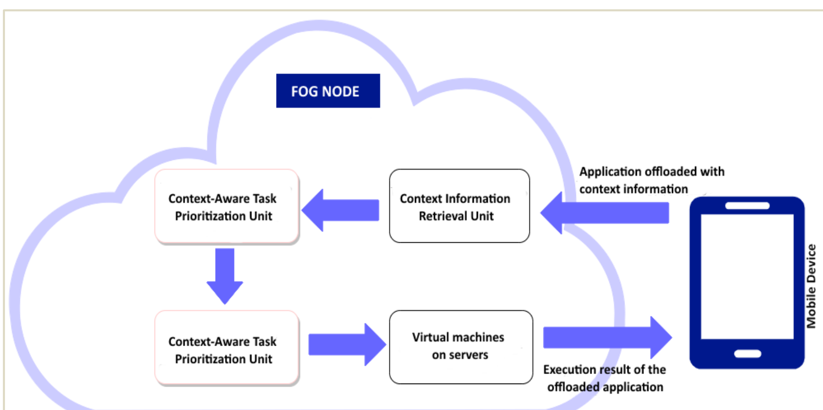

Figure 1: Proposed model.

3.2 Architecture of the proposed model

The fact that the context parameters associated with a request are heterogeneous makes it difficult to explore the context information in the scheduling. To solve this problem, in Han, Kamber and Jian [13], a context heterogeneity resolver is proposed, which processes several parameters, in a normalized interval, through Min-Max normalization, where each request is prioritized based on its context values.

The Context-Aware Task Prioritization Unit is composed by *Context Repository*, which stores context information of current and previously received tasks and *Context Forecasting Unit*, exploits the context information at a given time and feeds the *Forecast Table*. Thus, we manage to eliminate the heterogeneity of the context information in the feeding of the forecast table.

The *Forecast Table* provides a data set for the MLR analysis which aims to define the priority of requests.

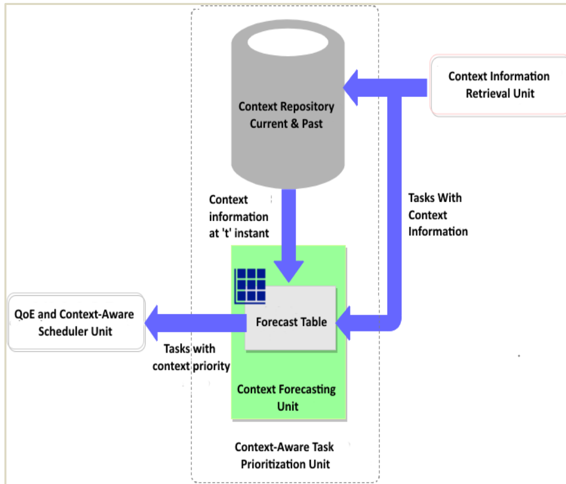

Figure 2: Context-Aware Task Prioritization Unit architecture of the proposed model.

Figure 2 shows the architecture of the Context-Aware Task Prioritization Unit of the proposed model.

3.3 Optimization of application scheduling

In order to optimize QoE, the proposed model explores the context priority (P_r) of the request $r \in R$ and its estimated execution time duration (T_r, v) to define the scheduling of this request $r \in R$ in an VM, $v \in V$. We also explore the number of scheduling intervals (I_r), in which a request is delayed its scheduling since its arrival, in order to avoid the starvation situation.

One of the objectives of this paper consists of scheduling requisitions, $r \in R$ in VM, $v \in V$, in order to optimize the QoE for all the requisitions in a certain scheduling interval.

The OF is defined according to equation 11.

$$\boldsymbol{min_{r,v}} \sum_{\boldsymbol{r \in R}} \sum_{\boldsymbol{v \in V}} \frac{\boldsymbol{P_r * \psi_{r,v}}}{\boldsymbol{I_r}} \tag{1}$$

This equation indicates that the QoE can be optimized by minimizing the sum of their execution times. It also takes into account the priority execution of the tasks with higher priorities by minimizing the sum of the priorities of all requests, since the lower the result, the higher the priority obtained. Moreover, the sum of the inverse values of (I_r), $\forall r \in R$ shows that the requisitions, in which their scheduling has been postponed in a given interval, will have higher priority to be scaled in the current intervals, thus mitigating the starvation situation.

4 Conclusions

The main purpose of this paper of this paper is proposing a model of context-aware task scheduling algorithm for the fog-computing paradigm. To accomplish the main objective, the following piecemeal objectives were achieved:

- We defined some concepts such as fog computing paradigm, context-aware and task scheduling. We intend to contextualize the main theories and concepts related to this paper.
- We propose a model that uses Min-Max normalization, to normalize the different context parameters and solve the problem of heterogeneity of device and application contexts. The MLR analysis was used to define the priority of the context of the requests, which allows the

availability of a set of hypothetical data. The optimal scheduling of requests to optimize QoE was solved by using the MONLP technique.

All proposed objectives were achieved. We consider several context parameters. Others, however, can still be pondered in order to analyze their influences on the scheduling.

Acknowledgments

The authors are grateful to the Calouste Gulbenkian Foundation for its funding of this research through the Ph.D. scholarship under the reference No. 234242, 2019 - Postgraduate Scholarships for students from PALOP and Timor-Leste.

References

[1] OpenFog, "OpenFog Reference Architecture for Fog Computing," *OpenFog*, 2017. https://www.iiconsortium.org/pdf/OpenFog_Reference_Architecture_2_09_17.pdf (accessed Feb. 24, 2020). URL: https://www.iiconsortium.org/pdf/OpenFog_Reference_Architecture_2_09_17.pdf.

[2] C. Barros, V. Rocio, A. Sousa, and H. Paredes, "Survey on Job Scheduling in Cloud-Fog Architecture," in *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, 2020, pp. 1–7.

[3] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 50–55, Dec. 2008. DOI: 10.1145/1496091.1496100. URL: https://doi.org/10.1145/1496091.1496100.

[4] T. H. Luan, L. Gao, Z. Li, Y. Xiang, G. We, and L. Sun, "A View of Fog Computing from Networking Perspective," Feb. 2016, [Online]. Available: http://arxiv.org/abs/1602.01509. URL: http://arxiv.org/abs/1602.01509.

[5] A. K. Dey, D. Salber, G. D. Abowd, and M. Futakawa, "The Conference Assistant: combining context-awareness with wearable computing," in *Digest of Papers. Third International Symposium on Wearable Computers*, 1999, pp. 21–28. DOI: 10.1109/ISWC.1999.806639. URL: http://ieeexplore.ieee.org/document/806639/.

[6] M. Bazire and P. Brézillon, "Understanding Context Before Using It," in *Modeling and Using Context*, A. Dey, B. Kokinov, D. Leake, and R. Turner, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 29–40. DOI: 10.1007/11508373_3. URL: http://link.springer.com/10.1007/11508373_3.

[7] A. K. Dey, "Understanding and Using Context, Personal and Ubiquitous Computing, Vol. 5," vol. 5, pp. 4–7, 2001.

[8] G. W. Musumba and H. O. Nyongesa, "Context awareness in mobile computing: A review," *Int. J. Mach. Learn. Appl.*, vol. 2, no. 1, May 2013. DOI: 10.4102/ijmla.v2i1.5. URL: https://ijmla.net/index.php/ijmla/article/view/5.

[9] E. Cuervo *et al.*, "MAUI: Making Smartphones Last Longer with Code Offload," in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, 2010, pp. 49–62. DOI: 10.1145/1814433.1814441. URL: https://doi.org/10.1145/1814433.1814441.

[10] M. Gordon, D. Jamshidi, S. Mahlke, Z. Mao, and X. Chen, "COMET: code offload by migrating execution transparently," in *Proceedings of the 10th USENIX Conference on Operating Systems Design and Implementation*, 2012, pp. 93–106.

[11] F. Berg, F. Dürr, and K. Rothermel, "Increasing the Efficiency and Responsiveness of Mobile Applications with Preemptable Code Offloading," in *2014 IEEE International Conference on Mobile Services*, 2014, pp. 76–83. DOI: 10.1109/MobServ.2014.20.

[12] H. J. La and S. D. Kim, "A Conceptual Framework for Provisioning Context-aware Mobile Cloud Services," in *2010 IEEE 3rd International Conference on Cloud Computing*, Jul. 2010, pp. 466–473. DOI: 10.1109/CLOUD.2010.78. URL: http://ieeexplore.ieee.org/document/5557960/.

[13] J. Han, M. Kamber, and P. Jian, *Data Mining: Concepts and Techniques*. Elsevier, 2012. DOI: 10.1016/C2009-0-61819-5. URL: https://linkinghub.elsevier.com/retrieve/pii/C20090618195.