Container Migration Performance Evaluation: An Approach Based on Stochastic Petri Nets

Leonel Feitosa[1], Paulo Rego[2] [Co-advisor], Francisco Airton Silva[1] [Advisor]

[1] Universidade Federal do Piauí – PI – Brasil

[2]Universidade Federal do Ceará – CE – Brasil

{leonelfeitosa,faps}@ufpi.edu.br, paulo@dc.ufc.br

Abstract. *Containerization has emerged as a transformative technology in modern data centers, enabling efficient resource management and improving operational flexibility across various applications. While often associated with platforms like Docker and Kubernetes, container-based solutions are widely integrated into cloud environments such as AWS, Microsoft Azure, and Google Cloud. In large-scale distributed systems, efficient container migration is crucial to manage server downtime, consolidate resources, and ensure reliability in mobile edge computing scenarios. The problem with evaluating container migration performance lies in the high cost and computational demand of real-world experiments. Assessing different migration strategies efficiently remains a challenge, particularly for stateful containers, which require structured modeling approaches to quantify their impact on system performance. The proposal of this study is to develop Stochastic Petri Net (SPN) models to assess container migration strategies. The approach includes two models—one incorporating an absorbing state and another without—analyzing key migration techniques: Cold, PreCopy, PostCopy, and Hybrid. These models evaluate critical performance metrics, including Migration Total Time (MTT), Mean Migration Time (MMT), utilization, discard probability, and migration rate. Furthermore, a sensitivity analysis based on the Design of Experiments (DoE) was conducted for the Hybrid migration strategy to identify key performance factors. The conclusion of this research is that by providing an analytical framework for container migration evaluation, it enhances the understanding of migration performance dynamics and supports decision-making in cloud and edge computing infrastructures.*

1. Introduction

Containerization is a widely adopted method of operating system virtualization that isolates applications within lightweight, portable environments. While container technology has existed since 2008 [Turnbull 2014], its rapid adoption in cloud computing and distributed systems has been driven by the flexibility and scalability it offers. Major cloud providers such as AWS, Microsoft Azure, and Google Cloud have integrated containerized workloads into their platforms [Statista 2023], enabling efficient resource utilization and deployment automation. Large-scale virtualized computing platforms, particularly those in cloud and edge computing environments, require efficient mechanisms for resource migration [Varasteh and Goudarzi 2015]. Container migration is important for ensuring service continuity, load balancing, and disaster recovery, particularly in scenarios where proximity to end users affects performance, such as mobile edge computing [Conforti et al. 2021, Junior et al. 2020]. Kubernetes [Authors 2021] and OpenStack [Vaughan-Nichols 2022] are two widely used platforms that provide container orchestration and migration capabilities to optimize resource allocation dynamically.

Despite its advantages, container migration presents several challenges, mainly when dealing with stateful applications. Stateless containers can be easily relocated across nodes, but migrating stateful containers requires preserving and synchronizing their states, which increases the complexity of the process [Kotikalapudi 2017]. Checkpoint/Restore in Userspace (CRIU) is a key tool that facilitates container migration by allowing the freezing of running applications and saving their states, enabling seamless restoration on different nodes [Torre et al. 2019]. CRIU supports multiple migration policies, including Cold, PreCopy, PostCopy, and HybridCopy, each with distinct trade-offs in terms of service downtime and migration efficiency [Pickartz et al. 2016]. SPNs provide a robust modeling approach for evaluating container migration performance. SPNs extend classical Petri nets by incorporating stochastic timing, making them suitable for analyzing systems with probabilistic behaviors [Malhotra and Trivedi 1995]. These models allow for a detailed assessment of migration-related performance metrics, such as response time, throughput, and resource utilization. By integrating randomness with structured system representation, SPNs facilitate performance evaluation in scenarios like mobile edge computing (MEC) [Carvalho et al. 2020] and real-time data processing [Requeno et al. 2017]. They also enable predictive analysis of system behavior under varying conditions, reducing the need for expensive real-world experimentation [Silva et al. 2021].

Given these challenges, this study aims to provide a structured performance evaluation framework for container migration strategies. The proposed methodology includes the development of SPN models to assess migration performance, allowing for a detailed analysis of key performance indicators. Additionally, a Design of Experiments (DoE)-based sensitivity analysis is employed to identify the most influential factors affecting migration efficiency. Through empirical validation, the study contributes to a better understanding of how migration strategies can be optimized in cloud and edge environments, ultimately supporting improved decision-making in distributed computing infrastructures.

2. Research Questions

The objective of this research is to evaluate container migration strategies by modeling and analyzing their performance under different scenarios. To achieve this goal, the study addresses the following research questions (RQ):

- **RQ1:** How do different container migration policies (Cold, PreCopy, PostCopy, and Hybrid) impact key performance metrics such as Migration Total Time (MTT), Mean Migration Time (MMT), utilization, discard probability, and migration rate?
- **RQ2:** How do the number of migrating containers and the system's parallel migration capacity affect overall migration performance?
- **RQ3:** How well do the proposed SPN models predict real-world container migration performance?

- **RQ4:** What is the influence of different parameter variations on container migration efficiency, as assessed through a Design of Experiments (DoE)-based sensitivity analysis?

Particularly, this study employs SPN modeling to assess the performance of different migration strategies. Additionally, a DoE approach is used to analyze the impact of key factors on migration efficiency. By addressing these research questions, the study provides insights into container migration dynamics and offers strategies for optimizing resource management in distributed computing environments.

3. Publications and Contributions

The dissertation was structured as a series of published papers (three papers in total), grouped according to their contributions. Figure 1 presents the research questions, the objectives of the studies, and the corresponding publications. Additionally, it includes publication metrics evaluated using the Qualis Capes platform [1].

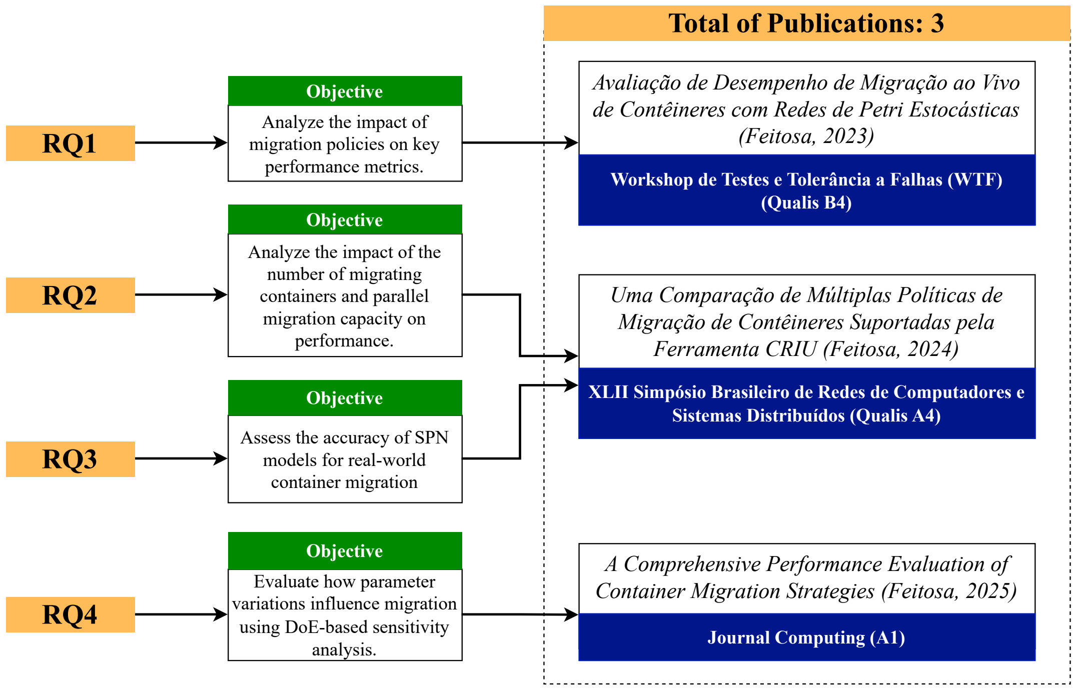

Figure 1. The research questions, highlighting the features employed and the published papers during the research

3.1. Performance Evaluation of Live Container Migration Using Stochastic Petri Nets

The first paper, titled *"Avaliação de Desempenho de Migração ao Vivo de Contêineres com Redes de Petri Estocásticas"* [Feitosa et al. 2023], introduces a SPN model designed to evaluate the performance of various container migration strategies. The main contributions of this research are as follows:

[1] https://qualis.capes.gov.br/

- **Development of an analytical SPN model** to represent and assess different container migration policies, including Cold, PreCopy, PostCopy, and Hybrid.
- **Quantification of migration performance metrics**, such as Migration Total Time (MTT), probability of completion within a specific time window, and system utilization.
- **Evaluation of parallel migration impact**, analyzing how the number of migrating containers and system migration capacity affect overall performance.
- **Comparison between modeled results and empirical measurements**, demonstrating the accuracy of the proposed model in predicting real-world migration behaviors.

3.2. Comparison of Multiple Container Migration Policies Using CRIU

The second paper, titled *"Uma Comparação de Múltiplas Políticas de Migração de Contêineres Suportadas pela Ferramenta CRIU"* [Feitosa et al. 2024], presents a comparative analysis of container migration policies supported by the CRIU tool. The main contributions of this research are as follows:

- **Proposal of two stochastic Petri Net models**, with and without absorbing state, to evaluate container migration performance.
- **Analysis of migration performance metrics**, including Migration Total Time (MTT), Mean Migration Time (MMT), discard probability, and system utilization.
- **Evaluation of cumulative distribution function (CDF)**, allowing the estimation of migration completion probability within a given time window.
- **Assessment of different migration policies**, including Cold, PreCopy, PostCopy, and Hybrid, in terms of their impact on migration efficiency.

3.3. A Comprehensive Performance Evaluation of Container Migration Strategies

The third paper, titled *"A Comprehensive Performance Evaluation of Container Migration Strategies"* [Feitosa et al. 2025], represents the culmination of the research by integrating all contributions from previous works and expanding the analysis. The main contributions of this study are as follows:

- **Development of an advanced SPN model** incorporating both absorbing and non-absorbing states, enabling a broader analysis of migration policies under different operational conditions.
- **Comprehensive performance evaluation of migration policies** (Cold, PreCopy, PostCopy, and Hybrid) using metrics such as MTT, MMT, utilization, discard probability, and migration rate.
- **Implementation of a DoE-based sensitivity analysis** to determine the impact of key system parameters on migration efficiency.
- **Empirical validation** of the proposed models through real-world experiments, comparing analytical predictions with observed results.
- **Discussion on the scalability and optimization of container migration** in dynamic environments, offering insights into improving migration strategies in distributed computing infrastructures.

4. Related Work

This section categorizes related studies based on migration policies, which significantly impact migration time, discard rates, and downtime. Table 1 summarizes the contributions.

PreCopy migration is widely studied for minimizing downtime and improving predictability. Xu et al. [Xu et al. 2020] and Benjaponpitak et al. [Benjaponpitak et al. 2020] introduced Sledge and CloudHopper, respectively, automating live container migration and managing component integrity. Fan et al. [Fan et al. 2019] optimized local PreCopy migration for Docker, while Smimite et al. [Smimite and Afdel 2019] explored hybrid virtualization for monolithic applications. Al-Najjar et al. [Al-Dhuraibi et al. 2017] developed ELASTICDOCKER for autonomous resource-based live migration, Bhardwaj et al. [Bhardwaj and Rama Krishna 2022] compared LXD/CR and PreCopy VM migration, and Ramanathan et al. [Ramanathan et al. 2021b] examined live migration for NFV. Studies on multiple migration policies include Stoyanov et al. [Stoyanov and Kollingbaum 2018] and Govindaraj et al. [Govindaraj and Artemenko 2018], who enhanced CRIU-based migrations to reduce downtime. Chou et al. [Chou et al. 2019] optimized checkpoint-based migration, while Ramanathan et al. [Ramanathan et al. 2021a] compared VM and container migrations, and Pecholt et al. [Pecholt et al. 2021] ensured secure live migrations with encrypted VMs.

Several studies do not specify migration policies but focus on efficiency. Ma et al. [Ma et al. 2018] introduced a layered storage structure for mobile service migration, and Di et al. [Di et al. 2021] optimized multi-container migration. Tay et al. [Tay et al. 2017] analyzed workload placement in data centers, while Gonzalez et al. [González and Arzuaga 2020] developed Herd-Monitor for performance tracking, and Abdullah et al. [Abdullah et al. 2022] proposed a migration management algorithm based on resource constraints. Cold migration, though less explored, reduces availability impact by migrating containers offline. Torre et al. [Torre et al. 2019] analyzed migration performance across various conditions, while Karhula et al. [Karhula et al. 2019] leveraged Docker and CRIU for checkpointing long-running IoT functions, improving resource efficiency.

5. Discussion

This section discusses the challenges and the significance of the results obtained in evaluating container migration strategies.

5.1. Challenges

Several challenges arise when designing and implementing effective container migration strategies. One major concern is **performance trade-offs** among migration policies. While Cold Migration minimizes resource consumption, it results in extended downtime. In contrast, PostCopy minimizes downtime but increases the probability of discard under network instability. These trade-offs make it difficult to define a universally optimal strategy.

Another critical challenge is **scalability and parallel migration**. As the number of migrating containers increases, system resources become a bottleneck, affecting

Table 1. Related work.

Work	Policies	Metrics	Use of CRIU	Sensitivity Analysis	Evaluation Method
[Torre et al. 2019]	Cold	MRT	✓	No	Measurement
[Stoyanov and Kollingbaum 2018]	PreCopy, PostCopy, Hybrid	MRT	✓	No	Measurement
[Ma et al. 2018]	Non-explicit	MRT, Network Bandwidth, Network Latency	✓	No	Measurement
[Di et al. 2021]	Non-explicit	MRT	✓	No	Measurement
[Kotikalapudi 2017]	Non-explicit	MRT, Downtime, Usage	✓	No	Measurement
[Tay et al. 2017]	Non-explicit	MRT	No	No	Measurement and Modeling
[Xu et al. 2020]	PreCopy	MRT, Downtime, Image Extraction Time	✓	No	Measurement
[González and Arzuaga 2020]	Non-explicit	Usage	✓	No	Measurement
[Govindaraj and Artemenko 2018]	PreCopy, PostCopy, Hybrid	MRT, Downtime	✓	No	Measurement
[Benjaponpitak et al. 2020]	PreCopy	MRT, Throughput	✓	No	Measurement
[Abdullah et al. 2022]	Non-explicit	Usage, Response Time	✓	No	Measurement
[Ramanathan et al. 2021a]	Cold, PreCopy	MRT, Downtime	✓	No	Measurement
[Fan et al. 2019]	PreCopy	MRT, Usage	No	No	Measurement
[Machen et al. 2017]	Non-explicit	MRT, Downtime	No	No	Measurement
[Pecholt et al. 2021]	Cold, PreCopy, PostCopy	MRT, Downtime, Confidentiality, Integrity	No	No	Measurement
[Smimite and Afdel 2019]	PreCopy	MRT, Memory Consumption, Network Traffic	✓	No	Measurement
[Al-Dhuraibi et al. 2017]	PreCopy	Usage, Response Time	✓	No	Measurement
[Chou et al. 2019]	PreCopy, PostCopy	MRT, Downtime, Aging	✓	No	Measurement
[Das and Sidhanta 2023]	Non-explicit	Throughput, Latency	✓	No	Measurement
[Majeed et al. 2020]	Non-explicit	R2, MAPE, MAE	✓	No	Modeling
[Karhula et al. 2019]	Cold	Usage	✓	No	Measurement
[Bhardwaj and Rama Krishna 2022]	PreCopy	MRT, Downtime, Usage	✓	No	Measurement
[Ramanathan et al. 2021b]	PreCopy	MRT, Downtime	✓	No	Measurement
[Kakakhel et al. 2018]	Non-explicit	MRT, Downtime	✓	No	Measurement
[Baccarelli et al. 2018]	Cold, PreCopy, PostCopy	Migration Rate, Energy	✓	No	Measurement
This work	Cold, PreCopy, PostCopy, Hybrid	MRT, MMT, Discard Probability, Utilization, Migration Rate	✓	✓	Measurement and Modeling

migration total time (MTT) and overall system performance. While increasing parallel migration capacity can mitigate these effects, efficient scheduling and allocation mechanisms are required. Additionally, **system heterogeneity** poses an obstacle, as migration efficiency depends on the underlying infrastructure, such as network bandwidth, CPU availability, and memory constraints. In cloud and edge computing environments, dynamic workloads further complicate migration decisions, necessitating adaptive policies.

5.2. The Application of Stochastic Petri Net Models

To address these challenges, this study employed **SPN models** to analyze the performance of Cold, PreCopy, PostCopy, and Hybrid migration strategies. The models provided a structured approach to assess migration metrics, including **MTT, mean migration time (MMT), discard probability, and utilization rates**.

- The **absorbing SPN model** was used to estimate the probability distribution of migration completion within a given time window.

- The **non-absorbing model** allowed the evaluation of migration rates under continuous arrivals, simulating real-world workload conditions.
- A **Design of Experiments (DoE) sensitivity analysis** identified the factors most influencing migration performance, particularly the number of containers being migrated and parallel migration capacity.

These methodologies enabled **a predictive approach** to optimize migration performance without relying solely on costly real-world experimentation.

5.3. Importance of the Results

The findings of this study provide significant insights into optimizing container migration in large-scale computing environments:

- **Improved decision-making for migration policies:** The results indicate that Cold Migration is preferable for **high-load environments** due to its lower MTT. At the same time, PostCopy is more suitable for scenarios requiring **low discard probability**.
- **Scalability optimization:** By analyzing parallel migration capabilities, the study highlights the optimal number of containers that can be migrated concurrently before reaching diminishing returns.
- **Performance prediction for real-world scenarios:** The integration of **SPN models with empirical validation** ensures that migration performance can be predicted with high accuracy, aiding in resource planning for cloud and edge computing platforms.

6. Conclusion

This dissertation presented a comprehensive analysis of container migration strategies, focusing on performance modeling using SPNs and empirical validation. By addressing key challenges in live container migration, this research contributes to a deeper understanding of migration policies and their impact on service availability, system efficiency, and resource utilization in cloud and edge computing environments. The study investigated four main research questions (RQ1–RQ4), each contributing to the broader goal of optimizing container migration strategies. First, we examined the impact of different migration policies—Cold, PreCopy, PostCopy, and Hybrid—on critical performance metrics, demonstrating the trade-offs between migration total time, downtime, discard probability, and resource utilization. Second, we analyzed how parallel migration capacity influences migration efficiency, highlighting scalability limits and optimal configurations. Third, the accuracy of the proposed SPN models was validated through empirical experiments, confirming their predictive capabilities in real-world scenarios. Finally, we conducted a DoE-based sensitivity analysis, identifying the most influential factors affecting migration performance. The findings of this dissertation contribute to container migration research. The proposed SPN models provide a structured approach for evaluating migration efficiency, offering a predictive framework that aids system administrators and researchers in decision-making. The integration of theoretical modeling with empirical validation ensures that the results are not only analytically sound but also applicable to real-world deployments.

References

Abdullah, D. B., Hadeed, W., et al. (2022). Container live migration in edge computing: a real-time performance amelioration. *International Journal of Applied Science and Engineering*, 19(3):1–8.

Al-Dhuraibi, Y., Paraiso, F., Djarallah, N., and Merle, P. (2017). Autonomic vertical elasticity of docker containers with elasticdocker. In *2017 IEEE 10th international conference on cloud computing (CLOUD)*, pages 472–479. IEEE.

Authors, T. K. (2021). *Kubernetes Documentation*. Acessado em: 2 de novembro de 2024.

Baccarelli, E., Scarpiniti, M., and Momenzadeh, A. (2018). Fog-supported delay-constrained energy-saving live migration of vms over multipath tcp/ip 5g connections. *IEEE Access*, 6:42327–42354.

Benjaponpitak, T., Karakate, M., and Sripanidkulchai, K. (2020). Enabling live migration of containerized applications across clouds. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 2529–2538. IEEE.

Bhardwaj, A. and Rama Krishna, C. (2022). A container-based technique to improve virtual machine migration in cloud computing. *IETE Journal of Research*, 68(1):401–416.

Carvalho, D., Rodrigues, L., Endo, P. T., Kosta, S., and Silva, F. A. (2020). Mobile edge computing performance evaluation using stochastic petri nets. In *2020 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6. IEEE.

Chou, C. C., Chen, Y., Milojicic, D., Reddy, N., and Gratz, P. (2019). Optimizing post-copy live migration with system-level checkpoint using fabric-attached memory. In *2019 IEEE/ACM Workshop on Memory Centric High Performance Computing (MCHPC)*, pages 16–24. IEEE.

Conforti, L., Virdis, A., Puliafito, C., and Mingozzi, E. (2021). Extending the quic protocol to support live container migration at the edge. In *2021 IEEE 22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 61–70. IEEE.

Das, R. and Sidhanta, S. (2023). Live migration of containers in the edge. *SN Computer Science*, 4(5):479.

Di, Z., Shao, E., and Tan, G. (2021). High-performance migration tool for live container in a workflow. *International Journal of Parallel Programming*, 49:658–670.

Fan, W., Han, Z., Li, P., Zhou, J., Fan, J., and Wang, R. (2019). A live migration algorithm for containers based on resource locality. *Journal of Signal Processing Systems*, 91:1077–1089.

Feitosa, L., Barbosa, V., Sabino, A., Lima, L. N., Fé, I., Silva, B., and Silva, F. A. (2024). Uma comparação de múltiplas políticas de migração de contêineres suportadas pela ferramenta criu. In *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, pages 742–755. SBC.

Feitosa, L., Barbosa, V., Sabino, A., Lima, L. N., Fé, I., Silva, L. G., Callou, G., Carvalho, J., Leão, E., Nguyen, T. A., et al. (2025). A comprehensive performance evaluation of container migration strategies. *Computing*, 107(2):1–39.

Feitosa, L., Rego, P. A., and Silva, F. A. (2023). Avaliaçao de desempenho de migraçao ao vivo de contêineres com redes de petri estocásticas. In *Workshop de Testes e Tolerância a Falhas (WTF)*, pages 94–107. SBC.

González, A. E. and Arzuaga, E. (2020). Herdmonitor: monitoring live migrating containers in cloud environments. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 2180–2189. IEEE.

Govindaraj, K. and Artemenko, A. (2018). Container live migration for latency critical industrial applications on edge computing. In *2018 IEEE 23rd international conference on emerging technologies and factory automation (ETFA)*, volume 1, pages 83–90. IEEE.

Junior, P. S., Miorandi, D., and Pierre, G. (2020). Stateful container migration in geo-distributed environments. In *2020 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 49–56. IEEE.

Kakakhel, S. R. U., Mukkala, L., Westerlund, T., and Plosila, J. (2018). Virtualization at the network edge: A technology perspective. In *2018 Third International Conference on Fog and Mobile Edge Computing (FMEC)*, pages 87–92. IEEE.

Karhula, P., Janak, J., and Schulzrinne, H. (2019). Checkpointing and migration of iot edge functions. In *Proceedings of the 2nd International Workshop on Edge Systems, Analytics and Networking*, pages 60–65.

Kotikalapudi, S. V. N. (2017). Comparing live migration between linux containers and kernel virtual machine: investigation study in terms of parameters.

Ma, L., Yi, S., Carter, N., and Li, Q. (2018). Efficient live migration of edge services leveraging container layered storage. *IEEE Transactions on Mobile Computing*, 18(9):2020–2033.

Machen, A., Wang, S., Leung, K. K., Ko, B. J., and Salonidis, T. (2017). Live service migration in mobile edge clouds. *IEEE Wireless Communications*, 25(1):140–147.

Majeed, A. A., Kilpatrick, P., Spence, I., and Varghese, B. (2020). Modelling fog offloading performance. In *2020 IEEE 4th International Conference on Fog and Edge Computing (ICFEC)*, pages 29–38. IEEE.

Malhotra, M. and Trivedi, K. S. (1995). Dependability modeling using petri-nets. *IEEE Transactions on reliability*, 44(3):428–440.

Pecholt, J., Huber, M., and Wessel, S. (2021). Live migration of operating system containers in encrypted virtual machines. In *Proceedings of the 2021 on Cloud Computing Security Workshop*, pages 125–137.

Pickartz, S., Eiling, N., Lankes, S., Razik, L., and Monti, A. (2016). Migrating linux containers using criu. In *High Performance Computing: ISC High Performance 2016 International Workshops, ExaComm, E-MuCoCoS, HPC-IODC, IXPUG, IWOPH, P^ 3MA, VHPC, WOPSSS, Frankfurt, Germany, June 19–23, 2016, Revised Selected Papers 31*, pages 674–684. Springer.

Ramanathan, S., Kondepu, K., Razo, M., Tacca, M., Valcarenghi, L., and Fumagalli, A. (2021a). Live migration of virtual machine and container based mobile core network components: A comprehensive study. *IEEE Access*, 9:105082–105100.

Ramanathan, S., Kondepu, K., Zhang, T., Mirkhanzadeh, B., Razo, M., Tacca, M., Valcarenghi, L., and Fumagalli, A. (2021b). A comprehensive study of virtual machine and container based core network components migration in openroadm sdn-enabled network. *arXiv preprint arXiv:2108.12509*.

Requeno, J.-I., Merseguer, J., and Bernardi, S. (2017). Performance analysis of apache storm applications using stochastic petri nets. In *2017 IEEE International Conference on Information Reuse and Integration (IRI)*, pages 411–418. IEEE.

Silva, F. A., Fé, I., and Gonçalves, G. (2021). Stochastic models for performance and cost analysis of a hybrid cloud and fog architecture. *The Journal of Supercomputing*, 77(2):1537–1561.

Smimite, O. and Afdel, K. (2019). Impact of hybrid virtualization using vm and container on live migration and cloud performance. In *Lecture Notes in Real-Time Intelligent Systems*, pages 196–208. Springer.

Statista (2023). Container technology - statistics and facts.

Stoyanov, R. and Kollingbaum, M. J. (2018). Efficient live migration of linux containers. In *High Performance Computing: ISC High Performance 2018 International Workshops, Frankfurt/Main, Germany, June 28, 2018, Revised Selected Papers 33*, pages 184–193. Springer.

Tay, Y., Gaurav, K., and Karkun, P. (2017). A performance comparison of containers and virtual machines in workload migration context. In *2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pages 61–66. IEEE.

Torre, R., Urbano, E., Salah, H., Nguyen, G. T., and Fitzek, F. H. (2019). Towards a better understanding of live migration performance with docker containers. In *European Wireless 2019; 25th European Wireless Conference*, pages 1–6. VDE.

Turnbull, J. (2014). *The Docker Book: Containerization is the new virtualization*. James Turnbull, San Francisco, CA.

Varasteh, A. and Goudarzi, M. (2015). Server consolidation techniques in virtualized data centers: A survey. *IEEE Systems Journal*, 11(2):772–783.

Vaughan-Nichols, S. (2022). *Migrating from VMware to OpenStack: Optimizing your Infrastructure to Save Money and Avoid Vendor-Lock-in*. Acessado em: 2 de novembro de 2024.

Xu, B., Wu, S., Xiao, J., Jin, H., Zhang, Y., Shi, G., Lin, T., Rao, J., Yi, L., and Jiang, J. (2020). Sledge: Towards efficient live migration of docker containers. In *2020 IEEE 13th International Conference on Cloud Computing (CLOUD)*, pages 321–328. IEEE.