Evaluation Model and Performance Analysis of NIC Aggregations in Containerized Private Clouds

Anderson M. Maliszewski*‡, Dalvan Griebler‡§, Eduardo Roloff*, Rodrigo da Rosa Righi¶, Philippe O. A. Navaux*
*Informatics Institute, Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, Brazil
‡Laboratory of Advanced Research on Cloud Computing (LARCC), Três de Maio Faculty (SETREM), Três de Maio, Brazil
§School of Technology, Pontifical Catholic University of Rio Grande do Sul (PUCRS), Porto Alegre, Brazil
¶Software Innovation Lab (SoftwareLab), University of Vale do Rio dos Sinos (UNISINOS), São Leopoldo, Brazi
Email: {ammaliszewski,eroloff,navaux}@inf.ufrgs.br*,
dalvan.griebler@pucrs.br, §, rrrighi@unisinos.br¶

***Abstract*—The availability of computational resources changed significantly due to cloud computing. In addition, we have witnessed efforts to execute High-Performance Computing (HPC) applications in the cloud attracted by the advantages of cost savings and scalable/elastic resource allocation. Allocating more powerful hardware and exclusivity allocating resources such as memory, storage, and CPU can improve performance in the cloud. For network interconnection, significant noise, and other inferences are generated by several simultaneous instances (multi-tenants) communicating using the same network. As increasing the network bandwidth may be an alternative, we designed an evaluation model, and performance analysis of NIC aggregation approaches in containerized private clouds. The experiments using NAS Parallel Benchmarks revealed that NIC aggregation approach outperforms the baseline up to ≈98% of the executions with applications characterized by intensive network use. Also, the Balance Round-Robin aggregation mode performed better than the 802.3ad aggregation mode in most assessments.**

***Index Terms*—Cloud Computing; Private Cloud; High Performance Computing; Network Performance; NIC Aggregation; Performance Analysis**

I. Introduction

With an increase in complexity and the number of computational problems as well as in the acquisition value of datacenter infrastructures, there was a significant migration from these traditional environments to those that provide resources in a fast, scalable, and pay-for-use manner, such as cloud computing [1]. Cloud environments are built on several technologies (e.g., virtualization) and concepts from distributed to parallel computing. Nowadays, it is a model capable of providing on-demand computing resources (e.g., CPU, GPU, memory, storage, network) without upfront investments through three service layers, known as IaaS, PaaS, and SaaS [2].

High-performance computing, which clusters and cloud models supply, has historically been used to speed up data processing. The improvements in the technology stack and the possibility to provide an alternative to the usual computing methods, both in resource scalability and cost reduction, make cloud computing suitable for the service-oriented world we live in now. There are also HPC applications, which have intensive computing resource usage behavior. It has been shared by other research works that these kinds of applications executing in cloud environments typically came at the price of performance losses due to the negative impact of the virtualization (compared with the native environment) and the overhead of multi-tenants sharing for resources [3], [4].

Moreover, HPC applications executed in cloud computing environments are mostly developed using Message Passing Interface (MPI), and the communication characteristics of applications vary due to their purpose. Thus, network interconnection impacts the overall application performance. The computing environment must ensure some levels of high-performance communication (high throughput and low latency) to address the application's requirements and not become the system bottleneck. As previous research presented [5], [6], [7], [8], network interconnection is still a considerable challenge for HPC applications executing in clouds, in addition to other application domains [9], [10].

HPC applications aim to use as many as possible of the available resources. This condition is only achieved in theory with the guarantee of exclusivity of resources. Although the cloud can assure levels of priority allocation in memory, processor, and storage, it is not in the case of network interconnection since switching equipment is inevitably shared among servers. In this study, we used NPB applications to represent real-world HPC workloads and deployed an OpenNebula cloud that orchestrates LXD containers [11], [12]. This environment was configured over identical physical hosts and tested over different NIC (Network Interface Cards) aggregation approaches. This scenario is different from previous works and will present new performance insights. Besides using a private cloud setup, we also varied the number of aggregated NICs (up to four), the aggregation types (802.3ad and Balanced-RR (Round-Robin)), and considered up to three simultaneous LXD instances running specific applications to create noise in the network concerning the fourth and central LXD instances. In this way, we provide a representative environment to evaluate the type of interference that may occur.

Our investigations are motivated by the following research questions: *Can the NIC aggregation approach improve HPC applications' performance on the private cloud? Which is the aggregation mode that provides better performance?* In this

979-8-3503-8160-3/23/$31.00 ©2023 IEEE
DOI 10.1109/SBAC-PADW60351.2023.00025

vein, our goal is to create an evaluation model, that could be adopted by other research, to analyze performance impacts regarding network (not just the NIC aggregation tested in this work) configurations integrated to private cloud (not only containerized) as well as considering different scenarios with simultaneous users (multi-tenants) executing HPC applications or other network bounded applications. The contributions are summarized below:

- An evaluation model for performance assessment in private cloud environments.
- A container-based cloud deployment approach using NIC aggregation with the comparison of Balance Round-robin and IEEE 802.3ad modes.
- A representative performance analysis on different network configurations based on NIC aggregation approaches and with different number of noises and parallel VM instances.

The remainder of this paper is organized as follows: Section II describes the work background. Section III presents our evaluation model concerning hardware/software details. Section IV presents our performance analysis. Section V presents the related work. Section VI summarizes our conclusions and outlines ideas for future works.

II. Background

This section presents the basic concepts and technologies used for describing this work.

A. NIC Aggregation

Also known as Link Aggregation (LA) or Bonding, it is a technique that combines several NICs into a logical link. It is commonly used to interconnect pairs of network devices (i.e., switches, routers, etc.) to improve bandwidth and resilience in a cost-effective way by merely adding new links together with existing ones instead of replacing equipment [13], [14]. The specific behavior of connected interfaces is based on choosing a mode of use among seven existing modes. Another equally important benefit of NIC aggregation is to fail over transparently. This is preferred for deployments where high availability is critical. The same idea can be further extended to provide the combination of increased bandwidth and transparent failover with degraded performance in a NIC failure event.

B. Private Clouds

In private clouds, computational infrastructures are managed and owned by a private identity, for instance, a company or a research laboratory. Contrary to public clouds, private cloud environments are not fully adherent to the essential characteristics of CC defined by NIST. This happens because the number of resources, the elasticity, and the pay-per-use billing model are inconsistent. Also, this is the cloud model with a higher price involved since the organization needs to maintain and buy the computation infrastructure. On the other hand, it also provides more security to the organization and higher low-level management to perform specific upgrades or changes. These characteristics of private clouds make it the most suitable model to conduct our evaluation since we must have access to low-level configurations and also physical access to the servers (e.g., cables and switch modifications).

III. Evaluation Model

This section describes our evaluation model that consists of several hardware/software specifications alongside with the NIC aggregation approach, in which the aggregation modes Balanced RR and 802.3ad are covered. Also, there are the private cloud platform, benchmarks, experimental setup and execution model.

A. Hardware/Software Specifications

The computational environment that supported the experiments was composed of four HP ProLiant servers with identical hardware. Each has two six-core AMD Opteron processors 2425 HE, 32 GB of RAM, 4 Intel Gigabit network interface cards (NICs) interconnected by a Gigabit Switch. The software specification has Ubuntu Server 18.04 64-bit (kernel 4.15.0-99) as the operating system (OS), MPI Open MPI 2.1.1 library, GCC/GNU Fortran compiler version 7.5.0. Besides, OpenNebula cloud manager was used with version 5.10.1 and the Ethernet Channel Bonding Driver with version 3.7.1. All softwares involved in the evaluation process were used with their last stable available version. The LXD instances were created using the LXC version 3.0.3 and used the same OS, MPI wrapper, and GCC version as the physical servers.

B. NIC Aggregation

We used up to 4 NICs and up to 4 VMs with the IEEE 802.3ad Dynamic link aggregation and Balanced Round Robin modes. IEEE 802.3ad creates aggregation groups that share the same speed and duplex settings. The selection of the slave for outgoing traffic is made according to the transmission hash policy, which can be changed from the standard simple XOR policy using the xmit_hash_policy option. On the other hand, Balanced RR implements a Round-robin policy, transmitting all packets in sequence from the first node to the last, providing load balancing and fault tolerance.

C. Containerized Private Cloud

In this work, we deployed a private cloud using the OpenNebula cloud manager to create LXD clusters. It was chosen because of being one of the popular private cloud managers. Also, containers were used because of their representation as lightweight virtualization. We used four servers, each one with four NICs connected to the same switch. NICs are grouped into a logical link called bond0 and bridged to the containers. OpenNebula manages the containers and creates a cluster, establishing the communication over the underlying bonded NICs. We also evaluated with two network interface cards.

A representation of the containerized private cloud environment using NICs aggregated is depicted in Figure 1. It highlights the possibility of multiple instances using containers, the NIC aggregation, and the usage of several NICs in the form of computer infrastructure supporting potentially large-scale cloud environments. It is possible to view how instances of applications from our evaluation model are executed.

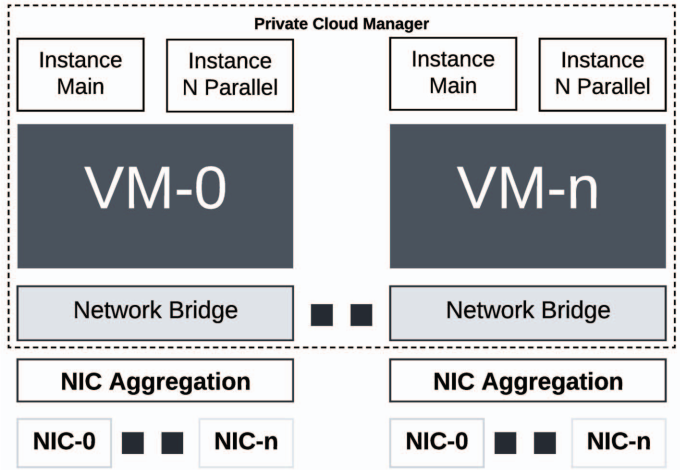

Fig. 1. Conceptual representation of the containerized private cloud environment. Adapted from [15].

D. Benchmarks

To represent real-work HPC application workload, we conduct our evaluation using the NAS Parallel Benchmarks (NPB) suite [16]. All NAS benchmarks were compiled with the workload size C with -O3 flag, mpifort, and mpicc for Fortran and C codes. We executed the applications with two variations in the number of processes. We choose 16 MPI processes (4 per instance) to evaluate only the network concurrency, because the instances have their hardware equally divided (parallel). We choose 64 MPI processes (16 per instance) to add one more factor of concurrency, resulting in more processes per instance than the physical server has.

E. Environment

The computational resources are equally divided between the instances (1 VM as a baseline, 2 VMs, and 4 VMs)[1]. For example, with four simultaneous LXD instances deployed, each one had 25% of the total computation resources available. Table I shows how we divide the executions into three environments regarding the network utilization by the applications used: High, Medium/High, and Low. These configurations were inspired by previous researches for choosing the application configuration [17], [18], [19], where the applications BT and SP presented medium/low network utilization and the application FT and IS presented high network utilization.

F. Experimental Setup

We employed a reproducible research methodology, using R programming language, GIT[2], and a laboratory notebook, making publicly available all the data of this work. Every time we ended the execution of the experiments for the baseline, each number of NICs aggregated (0, 2, and 4) and different aggregation modes (802.3ad and Balanced Round-Robin), we needed to restart the underlying servers. With the reboot process, we also ensured that there was no interference in the experiments related to various levels of cache (e.g., memory, processor instructions). The configurations and scenarios have 10 replications, and the reported execution times measurements are averages of the replications with the error bars calculated considering a confidence level of 95%, assuming a Gaussian distribution. We considered the results of the applications executed in the main instance. The parallel instances' results were discarded since they were just used to cause the noise in the network and consequently affect or not the main instance performance.

TABLE I
CONFIGURATIONS FOR THE NIC AGGREGATION EXPERIMENTS.

Computational resources divided between the two VMs				
Utilization	**Main**	**VMs**	**Apps**	**Result**
High	IS	2	IS	(IS) + Parallel (IS)
	FT	2	FT	(FT) + Parallel (FT)
	FT	2	IS	(FT) + Parallel (IS)
	IS	2	FT	(IS) + Parallel (FT)
Medium/High	BT	2	IS	(BT) + Parallel (IS)
	SP	2	IS	(SP) + Parallel (IS)
	BT	2	FT	(BT) + Parallel (FT)
	SP	2	FT	(SP) + Parallel (FT)
Low	BT	2	BT	(BT) + Parallel (BT)
	SP	2	SP	(SP) + Parallel (SP)
	BT	2	SP	(BT) + Parallel (SP)
	SP	2	BT	(SP) + Parallel (BT)
Computational resources divided between the four VMs				
Utilization	**Main**	**VMs**	**Apps**	**Result**
High	IS	4	IS	(IS) + Parallel (IS + IS + IS)
	FT	4	FT	(FT) + Parallel (FT + FT + FT)
	FT	4	IS	(FT) + Parallel (IS + IS + IS)
	IS	4	FT	(IS) + Parallel (FT +FT + FT)
Medium/High	BT	4	IS	(BT) + Parallel (IS + IS + IS)
	SP	4	IS	(SP) + Parallel (IS + IS + IS)
	BT	4	FT	(BT) + Parallel (FT + FT + FT)
	SP	4	FT	(SP) + Parallel (FT + FT + FT)
Low	BT	4	BT	(BT) + Parallel (BT + BT + BT)
	SP	4	SP	(SP) + Parallel (SP + SP + SP)
	BT	4	SP	(BT) + Parallel (SP + SP + SP)
	SP	4	BT	(SP) + Parallel (BT + BT + BT)

G. Execution Model

The designed execution model was implemented in Shell Script program for automating the execution of the experiments. It automatically start in all nodes (main and parallel ones). Figure 2 depicts this process using a flowchart, considering the colors gray for the processes in the main instance and dark gray for the process in the parallel instances. For instance, when we aim to evaluate the interference caused by three instances executing BT application, against the execution of BT in the main instance, our model first downloads and compiles the benchmarks in all instances. After, in the main instance, it reads the experiment project, which contains the applications' execution order, and creates an output file in an NFS folder. Next, in the parallel nodes, it calls a script that kills any executing applications, calls a script to read the output created by the main instance in the NFS folder and creates a confirmation file in the same folder to signal it is ready to

[1]In this work, when we mention VMs or instances, both are referred to as same virtualized machine hosted in the cloud.
[2]https://github.com/andermm/NEWP

cause noise. Then, it selects and executes an application in a loop. After confirming that the parallel instances are executing their applications, the application will execute on the main instance elected. Each time it executes in the main instance, finishes, and changes to another, a message is signalized using the NFS so that the execution loop in the parallel instances is killed. Finally, the program re-reads the experimental project and restart the previous steps with the correct applications.

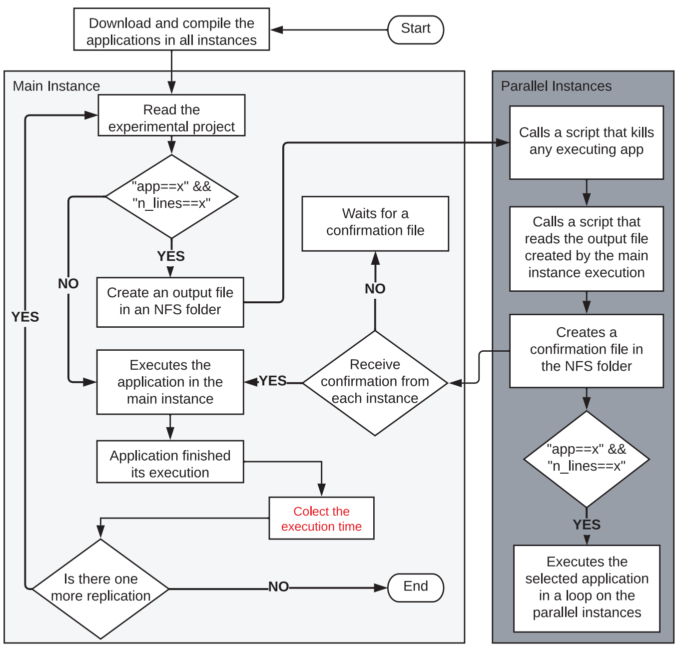

Fig. 2. Representations of the execution model.

IV. Performance Analysis

Below, we present the results of the experiment's methodology in bar graphs. In Y-axis, we present the aggregation mode performance gain. On the X-axis is the name of the applications group, considering its network utilization. The higher the percentage, the greater the gain. These results are an average of the four cases used by each group. For instance, in the high network utilization group, the results represent the average gain of the cases IS x IS, FT x FT, FT x IS, and IS x FT compared against the baseline.

We start analyzing RR aggregation mode against the baseline environment results with the applications combinations from the higher to the lowest network utilization executed with 16 and 64 processes. These results are plotted in Figure 3. As we can see, with 16 processes (left side graph), the higher gains happen with the execution of 2 VMs and 2 NICs aggregated, even with high, medium/high, and low network utilization, reaching up to ≈121% of gain against the baseline. Also, on average, the higher gains are in the high network utilization group. As expected, in the groups with less network utilization, the performance gains are smaller. The only slight loss (less than ≈1%) is seen in the low network utilization group with 4 VMs, and 4 NICs aggregated.

The performance gains are more singular in the executions with 64 processes (right side graph) than with 16 processes execution. For instance, the higher gains are up to ≈54% with 64 processes and ≈121% with 16 processes. In general, the results with 64 processes obtained lower gains than the results with 16 processes because of the concurrency, in which there is a bigger dispute of CPU. Again, the higher gains are in the high network utilization group, mostly with 1 VM and 4 NICs aggregated. Also, we can see that the executions with 4 aggregated NICs usually have better results than with 2 aggregated NICs. This previous affirmative is true in the high and medium/high network utilization groups in all execution and the low with one and two VMs. Compared to the baseline, the only loss in performance are in the low group with 4 VMs and 4 NICs. With both executions using 16 and 64 processes, we can see that the Round Robin aggregation mode can provide better results in most of the cases tested.

In Figure 4, we present the comparison of 802.3ad aggregation mode against the baseline environment with the applications combinations from the higher to the lowest network utilization executed with 16 and 64 processes. At a first look, we can see that different from the RR mode, 802.3ad mode can significantly gain and lose performance in specific results compared to the baseline. In the high network utilization group with 16 processes, five from six aggregation configurations performed better than baseline. The only aggregation that loss performance is with one VM and 4 NICs aggregated (≈-8.6% compared to baseline). With medium/high network utilization, five from six configurations performed better than baseline, with the only loss of performance in the executions with one VM and 4 NICs aggregated (≈-6.1% compared to the baseline). Only 2 from 6 aggregation configurations have outperformed the baseline in the last group with low network utilization. The outstanding results in this graph are with one VM and 4 NICs aggregated (≈-27.6%).

With 64 processes in the high network utilization group, all configurations performed better than baseline. It is also noticeable that the performance gain is higher as the number of VMs and NICs aggregated increases. In the Medium/High network utilization group, two from six configurations are outperformed by baseline, mostly with 1 VM and 4 NICs aggregated (≈-21.5%). In the last group, with low network utilization, three from six configurations performed worse than baseline, mostly seen with 1 VM and 4 NICs (≈-21.7%) and with 4 VMs and 4 NICs (≈-15.6%).What calls attention is that these three results happen in the aggregation with four NICs. As shown by the results of Figure 4, 802.3ad mode can provide better performance with applications that demand more network utilization. On the other hand, for other applications, the results may not be as expected.

In Figure 5, we depict the comparison between the aggregation modes Round Robin and 802.3ad. In the Y-axis, we have the RR performance gains, which means that in all positive percentages, RR outperformed 802.3ad. In the executions with 16 processes (left side graph), all configurations from the high network utilization group, RR obtained better results than

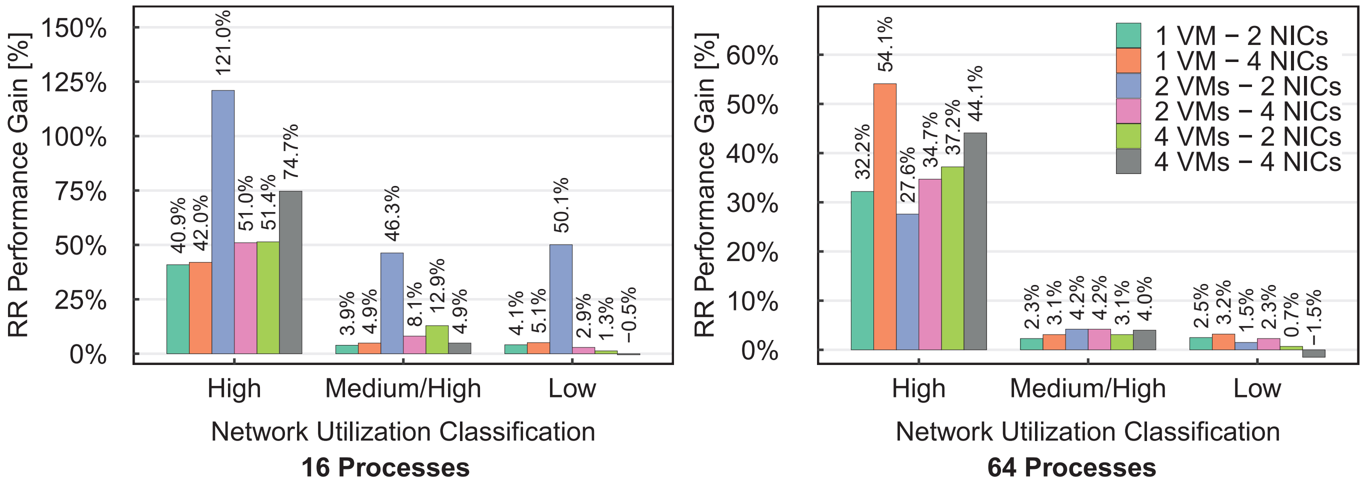

Fig. 3. Results from the comparison between RR aggregation mode and baseline without aggregation.

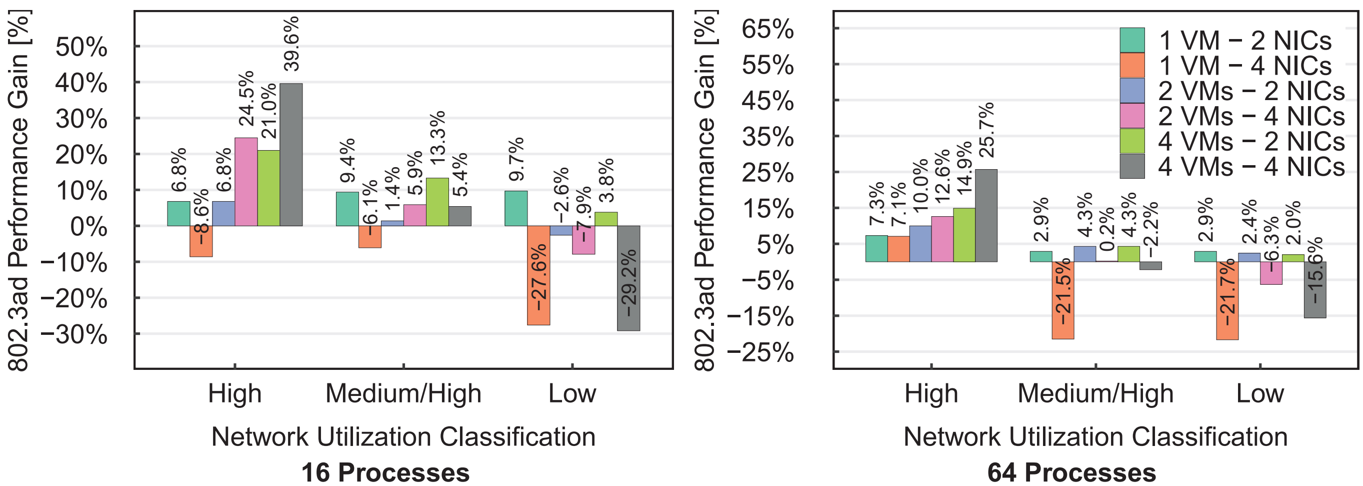

Fig. 4. Results from the comparison between 802.3ad aggregation mode and baseline without aggregation.

802.3ad, with outstanding results with 2 VMs and 2 NICs aggregated (≈106.9%) In the Medium/High network utilization group, RR outperformed 802.3ad in three of six configurations. As can be seen, performance losses are generally up to ≈4%, but the gains can reach more than ≈44%. In the low network utilization group, 802.3ad only outperformed RR in two of the six configurations.

With 64 processes executions, the outstanding results are the configuration with one VM and 4 NICs aggregated, which outperformed 802.3ad mode in all network utilization groups in more than ≈30% In the high network utilization group, again, all configurations have better results than 802.3ad mode. In the Medium/High network utilization group, three from six configurations outperformed RR mode by up to ≈1.2%. On the other hand, RR outperformed 802.3ad by up to ≈34%. In the low network utilization group, three from six configurations outperformed 802.3ad mode by up to ≈35%. As can be seen in both graphs, in the majority of the executions, the performance has obtained better results using RR aggregation mode than 802.3ad aggregation mode.

V. Related Work

In this section, we selected the state-of-the-art papers regarding evaluations in network performance for cloud environments. We considered as related work those that tackle network performance optimizations/evaluations in public/private clouds and clusters, using NIC aggregation approach. The selected related works are described below.With private clouds, Chakthranont et al. [20] integrated CloudStack with InfiniBand and conducted a performance evaluation in virtual and physical cluster using Intel MPI benchmarks, HPC Challenge, OpenMX, and Graph500.

Vogel et al. [21] conducted a network performance assessment using the CloudStack manager, deploying clouds based on KVM and LXC. They measured network throughput and latency and indicated alternatives for improvements in network performance using the vhost-net module. The results

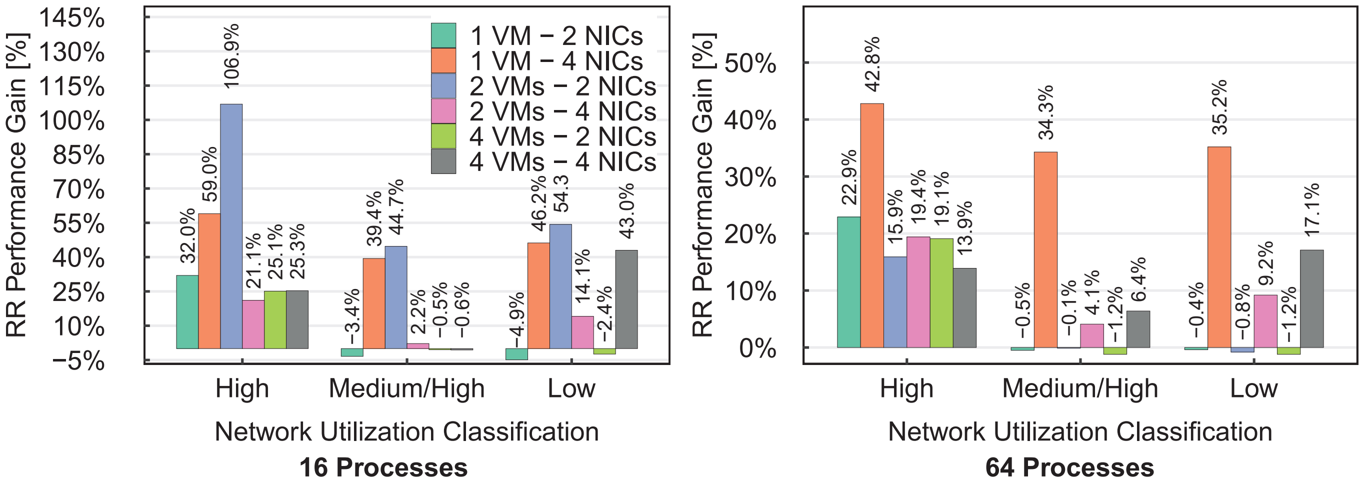

Fig. 5. Results from the comparison between RR aggregation mode and 802.3ad aggregation mode.

showed that the KVM achieves fair yield rates but performance degradation in latency. On the other hand, LXC performed better in latency, but lacked support and compatibility.

Other works have focused their investigations on network/link aggregation approaches. For instance, Watanabe et al. [22] investigated the impact of topology and link aggregation on a large-scale PC cluster with Ethernet. They performed several experiments with High-Performance LINPACK Benchmark (HPL) using 4-6 NICs aggregated using a torus topology. Their results have shown that the performance can be significantly improved in overall HPC applications up to 650%. This would allow cloud infrastructure using commodity hardware to improve network performance without significant additional investments in the hardware side.

Chaufournier et al. [23] created an assessment of the feasibility of using MPTCP (Multi Path TCP) to improve the performance of data center and cloud applications. Their results showed that while MPTCP provides useful bandwidth aggregation, congestion prevention, and improved resiliency for some cloud applications, these benefits do not apply uniformly across all. Similarly, Wang et al. [24] evaluated the applicability of MultiPath TCP (MPTCP) to improve the performance of the MapReduce application. Its scenario explored the capabilities of GPUs and showed the impact of network bottlenecks on applications' performance. As a result, it demonstrated that aggregation of network links reduced the data transfer time and improved the overall performance.

Rista et al. [15] created a methodology for evaluating performance measures such as bandwidth, throughput, latency, and execution times for Hadoop applications. The assessment also employed the Network Bonding 4 (IEEE 802.3ad) mode, but mainly explored the benefits that aggregation brings with up to 3 instances simultaneously in LXC containers. As a result, they achieved performance improvements by reducing application times of $\approx$33%. Although the results obtained are promising, the use of simultaneous instances, also known as multi-tenant, does not apply to HPC applications, as these require no competition for computational resources.

In contrast to the previous articles, our work focuses on creating an evaluation model applying to NIC aggregation, comparing network bonding IEEE 802.3ad and Balance Round-Robin modes to reduce the execution time of HPC applications. With our model, we could successfully emulate a multi-tenant cloud environment and apply different levels of interference that happens on intensive network usage environments. Also, this approach allow us to investigate how the aggregation modes react to the interference in the network. For instance, improving performance even when the network was under high usage. Finally, an important consideration of this section, was the lack in the state-of-art by recent works that considered and evaluated network aggregation methodologies. The advantage is that this approach is a cost-effective alternative to improve network performance in general.

VI. Conclusion

This article presented an evaluation model and an analysis of the performance of NIC aggregation configurations in a containerized private cloud. The general conclusion and main contribution are that network interconnection is crucial and can severely impact HPC applications' performance in the cloud. Using NIC aggregation, which is a low-cost approach, provides relevant performance improvements for HPC applications executing in containerized private cloud environment.

For applications with a high data dependency, which generates intensive communication among processes, the network reaches or even surpasses the same level of importance as computing power. Several efforts have already been made to develop new network technologies or specific approaches for HPC. Usually, such techniques are not widely accessible by the entire community because of their cost and complexity. The cloud offers a simplified environment with autoconfigured instances to overcome this limitation. It is recommended to perform a profile or characterization of the HPC applications to improve clouds' performance and cost-efficiency. Thus, it is possible to determine the requirements concerning the network and computational power and to be able to allocate the cloud environment correctly.

In this evaluation, our first question was, "*Can the NIC aggregation approach improve HPC applications performance on the cloud?*" With the obtained results, we argue that the NIC aggregation approach integrated into the containerized cloud improved the applications' performance concerning the high network usage scenario in most executions. For instance, RR and 802.3ad modes performed better than baseline in $\approx$98% of the executions. In the medium/high network usage scenario, RR and 802.3ad modes outperformed the baseline in $\approx$86% of the executions. Finally, RR and 802.3ad modes outperformed the baseline in $\approx$64% of the low network usage scenario. Secondly, "*Which is the aggregation mode that provides better performance?*" RR performed better than 802.3ad in the majority of the executions. As expected, the NIC aggregation technique tends to have better results when we execute more network-intensive applications.

Although NIC aggregating can improve performance, a considerable amount of the configurations were done manually. The maximum number of aggregated physical links were limited to four, and all network interfaces must operate at the same speed to be aggregated. The IEEE 802.3ad mode requires a switch with support to this aggregation mode.

For the future, we plan to apply our evaluation model to leading public cloud providers such as Amazon AWS and Google Cloud, comparing not just one provider against itself but between providers. Also, we plan to expand the analysis in private clouds using other virtualization technologies like KVM and assess this environment with a wide range of other representative application domains.

Acknowledgment

This work has been supported by the projects; 1) GREEN-CLOUD project (#16/2551-0000 488-9) from FAPERGS and CNPq Brazil, program PRONEX 12/2014. 2) Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. 3) FAPERGS 10/2020-ARD project SPar4.0 (Nº 21/2551-0000725-7), and Universal MCTIC/CNPq Nº 28/2018 project SParCloud (Nº 437693/2018-0). We thank the Laboratory of Advanced Research on Cloud Computing (LARCC) for the infrastructure.

References

[1] S. Bhowmik, *Cloud Computing*. Cambridge University Press, 2017.

[2] P. Mell, T. Grance *et al.*, "The NIST Definition of Cloud Computing," *National Institute of Standards and Technology (NIST)*, 2011.

[3] J. Emeras, S. Varrette, V. Plugaru, and P. Bouvry, "Amazon Elastic Compute Cloud (EC2) versus In-House HPC Platform: A Cost Analysis," *IEEE Transactions on Cloud Computing (TCC)*, 2019.

[4] A. Gupta, P. Faraboschi, F. Gioachin, L. V. Kale, R. Kaufmann, B. Lee, V. March, D. Milojicic, and C. H. Suen, "Evaluating and Improving the Performance and Scheduling of HPC Applications in Cloud," *IEEE Transactions on Cloud Computing (TCC)*, 2016.

[5] E. Roloff, M. Diener, L. P. Gaspary, and P. O. A. Navaux, "HPC Application Performance and Cost Efficiency in the Cloud," in *Euromicro Int. Conference on Parallel, Distributed and Network-based Processing (PDP)*, 2017.

[6] J. Moura and D. Hutchison, "Review and Analysis of Networking Challenges in Cloud Computing," *Journal of Network and Computer Applications (JNCA)*, 2016.

[7] I. Sadooghi, J. H. Martin, T. Li, K. Brandstatter, K. Maheshwari, T. P. P. de Lacerda Ruivo, G. Garzoglio, S. Timm, Y. Zhao, and I. Raicu, "Understanding the Performance and Potential of Cloud Computing for Scientific Applications," *IEEE Transactions on Cloud Computing*, 2017.

[8] G. Pretto, B. L. Dalmazo, J. A. Marques, Z. Wu, X. Wang, V. Korkhov, P. O. Navaux, and L. P. Gaspary, "Janus: A Framework to Boost HPC Applications in the Cloud Based on SDN Path Provisioning," *Cluster Computing*, 2021.

[9] Y. He, W. Wu, Y. Le, M. Liu, and C. Lao, "A generic service to provide in-network aggregation for key-value streams," in *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, 2023, pp. 33–47.

[10] J. Fang, G. Zhao, H. Xu, C. Wu, and Z. Yu, "Grid: Gradient routing with in-network aggregation for distributed training," *IEEE/ACM Transactions on Networking*, pp. 1–14, 2023.

[11] A. Vogel, D. Griebler, C. A. F. Maron, C. Schepke, and L. G. Fernandes, "Private IaaS Clouds: A Comparative Analysis of OpenNebula, CloudStack and OpenStack," in *Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, 2016.

[12] R. Keller Tesser and E. Borin, "Containers in hpc: a survey," *The Journal of Supercomputing*, vol. 79, no. 5, pp. 5759–5827, 2023.

[13] IEEE, "IEEE Standard for Information Technology - Local and Metropolitan Area Networks - Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications-Aggregation of Multiple Link Segments," *IEEE Std 802.3ad-2000*, 2000.

[14] T. Davis, W. Tarreau, C. Gavrilov, C. N. Tindel, J. Girouard, J. Vosburgh, J. Vosburgh, and M. Williams. (2011) Linux Ethernet Bonding. [Online]. Available: https://www.kernel.org/doc/Documentation/networking/bonding.txt

[15] C. Rista, D. Griebler, C. A. F. Maron, and L. G. Fernandes, "Improving the Network Performance of a Container-Based Cloud Environment for Hadoop Systems," in *International Conference on High Performance Computing & Simulation (HPCS)*, 2017.

[16] D. Bailey, E. Barszcz, J. Barton, D. Browning, R. L. Carter, L. Dagum, R. A. Fatoohi, P. O. Frederickson, T. A. Lasinski, R. S. Schreiber, H. D. Simon, V. Venkatakrishnan, and S. K. Weeratunga, "The NAS Parallel Benchmarks; Summary and Preliminary Results," in *ACM/IEEE Conference on Supercomputing (SC)*, 1991.

[17] A. M. Maliszewski, A. Vogel, D. Griebler, E. Roloff, L. G. Fernandes, and P. O. A. Navaux, "Minimizing Communication Overheads in Container-based Clouds for HPC Applications," in *IEEE Symposium on Computers and Communications (ISCC)*, 2019.

[18] A. M. Maliszewski, E. Roloff, D. Griebler, L. P. Gaspary, and P. O. A. Navaux, "Performance Impact of IEEE 802.3ad in Container-based Clouds for HPC Applications," in *International Conference on Computational Science and its Applications (ICCSA)*, 2020.

[19] A. M. Maliszewski, E. Roloff, E. D. Carreño, D. Griebler, L. P. Gaspary, and P. O. A. Navaux, "Performance and Cost-aware HPC in Clouds: A Network Interconnection Assessment," in *IEEE Symposium on Computers and Communications (ISCC)*, 2020.

[20] N. Chakthranont, P. Khunphet, R. Takano, and T. Ikegami, "Exploring the Performance Impact of Virtualization on an HPC Cloud," in *International Conference on Cloud Computing Technology and Science (CloudCom)*, 2014.

[21] A. Vogel, D. Griebler, C. Schepke, and L. G. Fernandes, "An Intra-Cloud Networking Performance Evaluation on CloudStack Environment," in *Euromicro Int. Conference on Parallel, Distributed and Network-based Processing (PDP)*, St. Petersburg, Russia, 2017.

[22] T. Watanabe, M. Nakao, T. Hiroyasu, T. Otsuka, and M. Koibuchi, "Impact of Topology and Link Aggregation on a PC cluster with Ethernet," in *International Conference on Cluster Computing*, 2008.

[23] L. Chaufournier, A. Ali-Eldin, P. Sharma, P. Shenoy, and D. Towsley, "Performance Evaluation of Multi-Path TCP for Data Center and Cloud Workloads," in *ACM/SPEC International Conference on Performance Engineering*, 2019.

[24] C. Wang, C. Yang, W. Liao, R. Chang, and T. Wei, "Coupling GPU and MPTCP to Improve Hadoop/MapReduce Performance," in *International Conference on Intelligent Green Building and Smart Grid (IGBSG)*, 2016.