The 12th Conference of PhD Students in Computer Science

Volume of short papers

CS^2

Organized by the Institute of Informatics of the University of Szeged

June 24 – June 26, 2020
Szeged, Hungary

Scientific Committee:

János Csirik (Co-Chair, SZTE)
Lajos Rónyai (Co-Chair, SZTAKI, BME)
András Benczúr (ELTE)
Tibor Csendes (SZTE)
László Cser (BCE)
Erzsébet Csuhaj-Varjú (ELTE)
József Dombi (SZTE)
István Fazekas (DE)
Zoltán Fülöp (SZTE)
Aurél Galántai (ÓE)
Zoltán Gingl (SZTE)
Tibor Gyimóthy (SZTE)
Katalin Hangos (PE)
Zoltán Horváth (ELTE)
Márk Jelasity (SZTE)
Tibor Jordan (ELTE)
Zoltán Kató (SZTE)
Zoltán Kása (Sapientia EMTE)
László Kóczy (SZE)
Andrea Kő (SZE)
János Levendovszki (BME)
Máté Matolcsi (BME)
Gyöngyvér Márton (Sapientia EMTE)
Branko Milosavljevic (UNS)
Valerie Novitzka (TUKE)
László Nyúl (SZTE)
Marius Otesteanu (UPT)
Zsolt Páles (DE)
Attila Pethő (DE)
Sándor Szabó (PTE)
Gábor Szederkényi (PPKE)
Jenő Szigeti (ME)
János Sztrik (DE)
János Tapolcai (BME)
János Végh (ME)

Organizing Committee:
Attila Kertész, Balázs Bánhelyi, Tamás Gergely, Judit Jász, Zoltán Kincses
Address of the Organizing Committee
c/o. Attila Kertész
University of Szeged, Institute of Informatics
H-6701 Szeged, P.O. Box 652, Hungary
Phone: +36 62 546 781, Fax: +36 62 546 397
E-mail: cscs@inf.u-szeged.hu
URL: http://www.inf.u-szeged.hu/~cscs/

Sponsors
Supported by the project "Integrated program for training new generation of scientists in the fields of computer science", No. EFOP-3.6.3-VEKOP-16-2017-00002. The project has been supported by the European Union and co-funded by the European Social Fund.
University of Szeged, Institute of Informatics

Preface

This conference is the 12th in a series. The organizers aimed to bring together PhD students working on any field of computer science and its applications to help them publishing one of their first papers, and provide an opportunity to hold a scientific talk. As far as we know, this is one of the few such conferences. The aims of the scientific meeting were determined on the council meeting of the Hungarian PhD Schools in Informatics: it should

- provide a forum for PhD students in computer science to discuss their ideas and research results;
- give a possibility to have constructive criticism before they present the results at professional conferences;
- promote the publication of their results in the form of fully refereed journal articles; and finally,
- promote hopefully fruitful research collaboration among the participants.

The papers emerging from the presented talks will be invited to be considered for full paper publication the Acta Cybernetica journal.

This year, due to the unfortunate international situation caused by the COVID-19 pandemic, the organizers had no option but to adapt, so they decided to convert the conference completely into a web-based event. Nevertheless, they still hope that the conference will be a valuable contribution to the research of the participants, who can perform virtual interactions to get feedback on their research progress.

Szeged, June 2020

Attila Kertész
Balázs Bánhelyi
Tamás Gergely
Judit Jász
Zoltán Kincses

Contents

Task allocation possibilities in simulated Fog environments

András Márkus

Abstract: Nowadays billions of smart devices utilise the network and storage capacity of complex systems. These devices having various sensors are usually managed in Internet of Things systems, which are often supported by Fog Computing or Cloud Computing services. Within the Fog paradigm, nodes are installed close to the users (i.e their devices) to achieve faster and more reliable communications and sensor data management than former cloud solutions. In this paper, we present an extension of a simulation tool called DISSECT-CF-Fog, which provides a more detailed fog model by enhanced location awareness and multi-layer fog node management. We also exemplify the utilisation of these fog properties by developing and validating different task allocation strategies in simulated IoT-Fog-Cloud environments. We also show an evaluation of a scenario comparing four different approaches for task allocation in these systems.

Keywords: Fog Computing, Internet of Things, Simulation, Task allocation

Introduction

According to the paradigm of the Internet of Things (IoT), sensors, actuators and smart devices are connected through the Internet. Based on [1], the number of smart devices will overstep 75 billions all over the world by 2025. IoT is often encoupled with Cloud Computing, because the huge amount of sensed data require storing and processing for further analysis with cloud resources. In the past years a new paradigm called Fog Computing appeared (grown out of Cloud Computing), where the generated sensor data are stored and processed on so-called fog nodes, which are located geographically closer to the end users for minimising latency and ensuring privacy of the data [2]. These fog nodes usually have less computing power and limited functionality as well (due to their constrained nature). Different type of applications, such as real time or forecasting systems, can utilise fog architectures, which typically have more layers of fog or cloud nodes with different resource constraints and characteristics. These nodes can process bag of task applications composed of the generated data of sensors, hence one of the open issues of Fog Computing is how the operating costs and the node response times can be minimised by an appropriate allocation of the tasks.

The investigation of real-world fog systems and topologies are rarely feasible, thus different simulation environments are utilised by the researchers for such purposes. In this paper we chose the DISSECT-CF-Fog simulator[1] to be extended with a more realistic fog model, and for further analysing task allocation possibilities. The rest of this paper we give a brief introduction to related works in Section , in Section we present our simulator extension, the proposed strategies and their evaluation. Finally, Section concludes our work.

Related work

The literature of Fog, Cloud and IoT systems contains numerous articles with diverse approaches aiming at task and virtual machine (VM) placement for IoT applications. Mann et al. [3] compared seven different VM placement algorithms for clouds. They executed their experiments in the well-known CloudSim simulator, relying on cloud properties such as CPU speed, resource capacity, utilisation ratio and energy consumption of the available nodes.

[1]The DISSECT-CF-Fog simulator is available at: https://github.com/andrasmarkus/dissect-cf/

Vasconcelos et al. [4] presented a different approach, where the Fog topology is modelled as a weighted graph, and they considered resource cost, bandwidth, and latency of the smart devices. Resource availability and the topology of the network are also considered during their validations performed in the Cooja simulator.

Xia et al. [5] proposed four heuristics for application placement in SimGrid with the following parameters: required CPU, RAM, bandwidth, and latency of the fog nodes. Three allocation policies were presented by Bittencourt et al. [6], which were evaluated in the one of the most commonly used Fog simulators called iFogSim. The proposed strategies took into account CPU capacity, the arrival time and the delay-priority of the tasks. Skarlat et al. [7] presented a framework, where fog nodes are utilised when possible. Their optimisation solution considers the resource capacity of fog nodes, the resource demand of a service, the link delay between nodes, and different properties of the application (e.g. deployment and response time).

Based on these works we can state that the evaluation of task scheduling algorithms have been performed in different simulation environments, and most approaches consider almost the same characteristics of the Fog and IoT systems. Beside the former commonly used cloud properties, location and mobility-related properties are used in fogs, but more heterogeneous and multi-layered Fog-Cloud systems are rarely analysed.

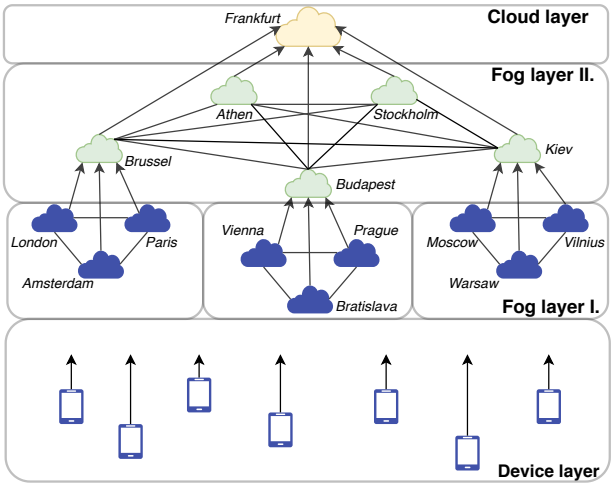

Figure 1: The considered Fog topology in the evaluation

The proposed simulator extension and its evaluation

An algorithm for the allocation of a task of an IoT application in our proposed model can consider the energy consumption of the execution environment (i.e. fog or cloud node), the load of the network used for communication and data transfers, and the transfer, storage and execution costs. When a selected fog node is overloaded, the execution of the appropriate task will be delayed, which has a negative effect on the makespan of it application. To overcome this problem, we also introduce the possibility of multi-layer fog node management by enabling task offloading from (possibly overloaded) nodes to others. A typical fog topology can contain numerous nodes, some of them are grouped as a Fog cluster, which restricts the access and visibility of other nodes. These nodes can be ordered into layers, where higher level fog layers usually contain stronger physical resources. We implemented this envisioned model in the DISSECT-CF-Fog simulator. We developed a solution that is able to handle fog properties dynamically, hence performing task allocation and reallocation during the experiments. In this way, different approaches can be created and implemented by extending the new, *Application-Strategy* abstract class.

We defined four strategies to validate the usability of our proposal. The *Random* strategy is the default, which always chooses one from the connected nodes randomly. The *Pushing Up* strategy always chooses the parent node (i.e. a node from a higher layer), if available. The disadvantage of these strategies is the disability to consider increased network traffic and costs of the operation. The third strategy called *Holding Down* considers data protection, because the application keeps the data as close to the end-user as possible. In this way the network traffic is minimal, but the execution time of the application can increase dramatically (due to the overload of the lowest layer). Finally, the *Load Balancing* strategy ranks the parent nodes (if available) and all neighbour nodes (from its own cluster) by network latency and the ratio of the available CPU capacity and the total CPU capacity. The algorithm picks the node with the highest rank (i.e. closest and least loaded).

We evaluated these proposed strategies in a European-wide weather forecasting scenario with a fog topology having 10000 weather stations. Each station is equipped with five sensors (e.g. measuring weather conditions (e.g. temperature, humidity) with 50 bytes of sensor data). The time interval between two measurements was set to one minute, and the whole simulation period took one day. The topology contains two Fog layers with 14 different Fog nodes organised into clusters, and one cloud layer having a single cloud node. Each node has at lest 40 CPU cores and 44 GB RAMs, and they are mapped to different cities, with exact latency values defined between them using WonderNetwork[2]. The defined topology can be seen in Figure 1, which also depicts the Fog clusters and layers with different colour. The arrows represent routes responsible for communication between the layers, while the (undirected) edges are for the message exchanges inside a cluster. The network capability of the smart devices (or stations) are modelled with a 4G network with an average 50 ms of latency. The fog nodes are modelled with real VM specifications according to the Amazon Web Services (AWS): the lowest Fog layer has VMs with 2 CPU cores, 4 GB RAMs and 0.051$ hourly price, the top fog layer has VMs with 4 CPU cores, 8 GB RAMs and 0.101$ hourly price, finally the cloud layer has VMs with 8 CPU cores, 12 GB RAMs and 0.204$ hourly price. Each VM can process only one task (represented by 250 Kilobytes of data) at the same time.

Table 1: Evaluation results of the proposed strategies

Strategies	Random	Pushing Up	Holding Down	Load Balancing
Num. of VM	78	73	60	78
Network utilisation (sec.)	54	40	0	43
Data transferred (MB)	1076	279	0	346
Timeout (min.)	42.04	4.88	175.16	4.66
Fog Layer I. cost ($)	66.41	66.40	72.05	66.40
Fog Layer II. cost ($)	21.71	22.02	0	22.01
Cloud cost ($)	13.10	0.74	0	0.81
Sum. of cost ($)	101.22	89.16	72.05	89.22

Table 1 summarises the average results after executing the scenario with all strategies five times. For the comparison of the strategies, we measured how many VMs were required for processing the tasks (and their data) during the operating hours. The Network utilisation metric reflects the network load, and it represents the time taken to transfer the sensor data from the source to the actual processing node, while the Data transferred metric represents its size. The Timeout value means the time taken to finish data processing after the last sensor

[2]WonderNetwork website is available at: https://wondernetwork.com/pings. Accessed in January, 2020.

measurement was performed. According to the used AWS pricing models, we could calculate the exact costs of the usage of the resources, separating each layer.

Concerning the results, the *Random* strategy occasionally used the network unnecessarily (1074 MB for 54 seconds), and the task were often moved to the cloud layer resulting in the highest cost (101.22 $). The *Holding Down* strategy used the least VMs (60), however its timeout value extremely increased (175.16 minutes), pointing out the need of the upper layers. Nevertheless, it is obviously the cheapest solution with 72.05 $. The *Pushing Up* approach secured us average cost, but its timeout is the best so far. Finally, the *Load Balancing* algorithm showed slightly better task allocation for almost the same price, with the timeout value of 4.66 minutes.

Conclusion

In this paper we presented a simulation solution for investigating task allocation problems in Fog environments. We developed an extension to DISSECT-CF-Fog with a revised fog model, and proposed four strategies to exemplify its utilisation. We also compared the performance of these strategies. In our future work we plan to extend our strategies with more sophisticated methods, and perform evaluations of larger-scale IoT systems.

Acknowledgements

This research was supported by the Hungarian Scientific Research Fund under the grant number OTKA FK 131793, and by the Hungarian Government under the grant number EFOP-3.6.1-16-2016-00008.

References

[1] Taylor, Robin and Baron, David and Schmidt, Daniel. (2015). The world in 2025 - predictions for the next ten years. 192-195. 10.1109/IMPACT.2015.7365193.

[2] Mahmud, Md and Buyya, Rajkumar. (2016). Fog Computing: A Taxonomy, Survey and Future Directions. 10.1007/978-981-10-5861-5_5.

[3] Mann, Zoltan and Szabo, Mate. (2017). Which is the best algorithm for virtual machine placement optimization?. Concurrency and Computation: Practice and Experience. e4083. 10.1002/cpe.4083.

[4] Vasconcelos, Danilo and Andrade, R and Severino, Valdenir and De, J and Souza,. (2019). Cloud, Fog, or Mist in IoT? That Is the Question. ACM Transactions on Internet Technology. 19. 25. 10.1145/3309709. 10.1145/3309709.

[5] Xia, Ye and Etchevers, Xavier and Letondeur, Loic and Lebre, Adrien and Coupaye, Thierry and Desprez, Frederic. (2018). Combining Heuristics to Optimize and Scale the Placement of IoT Applications in the Fog. 153-163. 10.1109/UCC.2018.00024.

[6] Bittencourt, Luiz Fernando and Diaz-Montes, Javier and Buyya, Rajkumar and Rana, Omer and Parashar, Manish. (2017). Mobility-Aware Application Scheduling in Fog Computing. IEEE Cloud Computing. 4. 26-35. 10.1109/MCC.2017.27.

[7] Skarlat, Olena and Nardelli, Matteo and Schulte, Stefan and Borkowski, Michael and Leitner, Philipp. (2017). Optimized IoT Service Placement in the Fog. Service Oriented Computing and Applications. 11. 427-443. 10.1007/s11761-017-0219-8.