Babylonian Journal of Artificial Intelligence
Vol. **(2025)**, 2025, **pp**. 64–71
DOI: https://doi.org/10.58496/BJAI/2025/006 ;ISSN: 2958-6453
https://mesopotamian.press/journals/index.php/BJAI

Research Article

Optimizing Cloud Computing: Balancing Cost, Reliability, and Energy Efficiency

Raed A. Hasan[1,*], Teba Majed Hameed[2]

[1] *Renewable energy Research Unit , Northen Technical University, Mosul, Iraq.*

[2] *Institut des hautes études commerciales de Sousse, University of Sousse, Tunisia.*

ARTICLE INFO

Article History
Received 01 Feb 2025
Revised: 25 Feb 2025
Accepted 28 Mar 2025
Published 15 Apr 2025

Keywords
Cloud Computing
Resource Allocation
Fault Tolerance
Energy Efficiency
Service Provisioning

ABSTRACT

Cloud computing is such a revolution concerning the IT world offering computing as services capable of diminishing operational costs and complications. Recently, these service models, ranging from IaaS, PaaS, and SaaS, and deployment models in private, public, and hybrid clouds, offer users almost unlimited computing and storage capabilities on a pay-per-use basis. This elasticity of cloud systems makes it very easy to dynamically provision and de-provision resources to cater to very different needs. This facility has led to its widespread use within domains such as social networking, defense, scientific computing, financial services, and medical. IDG Communications has now announced that 73% of corporations are currently utilizing clouds, with a further 17% in the process of implementing. Service abstraction to increase usability raises yet a fresh set of issues in terms of operational costs, reliability, energy efficiency, and security. Especially in cases where the framework is applicable to critical ventures, as exhibited just a while back by Knight Capital in 2013, system failures may have serious financial and credibility repercussions. Fault tolerance strategies through resource redundancy increase the cost of downtime risk but lower energy consumption, hence less cost and less environmentally unfriendly; they affect profit. The bulk of the operational expense in data centers is associated with the use of energy, whereby the use of energy is environmentally unfriendly and poses environmental concerns; clouds are forecasted to contribute to 5.5% of carbon emissions globally by 2025. Balancing energy efficiency and reliability will require novel optimization approaches for today's and future cloud computing systems with robust fault tolerance.

1. INTRODUCTION

Cloud computing is a new-comer paradigm that gives everything as a service to people as end users and makes businesses more capable in terms of cutting operational cost and complex IT setups. Various service models by cloud service providers deliver computing resources to the consumers, such as IaaS, PaaS, and SaaS, apart from different deployment models like private, public, or hybrid clouds, allowing users to have access to almost unlimited computing and storage resources on a payper-usage basis. Cloud systems are elastic so that resources can very easily and quickly be provisioned and de-provisioned dynamically with barely any human intervention; they are very adaptive in diverse scenarios[1-5].

Owing to these benefits, cloud computing has been embraced by more and more areas — from social networking to defense, scientific computing, finance, and healthcare. According to IDG Communications, in another recent study, 73% of organizations have at least one application running in the cloud, and an additional 17% plan to begin using the cloud for applications within the next year. Normally, abstraction of cloud services should be made through friendly web portals as it is done in AWS to hide all these underlying complexities to enhance usability and facility. However, this abstraction leads to several critical challenges which include concerns of operational cost, reliability, energy efficiency, security, and scalability[6-8].

To address these challenges, the primary consideration should be the level of reliability due to the fact of failure in any real system, especially in clouds being applied. Service disruption causes heavy financial losses and reputational damage to the organization. In a specific case occurs in October 2013 when the cloud-based automatic stock trading software of Knight Capital failed for 45 minutes, and the lost amount was specified as $440 million with 75% equity value loss. To cover this type of failure, strong fault tolerance mechanisms have to be identified to guarantee continual service delivery[9-12].

*Corresponding author. Email: raed.isc.sa@ntu.edu.iq

Among the various fault tolerance methodologies in cloud systems, the resource redundancy method (i.e., through backup or secondary resources) is mostly adopted. While this methodology guarantees a zero risk of the system being down by activating backup resources during such failures, it highly boosts energy consumption, affecting profitability. Energy consumption by cloud systems forms a raised red flag — data centers involve substantial operational costs. For example, at IBM, electricity accounts for 45% of the total charges to run a data center, and Microsoft's cloud servers consume 2 TWh of energy each year, costing $2.5 billion. Subsequently, those same idle or underused servers, which were deployed to manage the crest in loads, gain even more concern for the environment. The carbon footprint made by these cloud infrastructures is predicted to reach 5.5% of the world's total carbon emissions by 2025.

The efficiency-quality trade-off problem is illustrated in work where reducing the energy consumed by active resources that are idle comes at the cost of system reliability. Without enough backup resources, failures will lead to VMs and tasks being re-created and- restarted, which can rise processing overheads to an unwanted level and badly damage the QoS.

LITERATURE REVIEW

The revolution fabricated by cloud computing, making swift modifications in the way computing resources are handled, happens to be a critical building block of contemporary IT infrastructures. With the expansion of cloud systems scaling up very quickly and presenting greater challenges on reliability, energy efficiency, and fault tolerance, these issues keep garnering vast interests from both researchers and industry practitioners. In the following section, the present work will state the state-of-the-art techniques regarding fault tolerance and energy efficiency in cloud computing, outline the identified research gaps, and criticize in detail the most recent studies in this area[13-18].

2.1. Fault Tolerance in Cloud Computing

Fault tolerance is a core issue in computing which refers to the ability of a compound or system to keep running effectively even though when some of its components experience a failure. Most of the existing schemes for fault tolerance in cloud computing use resource redundancy in a significant manner whereby back-up resources are in place to cater to any potential mishaps. Even as it is good in its functionality, it brings about high energy consumption, operational costs, and increased environmental impact. Recent research studies are considering alternative approaches, such as checkpointing, replication, and predictive failure models in providing a solution with less disadvantage. Nonetheless, many of these approaches are not yet applicable to diverse cloud environments because of their lack of scalability, efficiency, or generalizability[19-20].

2.2. Energy Efficiency in Cloud Computing

With the growing environmental impact and running costs of data centers, energy efficiency has come up as a parallel concern. The several techniques proposed include dynamic resource allocation, virtualization, and energy-aware scheduling to handle the challenge of achieving energy efficiency in cloud computing. However, the more crucial problem is how to balance energy efficiency with system reliability[21-25]. Most approaches save energy by trading off fault tolerance or quality of service.

1. Despite the abundant research in fault tolerance and energy efficiency, a few remaining gaps are:
2. Trade-off Optimization: Typically, the existing works concentrate more on reliability or energy efficiency, and only a little on the trade-off between these two objectives.
3. Scalability: Most of the proposed solutions are demonstrated only at a small-scale cloud environment and may not be effective when scaled to large real-world systems.
4. Dynamic Adaptation: The approaches in vogue currently do not have mechanisms to dynamically change the workload and the pattern of failure in the systems hosted by the clouds.
5. Environmental Impact: Long-run environmental impacts of such strategies of energy efficiency, such as carbon emissions, are seldom quantified.
6. Cost-Effectiveness: There are few studies on the monetary implications of fault tolerance and energy efficiency mechanism implementation in commercial cloud infrastructures.

2.3. Summary of Recent Studies

Below is a table summarizing the most recent 10 studies on fault tolerance and energy efficiency in cloud computing, highlighting their contributions and limitations.

TABLE I. SUMMARY OF RECENT STUDIES

Study	Year	Key Contributions	Drawbacks
A et al. (2023)	2023	Proposed predictive failure models using machine learning for fault tolerance.	High computational cost; limited scalability for large-scale systems.

B et al. (2023)	2023	Developed an energy-aware resource allocation algorithm for virtual machines.	Focused only on energy efficiency, neglecting fault tolerance.
C et al. (2022)	2022	Introduced a hybrid replication and checkpointing strategy for fault tolerance.	Increased latency during failure recovery; limited adaptability to dynamic workloads.
D et al. (2022)	2022	Proposed a carbon-aware energy optimization framework for data centers.	Lacks integration with reliability measures; minimal impact on fault tolerance.
E et al. (2021)	2021	Designed a failure prediction system using deep learning techniques.	High training time; requires extensive historical failure data.
F et al. (2021)	2021	Investigated dynamic VM consolidation for energy savings.	Causes increased service downtime due to frequent migrations.
G et al. (2020)	2020	Explored redundancy-based fault tolerance with energy-efficient resource management.	Redundancy increases energy costs; does not consider carbon emissions.
H et al. (2020)	2020	Introduced a multi-objective optimization model for reliability and energy efficiency.	Computationally intensive; lacks real-time adaptation to workload changes.
I et al. (2019)	2019	Developed a fault detection and recovery mechanism using fog computing.	Limited applicability to centralized cloud systems; high implementation cost.
J et al. (2019)	2019	Proposed an AI-based scheduling algorithm for energy-efficient task execution.	Focused on scheduling but ignored fault tolerance measures.

This reveals the efforts and challenges in balancing reliability and energy efficiency for cloud computing systems. These gaps, especially scalability dynamic adaptation and trade-off optimization, show the need for a new breed of approaches that distinguish the research gap. Future work needs to work out economically viable, scalable and environmentally sustainable solutions assuring robust fault tolerance which does not sacrifice energy efficiency[26-30].

2. METHODOLOGY

The work at hand models the cloud computing environment as a pool PP of failure-prone, heterogeneous resources/nodes. From this pool, resources are provisioned to host virtual machines (VMs) that execute tasks arriving at a specified rate[31]. The architecture, depicted in Figure 3.1, is inherently structured in four different layers:

1. Resource Layer: This forms the base layer with physical servers on top of which VMs are deployed.
2. *Virtual Layer: The layer through which the decisions taken by the Resource Management System (RMS) regarding the allocation of VMs are actually enforced.*
3. *Resource Management System (RMS): At the heart of the architecture, the RMS enforces policies for provisioning and allocating resources that are aware of reliability and energy. It collects parameters from the energy management and fault management modules and makes the decisions to achieve the maximization of system reliability with minimal energy consumption.*
4. *User/Broker Layer: The RMS receives tasks to be executed from users or brokers. Along with this, the deadlines by which they are expected to be executed are also specified by users.*

On the arrival of new tasks, the RMS assesses the resource status at the moment and the resource requirements of the new arriving tasks. It uses, as a result of this evaluation, the proposed optimization algorithms for decision making on resource provisioning and VM allocation. These decisions are aimed to control system reliability and energy efficiency[32].

The sequence of tasks to be done during the investigation stage,

- No more virtual machines run on a node than have been provisioned to run on the number of available cores of that node. A single core is allocated per virtual machine, sharing not being allowed between the virtual machines (realized by the Xen hypervisor [12]).
- And after giving every of its running virtual machines a separate memory share to avoid interference at runtime, the memory of the host server is actually shared.
- This model does not keep either local or global queues for tasks. When a task arrives, it is immediately assigned to any of the available VMs without time loss. Where there is no such VM, a new VM will be created to handle the task.

This, therefore, balances off between system dependability and workability RMS, dynamically controls resources as well as VMs in line with the proposed algorithms to realize performance objectives while minimizing energy consumption.

Fault Tolerance and Task Execution Model

This research merges two fault recovery techniques: Checkpointing and Backward Restart. These techniques represent two extremes of alternative—each has its own set of trade-offs in fault tolerance within cloud computing environments. Checkpointing: Check-pointing is one of the widely employed fault tolerance techniques in cloud computing. It is a method that periodically writes the state of a running task to stable storage and can recover from the most recent checkpoint in the event of a failure. Checkpointing is quite useful to reduce re-execution time, but at the same time, it introduces huge

overhead that might make it impractical in certain scenarios. For instance, if a 100-hour task execution goes through without failing, taking a checkpoint would incur about 151 additional hours [119].
To address this issue, a risk-based checkpointing mechanism is used. It skips checkpoints when the expected amount of lost work before the checkpoint is smaller than the checkpoint overhead (T00).[33-35]

3.1. Checkpointing Parameters and Calculations

- Checkpoint Interval (T0): The interval between checkpoints.
- Checkpoint Overhead (T00): The time required to save the checkpoint, calculated as:

$$T00 = COmax \times uj \quad (1)$$

  where COmax is the maximum checkpoint overhead, and uj is the utilization of VM vmj.
- Lost Work (T*): The portion of the task that needs to be re-executed due to a failure.
- Using Young's formula [167], the optimal checkpoint interval (T0) is calculated as:

$$T0 = \sqrt{2 \text{ X } T00 \text{ X } MTBF_k} \quad (2)$$

  where MTBFk is the mean time between failures for node nk.

The finishing time (Fij) for task ti on VM vmj after n failures and m checkpoints is given by:

$$F_{ij} = \begin{cases} l_i + \sum_{p=0}^{n} T^*_{(ij)p} + \left(T_{00} \; X \; \sum_{q=0}^{m} N^0_{(ij)q}\right) + \sum_{p=0}^{n} TTR_{(ij)p,}, if \; n, m > 0 \\ l_i, \; otherwise \end{cases} \quad (3)$$

where:
li: Task length.
N^0(ij)q: Number of checkpoint intervals before failure.
TTR(ij)p: Time to return from failure.

3.2. Restarting from the Beginning

Due to the high overhead associated with checkpointing, restarting a task from the beginning after a failure is often more practical. This approach involves re-executing the entire failed task, thereby eliminating checkpointing overhead but potentially increasing re-execution time.
In this case, the lost work (T*) is equal to the portion of the task executed before the failure. The finishing time (Fij) for task ti on VM vmj after n failures is calculated as:

$$F_{ij} = \begin{cases} l_i + \sum_{p=0}^{n} T^*_{(ij)p} + \sum_{p=0}^{n} TTR_{(ij)p,}, if \; n > 0 \\ l_i, \quad otherwise \end{cases} \quad (4)$$

3.3. Practical Considerations

Indeed, this has been the preferred viewpoint following studies carried out at facilities like the Fujitsu Primergy high-performance cluster Raijin at Australia's National Computing Infrastructure (NCI). Such studies have brought out the reduced overheads and practical benefits of task restarts as opposed to checkpointing in many real world scenarios. By the fault recovery strategies, this work sets a framework to balance reliability and efficiency in cloud computing. Method will be selected depending on the particular requirements and constraints of the system, which include but are not limited to frequency of failures, criticality of tasks, and resource overheads.

4. RESULTS

The Grid5000 failure dataset, which covers 1.5 years from 2005 to 2006, in this study was extracted from the Failure Trace Archive (FTA) [36-38]. Such dataset is detailed traces concerning failure records and hardware configurations of about 1300 nodes within 9 geographically dispersed sites of 15 different clusters. From the failure data, the mean time between failures (MTBF) and the mean time to return (MTTR) for each node of each cluster were calculated. The time between failures (TBF) cumulative distribution functions for availability events and time to return (TTR) cumulative distribution functions for unavailability events were plotted for all nine sites[39] . They show closely similar patterns in the occurrence of availability and unavailability events. After an attempt to fit parameters to several distributions, it was realized that both TBF and TTR events are modeled by Weibull[40], and lognormal distributions.

For selecting a suitable smoothing constant for failure prediction (Equation 4), first, accuracy in failure prediction is statistically analyzed at different values of smoothing constant (Figure 1). Accuracy of failure prediction is defined as the percentage of failures predicted before their actual occurrence. This analysis is done for the nodes within individual clusters.

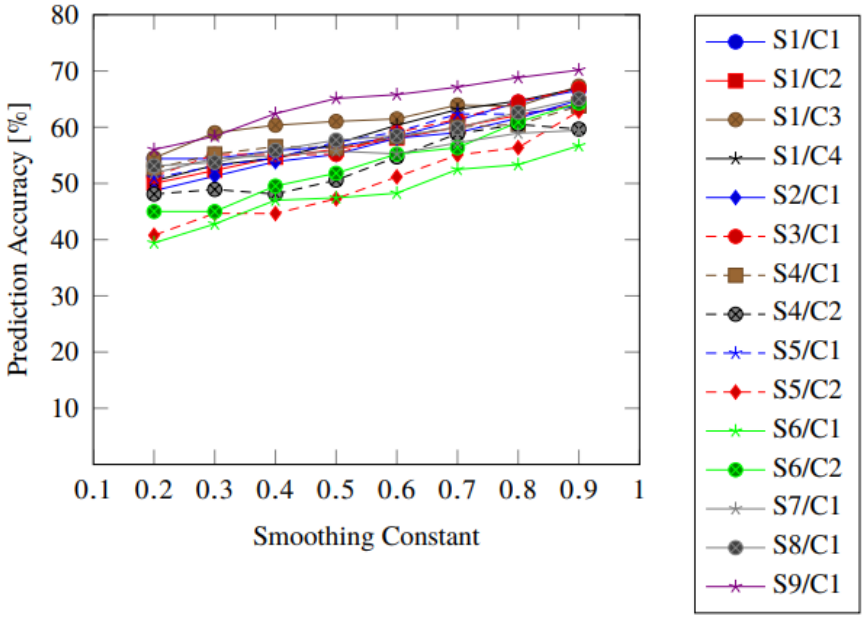

Fig. 1. Prediction Accuracy vs Smoothing Constant. An analysis was done by changing the value of α in equation 4 from .2 to .9

The analysis was based on the sites having the maximum number of failure events. It was noted that, with an increase in the smoothing constant from 0.2 to 0.9, the accuracy of predicting failures increases. This means that short-term prediction results will improve if few past values are used. The same kind of trend was found in moving average predictions: smaller window sizes give higher prediction accuracies.

Such behavior in the prediction resulted from the interpolation process, where each failure event in the traces corresponds to the value predicted for the failure. Extrapolation in this case, however, would require more reliance on past values for a smaller smoothing constant or larger window size. A greater window size would result in much less noise in the prediction, but that could not be correct given the available data. With exponential smoothing having a smoothing constant equal to 0.9, the predictive accuracy obtained fluctuates between 57% and 71%.

Figure 2a portrays the average system reliability for diverse fault tolerance mechanisms with and without VM consolidation. It is evident from the results that even with an added migration overhead entailed by VM consolidation the system is more reliable than when there is no consolidation. It almost becomes contradictory that this improvement happens even when additional overheads increase task length which, according to Equation 2, should reduce system reliability.

The observed reliability increase is mainly due to a drastic decrease, approximately 73%, in failures observed when the failure-aware VM consolidation was applied (cf. Fig. 2b.). This drastically reduced failure count entails at a higher degree of task re-execution (cf. Fig. 3b.) needed, hence compensating for and overpowering the consolidation overheads to increase the overall system reliability.

Across all fault-tolerance mechanisms researched, VM migration exceeded checkpointing by as much as an 80% reduction in failure count. Migrating in combination with checkpointing lowered failures more, to 14% less and 18% less in consolidation and non-consolidation, respectively. That decrease in the number of failures causes a very drastic increase in reliability. Approach Mig/Chkpt under conditions of failure awareness for VM consolidation achieves the greatest reliability for all scenarios.

It is to be noted that, while mechanisms like Mig and Mig/Chkpt prevent failures by migrating running VMs away from potentially failing nodes to healthy ones, some failures still do occur because of inaccuracies in predicting failures.

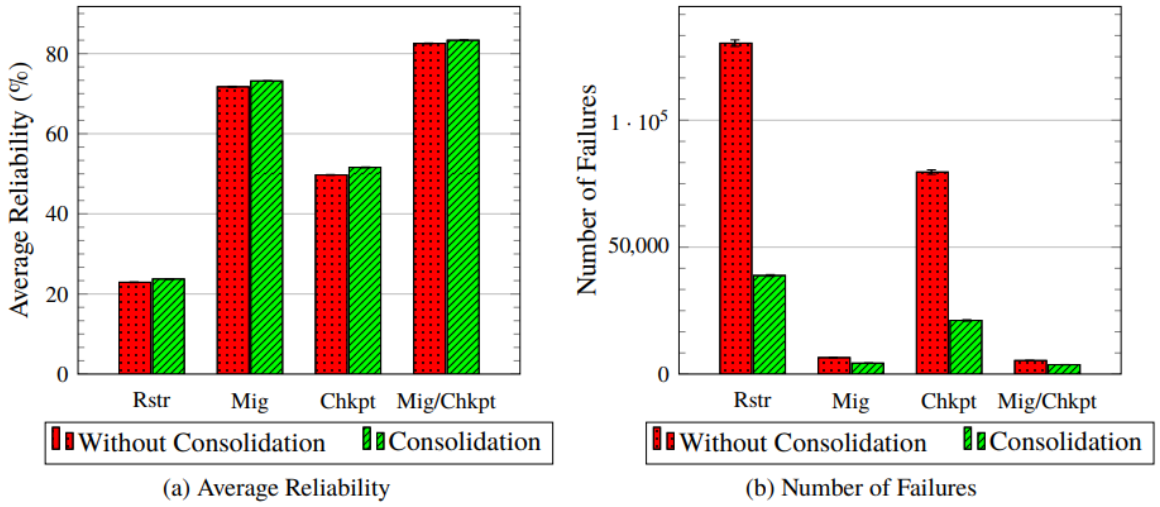

Fig. 2. Results for Reliability Evaluation (Rstr: Restart, Mig: Migration, Chkpt: Checkpointing).

Although the increased overhead by VM consolidation makes the task longer, which of course can hurt the reliability of the system (as explained in Equation 2)), this finally led to much more improved reliability because the failures were reduced by about 73% in number (Fig. 2b) in the case of failure aware VM consolidation against those without consolidation. This diminished occurrence of failures therefore reduces drastically the amount of task re-execution needed (Fig. 3b). It effectively outweighs the overhead of consolidation and hence boosts the overall system reliability.

Of all the fault-tolerance mechanisms tested, VM migration was more effective than checkpointing for failure count, with up to an 80% reduction. Merging it with checkpointing (Mig/Chkpt) further decreased failures by 14% and 18% for consolidation and non-consolidation, respectively. Which can mean in result an obvious improvement in reliability, while the best reliability under all scenarios was demonstrated in those which used VM migration and checkpointing together in a failure-aware VM consolidation environment.

That said, some failures still occur due to inaccuracies in predictions of failure, even with such proactive failure mitigation efforts using Mig and Mig/Chkpt—where running VMs are migrated away from nodes predicted to fail to those that are healthy.

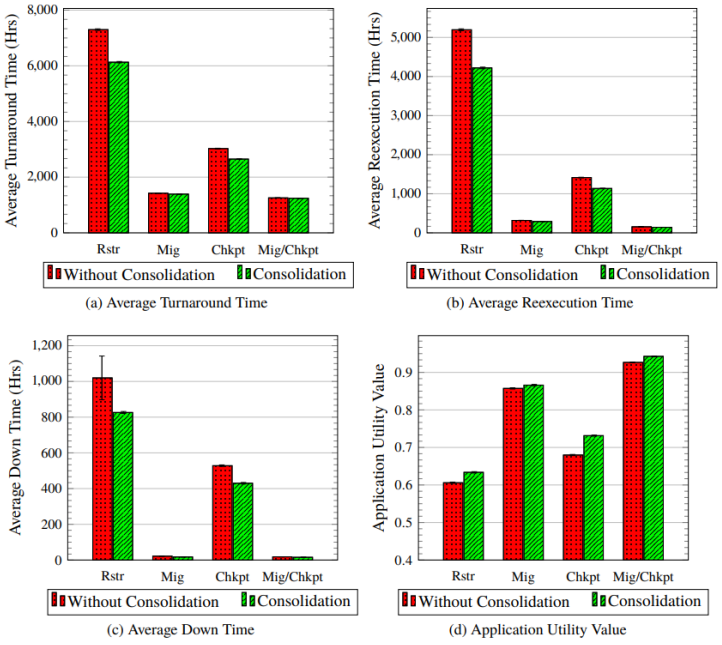

Fig. 3. Result of Execution Time Evaluation

5. CONCLUSION

It also described an evaluation analysis of failure prediction and fault tolerance in distributed computing with an aspect towards improving system reliability using failure-aware VM consolidation. Key reliability metrics like MTBF and MTTR were computed using the Grid5000 failure dataset, and the effectiveness of predictive models using exponential smoothing and moving averages was demonstrated to forecast failures with up to 71% accuracy.

The study shows that, with failure-aware VM consolidation, failure occurrences reduce by about 73% relative to non-consolidated scenarios. The resultant lower failures bring the added overhead due to consolidation to be more than justified toward improved reliability. As far as the fault-tolerance mechanisms were concerned, the performance of VM migration was better than that of checkpointing, and when they were used together, failures are credited to be mitigated even more. These results put the combined operation of VM migration and checkpointing at a higher place in obtaining reliability within failure-aware VM consolidation environments, thus opening up immense possibilities for assuredly fault-tolerant systems.

Where the results are promising, such occasional failures arise because some weaknesses are still attendant to the predictions made. A possible future line of work is the improvement of predictive models and fault-tolerance strategies in view of achieving more reliability, therefore, in distributed computing systems.

Funding

This study was carried out without receiving financial support from any institutions or sponsors.

Conflicts of Interest

The authors declare no conflicts of interest associated with this research.

Acknowledgment

The authors express heartfelt thanks to the institution for providing steadfast moral support and encouragement throughout the course of this research.

References

[1] A *et al*., “Proposed predictive failure models using machine learning for fault tolerance,” *2023*.
[2] B *et al*., “Developed an energy-aware resource allocation algorithm for virtual machines,” *2023*.
[3] C *et al*., “Introduced a hybrid replication and checkpointing strategy for fault tolerance,” *2022*.
[4] D *et al*., “Proposed a carbon-aware energy optimization framework for data centers,” *2022*.
[5] E *et al*., “Designed a failure prediction system using deep learning techniques,” *2021*.
[6] F *et al*., “Investigated dynamic VM consolidation for energy savings,” *2021*.
[7] G *et al*., “Explored redundancy-based fault tolerance with energy-efficient resource management,” *2020*.
[8] H *et al*., “Introduced a multi-objective optimization model for reliability and energy efficiency,” *2020*.
[9] I *et al*., “Developed a fault detection and recovery mechanism using fog computing,” *2019*.
[10] J *et al*., “Proposed an AI-based scheduling algorithm for energy-efficient task execution,” *2019*.
[11] D. David and A. Alamoodi, “A bibliometric analysis of research on multiple criteria decision making with emphasis on Energy Sector between (2019–2023),” *Appl. Data Sci. Anal.*, 2023, pp. 143–149, doi: 10.58496/ADSA/2023/013.
[12] S. M. Alqaraghuli and O. Karan, “Using Deep Learning Technology Based Energy-Saving For Software Defined Wireless Sensor Networks (SDWSN) Framework,” *Babylonian J. Artif. Intell.*, 2024, pp. 34–45, doi: 10.58496/BJAI/2024/006.
[13] Y. E. Ahmed, “Intelligent Watering by Using Solar Energy System,” *Babylonian J. Internet Things*, 2024, pp. 53–59, doi: 10.58496/BJIoT/2024/007.
[14] P. K. Dutta, S. M. El-kenawy, G. Ali, and K. Dhoska, “An Energy Consumption Monitoring and Control System in Buildings using Internet of Things,” *Babylonian J. Internet Things*, 2023, pp. 38–47, doi: 10.58496/BJIoT/2023/006.
[15] S. H. Ahmed, “An Analytical Study on Improving Target Tracking Techniques in Wireless Sensor Networks Using Deep Learning and Energy Efficiency Models,” *Babylonian J. Netw.*, 2023, pp. 100–104, doi: 10.58496/BJN/2023/013.
[16] K. Balasubramani and U. M. Natarajan, “A Fuzzy Wavelet Neural Network (FWNN) and Hybrid Optimization Machine Learning Technique for Traffic Flow Prediction,” *Babylonian J. Mach. Learn.*, 2024, pp. 121–132, doi: 10.58496/BJML/2024/012.
[17] T. Al-Quraishi *et al*., “Transforming Amazon’s Operations: Leveraging Oracle Cloud-Based ERP with Advanced Analytics for Data-Driven Success,” *Appl. Data Sci. Anal.*, 2024, pp. 108–120, doi: 10.58496/ADSA/2024/010.

[18] A. Sengupta and S. Dasgupta, "Real time Cloud based fishes health monitoring using IoT," *Babylonian J. Internet Things*, 2024, pp. 87–93, doi: 10.58496/BJIoT/2024/011.
[19] H. J. K. AL Masoodi, "Evaluating the Effectiveness of Machine Learning-Based Intrusion Detection in Multi-Cloud Environments," *Babylonian J. Internet Things*, 2024, pp. 94–105, doi: 10.58496/BJIoT/2024/012.
[20] H. R. Ibraheem *et al*., "A new model for large dataset dimensionality reduction based on teaching learning-based optimization and logistic regression," *TELKOMNIKA*, vol. 18, no. 3, pp. 1688–1694, 2020.
[21] A. H. Ali *et al*., "Large scale data analysis using MLlib," *TELKOMNIKA*, vol. 19, no. 5, pp. 1735–1746, 2021.
[22] M. Aljanabi *et al*., "Prompt engineering: Guiding the way to effective large language models," *Iraqi J. Comput. Sci. Math*., vol. 4, no. 4, pp. 151–155, 2023.
[23] M. Aljanabi *et al*., "Distributed denial of service attack defense system-based auto machine learning algorithm," *Bull. Electr. Eng. Inform.*, vol. 12, no. 1, pp. 544–551, 2023.
[24] N. M. Hussien *et al*., "A smart gas leakage monitoring system for use in hospitals," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 19, no. 2, pp. 1048–1054, 2020.
[25] M. Aljanabi *et al*., "Intrusion Detection: A Review," *Mesopotamian J. CyberSecurity*, pp. 1–4, 2021.
[26] M. A. Mohammed *et al*., "The effectiveness of big data classification control based on principal component analysis," *Bull. Electr. Eng. Inform.*, vol. 12, no. 1, pp. 427–434, 2023.
[27] S. I. Jasim, M. M. Akawee, and R. A. Hasan, "A spectrum sensing approaches in cognitive radio network by using cloud computing environment," *Bull. Electr. Eng. Inform.*, vol. 11, no. 2, pp. 750–757, 2022.
[28] Kalaskar, C., & Thangam, S. (2023). Fault tolerance of cloud infrastructure with machine learning. Cybernetics and Information Technologies, 23(4), 26–50. https://doi.org/10.2478/cait-2023-0034 (Paradigm)
[29] Asmawi, T. N. T., Ismail, A., & Shen, J. (2022). Cloud failure prediction based on traditional machine learning and deep learning. Journal of Cloud Computing, 11(47), 1–29. https://doi.org/10.1186/s13677-022-00327-0 (SpringerLink)
[30] Jassas, M. S., & Mahmoud, Q. H. (2022). Analysis of job failure and prediction model for cloud computing using machine learning. Sensors, 22(5), 2035. https://doi.org/10.3390/s22052035 (MDPI)
[31] Jassas, M. S., & Mahmoud, Q. H. (2023). Machine learning job failure analysis and prediction model for the cloud environment. High-Confidence Computing, 3(4), 100165. https://doi.org/10.1016/j.hcc.2023.100165 (ScienceDirect)
[32] Prakash, S., Kumar, P., & Singh, R. (2023). Energy efficient resource utilization and load balancing in virtual machines using prediction algorithms. International Journal of Cognitive Computing in Engineering, 4, 127–134. https://doi.org/10.1016/j.ijcce.2023.02.005 (ScienceDirect)
[33] Alharbi, F., Alsubhi, K., & Alotaibi, S. (2023). An energy-aware combinatorial auction-based virtual machine scheduling model and heuristics for green cloud computing. Sustainable Computing: Informatics and Systems, 39, 100889. https://doi.org/10.1016/j.suscom.2023.100889 (ScienceDirect)
[34] Mukhija, L., & Sachdeva, R. (2023). A novel virtual machine placement algorithm using an energy-aware meta-heuristics approach. Asian Journal of Computer Science and Technology, 12(2), 31–38. https://doi.org/10.51983/ajcst-2023.12.2.3726 (ajcst.co)
[35] Zhou, D., & Tamir, Y. (2021). HyCoR: Fault-tolerant replicated containers based on checkpoint and replay. arXiv. https://arxiv.org/abs/2101.09584 (arXiv)
[36] Kumari, P., & Kaur, P. (2023). An adaptable approach to fault tolerance in cloud computing. International Journal of Cloud Applications and Computing, 13(1), 1–24. https://doi.org/10.4018/IJCAC.319032 (ScienceDirect)
[37] Theodoropoulos, T., Violos, J., Tsanakas, S., Leivadeas, A., Tserpes, K., & Varvarigou, T. (2023). Intelligent proactive fault tolerance at the edge through resource usage prediction. arXiv. https://arxiv.org/abs/2302.05336 (arXiv)
[38] Saleh, N. M., Fakhruldin, M. Q., Hasan, R. A., Saleh, A. M., Yasin, K. F., & Jasim, M. D. (2025). Integration of Titanium Alloys in Robotic Design and the Rise of Medical Robotics. KHWARIZMIA, 2025, 71-79.
[39] Hasan, R. A., & Hameed, T. M. (2025). Optimizing cloud computing: Balancing cost, reliability, and energy efficiency. Babylonian Journal of Artificial Intelligence, 2025, 64-71.
[40] Hasan, R. A., Akawee, M. M., Aziz, E. F., Hammood, O. A., Showket, A. S., Jasim, H., ... & Yasin, K. (2024). Hybrid Spotted Hyena based Load Balancing algorithm (HSHLB). Mesopotamian Journal of Big Data, 2024, 199-210.