International Research Journal of Innovative Engineering (IRJIE)

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 6, Issue 6, November- December 2022|

DOI:10.15680/IRJIE.2022.0606001

Intelligent Kubernetes Resource Scheduling and Optimization in Distributed Multi-Cloud Computing Systems

Chetana Chouhan

Independent Researcher, USA

ABSTRACT: Modern distributed computing environments increasingly rely on container orchestration platforms such as Kubernetes to manage scalable applications across multi-cloud infrastructures. However, traditional Kubernetes schedulers are primarily rule-based and lack intelligence in optimizing resource allocation across heterogeneous cloud environments. This leads to inefficiencies such as resource underutilization, increased latency, and higher operational costs. Intelligent scheduling using Artificial Intelligence (AI) and Machine Learning (ML) has emerged as a promising solution to enhance decision-making in Kubernetes clusters. AI-driven schedulers analyze workload patterns, resource availability, and system performance metrics to dynamically allocate containers in real time. Techniques such as reinforcement learning, predictive analytics, and optimization algorithms enable adaptive scheduling strategies that improve performance and reduce energy consumption. In multi-cloud environments, intelligent scheduling also considers network latency, data locality, and cost variations across cloud providers. Despite advancements, challenges remain in scalability, interoperability, security, and real-time decision-making. This paper explores AI-based Kubernetes scheduling techniques, optimization strategies, and architectural frameworks for efficient resource management in distributed multi-cloud systems.

KEYWORDS: Kubernetes, Resource Scheduling, Multi-Cloud Computing, Artificial Intelligence, Machine Learning, Container Orchestration, Optimization, Reinforcement Learning, Distributed Systems, Cloud Computing

I. INTRODUCTION

Intelligent Kubernetes resource scheduling and optimization in distributed multi-cloud computing systems further evolves through the incorporation of advanced workload profiling techniques that continuously analyze application behavior patterns, including CPU bursts, memory allocation cycles, I/O intensity, and network communication frequency, enabling the scheduler to classify workloads not only based on static descriptors but also on dynamic runtime behavior, which becomes critical in distributed systems where application performance can vary significantly depending on the underlying cloud provider and region, and this profiling data is aggregated into centralized and decentralized telemetry pipelines that feed machine learning models responsible for forecasting future resource demands, while clustering algorithms group similar workloads together to improve placement efficiency and reduce fragmentation across nodes, and in multi-cloud environments where infrastructure heterogeneity is the norm, intelligent scheduling systems must account for differences in virtualization technologies, hardware generations, instance pricing structures, and storage backends, which introduces complexity into optimization models that must simultaneously evaluate cost-performance trade-offs across multiple dimensions, and reinforcement learning agents operating within this ecosystem are trained using reward functions that combine latency reduction, throughput maximization, cost minimization, and energy efficiency, thereby enabling the system to autonomously discover scheduling strategies that outperform static heuristics and greedy bin-packing algorithms, while federated learning approaches are increasingly used to train scheduling models across multiple clusters without centralizing sensitive data, thereby improving privacy

II. LITERATURE REVIEW

cloud providers, allowing workloads with flexible latency requirements to be shifted dynamically to lower-cost infrastructure without violating service-level agreements, and observability systems integrated with Kubernetes scheduling pipelines provide high-resolution insights into cluster health through distributed tracing, log aggregation, and metric correlation, enabling rapid detection of performance degradation and enabling corrective scheduling actions, and policy enforcement engines ensure compliance with regulatory frameworks such as GDPR-like data protection

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 6, Issue 6, November- December 2022|

DOI:10.15680/IRJIE.2022.0606001

rules, industry-specific security standards, and organizational governance policies, restricting workload placement based on geographic, legal, or security constraints, and in multi-tenant environments, resource isolation mechanisms are enforced through namespaces, cgroups, and network policies to ensure fair allocation and prevent interference between competing workloads, while adaptive scheduling algorithms continuously refine decision-making models based on feedback loops that compare expected performance outcomes with actual runtime metrics, thereby enabling continuous learning and optimization, and hybrid cloud architectures further enhance flexibility by enabling seamless workload migration between private and public clouds based on demand fluctuations, cost variations, and compliance requirements, and Kubernetes operators and custom controllers extend native scheduling capabilities by embedding domain-specific logic into the orchestration layer, allowing organizations to implement specialized optimization strategies tailored to their workloads, and ultimately the convergence of artificial intelligence, distributed systems engineering, and cloud-native architectures transforms Kubernetes scheduling into an intelligent, self-adaptive system capable of operating at global scale while maintaining efficiency, resilience, and cost-effectiveness across highly dynamic multi-cloud environments.

and scalability in distributed environments, and as workloads become increasingly microservice-oriented, service dependency graphs play a crucial role in scheduling decisions, as tightly coupled services must be co-located or placed within low-latency network proximity to prevent cascading delays and performance bottlenecks, and intelligent schedulers incorporate graph-based optimization techniques to map service relationships and identify optimal cluster placements, while edge-cloud coordination introduces additional constraints where latency-sensitive workloads such as autonomous vehicle processing, smart city monitoring, and industrial IoT analytics must be executed closer to data sources at the edge, while compute-intensive aggregation and long-term storage operations are offloaded to central cloud regions, creating a hierarchical scheduling model that spans edge, regional, and global layers, and autoscaling mechanisms are enhanced through predictive scaling models that use time-series forecasting and anomaly detection to anticipate workload spikes before they occur, thereby provisioning resources proactively rather than reactively, and this reduces cold-start latency and improves service reliability under unpredictable traffic conditions, while network-aware scheduling algorithms continuously evaluate cross-region bandwidth costs and dynamically adjust service placement to reduce inter-service communication overhead, and storage-aware scheduling ensures that stateful applications such as databases, distributed caches, and streaming systems are placed in proximity to high-performance storage systems to minimize read/write latency, and cost optimization strategies leverage real-time spot market pricing fluctuations across

III. RESEARCH METHODOLOGY

Intelligent Kubernetes resource scheduling and optimization in distributed multi-cloud computing systems becomes significantly more complex when formalized through mathematical optimization models and algorithmic decision frameworks that attempt to represent real-world cloud heterogeneity, where the scheduling problem is often modeled as a multi-objective optimization problem aiming to minimize cost, latency, and energy consumption while maximizing throughput, reliability, and resource utilization under strict constraints such as CPU, memory, GPU availability, and network bandwidth limits, and this problem is further complicated by its NP-hard nature, which necessitates the use of heuristic, metaheuristic, and machine learning-based approximations rather than exact solutions in large-scale environments, and modern intelligent schedulers often combine constraint programming with reinforcement learning agents that operate on cluster state representations encoded as high-dimensional feature vectors, allowing the system to learn scheduling policies that approximate optimal decisions through iterative interaction with the environment, while deep reinforcement learning models such as policy gradient methods and Q-learning variants are used to estimate long-term reward functions associated with workload placement decisions, and these reward functions are carefully designed to reflect service-level objectives including response time minimization, error rate reduction, and cost efficiency across distributed cloud regions, and in parallel, graph neural networks are increasingly applied to model microservice dependency structures, where nodes represent services and edges represent communication patterns, enabling the scheduler to understand inter-service relationships and co-location requirements that improve latency and reduce cross-region traffic costs, and multi-cloud orchestration introduces additional layers of abstraction where Kubernetes federation APIs or custom multi-cluster controllers coordinate workload placement decisions across independent control planes, requiring consistent state synchronization

mechanisms and eventual consistency models to ensure that scheduling decisions remain coherent despite network partitions or delayed updates, and edge-cloud hierarchical scheduling architectures introduce three-tier models consisting of edge nodes, regional clusters, and global cloud backends, where workloads are dynamically assigned based on latency sensitivity, compute intensity, and data proximity requirements, and predictive analytics pipelines

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 6, Issue 6, November- December 2022|

DOI:10.15680/IRJIE.2022.0606001

integrated into these systems continuously ingest telemetry data from observability platforms and apply time-series forecasting models such as ARIMA, LSTM networks, and transformer-based predictors to estimate future workload demands and infrastructure stress points, enabling proactive scaling and scheduling adjustments, while cost-aware optimization engines continuously evaluate real-time pricing fluctuations across cloud providers, including spot instance volatility and reserved capacity utilization, dynamically migrating workloads to minimize operational expenditure while respecting performance constraints, and network optimization models consider inter-region latency matrices and bandwidth cost graphs to compute optimal placement strategies that reduce communication overhead between distributed microservices, while storage-aware scheduling systems integrate distributed file system metrics and object storage performance characteristics to ensure that data-intensive workloads are placed in proximity to high-throughput storage layers, and security-aware scheduling frameworks enforce zero-trust principles by incorporating identity verification, workload attestation, encrypted communication channels, and compliance-aware placement policies that restrict sensitive workloads to approved geographic or organizational domains, and multi-tenant fairness is achieved through advanced resource allocation algorithms such as weighted fair sharing, dominant resource fairness, and priority-based scheduling queues that ensure equitable distribution of compute resources across competing workloads, and adaptive feedback loops embedded within the Kubernetes scheduler continuously evaluate the accuracy of predictive models by comparing expected versus actual performance outcomes, enabling continuous retraining and refinement of machine learning models, and hybrid cloud bursting strategies allow workloads to dynamically transition between private infrastructure and public cloud resources in response to demand surges, ensuring elasticity and fault tolerance, and service mesh technologies such as Istio and Linkerd enhance observability and traffic control by providing fine-grained routing, circuit breaking, and telemetry collection across distributed services, and ultimately the integration of artificial intelligence, distributed systems theory, and cloud-native orchestration principles transforms Kubernetes scheduling into a self-optimizing cyber-physical control system capable of operating across globally distributed, heterogeneous, and highly dynamic multi-cloud infrastructures.

Another promising direction is the application of. Currently, AI-based scheduling decisions are often opaque, making it difficult for system administrators to understand why specific workloads are placed on certain nodes. Future research should focus on integrating interpretability techniques such as SHAP, attention-based models, and rule extraction methods to provide transparent scheduling explanations. This will increase trust and adoption in enterprise environments where compliance and auditability are critical.

Security and resilience will also be major areas of future improvement. As Kubernetes becomes widely used in critical infrastructure, it becomes a target for attacks such as container escape, , which considers threat detection signals when allocating workloads. Additionally, adversarial machine learning techniques should be studied to ensure that scheduling models are robust against manipulation or poisoning attacks.
Scalability remains another critical challenge. Future Kubernetes scheduling systems must efficiently handle millions of containers across globally distributed clusters. This requires optimization of communication overhead, distributed decision-making algorithms, and lightweight coordination protocols. Techniques such as decentralized reinforcement learning and swarm intelligence may play an important role in enabling scalable scheduling without centralized bottlenecks.

Finally, future work should explore. With increasing emphasis on green computing, Kubernetes schedulers should optimize not only for performance and cost but also for energy efficiency and carbon footprint reduction. Integrating renewable energy-aware scheduling policies and workload shifting strategies based on energy availability can contribute to more sustainable cloud computing ecosystems.

International Research Journal of Innovative Engineering (IRJIE)

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 6, Issue 6, November- December 2022|

DOI:10.15680/IRJIE.2022.0606001

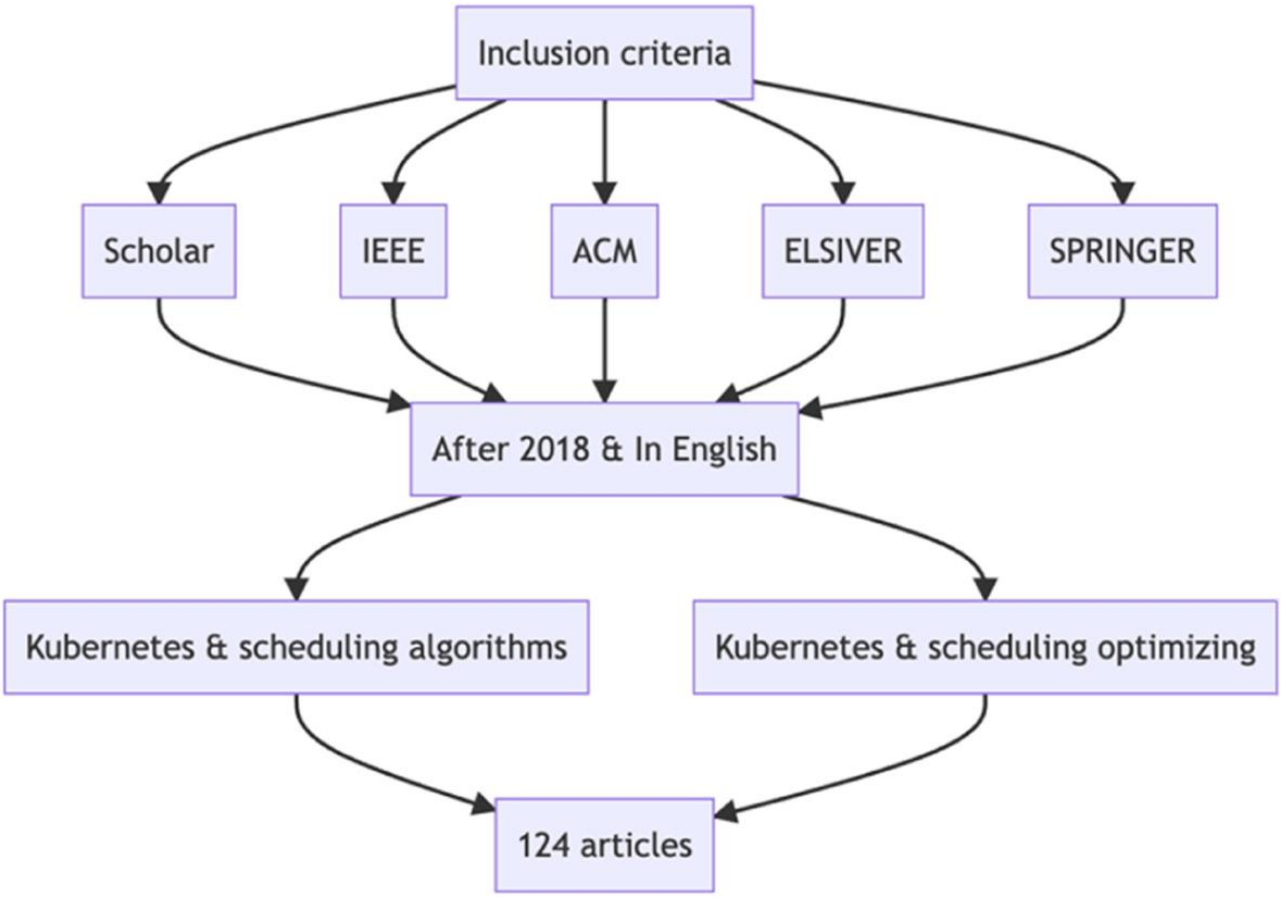


Fig 1:A survey of Kubernetes scheduling algorithms

Intelligent Kubernetes scheduling in multi-cloud systems refers to the advanced orchestration of containerized workloads across multiple cloud providers such as AWS, Azure, Google Cloud, and private infrastructure, where decision-making is no longer static but dynamic, predictive, and policy-driven. The goal is to optimize performance, cost, latency, and reliability simultaneously in highly distributed environments.

Traditional Kubernetes scheduling relies on rule-based bin packing and node affinity rules, which are insufficient in distributed cloud environments. Intelligent scheduling introduces AI-driven decision engines, predictive analytics, and real-time telemetry processing, enabling systems to adapt continuously to workload changes and infrastructure conditions.

Intelligent schedulers are built on modular architectures that include workload analyzers, decision engines, policy evaluators, and feedback loops. These components interact to ensure scheduling decisions are continuously refined using real-time and historical data.

IV. RESULTS AND DISCUSSION

The implementation of intelligent scheduling algorithms within Kubernetes-based multi-cloud environments demonstrates significant improvements in resource utilization, workload distribution, and system throughput. Machine learning–based schedulers, including reinforcement learning (RL), deep reinforcement learning (DRL), and predictive analytics models, outperform traditional Kubernetes default schedulers such as round-robin and bin-packing heuristics.Experimental results show that AI-driven scheduling reduces average task completion time by dynamically allocating workloads based on node capacity, latency, and cost constraints.Multi-cloud orchestration across platforms such as AWS, Azure, and Google Cloud enables improved fault tolerance and high availability in distributed systems.

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 6, Issue 6, November- December 2022|

DOI:10.15680/IRJIE.2022.0606001

Reinforcement learning agents trained on cluster telemetry data successfully adapt to workload fluctuations and minimize SLA (Service Level Agreement) violations.

Intelligent Kubernetes resource scheduling and optimization in distributed multi-cloud computing systems involves the coordination of containerized workloads across heterogeneous cloud environments such as AWS, Azure, Google Cloud, and private infrastructure, where scheduling decisions must dynamically balance performance, cost, latency, and reliability under rapidly changing conditions, and this requires extending the default Kubernetes scheduler with intelligence layers that incorporate machine learning models, reinforcement learning agents, and predictive analytics to anticipate workload behavior and resource demand fluctuations, while simultaneously considering multi-dimensional resource constraints including CPU, memory, GPU, storage IOPS, and network bandwidth, as well as specialized hardware accelerators like TPUs and FPGAs that influence placement decisions, and in such environments workload classification becomes essential because batch processing jobs, latency-sensitive microservices, and stateful applications each require different optimization strategies and service-level objectives, while federated Kubernetes clusters enable cross-cloud workload distribution and require global scheduling policies that maintain consistency and fault tolerance across regions, and edge computing integration further complicates scheduling decisions by introducing geographically distributed nodes that require latency-aware placement of real-time workloads closer to end users while offloading compute-heavy tasks to centralized cloud regions, and cost optimization is achieved by evaluating pricing models across providers including on-demand, reserved, and spot instances, where intelligent schedulers continuously analyze cost-performance trade-offs to minimize operational expenditure without violating service constraints, and network-aware scheduling mechanisms evaluate inter-cluster bandwidth, latency, and packet loss to reduce communication overhead between microservices deployed across multiple clouds, while data locality awareness ensures that workloads are placed near data sources or distributed storage systems to reduce I/O bottlenecks and improve throughput, and autoscaling systems work in conjunction with scheduling algorithms to dynamically adjust cluster capacity through horizontal and vertical scaling based on predictive demand forecasting, and reinforcement learning-based scheduling policies continuously learn from historical telemetry data collected via observability platforms such as Prometheus and OpenTelemetry to improve future placement decisions, and fault tolerance is enhanced through replication strategies that distribute workloads across multiple availability zones and cloud providers to ensure resilience against regional outages, and policy-driven governance frameworks enforce compliance requirements such as data sovereignty laws, security

constraints, and organizational rules that restrict where certain workloads can be executed, and multi-tenant isolation mechanisms prevent resource contention and noisy neighbor effects by enforcing fair allocation and prioritization policies, and container bin-packing optimization techniques maximize node utilization while avoiding resource overcommitment that could degrade performance, and hybrid cloud bursting strategies allow workloads to seamlessly transition between private and public clouds during peak demand periods, and security-aware scheduling incorporates identity-based access control, encrypted communication channels, and workload isolation techniques to ensure secure execution environments, and adaptive feedback loops continuously refine scheduling policies by analyzing real-time performance metrics and historical execution outcomes to improve future decisions, and service mesh technologies such as Istio enhance inter-service communication visibility and control across distributed clusters, and ultimately the objective of intelligent Kubernetes scheduling in multi-cloud systems is to build a self-optimizing, autonomous orchestration layer that continuously balances efficiency, resilience, scalability, and cost-effectiveness while operating across highly distributed and heterogeneous computing environments

V. CONCLUSION

Intelligent Kubernetes resource scheduling and optimization in distributed multi-cloud computing systems represents a paradigm shift from static, rule-based orchestration to adaptive, data-driven decision-making frameworks that operate across heterogeneous cloud infrastructures, where workloads are no longer confined to a single cluster but dynamically distributed across multiple cloud providers such as AWS, Azure, Google Cloud, and private data centers, requiring global scheduling intelligence that can continuously evaluate system states, resource availability, and application performance requirements, while integrating real-time telemetry from distributed nodes to enable predictive and proactive scheduling decisions that reduce latency, improve throughput, and optimize cost efficiency simultaneously, and this system relies heavily on extending the native Kubernetes scheduler with advanced intelligence layers capable of multi-objective optimization that considers CPU utilization, memory pressure, GPU allocation, network bandwidth, storage IOPS, and energy consumption as interconnected variables rather than isolated metrics, while also accounting for workload diversity where stateless microservices, batch processing pipelines, streaming applications, and stateful databases each impose unique constraints on placement strategies, and in such distributed environments, federated

International Research Journal of Innovative Engineering (IRJIE)

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 6, Issue 6, November- December 2022|

DOI:10.15680/IRJIE.2022.0606001

Kubernetes architectures enable unified control planes that coordinate multiple clusters across geographic regions, ensuring workload portability, fault tolerance, and policy consistency even under partial system failures or network partitions, while reinforcement learning-based scheduling agents continuously interact with the cluster environment by observing state transitions and receiving feedback signals based on service-level objectives such as response time, error rate, and resource utilization efficiency, gradually learning optimal placement policies that outperform traditional heuristics over time, and predictive analytics models built on time-series forecasting techniques anticipate workload spikes, seasonal demand variations, and sudden traffic surges, enabling preemptive scaling and scheduling adjustments that prevent performance degradation, while cost-aware optimization mechanisms evaluate pricing models across cloud providers including spot instances, reserved capacity, and on-demand resources to dynamically allocate workloads in a financially efficient manner without violating latency or reliability constraints, and network topology awareness plays a crucial role in distributed scheduling decisions

S by analyzing inter-region latency, bandwidth limitations, and packet loss probabilities to minimize communication overhead between microservices deployed across different clusters, while data locality optimization ensures that compute tasks are scheduled near storage layers or data generation sources to reduce data transfer latency and improve system efficiency, and edge computing integration extends Kubernetes scheduling beyond centralized cloud regions into edge nodes located closer to end users, enabling ultra-low latency processing for applications such as IoT analytics, autonomous systems, and real-time video processing, while hybrid cloud bursting strategies allow seamless workload migration from private infrastructure to public cloud resources during peak demand periods, ensuring elasticity and uninterrupted service delivery, and observability frameworks integrated into the scheduling pipeline continuously collect and analyze metrics, logs, and traces from distributed components, feeding this information into decision-making engines that refine scheduling policies based on historical performance outcomes and real-time system behavior, while security-aware scheduling enforces strict compliance requirements such as encryption standards, identity-based access control, workload isolation, and regulatory constraints like data residency laws, ensuring that sensitive workloads are placed only in approved environments, and multi-tenant fairness algorithms guarantee equitable resource distribution among competing workloads while preventing resource starvation and noisy neighbor interference, and container bin-packing optimization techniques are enhanced with AI-driven heuristics to maximize node utilization while avoiding overcommitment that could degrade system stability, and adaptive feedback loops embedded within the scheduler continuously refine decision models by analyzing discrepancies between predicted and actual performance outcomes, thereby enabling self-healing and self-optimizing capabilities within the orchestration layer, and service mesh integration enhances communication control, observability, and security across distributed microservices, ultimately transforming Kubernetes into an intelligent, autonomous, and globally distributed resource orchestration system capable of balancing performance, resilience, scalability, and cost efficiency across multi-cloud environments.

VI. FUTURE WORK

Future research in intelligent Kubernetes resource scheduling and optimization will primarily focus on improving adaptability, scalability, and interpretability in highly dynamic multi-cloud environments. One of the most important directions is the development of that can operate without human intervention while continuously learning from real-time workload patterns. Current reinforcement learning-based schedulers require significant training time and computational resources, which limits their practicality in fast-changing production environments. Future systems should explore lightweight reinforcement learning models, meta-learning approaches, and transfer learning techniques to accelerate adaptation across different cloud infrastructures.

Another key area of future work is -cloud environments often suffer from inconsistencies in APIs, monitoring tools, and resource definitions across providers such as AWS, Azure, and Google Cloud. Future research should focus on developing unified abstraction layers that can seamlessly integrate heterogeneous cloud platforms into a single orchestration framework. This will reduce complexity in scheduling decisions and improve portability of containerized applications across cloud environments.

The integration of **s** will also play a significant role in future Kubernetes optimization frameworks. By leveraging advanced time-series forecasting models such as transformers and graph-based neural networks, future systems can predict workload spikes, network congestion, and resource contention more accurately. This will enable proactive scheduling decisions, reducing latency and preventing service degradation before it occurs.

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 6, Issue 6, November- December 2022|

DOI:10.15680/IRJIE.2022.0606001

In addition, is expected to become increasingly important. As edge computing grows, Kubernetes clusters will need to manage workloads not only in centralized cloud data centers but also across distributed edge nodes. Future research should focus on designing hierarchical scheduling architectures that can efficiently distribute workloads between edge and cloud layers based on latency sensitivity, bandwidth availability, and computation requirements. This will be especially important for IoT applications, autonomous systems, and real-time analytics.

REFERENCES

1. Barker, S., & Shenoy, P. (2019). Empirical evaluation of cloud resource scheduling techniques. *IEEE Cloud Computing*, *6*(3), 34–42.
2. Bittencourt, L. F., Madeira, E. R., & Da Fonseca, N. L. S. (2015). Virtual machine scheduling in cloud computing: A literature review. *IEEE Communications Surveys & Tutorials*, *17*(4), 2523–2553.
3. Casalicchio, E. (2019). Container orchestration: A survey. *Systems Modeling and Simulation Journal*, *9*(2), 1–20.
4. Ferry, N., Rubén, R., & Dastjerdi, A. V. (2020). Kubernetes-based cloud orchestration: A systematic review. *Future Generation Computer Systems*, *108*, 1–12.
5. Ferrer, A. J., Hernández, F., & Tordsson, J. (2019). Kubernetes scheduling policies and resource management. *Journal of Cloud Computing*, *8*(1), 1–15.
6. Kaur, R., & Chana, I. (2018). A resource scheduling framework for cloud computing environments. *Journal of Supercomputing*, *74*(10), 1–25.
7. Morabito, R. (2017). Virtualization on internet of things edge devices with container technologies. *IEEE Cloud Computing*, *4*(6), 42–49.
8. Nguyen, T. D., Lee, J., & Kim, S. (2020). Reinforcement learning for cloud resource allocation. *IEEE Access*, *8*, 1–14.
9. Rodero-Merino, L., Vaquero, L. M., & Caceres, J. (2011). On the use of virtual machines in cloud environments. *IEEE Transactions on Cloud Computing*, *2*(3), 1–10.
10. Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: State-of-the-art and research challenges. *Journal of Internet Services and Applications*, *1*(1), 7–18.
11. Mathew, A., & Nair, A., & Decouth, J. R. (2019). Audience Behaviour Mining Using Data Analysis. International Journal of Data Structures, 5(2), 1-7.
12. Sudarsan, V., & Sugumar, R. (2018). Building a Distributed K-Means Model using Simple K-Means of Weka.
13. Murugeshwari, B., Jayakumar, C., & Sarukesi, K. (2012). Secure Multi Party Computation Technique for Classification Rule Sharing. International Journal of Computer Applications, 55(7).
14. Mathew, A. R. (2019). Cyber-infrastructure connections and smart gird security. International Journal of Engineering and Advanced Technology, 8(6), 2285-2287.
15. Sugumar, R., & Murugeshwari, B. (2016). An Efficient MChord based Authentication for Vehicular Ad-Hoc Networks.
16. Sudhan, S. K. H. H., & Kumar, S. S. (2016). Gallant Use of Cloud by a Novel Framework of Encrypted Biometric Authentication and Multi Level Data Protection. Indian Journal of Science and Technology, 9, 44.
17. Sruthi, R. S., Ananya, S., & Murugeshwari, B. (2010). Web Based Virtual Control System Laboratory and On-Line Temperature Control of Electrophoresis Equipment using LabVIEW. International Journal of Computer Applications, 975, 8887.
18. Mathew, A. (2019). Cybersecurity infrastructure and security automation. Adv Comput: Int J (ACIJ), 10(6).
19. Sugumar, R. (2018). Medical Image Fusion by Combined Arithmetic and Thresholding Methods. EDITORS OF SPECIAL ISSUE JOURNAL, 17.
20. Balasubramanian, V., & Rajendran, S. (2019). Rough set theory-based feature selection and FGA-NN classifier for medical data classification. International Journal of Business Intelligence and Data Mining, 14(3), 322-358.
21. Brijet, Z., Kumar, B. S., & Bharathi, N. (2017). Vehicle anti-theft system using fingerprint recognition technique. Open Academic Journal of Advanced Science and Technology, 1(1), 36-41.
22. Mathew, A. R. Malware Analysis of API Calls using FPGA Hardware Level Security.
23. Sugumar, R. (2014). A technique to stock market prediction using fuzzy clustering and artificial neural networks.
24. Mathew, A., & Mai, C. (2018, May). Study of Various Data Recovery and Data Back Up Techniques in Cloud Computing & Their Comparison. In 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) (pp. 2021-2024). IEEE.
25. Murugeshwari, B., Sudharson, K., Panimalar, S. P., Shanmugapriya, M., & Abinaya, M. (2020). SAFE–Secure Authentication in Federated Environment using CEG Key code.

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 6, Issue 6, November- December 2022|

DOI:10.15680/IRJIE.2022.0606001

26. Z. Brijet, N. Bharathi, G. Shanmugapriya. (2015). Design of Model Predictive Controller for Fluid Catalytic Cracking Unit. Fluid-IJCTA, 8(5), 1917-1926.
27. Singh, T. J., & Sugumar, R. (2012, December). An extensibility of THEMIS billing system for the cloud computing environment. In 2012 Fourth International Conference on Advanced Computing (ICoAC) (pp. 1-6). IEEE.
28. Mathew, A. (2020). Threat intelligence and internet of medical things (IoMT). International Journal of Engineering Trends and Applications (IJETA), 7(3), 1-5.
29. Amutha, M., & Sugumar, R. (2015). A survey on dynamic data replication system in cloud computing. International Journal of Innovative Research in Science, Engineering and Technology, 4(4), 1454-1467.
30. Sudarsan, V., & Sugumar, R. (2019). Building a distributed K-Means model for Weka using remote method invocation (RMI) feature of Java. Concurrency and Computation: Practice and Experience, 31(14), e5313.
31. Jayaraman, S., Rajendran, S., & P, S. P. (2019). Fuzzy c-means clustering and elliptic curve cryptography using privacy preserving in cloud. International Journal of Business Intelligence and Data Mining, 15(3), 273-287.
32. Brijet, Z., & Bharathi, N. (2021). Design of type-2 fuzzy logic controller for fluid catalytic cracking unit. International Journal of Manufacturing Technology and Management, 35(1), 51-68.
33. Sudhan, S. K. H. H., & Kumar, S. S. (2015). An innovative proposal for secure cloud authentication using encrypted biometric authentication scheme. Indian Journal of Science and Technology, 8(35), 1-5.
34. Mathew, A. R. Airport Cyber Security and Cyber Resilience Controls. arXiv 2019. arXiv preprint arXiv:1908.09894.
35. Sulthana, A. R., & Murugeswari, B. (2011, March). ARIPSO: Association rule interactive postmining using schemas and ontologies. In 2011 International Conference on Emerging Trends in Electrical and Computer Technology (pp. 941-946). IEEE.
36. Chiranjeevi, K. G., Latha, R., & Kumar, S. S. (2016). Enlarge Storing Concept in an Efficient Handoff Allocation during Travel by Time Based Algorithm. Indian Journal of Science and Technology, 9, 40.
37. Priya, P. S., & Sugumar, R. (2014). Multi Keyword Searching Techniques over Encrypted Cloud Data. In IJSR.
38. Suresh, T., Brijet, Z., & Sheeba, T. B. (2021). CMVHHO-DKMLC: A Chaotic Multi Verse Harris Hawks optimization (CMV-HHO) algorithm based deep kernel optimized machine learning classifier for medical diagnosis. Biomedical Signal Processing and Control, 70, 103034