Review of ”Secure and Sustainable Load Balancing of Edge Datacentres in Fog Computing - Deepak Puthal, Mohammad S. Obaidat, Priyadarsi Nanda, Mukesh Prasad, Saraju P. Mohanty, and Albert Y. Zomaya”

By Teodor Mircea Piron

In this paper review I will present the work done by Deepak Puthal, Mohammad S. Obaidat, Priyadarsi Nanda, Mukesh Prasad, Saraju P. Mohanty, and Albert Y. Zomaya in the fields of Load Balancing, Edge Datacentres and Fog computing.

As explained in the paper “Fog computing” is a term that describes a research trend to bring cloud services to the edge of the networks. Edge datacentres or EDC for short, are smaller facilities, than a cloud facility, that are deployed to decrease the latency and networks congestion by processing data streams and user requests in near real-time. The EDC’s have distributed deployment, positioned between the cloud datacentre and data sources.

The paper proposes a novel technique to authenticate the EDCs, for security reasons, and find less loaded EDC. The proposed technique proves to be more efficient than other approaches in finding less loaded EDC for task allocation and it improves the security because each EDC that is considered a destination must be authenticated before it can receive tasks.

Fog computing is similar to cloud computing, key differences being that Fog computing includes location awareness and EDC deployment. Many EDC are deployed geographically so that they can offer mobile, low latency data transparency over real-time request and responses. EDC are also used for the storage and provisioning of resources in accordance with the user requirements. Fog computing comes as a proposed solution to migrate the cloud resources to EDCs, which are deployed at the edge of networks. If we were to describe, where would we place Edge in a architecture, we would say that it belongs to a middle layer, at the bottom layer we would find several terminal devices such as wireless sensors, nodes or smart devices, devices that transmit date to upper layers. Next, as we said in the second layer, we would place Edge Datacentres devices, such as, routers switches and gateways. The third and last layer is usually a cloud datacentre made up of several high-end servers. This architecture is defined as a Fog computing architecture.

The security for EDC presents a problem due to their sparse distribution in a geographic scenario. Because the EDCs are deployed at the edges of the network and they are unattended, authentication of EDC with each other and with the cloud datacentre has become a key factor before load balancing. All EDCs are part of a distributed environment, this reasonably means that load balancing should be able to be applied in such an environment. There are two approaches when applying load balancing in a distributed environment, static and dynamic load balancing.

Static load balancing is achieved by providing a set of tasks to specific EDCs so that the performance function is minimized. We do this with either deterministic or a probabilistic means.

In a deterministic balancing technique EDC - A allocates the tasks that are overloaded to the EDC - B. In contrast in probabilistic balancing technique, EDC - A allocates the overloaded to EDC - B with the probability x and to EDC - C with the probability y.

One key difference between static and dynamic approaches is that static load balancing is that it does not consider the status of the destination EDC while making decision of load balancing. In the dynamic load balancing, the current load status of the individual EDCs is considered to decide the destination EDC. So, the assignment of tasks is done dynamically from an overloaded EDC to an underloaded EDC for efficient computing.

While there are several authentication methods available for the networks systems, the paper mentions that there is no authentication solution for the EDC, to the knowledge of the authors.

The proposed approach presented in the paper brings the following contributions:

- presents an adaptive EDC authentication technique with the help of a centralized cloud datacenter.
- brings a sustainable and dynamic load balancing technique by considering the load of the destination EDCs.
- combines both the authentication and load balancing technique to apply in the EDCs.

The work presented in the paper is related with the IoT. Mobile users with smart sensing devices which move randomly tend to offload tasks to their nearest EDC. That results in the load states of EDCs in various locations that differ greatly. Furthermore, unbalanced problem emerges, as some EDCs in the region could be overloaded while some other EDCs are in idle state. The paper inspires from other previous Load Balancing papers and EDC security papers, regarding the proposed Load Balancing solution and Security solution respectively.

Based on the current literature, there were no architectures that are used to authenticate the Edge datacentre before the task allocation. So, the paper proposed a novel architecture for the authentication and load balancing processes.

The paper first explains the security process, the "Secure Authentication". Based on the Fog computing architecture, all the data are stored and processed at the cloud layer, where EDCs work as the intermediate datacentres to reduce the latency of the user requests. Because cloud is always deployed in the secure environment, it has been considered that cloud should initiate the authentication process. Cloud initiates the process to assign initial ID (E_A) associated with the key (K_A) and shared key (K_C) for the individual EDCs during the EDCs deployment. EDCs use trusted modules to store the secret information from the cloud and the rekeying process. First step is the of the EDCs, each individual EDC starts to authenticate the rest of EDCs in its region. This is done to prevent malicious EDCs to participate in load balancing. Example: EDC-A is the first Edge Datacentre in his region to start the authentication process. Firstly, it combines its ID with the associate key and encrypts using the shared key initiated by the Cloud (E_{K_C} $(E_A \parallel K_A)$), then it broadcasts the generated packets by sending to all the EDCs in the region. When other EDCs get the authentication request packet, they decrypt it using the cloud shared key (D_{K_C} $(E_A \parallel K_A)$), the cloud offers a shared key before the load balancing process for every EDCs in the Fog architecture. Once destination EDC (EDC-B) gets the source ID and its associate key, it checks with the cloud to confirm authenticity of the source EDC. The cloud now responds to EDC-B request to verify if EDC-A is part of the architecture. Once the cloud confirms, EDC-B keeps a copy of the EDC-A details as an authenticated EDC. Then EDC-B concatenates its own ID with the associated, encrypts it using the associated key(E_{K_A} $(E_B \parallel K_B)$), and sends it to EDC-A with the format (E_{K_C} $(E_A \parallel E_{K_A}$ (E_B))), where E_B is encrypted with source EDC-B associate key. EDC-A decrypts the packet combines its own ID with it encrypts it again and sends it to the could so it will be verified again. When the cloud will receive the packet from EDC-A it will decrypt it using the shared key and then it will receive the associated key E_B, to validate EDC-B. When it is validated, cloud concatenates the associate key with E_B and sends it back to EDC-A. After it decrypts the packet, it compares the key with the one received earlier from EDC-B. If there is a match EDC-A will combine its ID and EDC-B ID with the associate key, encrypts it and send it to EDC-B. Once this combined packet is received by the EDC-B, it confirms that both EDC-A and EDC-B are now authenticated to each other for load balancing.

The paper follows Breadth First Search method to design the proposed load balancing technique. There have been used two parameters, m and n to maintain the load of all the EDCs, where m is the current load and n is the maximum capacity to process the tasks. A parameter p is used to compute the current load statues ($p = {}^{m}/_{n}$). Individual EDCs get load balancing requests from other EDCs to process their tasks.

If EDC-A is overloaded, then EDC-A broadcasts a control packet by sending requests to the other EDCs that are in the region with its own ID and the received load information i.e. (E_A, L_A). The ID of EDC-A is defined as E_A, whereas the received load information is defined as L_A. The neighbour EDC (named as EDC-B) checks the received ID and compares it with its own database. If it matches, EDC-B then looks for the load information from the control packets, otherwise, it ignores the control packet to avoid the DoS attack. While checking the EDC-A load information, EDC-B first checks its own load information using value of p. If p is less than or equal to 0.6, then it moves forward to get the available resources (m and n) to process the invited tasks. If the available resource is higher than the required resource to process the invited task, then EDC-B processes the positive response packet to the EDC-A. Otherwise, EDC-B becomes silent and does not send any response. While preparing the response, EDC-B includes its own identity (E_A), associate key and the current free resource of the datacentre, p. Finally, it generates the response packet, encrypts it with the public key of the destination EDC, Kpu_i, (E_{Kpu_i} (E_B $||K_B$ $||p$)) and sends it to the required destination EDC-I. After receiving the encrypted data packets, EDC-I uses its own private key, Kpr_i to decrypt the data packets (D_{Kpu_i} (E_B $||K_B$ $||p$)). After decryption, EDC-I checks the source ID and compares it with its own database to find if there is a match. If a match is found, it extracts the associate key with the ID (K_B'). Also, EDC-A compares the associated key with the received key (K_B' = K_B) and in case of a match, it accepts the response otherwise, it rejects it. In a similar way, EDC-A gets several responses from different EDCs in the region. EDC-A also compares the values of neighbor p and finds the lowest value to select destination. Finally, EDC-A sends tasks to the authenticated neighbor EDCs to process.

The paper briefly presents a definition of an “Attack on authentication”. An intruder “Ma” attacks on authenticity and is capable of monitoring, intercepting, and introducing itself as an authenticated EDC to start load balancing process. This is followed by a Claim that such an attack could be realized and a Proof that demonstrates that is close to impossible to pass the security put in place by the proposed architecture.

In the next part the paper presents the experimental part of the work, which is comprised of testing the performance of the proposed architecture on a PC using Matlab simulation environment. All simulations were executed for ten times each. The successful heat ratio was calculated in percentages (%), and it was found that the proposed load balancing solution gives 100% success rate to find the most suitable and less loaded EDC. The performance was compared with static, random and proportional allocation and it was found that proportional allocation gives always better result compared to others.

Lastly the paper presents the conclusion that was reached at the end of the work and the future development plans. The paper proposes a novel secured and sustainable load balancing solution for EDCs in Fog computing environment.