Distributed Container Management in Cloud Computing Using Kubernetes and Docker

Aphriam C Benoraj[1], Dr. Sudheer S Marar[2]
[1,2]*MCA, Nehru College of Engineering and Research Centre, Thrissur, Kerala, India*

***Abstract*—Container orchestration technologies play a crucial role in deploying large-scale artificial intelligence applications in cloud environments. Modern AI systems require scalable infrastructure capable of handling distributed workloads and high computational demand. This review analyzes orchestration platforms including Docker Swarm and Kubernetes, evaluating scheduling mechanisms, scalability, and resource utilization while identifying future research directions in AI infrastructure.**

***Index Terms*— Artificial Intelligence, Cloud Computing, Container Orchestration, Docker, Kubernetes**.

I. INTRODUCTION

Artificial intelligence applications increasingly rely on scalable distributed infrastructure. Containerization technologies enable consistent deployment of software systems across cloud environments. Orchestration frameworks automate container management, improving reliability and scalability of AI workloads.
Artificial intelligence applications increasingly rely on scalSSable distributed infrastructure. Containerization technologies enable consistent deployment of software systems across cloud environments. Orchestration frameworks automate container management, improving reliability and scalability of AI workloads.
Artificial intelligence applications increasingly rely on scalable distributed infrastructure. Containerization technologies enable consistent deployment of software systems across cloud environments. Orchestration frameworks automate container management, improving reliability and scalability of AI workloads.
Artificial intelligence applications increasingly rely on scalable distributed infrastructure. Containerization technologies enable consistent deployment of software systems across cloud environments. Orchestration frameworks automate container management, improving reliability and scalability of AI workloads.

II. BACKGROUND OF CONTAINER ORCHESTRATION

Container orchestration platforms manage clusters of containerized applications by automating scheduling, scaling, networking, and monitoring. Kubernetes and Docker Swarm are widely used orchestration technologies that support distributed cloud-native systems.
Container orchestration platforms manage clusters of containerized applications by automating scheduling, scaling, networking, and monitoring. Kubernetes and Docker Swarm are widely used orchestration technologies that support distributed cloud-native systems.
Container orchestration platforms manage clusters of containerized applications by automating scheduling, scaling, networking, and monitoring. Kubernetes and Docker Swarm are widely used orchestration technologies that support distributed cloud-native systems.
Container orchestration platforms manage clusters of containerized applications by automating scheduling, scaling, networking, and monitoring. Kubernetes and Docker Swarm are widely used orchestration technologies that support distributed cloud-native systems.

III. REVIEW OF RELATED RESEARCH

Previous studies highlight the advantages of Kubernetes in large-scale deployments due to advanced scheduling mechanisms and fault tolerance. Docker Swarm provides simplicity and ease of configuration, making it suitable for smaller deployments.
Previous studies highlight the advantages of Kubernetes in large-scale deployments due to advanced scheduling mechanisms and fault tolerance.

Docker Swarm provides simplicity and ease of configuration, making it suitable for smaller deployments.

Previous studies highlight the advantages of Kubernetes in large-scale deployments due to advanced scheduling mechanisms and fault tolerance.

Docker Swarm provides simplicity and ease of configuration, making it suitable for smaller deployments.

Previous studies highlight the advantages of Kubernetes in large-scale deployments due to advanced scheduling mechanisms and fault tolerance.

Docker Swarm provides simplicity and ease of configuration, making it suitable for smaller deployments.

IV. PERFORMANCE ANALYSIS

Performance evaluation of orchestration platforms focuses on metrics such as CPU utilization, memory efficiency, response latency, and throughput. Research indicates that Kubernetes performs better in large clusters while Docker Swarm offers easier setup.

Performance evaluation of orchestration platforms focuses on metrics such as CPU utilization, memory efficiency, response latency, and throughput. Research indicates that Kubernetes performs better in large clusters while Docker Swarm offers easier setup.

Performance evaluation of orchestration platforms focuses on metrics such as CPU utilization, memory efficiency, response latency, and throughput. Research indicates that Kubernetes performs better in large clusters while Docker Swarm offers easier setup.

Performance evaluation of orchestration platforms focuses on metrics such as CPU utilization, memory efficiency, response latency, and throughput. Research indicates that Kubernetes performs better in large clusters while Docker Swarm offers easier setup.

V. CHALLENGES IN AI WORKLOAD ORCHESTRATION

AI workloads introduce challenges including GPU scheduling, energy consumption, and dynamic resource allocation. Addressing these issues requires improved orchestration strategies and intelligent scheduling mechanisms.

AI workloads introduce challenges including GPU scheduling, energy consumption, and dynamic resource allocation. Addressing these issues requires improved orchestration strategies and intelligent scheduling mechanisms.

AI workloads introduce challenges including GPU scheduling, energy consumption, and dynamic resource allocation. Addressing these issues requires improved orchestration strategies and intelligent scheduling mechanisms.

AI workloads introduce challenges including GPU scheduling, energy consumption, and dynamic resource allocation. Addressing these issues requires improved orchestration strategies and intelligent scheduling mechanisms.

VI. FUTURE RESEARCH DIRECTIONS

Future research aims to integrate machine learning techniques into orchestration frameworks to enable predictive scaling and adaptive resource allocation for distributed AI systems.

Future research aims to integrate machine learning techniques into orchestration frameworks to enable predictive scaling and adaptive resource allocation for distributed AI systems.

Future research aims to integrate machine learning techniques into orchestration frameworks to enable predictive scaling and adaptive resource allocation for distributed AI systems.

Future research aims to integrate machine learning techniques into orchestration frameworks to enable predictive scaling and adaptive resource allocation for distributed AI systems.

VII. CONCLUSION

Container orchestration platforms play an essential role in supporting modern AI infrastructure. Kubernetes provides strong scalability and reliability while Docker Swarm offers simplicity for smaller systems.

Container orchestration platforms play an essential role in supporting modern AI infrastructure. Kubernetes provides strong scalability and reliability while Docker Swarm offers simplicity for smaller systems.

Container orchestration platforms play an essential role in supporting modern AI infrastructure. Kubernetes provides strong scalability and reliability

while Docker Swarm offers simplicity for smaller systems.

REFERENCES

[1] D. Merkel, "Docker: Lightweight Linux containers for consistent development and deployment," Linux Journal, vol. 2014, no. 239, 2014.

[2] B. Burns, B. Grant, D. Oppenheimer, E. Brewer, and J. Wilkes, "Borg, Omega, and Kubernetes: Lessons learned from three container-management systems over a decade," Communications of the ACM, vol. 59, no. 5, pp. 50–57, 2016.

[3] K. Hightower, B. Burns, and J. Beda, Kubernetes: Up and Running. Sebastopol, CA: O'Reilly Media, 2017.

[4] C. Pahl, "Containerization and the cloud," IEEE Cloud Computing, vol. 2, no. 3, pp. 24–31, 2015.

[5] R. T. Avireneni, S. Sharma, and A. K. Singh, "Comparative analysis of container orchestration frameworks for AI workloads," International Journal of Cloud Computing, 2021.

[6] C. Zhang and Y. Chen, "Resource management in cloud computing: A review," Journal of Network and Computer Applications, vol. 60, pp. 1–14, 2016.

[7] M. Mao and M. Humphrey, "A performance study on the VM startup time in the cloud," in Proc. IEEE Int. Conf. Cloud Computing, 2012, pp. 423–430.

[8] Verma, L. Pedrosa, M. Korupolu, D. Oppenheimer, E. Tune, and J. Wilkes, "Large-scale cluster management at Google with Borg," in Proc. ACM European Conf. Computer Systems (EuroSys), 2015, pp. 1–17.

[9] S. Newman, Building Microservices. Sebastopol, CA: O'Reilly Media, 2015.

[10] B. Hindman et al., "Mesos: A platform for fine-grained resource sharing in the data center," in Proc. USENIX Symp. Networked Systems Design and Implementation (NSDI), 2011, pp. 295–308.

[11] Y. Liu and M. Zhou, "Resource scheduling for cloud computing: A survey," Journal of Supercomputing, vol. 76, pp. 1–29, 2020.

[12] Balalaie, A. Heydarnoori, and P. Jamshidi, "Microservices architecture enables DevOps: Migration to a cloud-native architecture," IEEE Software, vol. 33, no. 3, pp. 42–52, 2016.

[13] K. Hwang, G. Fox, and J. Dongarra, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things. Burlington, MA: Morgan Kaufmann, 2012.

[14] Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," in Proc. Grid Computing Environments Workshop, 2008, pp. 1–10.

[15] M. Satyanarayanan, "The emergence of edge computing," Computer, vol. 50, no. 1, pp. 30–39, 2017.

[16] L. Chen, "Container orchestration: A survey," ACM Computing Surveys, vol. 53, no. 2, pp. 1–36, 2020.

[17] Gupta and P. Singh, "AI workload scheduling in cloud environments," Future Generation Computer Systems, 2022.

[18] R. Buyya, C. Vecchiola, and S. T. Selvi, Mastering Cloud Computing: Foundations and Applications Programming. Burlington, MA: Morgan Kaufmann, 2013.

[19] Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," Communications of the ACM, vol. 51, no. 1, pp. 107–113, 2008.

[20] X. Xu, H. Zhu, and L. Zhang, "Artificial intelligence in cloud infrastructure," IEEE Transactions on Cloud Computing, 2021.