RESOURCE ALLOCATION ALGORITHMS FOR UPF INSTANCES IN THE 5G CORE NETWORK

Tran Thi Xuan
TNU - University of Information and Communication Technology

ARTICLE INFO

Received: 04/10/2023
Revised: 24/10/2023
Published: 25/10/2023

KEYWORDS

5G core
UPF instance
Resource allocation
Load balancing
Energy saving

ABSTRACT

The service based architecture of 5G core network allows network services to flexibly and automatically scale up and down according to user traffic. Cloud-based implementation of 5G core network functions is a promising approach for serving increasing 5G user devices. However, allocating cloud resources for scaling 5G core network functions is an NP-hard problem due to the diversity of user demands and multiple criteria from both service providers and users' requirements of service quality. At present, the deployment of 5G network function on cloud is still limited. It is crucial to investigate the cloud-based approach for ensure full deployment effective and efficient in the near future. This study proposed two algorithms, Load-Balance and Energy-Saving, for cloud resource allocation to automatically scale up and down the 5G UPF instances. A simulation software is developed to model an Infrastructure-as-a-Service cloud and implement these resource allocation algorithms. Numerical results indicate that Energy-Saving can lead to better resource utilization and reduce over 33% energy consumption, while Load-Balance assures cloud server to not overloaded.

GIẢI THUẬT PHÂN BỔ TÀI NGUYÊN CHO THÀNH PHẦN UPF TRONG MẠNG LÕI 5G

Trần Thị Xuân
Trường Đại học Công nghệ Thông tin và Truyền thông – ĐH Thái Nguyên

THÔNG TIN BÀI BÁO

Ngày nhận bài: 04/10/2023
Ngày hoàn thiện: 24/10/2023
Ngày đăng: 25/10/2023

TỪ KHÓA

Mạng lõi 5G
Thành phần mạng UPF
Phân bổ tài nguyên
Cân bằng tải
Tiết kiệm năng lượng

TÓM TẮT

Kiến trúc dựa trên dịch vụ của mạng lõi 5G cho phép các dịch vụ mạng tăng giảm quy mô linh hoạt và tự động theo lưu lượng người dùng. Việc triển khai các chức năng mạng lõi 5G dựa trên đám mây là một phương pháp đầy hứa hẹn để phục vụ số lượng thiết bị người dùng 5G ngày càng tăng. Tuy nhiên, vấn đề phân bổ tài nguyên đám mây để mở rộng quy mô các chức năng mạng lõi 5G là một bài toán phức tạp do nhiều yêu cầu và tiêu chí từ cả nhà cung cấp dịch vụ cũng như yêu cầu của người dùng về chất lượng dịch vụ. Hiện tại, việc triển khai chức năng mạng 5G trên đám mây vẫn còn hạn chế. Nghiên cứu cách tiếp cận dựa trên đám mây rất quan trọng để đảm bảo triển khai đầy đủ hiệu quả và hiệu quả trong tương lai gần. Nghiên cứu này đề xuất hai thuật toán Cân bằng tải và Tiết kiệm năng lượng cho phân bổ tài nguyên đám mây nhằm tự động tăng và giảm quy mô phiên bản 5G UPF. Một phần mềm mô phỏng được viết để mô hình hóa đám mây IaaS và triển khai các thuật toán phân bổ tài nguyên này. Các kết quả số học chỉ ra rằng Tiết kiệm năng lượng có thể dẫn đến việc sử dụng tài nguyên tốt hơn và giảm hơn 33% mức tiêu thụ năng lượng, trong khi Cân bằng tải đảm bảo máy chủ đám mây không bị quá tải.

DOI: https://doi.org/10.34238/tnu-jst.8900

Email: ttxuan@ictu.edu.vn

1. Introduction

The 5th generation mobile network (5G network for short) has been the state-of-the-art technology and infrastructure of today mobile network. With a distributed and diverse architecture, the research on resource allocation and use for 5G network deployment has attracted great interest of researchers and businesses around the world [1] – [5]. In its architecture, an access network (NG-RAN), a collection of distributed base stations, is responsible for radio spectrum or radio access network management to meet the quality of service (QoS) required by users while a core network (NG-Core) provides Internet connectivity for both voice and data services, controls and ensures this connection to meet QoS requirements [6], [7].

The 5G core network achieves a significant improvement over the previous generation network with its service-based architecture; in which, components are network service software that can flexibly initialize, launch, scale up/down according to user traffic [4]. UPF (User Plane Function) is a major network service component of the 5G core and responsible for managing and maintaining Packet Data Unit (PDU) session which is an abstract data path between a user and a data network. UPF instances can run on virtual machines or as containerized software in containers (using packaging and package management technologies like Docker, Kubernetes, etc.) in cloud centers.

With the rapid increase in the number of 5G users, auto-scaling UPF to ensure connectivity between users and data networks plays an important role in service quality assurance. Therefore, the issue of allocating computing resources for scaling UPF instances has attracted researchers' attention increasingly. The problem is relatively new and open for new solutions. The authors of [8] proposed an automatic resource allocation solution for the virtual network function based on the resource computation pattern. In study [9], the authors proposed a queuing model to study the automatic scaling of UPF instances based on the threshold of the number of available UPFs.

Running the 5G core network efficiently and saving energy has always been an important goal for network operators. Therefore, this study explores the requirements for UPF implementation and implements algorithms to allocate resources for UPF scaling efficiently.

The paper is organized in what follows. Section 2 briefly explains the 5G core network, how to implement its components, and the proposed resource allocation algorithms for UPF instances. Section 3 presents the results and analysis on the efficiency of the resource allocation algorithms. The conclusion of the paper is given in Section 4.

2. Background and Research methodology

2.1. The 5G core network

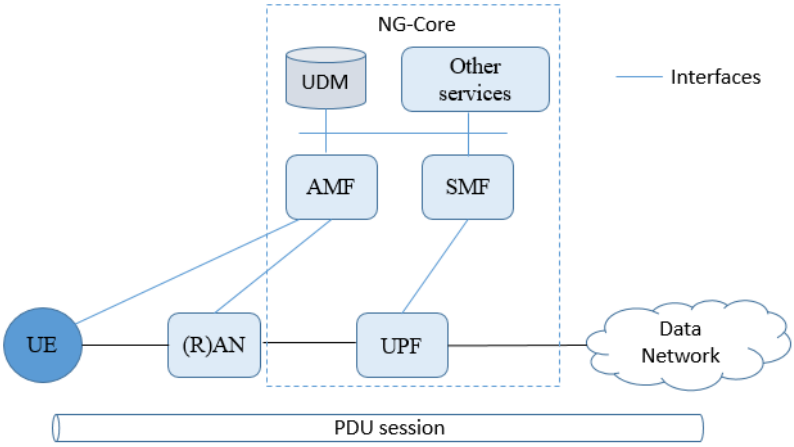

Figure 1. *A model of the 5G core network components*

A user equipment (UE) is connected to a Radio Access Network (RAN) via a 5G base station (gNB), which in turn connects to the 5G core network for data routing and forwarding between the UE and a data network [4]. In 5G networks, the connection between UE and a data network (Data center) is done through the establishment of a Packet Data Unit (PDU) session, which is an abstract data path and managed by a 5G core network.

The 5G core network consists of a set of functions that can run as micro-services. Each network function component provides one or more services to others through service-based interfaces as illustrated in Figure 1 [4]. These network functions can be classified into the control plane or the data plane.

2.1.1. Data plane

Base stations (gNB) and User Plane Function (UPF) instances perform data plane tasks. They provide the essential procedures for data transfer between UE and a data network. Before data transfer happens, an abstract data path - the PDU session- must be established and managed by the UPF. The UE initiates the PDU session establishment process by sending a request to the 5G core network. The request includes information about the type of service that the UE needs to use and the type of traffic. Once the PDU session has been established, the UE can use it to send and receive data. The UPF instance is responsible for maintaining PDU connectivity, transmitting packets and ensuring that the network is used efficiently and that the UE receives the appropriate QoS.

2.1.2. Control plane

The control plane of 5G systems has two main components involved in establishing and maintaining the connection between the UE and the data network: The Access and Mobility Management Function (AMF) and the Session Management Function (SMF). The AMF manages user authentication, establishes the data connection between the UE and the data network, and performs the necessary procedures to handle UE mobility. The SMF is responsible for user session management, assigning a session PDU to an appropriate UPF component.

2.2. UPF instance management and scaling

In practice, there are three implementations of the service components of the 5G core network: a dedicated hardware; software versions running on specialized hardware; or as virtualized/cloud-based software-as-a-service that is initialized and running on a cloud infrastructure platform.

As the number of users joining the 5G network increases, the number of PDU sessions being established increases. Therefore the number of UPF services also needs to be scaled up to maintain the connection of PDU sessions and manage transmission quality.. Therefore, deploying UPF instances in the form of cloud service is the main trend to satisfy the growth in 5G network users. The option of using UPF instances or other network functional services in the form of cloud service means that the cloud infrastructure will run virtualized components such as virtual machines (VMs) or containers (which are virtualized computing units that run independently and can run the application without being affected by system environment) [10].

When deploying UPF instances as cloud service, scaling up/down of UPF or more is easily done automatically by initializing/stopping a virtualized compute unit (VM, container). The management and allocation of cloud resource usage is done automatically at the cloud infrastructure. In the next part of the article, the author will present some considered algorithms to evaluate the performance of using computing resources to run UPF network services.

2.3. System modeling and Resource allocation for UPF instances

2.3.1. System modelling

Considering a 5G core network model is a computing system consisting of a set of *S* commercial-off-the-shelf (COTS) servers the Standard Performance Evaluation Corporation - SPEC (Standard Performance Evaluation Corporation) [11].

Each server *s* (*s* = 1, 2, ... ,S) is characterized by *N* (the number of CPUs) and *R* is the amount of main memory (RAM) of the computer in Gigabytes (GB).

Assume that the core network deploys cloud infrastructure services in the form of virtual machines (VMs). Each VM is provided with 1 CPU and 1GB of RAM. Each host *s* has *M* virtual machines: $\mathrm{M} = \min(N, R)$. We assume that each VM is capable to run an UPF instance. Therefore, each server is capable of running *M* UPF instances. For simplification, we also assume that each UPF instance can maintain and manage a PDU session between an UE and a data network. The system hence can simultaneously run $S \times M$ virtual machines to maintain PDU sessions. To perform VM allocation to UPF instances automatically, the scheduler needs to determine the appropriate VM to launch the 5G UPF service. The resource allocation for UPF instances is presented in what follows.

2.3.2. Resource allocation algorithms for UPF instances

The problem of resource allocation is always a challenging topic in all computing systems, including cloud computing. The reasonable and effective allocation should pay attention to ensuring the QoS for users and the efficient use of resources to reduce operating costs for the service provider.

For a straightforward establishment of the PDU session when requested, the cloud needs to ensure the UPF service availability. This means, there is always at least one UPF service running and ready to establish a new PDU session.

Let $vm(s)$ be the number of virtual machines that can be launched on the server *s* (s = 1,2,...S; $vm(s) \leq M$). When a new request to establish a PDU session arrives, the SMF will first select an available UPF instance to perform the setup. The scheduler will then consider launching a VM to run the new UPF for service availability. In the event that no UPF is available to establish a PDU session upon the user's request, the request will be blocked.

This paper examines two algorithms for scheduling and allocating cloud resources to UPF instances called Load-Balance algorithm and Energy-Saving policy.

Load-Balance algorithm (Alg. 1): The Load-Balance (LB) algorithm aims to balance the workload handled in different servers to avoid having some machines overloaded while the other machines are under loaded. The LB algorithm does the scheduling task as the following: First, it routes the PDU session establishment request to a ready UPF. The scheduler then launches a new UPF in a new initiated VM. To select a VM for UPF, the scheduler first calculates the utilization of cloud servers. If server s has load lower than the hard threshold θ, the scheduler will launch a new VM on server s and a UPF service will be made available on s. If all servers have load higher than θ, the lowest load server will be selected. In case there is none available UPF, the user's request will be blocked.

Alg. 1. Load-Balance allocation algorithm

```
for a new request of PDU session establishment do
    if FOUND an available UPF then
        ESTABLISH the PDU session by the UPF
    else
        BLOCK user request
    end if
    for each server s do
```

```
            CALCULATE load of server s: load(s) (s = 1,2,..S)
            if (load(s) < θ and vm(s) > 0) then
                LAUNCH a new VM and INITIATE a new UPF on server s
                STOP the algorithm
             end if
        end for
         if load of all servers is higher than θ then
            CHOOSE server with the lowest load
         end if
    end for
```

Energy-Saving algorithm (Alg. 2): Energy-Saving (ES) algorithm for scheduling aims to lessen the operation energy consumption of the system by reducing the active servers in alignment to user traffic and applying dynamic power management (DPM) technique. Applying the ES algorithm, the scheduler first sends the PDU session establishment to the available UPF instance. Then it seeks in sequence and selects a server that can launch a new VM but remains the lowest free resource. Afterward, a new UPF instance will be launched inside the selected server.

Alg. 2. Energy-Saving algorithm

```
  for a new request of PDU session establishment do
     if FOUND an available UPF then
         ESTABLISH the PDU session by the UPF
     else
         BLOCK user request
     end if
     for each server s in system do
         if (vm(s) > 0 and vm(s) is the smallest) then
             LAUNCH a new UPF instance on server s
             DECREMENT vm(s)
             STOP the algorithm
         end if
     end for
  end for
```

2.3.3. Simulation inputs and metrics

To evaluate the two aforementioned scheduling policies, a simulation written in C language is used to model the cloud system and the implementation of scheduling policies.

A system of S = 40 homogeneous COTS servers, of which each has 64 cores, 128 GBytes of RAM, and the full-load active power consumption of 269W [12]. Assuming that each server dedicates 2 cores for system operation task. The number of virtual machines per server is min(62,128)., i.e. 62. Assuming that the average service rate of user requests is 1 request/second. Hence, the system is capable to process in average of 40 x 62 = 2480 requests per second. We examine the system performance at diverse arrival rates (denoted by λ) ranging from in average 700 to 2200 incoming requests per second. Note that users' requests arrive the system following Poisson distribution and the service time of the system is exponentially distributed. For Load-Balance algorithm, the load threshold is 0.7.

The average load of each cloud server during the operation time is used as the performance metric to compare the studied algorithms.

Another metric considered in the evaluation is the average energy consumed by the system to run and manage an UPF instance during its operation time.

Let t is the operation time of the system that have completed to run and manange J UPF instances (jobs). Let t_s and P_s be the total active time ($t_s \leq t$) and the power consumption in the active state of the server s. We assume that a free server can be powered off for energy saving and powered on when it is needed without a cost of delay. Let AE denote the average energy consumption per job. The AE is calculated as follows:

$$AE = \frac{1}{J}\sum_{s=1}^{40} P_s * t_s \quad (1)$$

Table 1 gives the list of notations used in the work.

Table 1. *List of notations*

Notations	Description
S	The number of servers in considered system
N	The number of cores of a server s (s = 1, 2, ... , S)
R	The RAM capacity of a server s
M	The maximum virtual machine launched simultaneously in a server s
J	The total number of completed jobs
P_s	The total power consumption in the active state of the server s
t_s	The total active time of a server s
AE	The average energy consumption per job (Ws/job)
λ	The arrival rate of requests to the system

3. Results and discussion

To compare the two policies in their affect to system efficiency, we use the metric called the average load of cloud servers, which directly indicates how servers are utilized. The less busy servers, the less energy consumes. The reason is that idle servers can be powered off by DPM technique for energy saving.

Figure 2 and Figure 3 present the average load of cloud servers during operation when Load-Balance and Energy-Saving are applied, respectively.

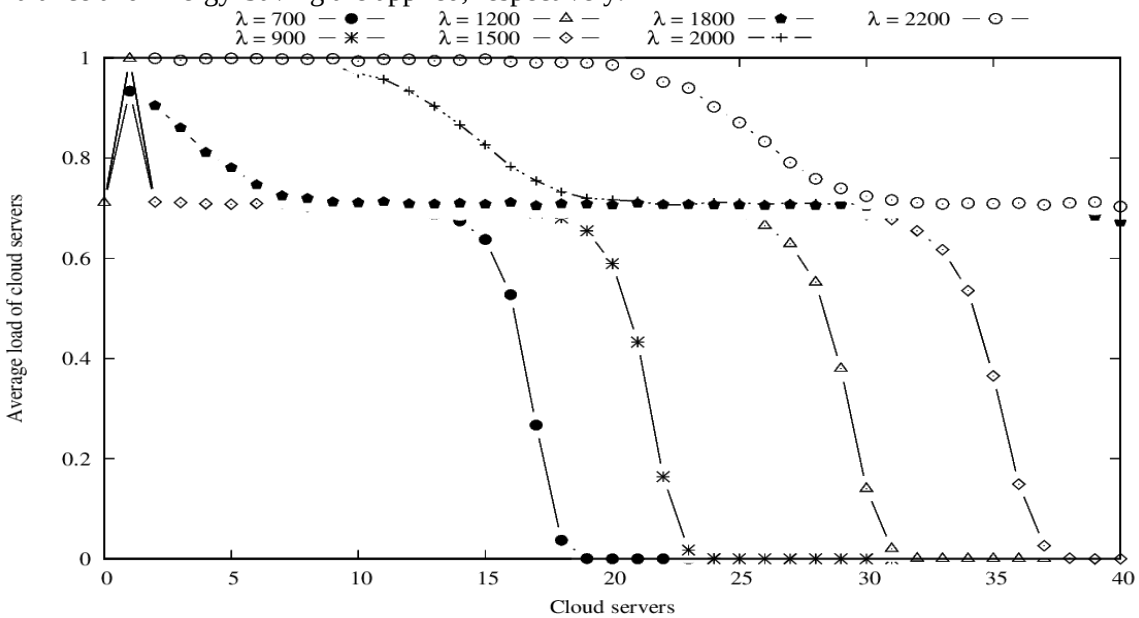

Figure 2. *The average load of cloud servers with Load-Balance algorithm*

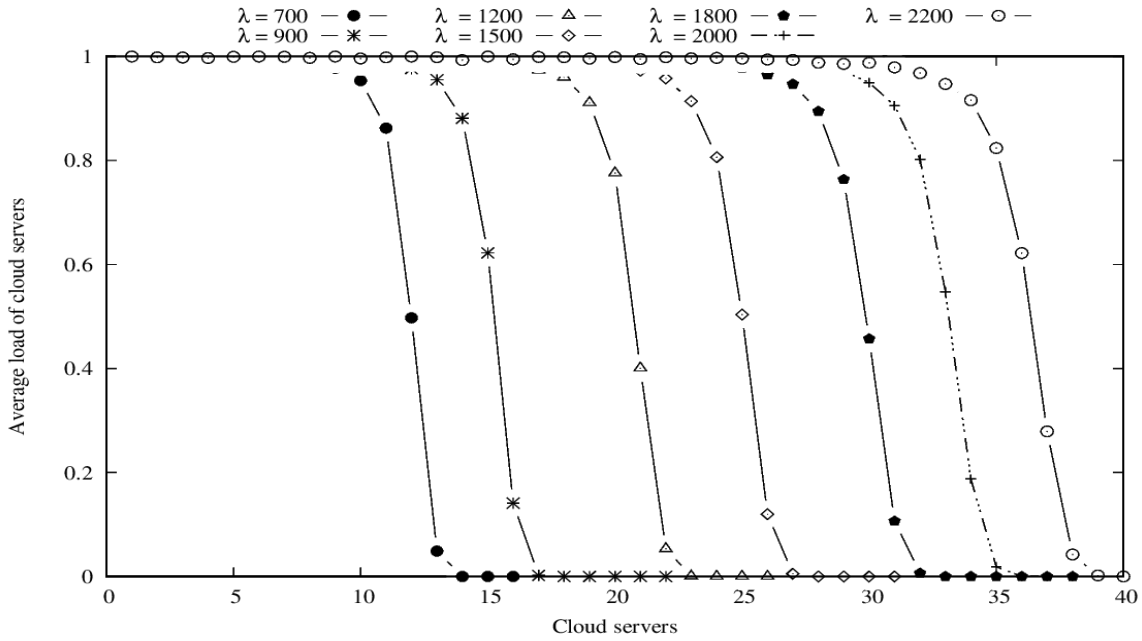

Figure 3. *The average load of cloud servers with Energy-Saving algorithm*

It can be observed that user requests of PDU establishment are distributed more evenly across cloud servers with Load-Balance (LB) as shown in Figure 2.

Figure 3 shows that at the low and medium traffic intensity there are a number of unutilized servers with the implementation of Energy-Saving (ES) policy, which is the condition for applying dynamic switching off technique to save energy.

For instance, with the traffic intensity of 700 requests/second, the number of servers in use is 14 and 19 with ES and LB policy, respectively. When the intensity increases up to 1500 requests/second on the average, 37 of 40 servers are in use if LB policy is applied while only 26 servers are busy if ES algorithm is implemented. That means those free servers can be switched off for energy saving.

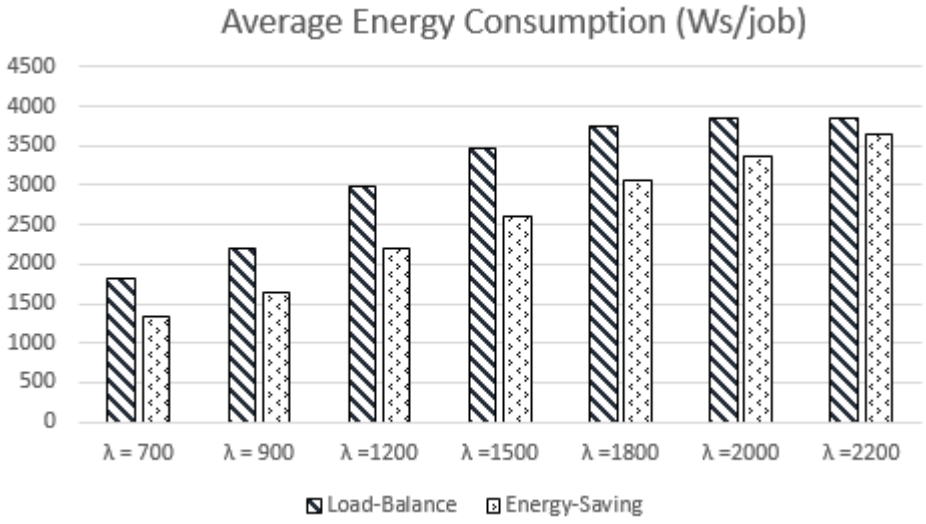

Figure 4. *The average energy consumption (W.s/job) with different arrival rates*

Figure 4 presents the average energy consumed by the system to run and manage an UPF instance. It can be observed that ES policy can save over 33% energy in comparison to LB at low to moderate workload intensity. The savings decrease and become insignificant at high workload rate, which we can observe only a reduction of roughly 5.5% in energy consumption.

4. Conclusion

Service-based architecture of 5G core network is a promising solution for rapidly growing numbers of 5G devices and the future 6G network. This study investigates the issue of allocating cloud resources for the most scalable service instance of the 5G core – the UPF instance.

We examine two algorithms called Load-Balance and Energy-Saving and evaluate their impact on the system resource utilization. Numerical results show that Energy-Saving yields better resource efficiency and can save energy in comparison to Load-Balance algorithm. However, other factors of system performance (e.g. data latency, throughput, etc.) should be involved in the consideration to have more sufficient view of system in future study.

Acknowledgement

This research is funded by Thai Nguyen University of Information and Communication Technology under grant number T2023-04-07.

REFERENCES

[1] N. Hassan, K. A. Yau, and C. Wu, "Edge Computing in 5G: A Review," *IEEE Access*, vol. 7, pp. 127276 – 127289, 2019.

[2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," *MCC '12: Mobile cloud comput.,* 2012, pp. 13-16.

[3] W. Chien, S. Huang, C. Lai, and H. Chao, "Resource Management in 5G Mobile Networks: Survey and Challenges*," Journal of Information Processing Systems*, vol. 16, no. 4, pp. 896-914, 2020.

[4] T. T. Xuan, "A review on resource management and allocation in the 5G mobile network," *TNU Journal of Science and Technology*, vol. 227, no. 18, pp. 44-52, 2022.

[5] D. D. Lieira, M. S. Quessada, A. L. Cristiani, and R. I. Meneguette, "Algorithm for 5G Resource Management Optimization in Edge Computing," *IEEE Latin America Transactions*, vol. 19, no. 10, pp. 1772 – 1780, 2021.

[6] L. Ma, X. Wen, L. Wang, Z. Lu, and R. Knopp, "An SDN/NFV based framework for management and deployment of service based 5G core network," *China Communications*, vol. 15, no. 10, pp. 86-98, 2018.

[7] T. V. K. Buyakar, A. K. Rangisetti, A. A. Franklin, and B. R. Tamma, "Auto scaling of data plane VNFs in 5G networks," *in Proceedings of the 2017 13th International Conference on Network and Service Management (CNSM)*, Tokyo, Japan, 2017, pp. 1-4.

[8] M. Asif, M. Afaq, A. K. Talha, J. D. R. Javier, I. Javed, I. Ihtesham, and S. Wang-Cheol, "Energy-efficient auto-scaling of virtualized network function instances based on resource execution pattern," *Computers & Electrical Engineering*, vol. 88, 2020, doi: 10.1016/j.compeleceng.2020.106814.

[9] R. Csaba and V. D. Tien, "A Queueing Model for Threshold-based Scaling of UPF Instances in 5G Core," *IEEE Access*, vol. 9, pp. 81443-81453, 2021, doi: 10.1109/ACCESS.2021.3085955.

[10] C. Pahl, A. Brogi, J. Soldani, and P. Jamshidi, "Cloud Container Technologies: A State-of-the-Art Review," *IEEE Trans. Cloud Comput.*, vol. 7, no. 3, pp. 677–692, 2019.

[11] Standard Performance Evaluation Corporation, "The SPECpower_ssj2008 benchmark," 2022. [Online]. Available: https://www.spec.org/. [Accessed July 18, 2023].

[12] Standard Performance Evaluation Corporation, "Hewlett Packard Enterprise ProLiant DL325 Gen10 Plus," 2022. [Online]. Available: https://www.spec.org/power_ssj2008/results/res2022q1/power_ssj2008-20220301-01168.html. [Accessed July 18, 2023].