INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 1, No 27 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-706 PAGE NO: 018-030

A MASTER NODE CONSENSUS APPROACH FOR ENHANCING LATENCY PERFORMANCE IN DISTRIBUTED CLOUD COMPUTING ENVIRONMENT

Ossai, Iyaye Mine
Information and Communication Technology Centre, Rivers State University, Port Harcourt.

Davies Isobo Nelson
Department of Computer Science, Rivers State University, Port-Harcourt, Nigeria.

Mini, Amobi Henry
Information and Communication Technology Centre, Rivers State University, Port Harcourt

Abstract

Efficient master node selection is critical for achieving low-latency performance in distributed cloud computing environments. Traditional election mechanisms often overlook node resource variability, leading to bottlenecks, increased execution times, and SLA violations. This paper introduces the Master Node Consensus Approach (MNCA), which integrates a Genetic Algorithm-based resource-profiling model with a fault-tolerant master-candidate failover mechanism to enhance latency performance in distributed cloud infrastructures. MNCA evaluates node fitness using CPU capacity, memory availability, bandwidth, and throughput, ensuring that only high-capacity nodes are selected as masters. The model was implemented in CloudSim Plus and benchmarked against the Heterogeneous Earliest Finish Time (HEFT) and Balanced Load Allocation (BLA) algorithms. Under node scaling from 20 to 100 hosts, MNCA reduced execution time from 5900 ms to 1800 ms, outperforming HEFT (11000–3800 ms) and BLA (7000–2800 ms). MNCA achieved the lowest SLA violation rates, decreasing from 200 to 130 violations, compared to HEFT (250–150) and BLA (300–180). Memory usage consistently dropped from 54 MB to 33 MB, surpassing HEFT (60–42 MB) and BLA (55–35 MB). Under workload scaling from 200 to 2000 tasks, MNCA maintained the fastest execution times (1500–4100 ms) versus HEFT (1700–11000 ms) and BLA (1600–8000 ms). These results confirm that MNCA enhances latency performance, reduces SLA violations, improves memory efficiency, and provides robust failover behavior in distributed cloud environments. The proposed approach offers a scalable, intelligent solution for optimizing coordination services in modern distributed systems.

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholorly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 1, No 27 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-705 PAGE NO: 006-017

Keywords: Master Node Selection, Genetic Algorithm (GA), Distributed Cloud Computing, Latency Optimization, Consensus Mechanism, Fault Tolerance.

1. Introduction

Cloud computing has emerged as a transformative paradigm in modern information technology, enabling on-demand access to shared computing resources over the internet. As organizations increasingly migrate their operations to cloud environments, the demand for efficient, reliable, and low-latency cloud services continues to grow exponentially. Distributed cloud computing architectures, which leverage multiple interconnected nodes to deliver scalable and resilient services, have become the backbone of contemporary digital infrastructure. However, the distributed nature of these systems introduces significant coordination challenges that directly impact system performance, particularly in terms of latency.

Latency, defined as the time delay between a request initiation and response delivery, remains one of the most critical performance metrics in cloud computing environments. High latency can severely degrade user experience, reduce application responsiveness, and compromise the quality of service (QoS) in real-time applications. Nonetheless, in distributed cloud architectures, latency is influenced by multiple factors including network congestion, geographical distribution of nodes, resource availability, and most critically, the efficiency of inter-node coordination mechanisms.

Central to the effective operation of distributed cloud systems is the concept of master node coordination. The master node serves as the central coordinator responsible for task allocation, resource management, workload distribution, and maintaining system coherence among worker nodes. The selection and performance of the master node significantly influence the overall system efficiency and latency characteristics. A poorly selected master node one with insufficient computational resources, limited network bandwidth, or inadequate reliability can become a bottleneck, leading to increased response times, inefficient resource utilization, and potential system failures.

Despite the critical importance of master node selection, existing approaches often suffer from inadequate node profiling mechanisms. Traditional master node election algorithms frequently rely on simplistic criteria such as node availability or arbitrary identifiers, without comprehensively evaluating the computational capabilities, network performance, and resource status of candidate nodes. This oversight can result in the selection of weak nodes that are ill-equipped to handle coordination responsibilities, thereby introducing latency issues and reducing overall system performance. Furthermore, the dynamic nature of cloud environments characterized by fluctuating workloads, node failures, and varying resource availability necessitates adaptive and intelligent master node selection

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholorly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 1, No 27 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-705 PAGE NO: 006-017

mechanisms that can respond to changing conditions in real-time.

The problem is further compounded by the inherent challenges of achieving consensus in distributed systems. Consensus mechanisms must balance multiple objectives such as ensuring fault tolerance, maintaining consistency, minimizing communication overhead, and achieving rapid decision-making. Traditional consensus protocols, while theoretically sound, often exhibit performance limitations in large-scale distributed cloud environments where network partitions, node failures, and variable latency are common occurrences.

To address these challenges, this research proposes a Master Node Consensus Approach (MNCA) that integrates intelligent node resource profiling with consensus-based master node election. The proposed model leverages genetic algorithm-based resource profiling to comprehensively evaluate participating nodes based on multiple performance criteria including CPU capacity, memory availability, network bandwidth, and historical reliability. By incorporating these metrics into the consensus mechanism, the MNCA model ensures that only nodes with sufficient resources and capabilities are eligible for master node selection, thereby reducing the likelihood of latency-inducing bottlenecks.

2. Related Works

The challenges of latency optimization, master node coordination, and resource allocation in distributed cloud computing have attracted significant research attention in recent years. Recent research has demonstrated the effectiveness of genetic algorithms in optimizing resource allocation and task scheduling in cloud environments. Proposed a hybrid algorithm leveraging genetic algorithms and neural networks to improve scheduling in cloud computing. Their method, which classifies tasks using Neural Network Task Classification (N2TC) and allocates resources through Genetic Algorithm Task Assignment (GATA), achieved significant improvements over state-of-the-art methods: 3.2% reduction in execution time, 13.3% decrease in costs, and 12.1% improvement in response time.

Developed a Growable Genetic Algorithm with Heuristic-based Local Search (GHW) for multi-dimensional resource scheduling in cloud computing. Tested on Microsoft Azure's workload traces, it focused on virtual machine allocation to optimize resource use and energy consumption. The GHW algorithms outperformed traditional NSGA-II and MOEA/D algorithms in convergence rates and optimality, especially with complex multi-dimensional resource coupling. Experimental results indicated better performance in hypervolume-over-time metrics and a higher probability of finding optimal Pareto solutions

Presented an energy-efficient virtual machine scheduling algorithm using genetic algorithms for cloud computing resource management. Their study tackled unbalanced resource loads and high-energy consumption, showing that GA2ND needed fewer

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 1, No 27 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-705 PAGE NO: 006-017

migrations (about 1,000) while achieving better energy efficiency than GA1ST. The results demonstrated that genetic algorithms can effectively minimize energy consumption with fewer virtual machine migrations.

Investigated resource allocation in cloud computing with a hybrid strategy combining Random Forest (RF) and Genetic Algorithm (GA) for virtual machine allocation. Testing with Planet Lab's workload traces, they achieved improved data center resource utilization, energy consumption, and execution time. Findings showed RF outperformed standalone GA, optimizing resource distribution and minimizing power consumption.

Research on consensus algorithms and leader election has focused on improving performance, reducing latency, and enhancing fault tolerance in distributed systems. Studied ways to enhance latency and throughput in ZooKeeper Atomic Broadcast, identifying communication bottlenecks and proposing optimizations to reduce consensus latency. Addressed performance enhancement in blockchain-based IoT data sharing using a lightweight consensus algorithm. Their research introduced Delegated Proof of Stake (DPoS) consensus combined with sharding techniques to improve scalability in blockchain-based IoT networks. Their experimental results indicated that system throughput increased synchronously with test load, addressing the scalability limitations that had previously hindered blockchain applications in IoT contexts. The study demonstrated that lightweight consensus mechanisms could maintain data transparency, integrity, and immutability while significantly improving performance.

Conducted a comprehensive survey on consensus algorithms for distributed wireless networks, comparing current trends and identifying emerging research opportunities. Their work emphasized that consensus algorithms are becoming essential for safe and reliable wireless communication systems, particularly in addressing channel decision-making, authentication problems, message routing, and fault tolerance. The survey highlighted that consensus-based algorithms could improve channel quality, signal strength, energy consumption.

Research on node profiling has progressed in understanding the value of comprehensive node evaluation. Introduced a Resource-Aware Federated Hybrid Profiling approach for edge node selection in federated patient similarity networks, using their Federated Workload Profiling (FWP) model to assess performance metrics like accuracy and execution time. Their findings showed that effective node selection could optimize Quality of Service (QoS), with FWP reducing memory usage and improving resource allocation efficiency while addressing the challenge of profile federation overhead.

Research on latency optimization in distributed systems has examined various aspects. Focused on distributed consensus estimation for networked multi-sensor systems facing hybrid attacks and missing measurements. Their modified consensus estimator combined characteristics of Denial

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholorly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 1, No 27 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-705 PAGE NO: 006-017

of Service and False Data Injection attacks, showing that it could maintain acceptable estimation accuracy despite adverse conditions, while emphasizing the importance of latency when managing compromised nodes.

Recent research on dynamic resource allocation has highlighted AI-driven strategies for optimizing workload distribution and cost efficiency. [20] Compared First-Come-First-Serve (FCFS), Round-Robin Scheduling (RRS), and Genetic Algorithm (GA) methods in cloud data centers, finding that GAs outperformed both FCFS and RRS in scheduling efficiency and workload distribution. Their study pointed out the limitations of traditional algorithms in managing dynamic workloads and resource heterogeneity, often resulting in inefficient resource utilization and increased processing delays.

3. System Design and Methodology

This study employed a Discrete Research Approach (DRA) and an Object-Oriented Design Method (OODM) for modeling the interaction of system components. It is important to note that the dataset utilized for this study is event driven, and it was produced at random.

The architectural design of the proposed MNCA model is captured in Figure 1. It presents an intelligent job-scheduling framework for cloud computing environments using a Genetic Algorithm (GA) integrated with a master-candidate fault-tolerant coordination mechanism. The proposed model operates within a distributed cloud data center composed of heterogeneous compute clusters, aiming to optimize job allocation, enhance load balancing, and maintain service continuity in the presence of node failures.

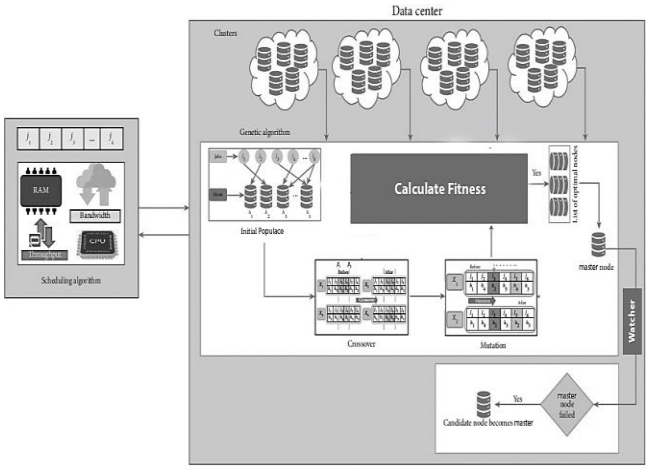

Figure 1: Architectural Design of the Proposed MNCA Model

The MNCA model architecture consists of four major components. These components are the Scheduling Algorithm Module, the Genetic Algorithm (GA) Engine, the Cloud Data Center Clusters, and the Watcher/Master Node Failure Detector. In this study, these components collectively ensure that user jobs are efficiently mapped to available hosts based on resource demands and system constraints.

3.1 Scheduling Algorithm Module

The Scheduling Algorithm Module is responsible for collection and preprocessing of all inputs required by the GA Engine. It captures the job parameters (j_1, j_2, j_3,..., jn),

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 1, No 27 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-705 PAGE NO: 006-017

and the resource attributes such as the CPU capacity, RAM availability, Bandwidth, and the Throughput metrics.

The scheduler then encapsulates these parameters into a formal representation suitable for evolutionary optimization. In this study, the representation defines the chromosome structure, where each gene indicates a mapping between a job and a specific host.

3.2 Genetic Algorithm (Ga) Engine

For this study, the GA engine forms the core of the system and performs heuristic optimization to derive the optimal job-to-host allocation. In the proposed system, the GA comprises several sequential processes such as the initial population generation, fitness evaluation, selection and crossover, mutation, and selection of optimal node list.

In the Initial Population Generation, the proposed system begins by generating an initial population of candidate solutions. Here, each candidate solution (chromosome) represents a potential allocation of jobs to cloud hosts. This can be presented mathematically as follows:

$$X\_k=\{(j\_1\rightarrow\lambda\_a),(j\_2\rightarrow\lambda\_b),\ldots,(j\_n\rightarrow\lambda\_z)\} \quad (1)$$

Where λ represents an available host in the data canter.

Note, for this study, the random initialization process ensures diversity and also enables the GA to explore a broad search space. Further, for the Fitness Evaluation, each chromosome is evaluated by a fitness function, which quantifies the quality of the allocation based on:

1. Load Balancing across hosts
2. Host CPU, RAM, and network usage
3. Job execution time and expected completion time
4. Minimization of resource contention
5. Maximization of throughput
6. Reduction of makespan (overall completion time)

Here, the fitness function returns a scalar value, which determine whether the allocation is optimal or suboptimal. Note, chromosomes with higher fitness values are selected as candidates for further evolution.

Furthermore, the proposed system applies a selection mechanism (tournament selection) to identify the fittest chromosomes. Pairs of selected chromosomes undergo crossover, where genetic material is exchanged to create improved offspring. This is represented as:

$$\text{Parent 1: } [\lambda\_1,\lambda\_4,\lambda\_3,\lambda\_2] \quad (2)$$
$$\text{Parent 2: } [\lambda\_3,\lambda\_2,\lambda\_1,\lambda\_4] \quad (3)$$

Equation 2 and 3 is then combined to form a new offspring that inherits attributes from both parents. Here, the crossover ensures exploration of the solution space and the formation of new high-quality solutions.

For this study, to maintain genetic diversity and avoid premature convergence, the system applies mutation to selected offspring. Here, it randomly alters the job-to-host mapping of one or more genes. This process is represented as follows:

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 1, No 27 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-705 PAGE NO: 006-017

Before mutation: [j_1→λ_1, j_2→λ_3, j_3→λ_2] (4)
After mutation: [j_1→λ_1, j_2→λ_1, j_3→λ_2] (5)

For the selection of optimal node process in this study, the random perturbation enables the GA engine to explore new regions of the search space, which may contain better solutions. Finally, after iterative cycles of crossover and mutation, the GA converges to an optimal or near-optimal solution.
The resulting chromosome represents the follow:

I. The best host configuration, and
II. The optimal execution plan for all jobs.

This information is then passed to the master node, which is responsible of coordinating job deployment across the data center.

3.3 Cloud Data Center Cluster Layer

The upper section of the Figure 1. represents a distributed cloud data center consisting of multiple clusters. Here, each cluster contains a pool of heterogeneous computing nodes with varying resource capacities.

The GA engine then interacts with these clusters to retrieve real-time resource availability and performance metrics. The optimal job-host mappings derived by the GA are then deployed across these clusters for execution.

Watcher and Master Node Coordination Mechanism

To maintain high availability and reliability, the system employs a Watcher module that continuously monitors the status of the master node. This usually involves two scenarios that exist. The scenarios are:

I. Master Node Active: If the master node is healthy, it processes the job allocation list and dispatches jobs to the selected hosts.
II. Master Node Failure: If the watcher detects a failure, a candidate node is selected from the optimal node list. Further, this selected candidate node will then assume the role of a master and the system will continue to operate seamlessly without service interruption thereby ensuring a fault-tolerant environment that supports or enhances reliable cloud operations.

4. Experiments and Results

This section presents the empirical evaluation of the proposed Master Node Consensus Approach (MNCA). All experiments were conducted using CloudSim Plus and executed under identical conditions for fair comparison with two state-of-the-art scheduling algorithms: Heterogeneous Earliest Finish Time (HEFT) and Balanced Load Allocation (BLA). Four major experiments were performed, focusing on execution time, Service Level Agreement (SLA) violations, memory utilization, and scalability under increasing workloads.

4.1 Experiment 1: Performance Evaluation with Increasing Number of Hosts

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 1, No 27 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-705 PAGE NO: 006-017

The first experiment examined how MNCA responds to increasing computational resources while keeping workload constant. The number of available nodes/hosts was varied from 20 to 100, execution time was recorded for all algorithms, and the results are tabulated in Table 1.

Table 1: Performance Evaluation through Node Increment

Nodes	MNCA	HEFT	BLA
20	5900	11000	7000
40	4000	7000	4500
60	3000	6000	4000
80	2000	4000	3000
100	1800	3800	2800

Across all node configurations, MNCA consistently achieved the lowest execution time, outperforming both BLA and HEFT. For example, at 20 nodes, MNCA recorded 5900 ms as compared to 7000 ms (BLA) and 11000 ms (HEFT).

Further, at 80 nodes, MNCA achieved 2000 ms, while BLA and HEFT produced 3000 ms and 4000 ms respectively. However, at 100 nodes, MNCA dropped to 1800 ms, reaffirming its superior scalability. The performance rate is visualized in Figure 2.

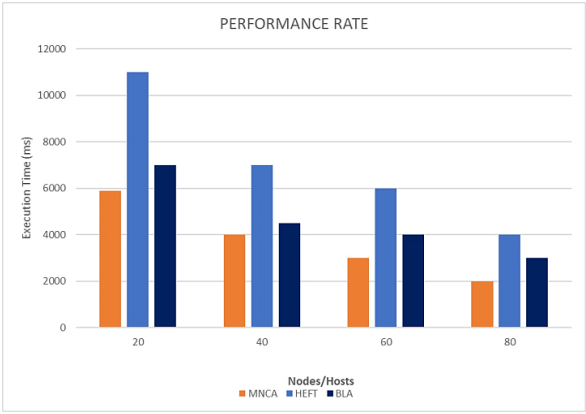

Figure 2: Performance Rate When Node is increased

These results indicate that MNCA benefits more from increased cluster size due to its optimized node selection and load-balancing strategy, validating its efficiency in heterogeneous cloud environments.

4.2 Experiment 2: SLA Violation Analysis under Host Scaling

The second experiment evaluated how host scaling influences SLA adherence. SLA violations were measured for each algorithm across 20-100 hosts, and the results are captured in Table 2.

Table 2: Performance Evaluation through Sla Violation

Nodes	MNCA	HEFT	BLA
20	200	250	300
40	180	220	280
60	160	200	240

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 1, No 27 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-705 PAGE NO: 006-017

80	140	160	200
100	130	150	180

From Table 2, the proposed MNCA demonstrated the lowest SLA violation rates across all host configurations. For instance, at 20 hosts, MNCA recorded 200 violations, compared to 250 (HEFT) and 300 (BLA). Further, at 80 host, MNCA violations reduced to 140, while HEFT and BLA recorded 160 and 200, respectively. Figure 3. Show the visual representation of the SLA violation.

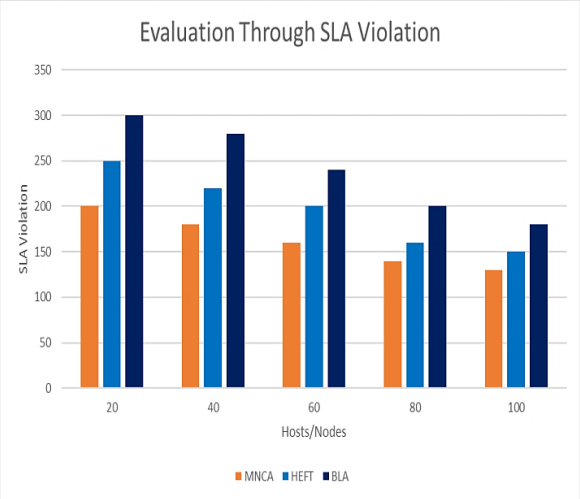

Figure 3: Service Level Agreement Violation

These results highlight MNCA's strong adaptability in maintaining service reliability as node availability increases.

4.3 Memory Utilization Analysis

Memory usage was evaluated to determine how efficiently MNCA handles resource consumption during scheduling operations. The outcome of this evaluation is tabulated in Table 3.

Table 3: Memory Utilization in Experiment 1

Hosts/Node	MNCA	HEFT	BLA
20	54	60	55
40	43	58	54
60	40	56	46
80	36	47	42
100	33	42	35

As shown in Table 3, the proposed MNCA consistently consumed less memory compared to HEFT and BLA. Nevertheless, the notable examples include the following:

I. At 20 hosts, MNCA used 54 MB, compared to 60 MB (HEFT) and 55 MB (BLA).
II. At 80 hosts, MNCA consumed 36 MB, significantly lower than HEFT (47 MB) and BLA (42 MB).

The declining memory usage as hosts increase shows that MNCA's resource-profiling mechanism allocates tasks more efficiently, reducing overhead. Figure 4 captures the bar chart representation of the various memory utilized.

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 1, No 27 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-705 PAGE NO: 006-017

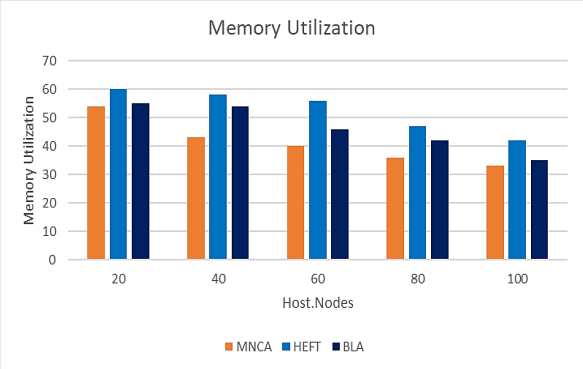

Figure 4: Memory Utilization

4.4 Experiment 3: Performance under Job/Task Increment

In this study, the third experiment evaluated scalability under increasing workloads while maintaining 50 fixed hosts. Task counts increased from 200 to 2000, and the results are tabulated in Table 4.

Table 4: Performance Evaluation through Job or Task Increment

Jobs/Tasks	MNCA	HEFT	BLA
200	1500	1700	1600
400	1700	2000	1800
600	1800	2800	1900
800	2000	4000	2500
1000	2500	5000	3500
1200	3000	6000	4000
1400	3700	7000	5000
1600	3900	8000	6000
1800	4000	9000	7000
2000	4100	11000	8000

Again, the proposed MNCA showed superior execution performance (Table 4.4). At 200 tasks, MNCA achieved 1500 ms, outperforming 1600 ms (BLA) and 1700 ms (HEFT). At 800 tasks, MNCA achieved 2000 ms while HEFT and BLA reached 4000 ms and 2500 ms respectively. At 2000 tasks, MNCA recorded 4100 ms, compared to 8000 ms (HEFT) and 11000 ms (BLA). Figure 5 shows the visual representation of the various execution times as the job or task increases.

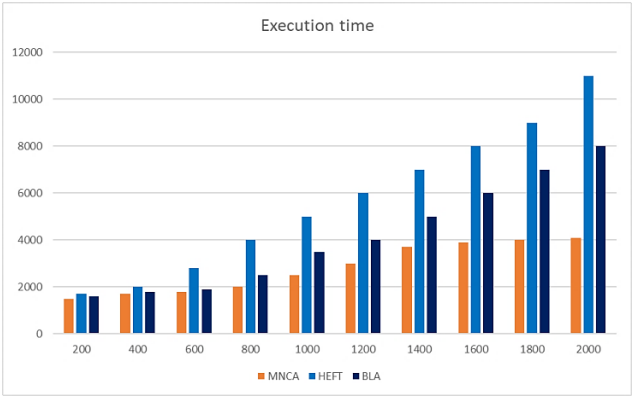

Figure 5: Execution Time When Job/Task Increase

This experiment demonstrates MNCA's capacity to handle high workloads without significant degradation, thanks to its GA-based profiling and optimized job-host mapping. In conclusion, the proposed MNCA consistently outperformed the benchmark algorithms across all experiments.

Conclusion

This paper presents the Master Node Consensus Approach (MNCA), an innovative framework aimed at optimizing the selection

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholorly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 1, No 27 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-705 PAGE NO: 006-017

and coordination of master nodes in distributed cloud computing environments. By integrating Genetic Algorithm-based resource profiling with a fault-tolerant master-candidate failover mechanism, MNCA addresses critical shortcomings of traditional election methods that often overlook variations in node resources and system dynamics.

Experimental evaluations conducted with CloudSim Plus showcased MNCA's exceptional performance. In node-scaling scenarios (20 to 100 hosts), MNCA reduced execution times from 5900 ms to 1800 ms, significantly surpassing HEFT and BLA algorithms. It also recorded the lowest SLA violation rates, decreasing from 200 to 130, while improving memory efficiency from 54 MB to 33 MB. Under workload, scaling conditions (200 to 2000 tasks), MNCA maintained optimal execution times between 1500 ms and 4100 ms, demonstrating robust scalability.

Key contributions of this research include: (1) a comprehensive node fitness evaluation model assessing CPU capacity, memory availability, bandwidth, and throughput; (2) an adaptive consensus mechanism that enables only high-capacity nodes to take on master roles; and (3) a fault-tolerant failover strategy that ensures service continuity during node failures. These features collectively enhance latency, mitigate resource bottlenecks, and boost overall reliability in diverse cloud environments.

Looking ahead, future work will explore several promising directions: incorporating machine learning for predictive resource profiling to facilitate proactive master node selection based on anticipated workloads; extending the framework to support multi-master architectures for improved fault tolerance; investigating the effects of geographic distribution and network topology on consensus latency; and validating the model's practical applicability through real-world deployments across various cloud platforms.

In summary, MNCA emerges as a scalable and intelligent solution for enhancing coordination services in distributed cloud systems, paving the way for greater performance, reliability, and efficiency in future cloud-computing infrastructures.

References

1. P. Sharma, R. Sharma, and K. Bardwaj, "Cloud Computing in Everyday Life: Revolutionizing How We Live, Work, and Connect," in *Driving Transformative Technology Trends With Cloud Computing*, ed: IGI Global, 2024, pp. 43-53.
2. N. H. Anh, "Hybrid Cloud Migration Strategies: Balancing Flexibility, Security, and Cost in a Multi-Cloud Environment," *Transactions on Machine Learning, Artificial Intelligence, and Advanced Intelligent Systems,* vol. 14, pp. 14-26, 2024.
3. [O. Oloruntoba, "Architecting Resilient Multi-Cloud Database Systems: Distributed Ledger Technology, Fault

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholorly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 1, No 27 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-705 PAGE NO: 006-017

Tolerance, and Cross-Platform Synchronization," *International Journal of Research Publication and Reviews,* vol. 6, pp. 2358-2376, 2025.

4. S. Shukla, M. F. Hassan, D. C. Tran, R. Akbar, I. V. Paputungan, and M. K. Khan, "Improving latency in Internet-of-Things and cloud computing for real-time data transmission: a systematic literature review (SLR)," *Cluster Computing,* vol. 26, pp. 2657-2680, 2023.
5. C. S. M. Babou, Y. Owada, M. Inoue, K. Takizawa, and T. Kuri, "Distributed Edge Cloud Proposal Based on VNF/SDN Environment," *IEEE Access,* vol. 12, pp. 124619-124635, 2024.
6. F. Cuares and J. I. Teleron, "Harmony in Nodes: Exploring Efficiency and Resilience in Distributed Systems," *Engineering and Technology Journal,* vol. 9, pp. 4127-4136, 2024.
7. Aditi, V. K. Prasad, V. C. Gerogiannis, A. Kanavos, D. Dansana, and B. Acharya, "Utilizing convolutional neural networks for resource allocation bottleneck analysis in cloud ecosystems," *Cluster Computing,* vol. 28, p. 22, 2025.
8. R. Ahmad, W. Alhasan, R. Wazirali, and N. Aleisa, "Optimization algorithms for wireless sensor networks node localization: An overview," *IEEE Access,* vol. 12, pp. 50459-50488, 2024.
9. M. M. Yakubu, F. B. Hassan, K. U. Danyaro, A. Z. Junejo, M. Siraj, S. Yahaya, *et al.*, "A Systematic Literature Review on Blockchain Consensus Mechanisms' Security: Applications and Open Challenges," *Computer Systems Science & Engineering,* vol. 48, 2024.
10. P. Mittal, D. S. Kumar, and D. S. Sharma, "Revolutionizing Cloud-Based Task Scheduling: A Novel Hybrid Algorithm for Optimal Resource Allocation and Efficiency in Contemporary Networked Systems," *International Journal of Computing and Digital Systems,* vol. 15, pp. 1551-1563, 2024.
11. M. Manavi, Y. Zhang, and G. Chen, "Resource allocation in cloud computing using genetic algorithm and neural network," in *2023 IEEE 8th International Conference on Smart Cloud (SmartCloud)*, 2023, pp. 25-32.
12. G. Zhou, W. Tian, R. Buyya, R. Xue, and L. Song, "Deep reinforcement learning-based methods for resource scheduling in cloud computing: A review and future directions," *Artificial Intelligence Review,* vol. 57, p. 124, 2024.
13. F. Shi, "A genetic algorithm-based virtual machine scheduling algorithm for energy-efficient resource management in cloud computing," *Concurrency and Computation: Practice and Experience,* vol. 36, p. e8207, 2024.
14. G. Senthilkumar, K. Tamilarasi, N. Velmurugan, and J. Periasamy, "Resource allocation in cloud computing," *Journal of Advances in*

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012
www.ijcsjournal.com Volume 14, Issue 1, No 27 2026 ISSN: 2348-6600
REFERENCE ID: IJCS-705 PAGE NO: 006-017

Information Technology, vol. 14, pp. 1063-1072, 2023.

15. H. Li, C. Lu, and C. D. Gill, "Rt-zookeeper: Taming the recovery latency of a coordination service," *ACM Transactions on Embedded Computing Systems (TECS),* vol. 20, pp. 1-22, 2021.
16. E. U. Haque, W. Abbasi, A. Almogren, J. Choi, A. Altameem, A. U. Rehman, *et al.*, "Performance enhancement in blockchain based IoT data sharing using lightweight consensus algorithm," *Scientific Reports,* vol. 14, p. 26561, 2024.
17. J. C. Ji, C. T. Lam, and B. Ng, "A survey on consensus algorithms for distributed wireless networks," in *The International Conference Optoelectronic Information and Optical Engineering (OIOE2024)*, 2025, pp. 294-303.
18. A. N. Navaz, H. T. E. Kassabi, M. A. Serhani, and E. S. Barka, "Resource-Aware Federated Hybrid Profiling for Edge Node Selection in Federated Patient Similarity Network," *Applied Sciences,* vol. 13, p. 13114, 2023.
19. Z. Cheng, G. Chen, X.-M. Li, and H. Ren, "Consensus-Based Power System State Estimation Algorithm Under Collaborative Attack," *Sensors,* vol. 24, p. 6886, 2024.
20. S. Bhatnagar and R. Mahant, "Dynamic Resource Allocation in Cloud Computing Environments: AI-Driven Approaches for Optimizing Workload Distribution and Cost Efficiency," in *International Conference on Paradigms of Communication, Computing and Data Analytics*, 2025, pp. 313-328.