JOIN (Jurnal Online Informatika)
p-ISSN: 2528-1682, e-ISSN: 2527-9165
Volume 9 Number 2 | June 2024: 179-189
DOI: 10.15575/join.v9i1.1327

The Effect of the Number of Nodes on Data Communication Performance in Nomad Clusters Using the Gossip Protocol

Ridwan Satrio Hadikusuma[1], Veronica Windha Mahyastuty[2], Lukas[3], Epril Moh Rizaludin[4]
[1,2,3]Master of Electrical Engineering, Universitas Katolik Indonesia Atma Jaya, Jakarta, Indonesia
[4]Platform Engineer, PT. Metrocom Global Solusi, Jakarta, Indonesia

Article Info

Article history:

Received April 13, 2024
Revised June 2, 2024
Accepted June 4, 2024
Published August 26, 2024

Keywords:

Gossip Protocol
Latency
Nomad Cluster
Packet Loss
Throughput

ABSTRACT

This research aims to understand the effect of the number of nodes on the performance of data communication in Nomad clusters using the gossip protocol. Through a series of tests, it can be concluded that data communication performance is greatly affected by the number of nodes in the cluster. Tests were conducted using two clusters, where one cluster consists of three nodes. The results show that when using a cluster with three nodes, no packet loss occurs in all data transmissions performed, indicating a reliable communication system. The average latency in one data communication cycle varied in each test, but generally remained within the acceptable range of below 100ms based on data communication quality of service parameters. CPU and disc usage remained relatively stable throughout the experiment. Although there were slight differences in throughput between clusters, the throughput generally remained above 100 Mbps, which is still in the good category according to the research parameters. These results show the importance of taking into account the number of nodes in the cluster in designing and managing data communication systems in a Nomad cluster environment with the gossip protocol.

Corresponding Author:

Ridwan Satrio Hadikusuma,
Master of Electrical Engineering, Universitas Katolik Indonesia Atma Jaya, Jakarta, Indonesia
Jl. Jend. Sudirman No.51 5, RT.004/RW.4, Karet Semanggi, Kecamatan Setiabudi, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12930
Email: ridwan.202200090017@student.atmajaya.ac.id

1. INTRODUCTION

Within the increasingly advanced digital era [1], the telecommunications industry faces challenges in managing and transferring data quickly and efficiently [2]. Data communication management (orcheastor) manages and coordinates tasks run on a distributed computing system [3]. These tasks can be applications [4], services [5], data, or computing jobs that need to be run and managed [6]. Some examples of popular orchestrators or task managers include Kubernetes [7], Nomad [8], Mesos [9], and Docker Swarm [10]. Each orchestrator has different approaches and features in managing tasks [11], depending on the needs and usage scenario [12]. Nomad clusters, as an innovative application management system [13], have become a popular solution in organizing and deploying application workloads in complex environments [14]. In Nomad clusters, the gossip protocol is used to maintain consistency and disseminate information between nodes in the network [15]. The gossip protocol has great potential to affect the performance of data communication in Nomad clusters [16], especially when it involves different numbers of nodes [17]. The effect of the number of nodes in a Nomad cluster on data communication performance has not been fully understood and studied in depth [18]. Therefore, this research will explore and analyze the effect of the number of nodes on data communication performance in a Nomad cluster using the gossip protocol.

p-ISSN: 2528-1682
e-ISSN: 2527-9165

The Nomad cluster management platform has emerged as a popular cluster management platform [19], enabling its use in a wide range of applications [20], including telecommunications [21]. The gossip protocol, used in communication between nodes in a Nomad cluster, offers flexibility and resilience to failures [22]. However, the effect of the number of nodes on data communication performance in a Nomad environment is still not fully understood. Previous studies have provided insights into data communication performance in Nomad clusters [8], but most focus on specific aspects such as packet loss, latency, CPU usage, and disc usage [23]. There has been no research that specifically explores how varying the number of nodes in a Nomad cluster can affect the overall performance of data communication. Therefore, this research aims to fill this knowledge gap by investigating the effect of the number of nodes on data communication performance in Nomad clusters using the gossip protocol. By understanding the impact of the number of nodes on performance parameters such as packet loss, latency, throughput, and resource usage, this research is expected to provide valuable guidance for telecommunication companies in designing and managing their Nomad cluster infrastructure more efficiently and effectively. In addition, the results of this research can provide valuable insights for the development of cluster technology and distributed communication protocols in general. These results are expected to help improve the efficiency and effectiveness of cluster infrastructure design and management while promoting advances in distributed network technology.

2. METHOD

2.1. Research Phases

The research phase begins with a Literature Study to understand the concepts and theories relevant to the research. After that, enter the Software Design stage, where the system is designed according to the research needs. The next phase is Testing the System, where the system that has been designed is tested to make sure it works well. If the system does not work as expected, return to the Software Design stage for improvement. However, if the system works well, the research proceeds to the Analysis and Conclusion stage, where the test result data is analyzed, and conclusions are drawn based on the research findings. This research will be carried out in stages and structured to ensure accurate and meaningful research results. The stages of this research can be observed in Figure 1.

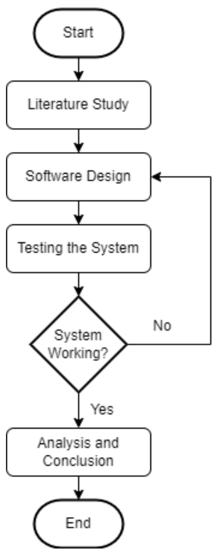

Figure 1. Research flowchart

2.2. Architectural Design

The research architecture design in this research method will include several important components. Firstly, a Nomad cluster consisting of several nodes will be formed and used in testing. Each node will have a defined role, such as leader nodes, follower nodes, and additional follower nodes, as listed in Figure 2. In Nomad clustering, determining follower nodes and leader nodes is an important step in organizing and managing dynamically moving computing resources [24].

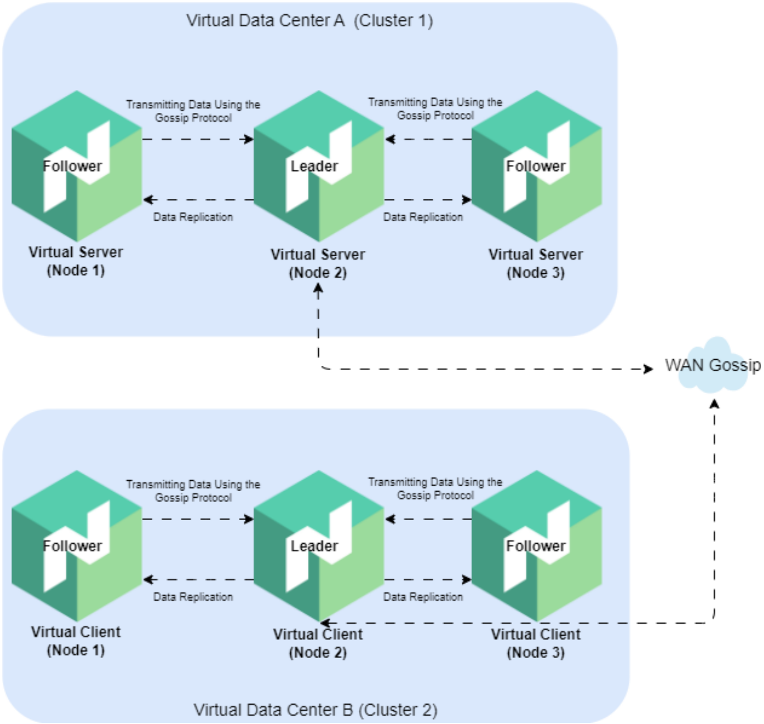

Figure 2. Architectural Design

Using various approaches [25], Nomad clustering can determine the appropriate follower nodes and leader nodes to optimize the management of mobile computing resources. Resources, reliability, availability, user preferences, and the dynamics of the computing environment can influence this decision. In this research, a resource characteristics approach is used to determine the leader and follower nodes. The best resource that will be used as a leader node is selected on a node with a maximum disk capacity of 60 GB with 2 GB of RAM (Random Access Memory). Meanwhile, the resource used as a follower node only has a maximum disk of 60 GB with 1 GB of RAM. While the two nodes are each installed using the Linux Centos 64-bit Operating System with 1 CPU core.

2.3. System Testing Method

The system testing method is carried out in several stages. First, Testing 3 Nomad Nodes in 1 Cluster was conducted to test the initial performance of the system. After that, a Test Scenario Evaluation is carried out to evaluate the stability conditions of the system. The test continues to the next stage if a stable condition is achieved. However, if it is unstable and there is an increase in disc and CPU usage, the test returns to the previous step for improvement. Next, the cluster is added with Added 1 Nomad Cluster with 3 Nodes, and the test scenario is repeated with the evaluation of stability conditions at the Test Scenario Evaluation stage. If a stable condition is achieved, the process continues, but if not, the previous steps are repeated for improvement. This process is repeated until the desired conclusion is reached at the Conclusion stage. Several relevant performance metrics will be taken during the test, such as data transfer time, latency, throughput, and CPU and memory utilization on each node. These measurements will provide a comprehensive picture of the data communication performance in Nomad clusters with varying numbers of nodes in the same cluster and different clusters. Furthermore, the test data will be evaluated using experimental methods.

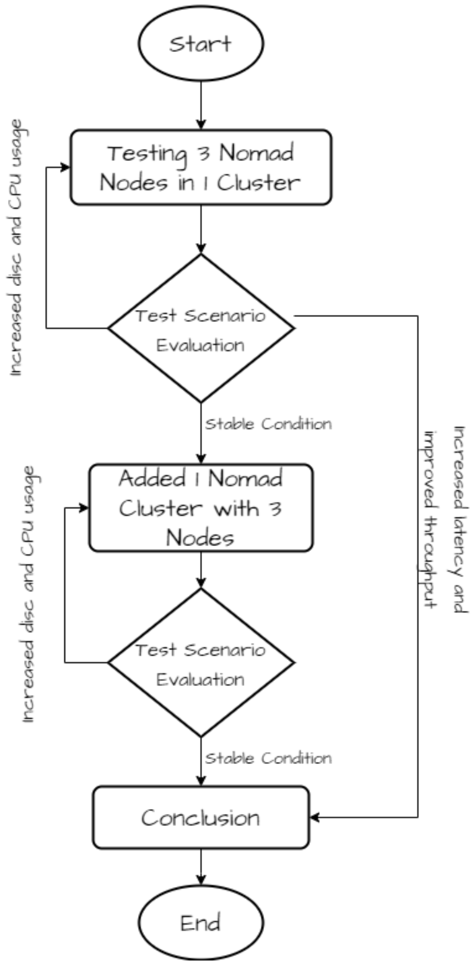

Figure 3. Testing Methods Flowchart

During the testing process, as described in Figure 3, other factors that can affect data communication performance will also be considered, such as network speed, network latency, and the workload of the application running on each node. This aims to ensure the tests are conducted under realistic conditions and obtain more accurate results. With this structured testing method, it is hoped that this research can provide an in-depth understanding of the effect of the number of nodes on data communication performance in Nomad clusters using the gossip protocol. The test results will serve as a basis for identifying the relationship between the number of nodes and data communication performance, as well as provide valuable insights into the future development of Nomad clusters.

2.4. Data Evaluation Method

The data evaluation method will be conducted using a structured and scalable approach. Firstly, communication performance data will be collected through experimental testing using various scenarios with varying numbers of nodes in the Nomad cluster. Tests will be conducted by sending simulated data and observing the data transfer time and throughput achieved. In addition, the evaluation method will also involve collecting data from logs and statistics generated by the Nomad Cluster, including information on latency, resource utilization, and data transfer reliability. The data obtained will be quantitatively analyzed using statistical methods such as regression analysis to determine the relationship between the number of nodes and communication performance. Although the data evaluation methods have been described in a structured manner and the scales used have been well described, the specific details regarding the use of the Tetris game and similar methods are not described in depth to keep the focus on the general data evaluation approach. The Tetris game and similar methods are only mentioned as a specific example of the test scenario. This lack of explanation is intended to keep the flow of explanation consistent and not fragmented.. This data evaluation will provide a better understanding of how the number of nodes in a Nomad cluster affects data communication performance, thus providing valuable insights for the telecoms industry. The parameters that will be investigated in this study can be seen in Table 1 [6].

Table 1. Parameters to be researched [6]

No	Parameter	Description
1	Number of Nodes	Number of nodes involved in the Nomad cluster
2	Gossip Protocols	Communication protocols used in Nomad clusters
3	Communication Latency	Time required to transfer data between nodes
4	Data Traffic	Volume of data sent and received by each node
5	Communication Throughput	Data transfer rate between nodes in the cluster
6	Packet Loss	Percentage of data packets lost during the communication process
7	Resource Usage	Utilisation of CPU, memory, and other resources on each node
8	Scalability	The ability of Nomad clusters to cope with an increasing number of nodes
9	Availability Level	Level of availability of Nomad clusters in servicing demand

The parameters commonly used in determining the quality of a data communication service can be seen in Table 2.

Table 2. Data communication service quality parameters [26]

No	Parameter	Minimum Standard
1	Latency	< 100 ms
2	Data Traffic	-
3	Throughput	> 100 Mbps
4	Packet Loss	< 1%

It should be noted that the minimum standards listed above are only general guidelines and may vary depending on the needs of a particular application or data communication environment [26]. Such minimum standards aim to maintain good data communication quality by minimizing latency, ensuring adequate throughput, and minimizing packet loss. By observing and evaluating these parameters, patterns, and relationships between the number of nodes and data communication performance in a Nomad cluster will be seen.

3. RESULT AND DISCUSSION

3.1. *Results of Data Communication Testing on Server Cluster*

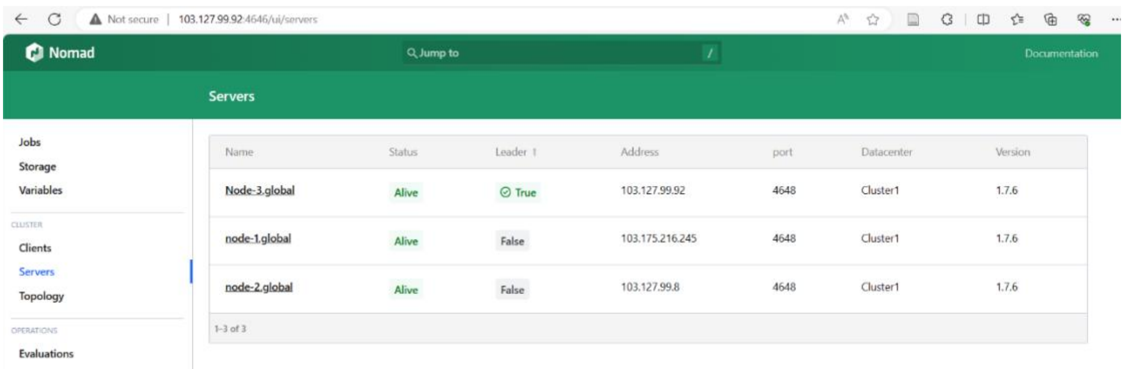

Figure 4. Cluster 1 test results with node leader

As shown in Figure 4, node-3 is used as the leader node because of its superior resource characteristics compared to other nodes, especially in the RAM capacity of 2 GB, where node-2 and node-1 only have 1 GB RAM capacity. After forming cluster 1 according to the architecture shown in Figure 5, the next step is to test the data communication between nodes in the server cluster. This test is done by running an application on the server that will observe the interaction of data communication between nodes in the cluster. This test aims to test the initial hypothesis regarding the effect of the number of nodes on data communication in Nomad, both communication that occurs within a cluster and between clusters. Thus, implementing the gossip protocol in the server cluster will be evaluated to understand how increasing or decreasing the number of nodes affects data communication performance in a Nomad environment. The results of these tests are expected to provide a deeper insight into the effect of the number of nodes on data communication performance on the Nomad platform.

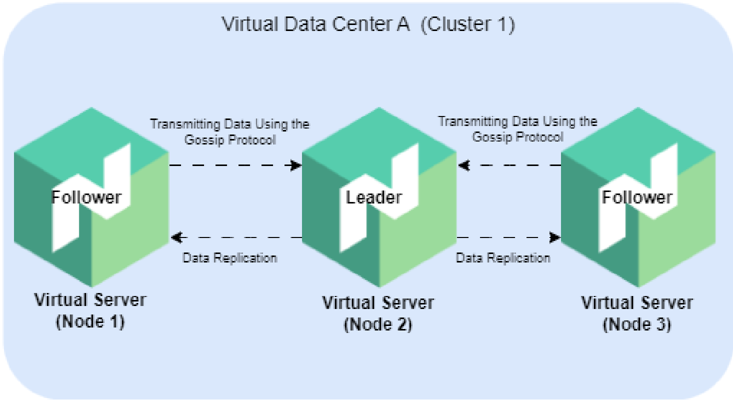

Figure 5. Implementation of Cluster 1 Architecture Design

The results of data communication testing on the server cluster show that an attempt was made to send ping data from the leader node to the member nodes to monitor parameters such as packet loss, latency, CPU usage, and disc usage. Based on the data recorded in Figure 6 (A), in 10 times of sending data from node 3 to node 1, there was no packet loss (0%), with an average CPU usage of 50 Mhz and disc usage of 0.001 Mb. In addition, the average latency in one data communication cycle was 0.9879 ms.

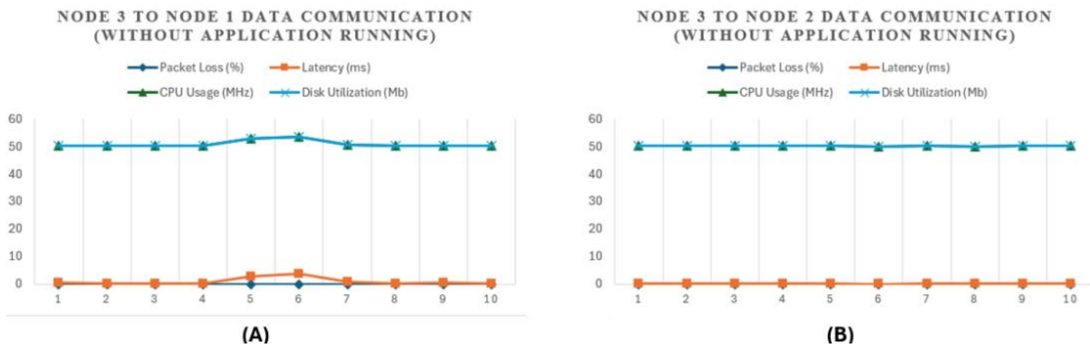

Figure 6. (A) Data Communication of Node 3 to Node 1 (Without Running Application) (B) Data Communication of Node 3 to Node 2 (Without Running Application)

Similarly, based on the data in Figure 6 (B), in the 10 times of sending data from node 3 to node 1, there was also no packet loss (0%), with the same average CPU usage and disc usage as in the previous data. The average latency in one data communication cycle in this experiment was 0.22555 ms. These results show that data transmission between leader nodes and member nodes in the server cluster is done without packet loss, with relatively low resource usage and fast latency, which supports the initial hypothesis regarding the number of nodes' effect on Nomad's data communication performance. Furthermore, testing was carried out by running the Tetris game application on the leader node. Furthermore, testing was carried out by running a job on Nomad as a Tetris game application on the leader node. This aims to evaluate Nomad's performance in distributing jobs to member nodes in a cluster.

Figure 7. (A) Data Communication of Node 3 to Node 1 (Running Application (B)Data Communication of Node 3 to Node 2 (Running Application)

From the data obtained in Figure 7 (A), in 10 times of sending data from node 3 to node 1, there was no packet loss (0%). The average CPU usage recorded was 50 Mhz, while the disc usage reached 256 Mb. The average latency in one data communication cycle was 3.4668 ms, with an average throughput of 256 Mbps. Meanwhile, in Figure 7 (B), the results of sending data from node 3 to node 1 also show no packet loss (0%) in 10 trials. CPU usage and disc usage also show similar numbers, which are 50 Mhz and 256 Mb, respectively. However, there was a difference in latency, which remained stable at 3.4668 ms, and throughput, which also remained consistent at 256 Mbps. The test results show a difference in data communication parameters before and after running the Nomad job. This shows the effect of job execution on data communication performance in the Nomad cluster. This data provides valuable insight into understanding how jobs or applications run in the cluster can affect the overall data communication performance.

3.2. Results of Data Communication Testing on Client Cluster

As shown in Figure 8, node-4 is used as the leader node because of its superior resource characteristics compared to other nodes, especially in RAM capacity of 2 GB compared to node-5 and node-6, which have only 1 GB RAM. Further tests will be conducted to monitor the performance of data communication in this client cluster, as well as to evaluate the effect of the number of nodes on data communication performance in a Nomad environment using the gossip protocol. After forming cluster 1 client according to the architecture shown in Figure 9, the next step is to test the data communication between nodes in the client cluster. This test is done by running an application on the client that will observe the interaction of data communication between nodes in the cluster.

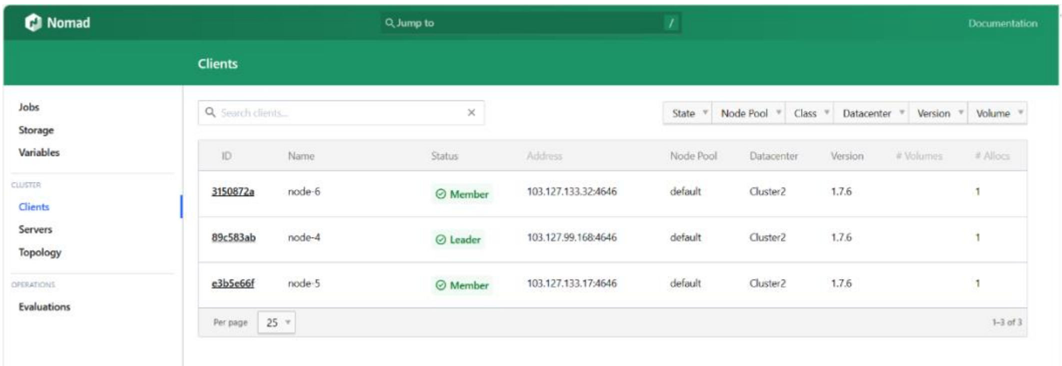

Figure 8. Cluster 2 test results with node leader

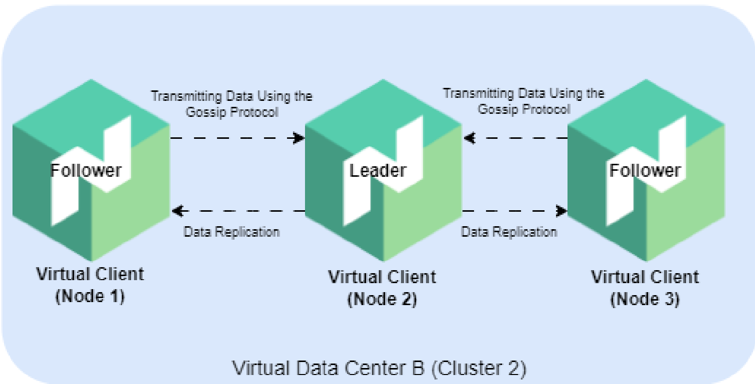

Figure 9. Implementation of Cluster 2 Architecture Design

From the test results recorded in Figure 10, in part (A), it can be seen that in the 10 times of sending data from node 4 to node 5, there was no packet loss (0%). The average CPU usage was 50 Mhz, and the disc usage was only 0.001 Mb. In addition, the average latency in one data communication cycle is 0.4392 ms. While in part (B), in 10 times of sending data from node 4 to node 6, there was also no packet loss (0%). CPU and disc usage were the same as in the previous experiment, at 50 Mhz and 0.001 Mb, respectively. However, the average latency in one data communication cycle was slightly lower, at 0.3579 ms.

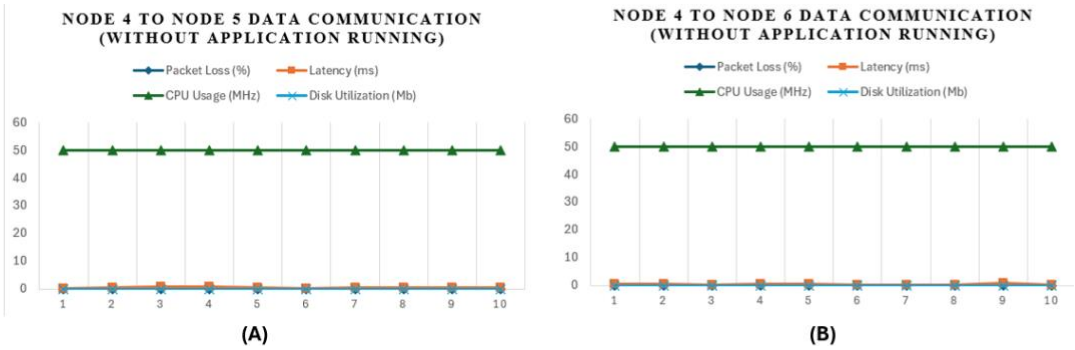

Figure 10. Data Communication of Node 4 to Node 5 (Without Application Running) (B)Data Communication of Node 4 to Node 6 (Without Application Running)

From the analysis of the results, it can be seen that although there is no significant difference in the usage of resources such as CPU and disk between the experiments from node 4 to node 5 and from node 4 to node 6, there is a notable difference in latency. The experiment from node 4 to node 6 shows a lower latency value compared to the previous experiment, indicating that the data communication between node 4 and node 6 has a faster response time. Furthermore, testing was carried out by running a job on Nomad in the form of a Tetris game application on the leader node. This aims to evaluate Nomad's performance in distributing jobs to member nodes in a cluster as well as in testing the server cluster.

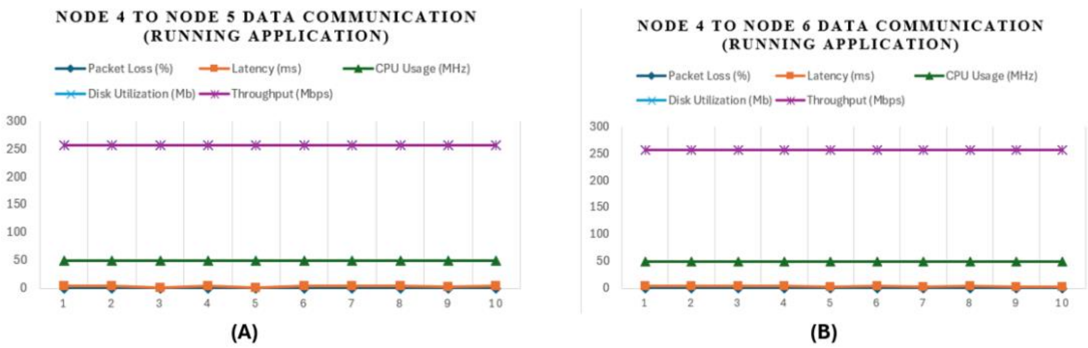

Figure 11. (A) Data Communication of Node 4 to Node 5 (Running Application) (B)Data Communication of Node 4 to Node 6 (Running Application)

Based on the data in Figure 11, in part A of sending data from node 4 to node 5, there is no packet loss (0%) with an average CPU usage of 50 MHz and disk utilization of 256 Mb. The average latency in one data communication cycle is 3.48 ms. While sending data from node 4 to node 6 in section B, there is also no packet loss (0%) with the same average CPU usage and disk utilization as in the previous transmission, namely 50 MHz and 256 Mb. However, there is a difference in the average latency, wherein the delivery from node 4 to node 6, the latency has increased to 3.648 ms. This shows the variation in data communication performance between different nodes in the client cluster, which can be affected by various factors such as the number of member nodes in a cluster, network traffic, and the density of resource usage on each node.

3.3. Testing Results of Data Communication between Clusters

In the test results of data communication between clusters, 2 clusters were tested with a total of 6 nodes, consisting of 3 server nodes and 3 client nodes divided into server clusters and client clusters, according to the architectural design in the architectural design implementation figure 12. The test was carried out by running the Nomad job containing the Tetris game application, where the job was distributed from the leader node on the server cluster. After that, the client node requests a job to the job server so that the job distribution is carried out from the leader node on the server cluster to all nodes in the client cluster using the gossip protocol.

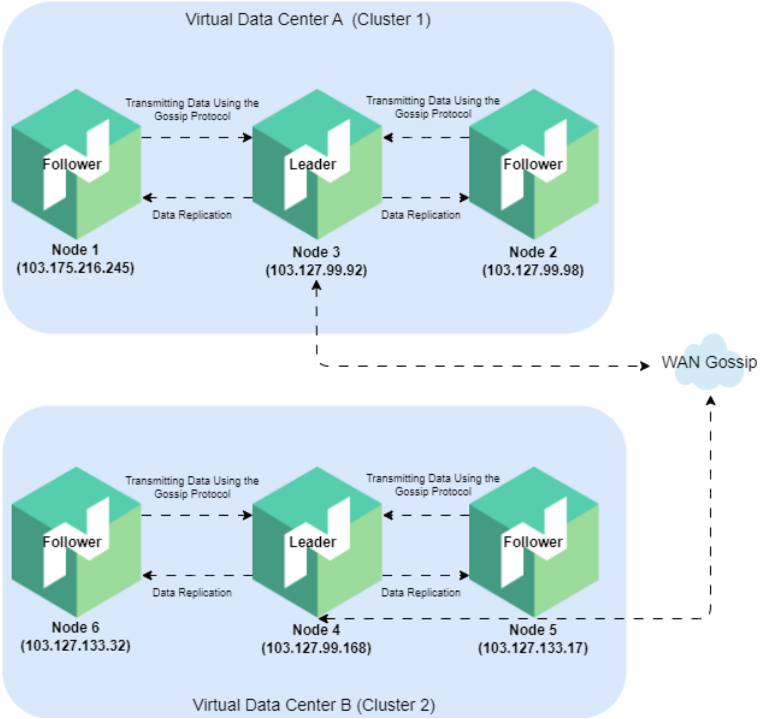

Figure 12. Architectural Design Implementation Results

The test results of data communication between clusters show that sending data from node 3, which serves as the leader node in the server cluster, to node 4, which serves as the leader node in the client cluster, results in an average latency of 3.505 ms in 10 times sending data, with a throughput of 255.1 Mbps. Packet loss in the test was 0%, and the average CPU usage was 39 Mhz, while the disc usage was 258 Mb, as shown in Figure 13. Analysis of the data shows that the average latency is slightly higher, and the average throughput is slightly lower compared to the results of previous data communication tests on server clusters and client clusters. Nevertheless, these results still show good performance in sending data between clusters, with latency still within an acceptable range and fairly high throughput when referring to Table 2 related to data communication service quality parameters. The correlation of these results with the research title suggests that increasing the number of nodes in a Nomad cluster can potentially affect data communication performance.

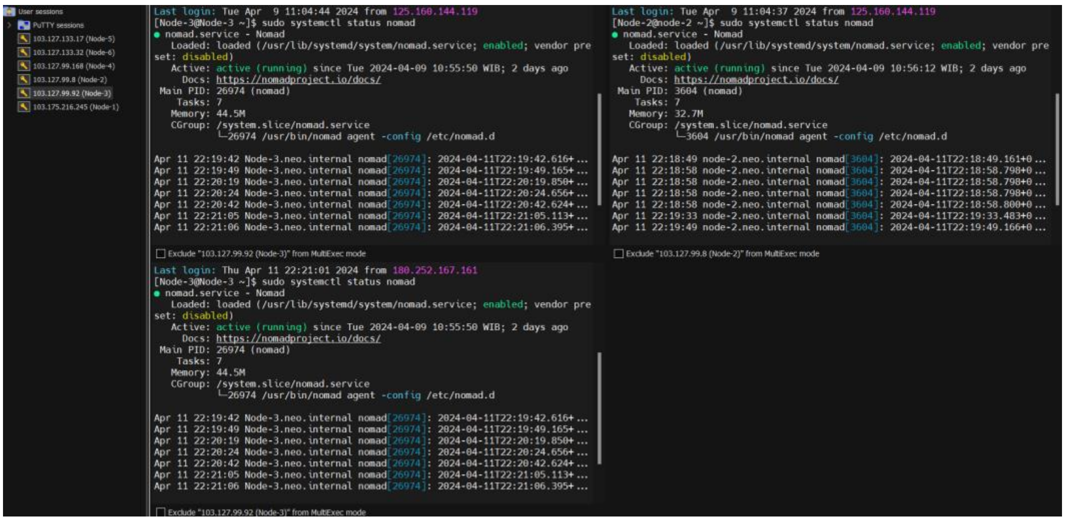
Figure 13. Observation results of the alert status log on Cluster 1 during testing

Testing client and server cluster data communication also involves checking the status of the alert log on each cluster. This aims to ascertain whether there is a possibility of an alert, such as a failure in communication between nodes or clusters or even a disconnection at one of the nodes. This checking process is recorded and documented in Figures 13 and 14. The test results also show that the job distribution process between clusters using the gossip protocol can run well. In each iteration of job delivery from the server cluster to the client cluster, there is no failure in data transmission and no packet loss. In addition, performance parameters such as latency, throughput, CPU usage, and disk utilization are also observed to evaluate overall system performance.

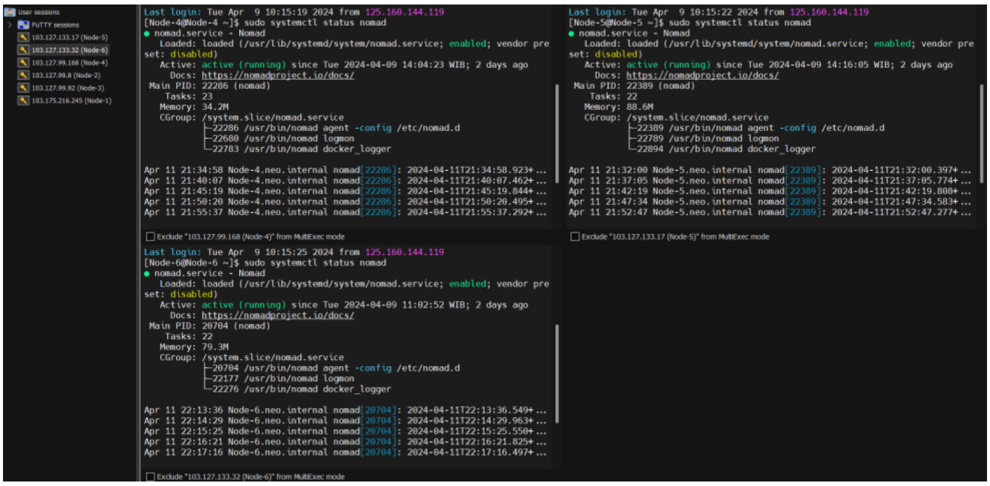

Figure 14. Observation results of the alert status log on Cluster 2 during testing

From the log observations made, it can be concluded that both clusters were in good condition during the test when referring to the parameters in Table 2, given that there were no alerts indicating packet loss or very high latency. Both clusters have been operating properly for 2 days, and no alerts were detected during the test. This indicates that the system has been functioning stably and reliably in carrying out its duties and did not experience any significant disruptions during the test. This research emphasizes the importance of ensuring system stability and availability in various situations, especially in a distributed environment such as a Nomad cluster. By regularly monitoring status logs, researchers can ensure that the system operates as intended and is ready to handle assigned tasks efficiently and reliably.

4. CONCLUSION

Based on the results of data communication testing on Nomad clusters using the gossip protocol, it can be concluded that the number of nodes in the cluster influences data communication performance. Test data using two clusters with one cluster consisting of three nodes shows no packet loss in all data transmissions made, so the communication system becomes reliable. The average latency in one data communication cycle varied in each test but generally remained within the acceptable range of <100ms based on data communication service quality parameters. CPU usage and disc usage were also relatively stable throughout the experiment. There were slight differences in throughput between clusters. However, the throughput generally had values > 100 Mbps and was still in the good category according to the research parameters.

REFERENCES

[1] "Data communications networking with TCPIP protocol suite (Behrouz A. Forouzan) (Z-Library).epub."

[2] F. B.-U. Team, "Microservices in the Cloud Native Era," in *Cloud-Native Application Architecture: Microservice Development Best Practice*, F. B.-U. Team, Ed., Singapore: Springer Nature, 2024, pp. 1–25. doi: 10.1007/978-981-19-9782-2_1.

[3] M. Straesser, J. Mathiasch, A. Bauer, and S. Kounev, "A Systematic Approach for Benchmarking of Container Orchestration Frameworks," in *Proceedings of the 2023 ACM/SPEC International Conference on Performance Engineering*, in ICPE '23. New York, NY, USA: Association for Computing Machinery, Apr. 2023, pp. 187–198. doi: 10.1145/3578244.3583726.

[4] S. Liu, M. He, Z. Wu, P. Lu, and W. Gu, "Spatial–temporal graph neural network traffic prediction based load balancing with reinforcement learning in cellular networks," *Inf. Fusion*, vol. 103, p. 102079, Mar. 2024, doi: 10.1016/j.inffus.2023.102079.

[5] K. Karmakar, S. Dey, R. K. Das, and S. Khatua, "Scheduling of Containerized Resources for Microservices in Cloud," in *Distributed Computing and Intelligent Technology*, S. Devismes, P. S. Mandal, V. V. Saradhi, B. Prasad, A. R. Molla, and G. Sharma, Eds., in Lecture Notes in Computer Science. Cham: Springer Nature Switzerland, 2024, pp. 34–49. doi: 10.1007/978-3-031-50583-6_3.

[6] E. Kalyvianaki and M. Paolieri, *Performance Evaluation Methodologies and Tools: 16th EAI International Conference, VALUETOOLS 2023, Crete, Greece, September 6–7, 2023, Proceedings*. Springer Nature, 2024.

[7] Y. Fu *et al.*, "Progress-based Container Scheduling for Short-lived Applications in a Kubernetes Cluster," in *2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA: IEEE, Dec. 2019, pp. 278–287. doi: 10.1109/BigData47090.2019.9006427.

[8] R. Bhardwaj and A. Ghouas, "Nomad - Hierarchical Computation Framework for IoT applications".

[9] P. Geetha, S. J. Vivekanandan, R. Yogitha, and M. S. Jeyalakshmi, "Optimal load balancing in cloud: Introduction to hybrid optimization algorithm," *Expert Syst. Appl.*, vol. 237, p. 121450, Mar. 2024, doi: 10.1016/j.eswa.2023.121450.

[10] V. Kaliappan, S. Yu, R. Soundararajan, S. Jeon, D. Min, and E. Choi, "High-Secured Data Communication for Cloud Enabled Secure Docker Image Sharing Technique Using Blockchain-Based Homomorphic Encryption," *Energies*, vol. 15, no. 15, p. 5544, Jul. 2022, doi: 10.3390/en15155544.

[11] K. Wang, Q. Zhou, S. Guo, and J. Luo, "Cluster Frameworks for Efficient Scheduling and Resource Allocation in Data Center Networks: A Survey," *IEEE Commun. Surv. Tutor.*, vol. 20, no. 4, pp. 3560–3580, 2018, doi: 10.1109/COMST.2018.2857922.

[12] V. Vassilev, "Data Platforms, Clouds and Spaces: Integration & Hybridization in Data Processing".

[13] "Tutorials | Nomad | HashiCorp Developer," Tutorials | Nomad | HashiCorp Developer. Accessed: Apr. 03, 2024. [Online]. Available: https://developer.hashicorp.com/nomad/tutorials

[14] N. Sabharwal, S. Pandey, and P. Pandey, *Infrastructure-as-Code Automation Using Terraform, Packer, Vault, Nomad and Consul: Hands-on Deployment, Configuration, and Best Practices*. Berkeley, CA: Apress, 2021. doi: 10.1007/978-1-4842-7129-2.

[15] P. Riti and D. Flynn, *Beginning HCL Programming: Using Hashicorp Language for Automation and Configuration*. Berkeley, CA: Apress, 2021. doi: 10.1007/978-1-4842-6634-2.

[16] B. Buyukates, M. Bastopcu, and S. Ulukus, "Version Age of Information in Clustered Gossip Networks." arXiv, Sep. 17, 2021. Accessed: Apr. 30, 2023. [Online]. Available: http://arxiv.org/abs/2109.08669

[17] Q.-H. Chen and C.-Y. Wen, "Optimal Resource Allocation Using Genetic Algorithm in Container-Based Heterogeneous Cloud," *IEEE Access*, pp. 1–1, 2024, doi: 10.1109/ACCESS.2024.3351944.

[18] S. R. Dira and M. A. F. Ridha, "Monitoring Kubernetes Cluster MenggunakanPrometheus dan Grafana," 2022.

[19] M. Straesser, J. Mathiasch, A. Bauer, and S. Kounev, "A Systematic Approach for Benchmarking of Container Orchestration Frameworks," in *Proceedings of the 2023 ACM/SPEC International Conference on Performance Engineering*, Coimbra Portugal: ACM, Apr. 2023, pp. 187–198. doi: 10.1145/3578244.3583726.

[20] M. Femminella and G. Reali, "Gossip-based Monitoring Protocol for 6G Networks," *IEEE Trans. Netw. Serv. Manag.*, pp. 1–1, 2023, doi: 10.1109/TNSM.2023.3263542.

[21] O. C. Ibe, *Fundamentals of Data Communication Networks*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2017. doi: 10.1002/9781119436294.

[22] R. Jin, Y. Huang, and H. Dai, "On the Privacy Guarantees of Gossip Protocols in General Networks." arXiv, Feb. 05, 2021. Accessed: Apr. 30, 2023. [Online]. Available: http://arxiv.org/abs/1905.07598

[23] M. A. Setiawan and I. A. N. Fathony, "Containerization of Shibboleth IdP as access management single sign-on (SSO) service based on integrated Kubernetes cluster with GitLab CI automation," presented at the VII INTERNATIONAL CONFERENCE "SAFETY PROBLEMS OF CIVIL ENGINEERING CRITICAL INFRASTRUCTURES" (SPCECI2021), Yekaterinburg, Russia, 2023, p. 020035. doi: 10.1063/5.0130139.

[24] W. Lu, L. Liang, B. Kong, B. Li, and Z. Zhu, "AI-Assisted Knowledge-Defined Network Orchestration for Energy-Efficient Data Center Networks," *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 86–92, Jan. 2020, doi: 10.1109/MCOM.001.1800157.

[25] "A VIRTUAL DATA CENTER COMPARISON OF DIFFERENT FIREWALLS PERFORMANCE_Hanane Aznaoui_Canan Batur Şahin_Journal of Advancement in Computing.pdf."

[26] B. C. Tedeschini, S. Savazzi, and M. Nicoli, "A Traffic Model based Approach to Parameter Server Design in Federated Learning Processes," *IEEE Commun. Lett.*, pp. 1–1, 2023, doi: 10.1109/LCOMM.2023.3272844.