Resource Allocation Influence on Application Performance in Sliced Testbeds

Rodrigo Moreira[1], **Larissa F. Rodrigues Moreira**[1,2], **Tereza C. Carvalho**[3],
Flávio de Oliveira Silva[2,4]

[1]Federal University of Viçosa (UFV)
38.810-000 – Rio Paranaíba – MG – Brazil

[2]Faculty of Computing – Federal University of Uberlândia (UFU)
38.408-100 – Uberlândia – MG – Brazil

[3]University of São Paulo (USP)
05.508-010 – São Paulo – SP – Brazil

[4]Department of Informatics – School of Engineering
University of Minho – Braga, Portugal

{rodrigo, larissa.f.rodrigues}@ufv.br, flavio@di.uminho.pt,

terezacarvalho@usp.br, {larissarodrigues, flavio}@ufu.br

Abstract. *Modern network architectures have shaped market segments, governments, and communities with intelligent and pervasive applications. Ongoing digital transformation through technologies such as softwarization, network slicing, and AI drives this evolution, along with research into Beyond 5G (B5G) and 6G architectures. Network slices require seamless management, observability, and intelligent-native resource allocation, considering user satisfaction, cost efficiency, security, and energy. Slicing orchestration architectures have been extensively studied to accommodate these requirements, particularly in resource allocation for network slices. This study explored the observability of resource allocation regarding network slice performance in two nationwide testbeds. We examined their allocation effects on slicing connectivity latency using a partial factorial experimental method with Central Processing Unit (CPU) and memory combinations. The results reveal different resource impacts across the testbeds, indicating a non-uniform influence on the CPU and memory within the same network slice.*

1. Introduction

Network slicing enables logical, service-tailored, and independent networks to coexist in a shared physical network [Feng et al. 2020, Moreira et al. 2021, Donatti et al. 2023]. Network slicing allows application verticals with different Service-Level Agreement (SLA) to be orchestrated under heterogeneous underlying infrastructures. The optimal allocation of resources to network slices is fundamental for cost reduction, energy harvesting, and compliance with SLA [Karbalaee Motalleb et al. 2023].

Many efforts, such as combinatorial optimization and computational intelligence, have been aimed at effectively managing resource allocation for network slices [Debbabi et al. 2020]. Although approaches to resource allocation are still under development, understanding the behavior and observability of this allocation on network slicing performance still poses challenges [S. et al. 2023].

In this paper, we propose and evaluate the influence of CPU and Random-Access Memory (RAM) resource allocation on the performance of a network slicing application deployed on Future Internet Brazilian Environment for Experimentation New Generation (FIBRE-NG) and Fabric testbeds. Using the partial factorial performance evaluation method, we built resource allocation templates. We combined them for allocation to the network slice and measured the influence of the combination on the latency response variable for Write (W) and Read (R) operations.

The remainder of this paper is organized as follows. In Section 2, we contextualize the related work on testbed experimentation. The proposed experimental method is presented in detail in Section 3, followed by a description of the experimental setup and results in Section 4. Section 5 discusses concluding remarks and future research directions.

2. Related Work

In this section, we present related works concerning the deployment of network slices in experimental testbeds.

[Dong et al. 2023] presents LinkLab 2.0, a multi-tenant IoT testbed with edge-cloud integration. The authors aim to address the challenges of programming and experimenting with heterogeneous IoT, edge, and cloud devices in a unified way. They design and implement a three-tiered architecture for managing the devices, a one-site programming framework for supporting serverless functions and computation offloading, and an anomaly detection system for ensuring reliability. They deploy LinkLab 2.0 with over 420 real devices of 14 types and support various research and educational experiments.

[Morel et al. 2023] introduce a method for managing network services in Visual Cloud Computing (VCC) applications that use video streaming across edge-to-cloud systems. It utilizes P4-enabled programmable data planes and In-band Network Telemetry (INT) to boost video delivery quality and performance. Tested on the FABRIC network, it uses a customized P4 program merging Multi-Hop Route Inspection (MRI) and port forwarding to monitor and control congested network traffic. Results show improvements in packet loss, throughput, and delay compared to standard switches.

[Arora et al. 2024] presents a Cloud-native Lightweight Slice Orchestration (CLiSO), a framework for managing network slices using Kubernetes and a CISM agent. It emphasizes Domain Specific Handlers (DSHs) for automated network slicing management. The framework is evaluated by orchestrating OpenAirInterface functions on different cloud platforms, showcasing efficient orchestration, low resource use, resilience, and quick deployment.

3. Evaluation Proposal

In this paper we designed and evaluated the performance of a network slicing application on real nationwide testbeds using partial factorial methodology. For this evaluation, we used the Cassandra application, a scalable, fault-tolerant, distributed key-value database system, for managing extensive data across multiple locations [Padalia 2015]. We deployed Cassandra in two experimental testbeds to measure the influence of allocating

CPU and RAM resources to the microservices of the network slice on the latency in read and write operations.

3.1. Experimental Setup

We illustrate the evaluation method and the experimental flow and testbed in Figure 1. According to step one ❶ , we deployed a Cassandra application on two different testbeds: Future Internet Brazilian Environment for Experimentation (FIBRE) New Generation and Fabric [Salmito et al. 2014, Baldin et al. 2019]. Our Cassandra application is based on the microservices model, in which different service parts (containers) run on different computing nodes of the testbed. The basic configuration of Cassandra in our experimental evaluation was three (3) replicas, each with 1024 data tokens.

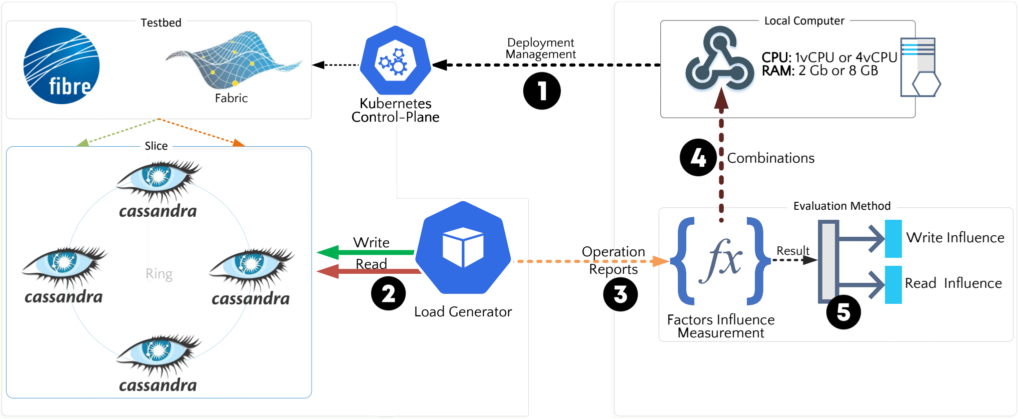

Figure 1. Evaluation Method.

In step two ❷ , there is a workload generator container that triggers operations towards the Cassandra ring. Internally, the workload generator container is equipped with the `cassandra-stress` application, where we configure the operation parameters (W or R) such as time, data volume (10,000 entries), distribution (replication factor 2), and consistency level (*quorum*). The workload generated on the Cassandra ring generates statistical outputs (operations per second, lines per second, latency and others). In step three ❸ we use these statistics in our partial factorial influence method on the response variable latency.

Variations in resource allocation were combined using a local script and reorganized in step four ❹ . In step five ❺ , we determine the influence of the CPU and RAM factors and their levels on the latency of W and R in different testbeds. In our experimental design, we built a slice on the FIBRE-NG and Fabric testbeds considering the computational nodes presented in Table 1.

3.2. Partial Factorial Model

The factorial method comprises K factors with n_i levels for each i factor. We used the CPU and acRAM factors allocated to the container of each node of the Cassandra ring; for each factor, the levels were 1vCPU, 4vCPU, 2 Gb RAM, or 8 GB RAM. Four experiments (2^2) were run on the combinations (CPU and RAM) to obtain the values y_1, y_2, y_3 and y_4, which are the averages of the write and read operations for each testbed. We performed

Table 1. Allocation of physical nodes to network slice service.

Testbed	Pod Name	Management IP	Node
FIBRE-NG	cassandra-0	10.50.103.245	Santa Catarina
	cassandra-1	10.50.79.144	Rio Grande do Sul
	cassandra-2	10.50.117.161	Paraíba
	loadgen	10.50.83.25	Rio Grande do Norte
Fabric	cassandra-0	192.168.135.18	Dallas
	cassandra-1	192.168.104.20	Salt Lake City
	cassandra-2	192.168.3.78	Lexington
	loadgen	192.168.135.13	Dallas

an analysis using the regression model generated by the experimental combinations as follows: $y = q_0 + q_A X_A + q_B X_B + q_{AB} X_{AB}$.

By replacing the four observations from the experiment with the model, we obtain $q_0 = \frac{1}{4} \times (y_1 + y_2 + y_3 + y4)$ which is the average of the latencies of the operations (W and R), $q_A = \frac{1}{4} \times (-y_1 + y_2 - y_3 + y4)$ which is the influence of the Factor $\mathcal{A}$ (CPU) on the response variable (latency). While $q_B = \frac{1}{4} \times (-y_1 - y_2 + y_3 + y4)$ is the influence of the Factor $\mathcal{B}$ (RAM) on the response variable (latency), and $q_{AB} = \frac{1}{4} \times (y_1 - y_2 - y_3 + y4)$ is the influence of the $\mathcal{AB}$ Factors simultaneously on the response variable.

From the values q_0, q_A, q_B and q_{AB} we determine the sum of squares that gives the total variation of the response variables and variations in the influences of the Factors $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{AB}$ simultaneously. With this, we calculated the total variance by following $SS_T = 2^2 q_A^2 + 2^2 q_B^2 + 2^2 q_{AB}^2$. Once we have the total variance, we calculate the variance of each factor by dividing by SS_T, where the factor $SS_A = 2^2 q_A^2$ is the influence of Factor $\mathcal{A}$ (CPU) on the response variable latency in operation (W and R); $SS_B = 2^2 q_B^2$ which is the influence of Factor $\mathcal{B}$ (RAM) on the response variable; and finally, $SS_{AB} = 2^2 q_{AB}^2$ which is the interaction of Factors $\mathcal{AB}$ (CPU and RAM) on the response variable.

4. Results and Discussions

Initially, we measured the overhead of deploying a network slice on both testbeds, as shown in Fig. 2. Fabric required less time to deploy the same network slice (with the template described in the manifest file). Quantitatively, FIBRE-NG required 66.36% more time (73.2 s) to deploy the network slice than Fabric (44 s) did. This variation in deployment time may or may not be associated with heterogeneity or the amount of computational resources available to the network slice.

It is assumed that testbeds have different heterogeneous resources with variations in both hardware and software, which can lead to differences in averages. However, the aim of the experiment was to verify that identical resource allocation profiles may not lead to the same behavior for network slices on different testbeds.

We carried out the experimental evaluation following the planned combinations (refer to Section 3.2) to measure the averages of the response variables for each operation (W or R). According to Table 2, we observed different averages for each experimental combination in both FIBRE-NG and Fabric.

Using the partial factorial method, we analyzed whether the allocation of CPU and RAM to the network had similar effects on the response variable in different testbeds. In Table 3, we note that the CPU allocation (Factor $\mathcal{A}$) has 93.32% influence on the latency

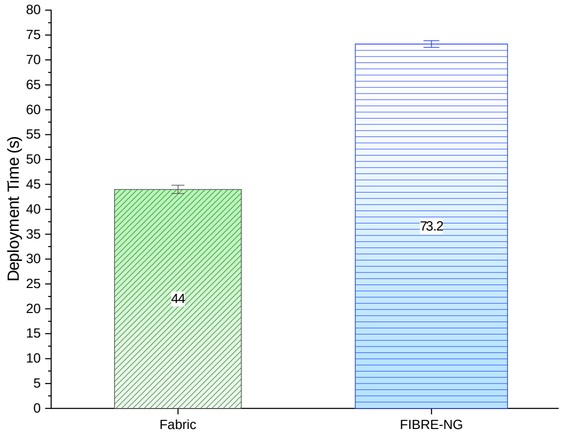

Figure 2. Deployment Time Comparison: FIBRE-NG and Fabric testbeds.

Table 2. Network Slice performance according to resource allocation combinations on testbeds.

				Measured on FIBRE-NG		Measured on Fabric	
Experiment	**CPU**	**RAM**	**Y: Write/Read Latency (ms)**	**Write Latency (ms)**	**Read Latency (ms)**	**Write Latency (ms)**	**Read Latency (ms)**
#1	1vCPU (-1)	2 Gb RAM (-1)	y_1	156.9	100.3	719.2	616.5
#2	4vCPU (1)	2 Gb RAM (-1)	y_2	93.5	99.1	913.3	830.8
#3	1vCPU (-1)	8 Gb RAM (1)	y_3	186.6	101.3	404.4	385.8
#4	4vCPU (1)	8 Gb RAM (1)	y_4	93.0	98.2	265.4	275.8

of the write operation, whereas the simultaneous influence of Factors $\mathcal{B}$ and $\mathcal{AB}$ on the latency is low.

We observed a similar pattern on the same testbed: CPU allocation (Factor $\mathcal{A}$) had 83.63% influence on latency. However, the simultaneous allocation of CPU and RAM (Factor $\mathcal{AB}$) contributed 16.33% to latency. This is likely due to the nature of the read operation, which includes input-output (IO) operations to access information from the input and output devices. This variation suggests that even identical resource allocation profiles, depending on the testbed, may not lead to the expected behavior of the network slice.

Table 3. Resource allocation influence for network slice performance on different testbeds.

		Influence		
Testbed	**Operation**	**Factor $\mathcal{A}$ (CPU)**	**Factor $\mathcal{B}$ (RAM)**	**Factor $\mathcal{AB}$ (CPU and RAM)**
FIBRE-NG	Write	93.32%	3.22%	3.45%
	Read	83.63%	0.04%	16.33%
Fabric	Write	0.29%	89.05%	10.66%
	Read	1.48%	84.18%	14.34%

In the Fabric testbed, we identified the different influences on resource allocation during operations W and R. In writing, memory allocation had 89.05% influence on latency, while resource interaction accounted for 10.66% in response. Analysis of `cassandra-stress` indicated timeouts for the *quorum* consistency level, increasing the response time, and possibly the use of RAM. In reading, memory allocation exerted 84.18% of influence, followed by 14.34% of the interaction between CPU allocation and

RAM on the latency.

The influence of CPU and RAM factors on the performance of the Cassandra application differed significantly between the two testbeds. This difference was primarily due to the high latency experienced by the slice deployed on the Fabric testbed. Although it launches network slices faster, our network slice on the Fabric testbed experienced higher latencies than the FIBER-NG testbed. This latency led to increased input–output usage and buffering in the application, which, in turn, required more RAM allocation. In contrast, the network slice deployed on FIBRE-NG experienced lower latency, making the impact of the CPU on the application response time.

5. Concluding Remarks

This study proposes an analysis of the influence of resource allocation on network slice behavior in distributed testbeds. We followed the partial factorial model, where we combined different resource allocations for network slicing and observed the response variable (latency) for the Write and Read operations. Looking at related works, we concluded that there was an opportunity to contribute from this perspective.

After analyzing the impact of resource allocation (CPU and RAM), we concluded that although there are time differences in the deployment of the network slice in the testbeds (FIBRE-NG and Fabric), this time does not directly influence the operation of the network slice. In addition, we found that the influence of resource allocation depends on the seasonal demand of the network slice; therefore, smart life-cycle orchestration still has opportunities.

One limitation of this study is that it focused its analysis on only two testbeds, making it difficult to generalize these results to other testbeds. In future work, we will evaluate the influence of allocating other types of resources and applying computational intelligence techniques for auto-scaling network slice resources.

Acknowledgments

We acknowledge the financial support of the FAPESP MCTIC/CGI Research project 2018/23097-3 - SFI2 - Slicing Future Internet Infrastructures. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. We also thank the National Council for Scientific and Technological Development (CNPq) under grant number 421944/2021-8 (call CNPq/MCTI/FNDCT 18/2021) and Centro ALGORITMI, funded by Fundação para a Ciência e Tecnologia (FCT) within the RD Units Project Scope 2020-2023 (UIDB/00319/2020) for partially support this work.

References

Arora, S., Ksentini, A., and Bonnet, C. (2024). Cloud native lightweight slice orchestration (cliso) framework. *Computer Communications*, 213:1–12.

Baldin, I., Nikolich, A., Griffioen, J., Monga, I. I. S., Wang, K.-C., Lehman, T., and Ruth, P. (2019). Fabric: A national-scale programmable experimental network infrastructure. *IEEE Internet Computing*, 23(6):38–47.

Debbabi, F., Jmal, R., Fourati, L. C., and Ksentini, A. (2020). Algorithmics and modeling aspects of network slicing in 5g and beyonds network: Survey. *IEEE Access*, 8:162748–162762.

Donatti, A., Correa, S. L., Martins, J. S. B., Abelem, A., Both, C. B., Silva, F., Suruagy, J. A., Pasquini, R., Moreira, R., Cardoso, K. V., and Carvalho, T. C. (2023). Survey on machine learning-enabled network slicing: Covering the entire life cycle. *IEEE Transactions on Network and Service Management*, pages 1–1.

Dong, W., Li, B., Li, H., Wu, H., Gong, K., Zhang, W., and Gao, Y. (2023). LinkLab 2.0: A multi-tenant programmable IoT testbed for experimentation with Edge-Cloud integration. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, pages 1683–1699, Boston, MA. USENIX Association.

Feng, J., Pei, Q., Yu, F. R., Chu, X., Du, J., and Zhu, L. (2020). Dynamic network slicing and resource allocation in mobile edge computing systems. *IEEE Transactions on Vehicular Technology*, 69(7):7863–7878.

Karbalaee Motalleb, M., Shah-Mansouri, V., Parsaeefard, S., and Alcaraz López, O. L. (2023). Resource allocation in an open ran system using network slicing. *IEEE Transactions on Network and Service Management*, 20(1):471–485.

Moreira, R., Rosa, P. F., Aguiar, R. L. A., and de Oliveira Silva, F. (2021). NASOR: A network slicing approach for multiple Autonomous Systems. *Computer Communications*, 179:131–144.

Morel, A. E., Calyam, P., Qu, C., Gafurov, D., Wang, C., Thareja, K., Mandal, A., Lyons, E., Zink, M., Papadimitriou, G., and Deelman, E. (2023). Network services management using programmable data planes for visual cloud computing. In *2023 International Conference on Computing, Networking and Communications (ICNC)*, pages 130–136.

Padalia, N. (2015). *Apache Cassandra Essentials*. Packt Publishing Ltd.

S., S., Mishra, S., and Hota, C. (2023). Joint QoS and energy-efficient resource allocation and scheduling in 5G Network Slicing. *Computer Communications*, 202:110–123.

Salmito, T., Ciuffo, L., Machado, I., Salvador, M., Stanton, M., Rodriguez, N., Abelem, A., Bergesio, L., Sallent, S., and Baron, L. (2014). Fibre-an international testbed for future internet experimentation. In *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos-SBRC 2014*, pages p–969.