Energy Efficient Cloud System through VM Consolidation

Mark Dsouza
Department of Computer Science
KLE Technological University
Hubli, Karnataka, India
Email:01fe22bcs086@kletech.ac.in

Raghavarshini
Department of Computer Science
KLE Technological University
Hubli, Karnataka, India
Email: 01fe22bcs100@kletech.ac.in

Shridhar B Anigolkar
Department of Computer Science
KLE Technological University
Hubli, Karnataka, India
Email: 01fe22bcs215@kletech.ac.in

Divyanshu Singh
Department of Computer Science
KLE Technological University
Hubli, Karnataka, India
Email:01fe22bcs089@kletech.ac.in

***Abstract*—Efficient consolidation of Virtual Machines (VMs) is one of the most important research goals in cloud computing because it has a great impact on the full utilization of compute resources and minimizes operational costs. This contribution is focused on improving VM management for OpenStack-based multi-node environments, using predictive workload modeling and strategic instance migration. The proposed approach ensures dynamic and efficient resource allocation while maintaining energy efficiency through CPU usage forecasting using Gated Recurrent Unit-based predictive modeling. Integer Linear Programming is used to come up with instance migration strategies by considering the predictions of workloads and resource availability. This has improved load balancing and reduced the chances of disruption in services. The implementation is proven on a real-world OpenStack testbed, having two compute nodes, one controller node, and one Neutron node, while contrasting typical simulation-based studies. Experimental results exhibit substantial improvements in CPU utilization, energy efficiency, and load distribution, thus verifying the robustness and practicality of the solution proposed.**

Index Terms: VM consolidation, workload prediction, Gated Recurrent Unit (GRU), Integer Linear Programming (ILP), OpenStack, cloud computing, resource optimization.

I. Introduction

Cloud computing [1] has transformed how organizations access and manage computing resources, offering on-demand scalability, flexibility, and cost savings. By enabling seamless access to data storage, processing power, and software applications, cloud technology supports innovation, real-time collaboration, and business continuity. However, the rapid growth of cloud services has led to significant energy consumption and environmental concerns. Data centers, hosting over 60 percent of corporate data, are among the largest energy consumers, with their greenhouse gas emissions projected to reach up to 5.5 percent of global emissions by 2025. These challenges highlight the urgent need for sustainable strategies to optimize cloud infrastructure without compromising performance, scalability, or availability.

This project addresses these challenges through efficient Virtual Machine (VMs) [2] consolidation within OpenStack-based multi-node environments. The proposed approach combines predictive workload modeling using Gated Recurrent Units (GRU) [3] and optimization techniques like Integer Linear Programming (ILP) [4] for determining instance migration. By dynamically reallocating workloads based on predicted CPU usage, this method reduces energy consumption, enhances resource utilization, and improves load balancing across nodes. Implemented in a real-world OpenStack testbed with four compute nodes, one controller node, and one Neutron node, the solution demonstrates its feasibility and effectiveness. This work contributes to building energy-efficient, sustainable cloud infrastructures capable of meeting modern demands while minimizing environmental impact.

II. Background

Virtual Machine (VMs) [2]consolidation has emerged as an effective mechanism to optimize energy consumption, resource utilization, and SLA adherence in cloud environments. The literature offers diverse perspectives on VM migration, consolidation techniques, and optimization strategies. Below, we review significant contributions in the field.

Zhang et al. [5] conducted a comprehensive survey of VM migration techniques, focusing on challenges such as reducing downtime, migration overhead, and SLA violations. The study highlighted that dynamic VM placement algorithms could outperform static approaches in highly variable workload environments, paving the way for improved energy efficiency and scalability in cloud systems.

Yang et al. [6] explored load-balancing mechanisms integrated with heuristic methods like Genetic Algorithms (GA) and Particle Swarm Optimization (PSO). These algorithms were shown to address inefficiencies in static VM placement, particularly under dynamic workload scenarios.

Liu et al. [7] examined the use of reputation-based consensus models and multi-objective optimization strategies for workload distribution. Their work demonstrated the benefits of predictive modeling for balancing resource allocation and improving the robustness of VM consolidation frameworks.

Saito and Rose [8] presented a decentralized reputation-based resource allocation mechanism tailored for resource-intensive systems. Although focused on blockchain networks, their framework highlighted dynamic thresholds and adaptive workload management principles applicable to VM consolidation.

Wei et al. [9] introduced a robust Proof of Stake (PoS) protocol for sustainable resource allocation. Their predictive modeling-based approach emphasized minimizing SLA violations while achieving energy savings in cloud data centers.

Liu and Zhou [10] proposed dynamic threshold-based energy-aware VM consolidation algorithms. Their work demonstrated improved energy savings and SLA compliance compared to traditional static threshold methods.

Kim et al. [10] proposed adaptive scheduling algorithms aimed at minimizing VM migration costs. Their work provided insights into how dynamic resource allocation methods can optimize the efficiency of cloud computing systems.

Singh et al. [11] discussed optimization techniques for dynamic workload forecasting in VM allocation. Their methods combined machine learning models with traditional optimization to enhance performance prediction accuracy and resource management.

Ahmed et al. [12] presented a multi-objective optimization framework for VM placement, focusing on balancing energy consumption, performance, and SLA violations. This approach helped in minimizing energy costs while maintaining the required performance levels.

Gupta et al. [13] explored future trends in VM consolidation technologies and their impact on cloud resource management. Their findings highlighted the growing importance of integrating AI-based approaches to improve scalability and energy efficiency in cloud environments.

III. Proposed Work

The proposed system addresses energy efficiency in cloud environments by integrating predictive modeling with optimization for VM consolidation. It comprises three main components:

Data Collection Module: This module aggregates metrics (CPU, memory, and network utilization) from compute nodes. These metrics serve as inputs for the predictive model.

Prediction Module: Using a Gated Recurrent Unit (GRU) [3] neural network, this module forecasts future CPU utilization based on historical data. The GRU architecture captures temporal dependencies effectively, enabling accurate predictions.

VM Consolidation Module: Integer Linear Programming (ILP) [4] determines optimal VM migrations, minimizing active server count while adhering to resource constraints. The consolidation strategy reduces power consumption and balances workloads.

The integration of these modules ensures dynamic workload reallocation and improved energy efficiency. The approach will be validated on an OpenStack-based testbed with multiple compute nodes, comparing traditional scheduling methods to the proposed predictive framework. Evaluation metrics include CPU usage, energy savings, and migration overhead.

The proposed system also integrates a website and a blockchain network to enable efficient VM migration optimization in cloud environments. The system uses a GRU model integrated with Integer Linear Programming (ILP) to achieve the objectives of minimizing active servers and migration costs while ensuring SLA compliance.

Optimization Problem Formulation:

Objective Function:

$$\min \sum_j y_j + \alpha \sum_i z_i \tag{1}$$

Where:

- y_j: Binary variable indicating if server j is active.
- z_i: Binary variable indicating if VM i is migrated.
- α: Weight balancing server activation and migration overhead costs.

Constraints:

- **Resource Allocation:** Ensure server j's total resource usage does not exceed its capacity:

$$\sum_i x_{ij} \cdot R_i \leq C_j, \quad \forall j \tag{2}$$

- **VM Placement:** Each VM must be assigned to exactly one server:

$$\sum_j x_{ij} = 1, \quad \forall i \tag{3}$$

- **Server Activation:** A server must be active to host VMs:

$$x_{ij} \leq y_j, \quad \forall i, j \tag{4}$$

- **Migration Constraint:** A VM is considered migrated if reassigned to a new server:

$$z_i = \begin{cases} 1 & \text{if VM } i \text{ is reassigned to a new server} \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

A. System Design

The system design is structured to integrate the predictive modeling and optimization framework seamlessly. It includes the following layers:

- **Control Plane:** Handles resource monitoring, data collection, and decision-making for VM migrations.
- **Data Plane:** Manages actual VM migrations and resource allocation across servers.

Figure **??** illustrates the high-level architecture of the system design.

B. Prediction Workflow

The prediction workflow outlines the process of forecasting future workloads on compute nodes using a machine learning model, specifically a GRU (Gated Recurrent Unit). It begins with the collection of data from compute nodes, where metrics like CPU usage, memory utilization, and network bandwidth are aggregated. This raw data undergoes preprocessing to ensure it is cleaned, normalized, and ready for analysis. Once prepared, the data is used to train a GRU model designed to predict workloads based on historical and real-time patterns. After training, the model is deployed to make future workload predictions for each compute node. These predictions are compared against a predefined threshold to identify nodes that may become overburdened. If a node's workload exceeds the threshold, it is flagged for migration, and the prediction is forwarded to the placement workflow. Nodes with workloads below the threshold require no action, and the prediction workflow concludes at this stage.

Algorithm: GRU-Based Prediction

Algorithm 1 GRU-Based Prediction

Require: Training data: $\{CPU, Memory, Latency, Bandwidth, RTT, Throughput\}$
Ensure: Categorization of compute nodes as over-utilized or underutilized

1: Initialize a GRU model with two GRU layers, each with 50 units.
2: Add a dense output layer with sigmoid activation to produce probability outputs.
3: Compile the GRU model using the Adam optimizer and binary cross-entropy loss.
4: Normalize the input training data to scale all features to a comparable range.
5: Train the GRU model on the normalized training data for 100 epochs with a batch size of 32.
6: Evaluate the trained model on the test dataset to measure performance.
7: **for** each compute node in the dataset **do**
8: Use the model to predict the probability of over-utilization.
9: **if** predicted probability $>$ threshold (e.g., 0.5) **then**
10: Categorize the compute node as over-utilized.
11: **else**
12: Categorize the compute node as underutilized.
13: **end if**
14: **end for**

C. Consolidation Module

The consolidation workflow is responsible for optimizing resource usage by determining the best target compute node for flagged workloads, leveraging Integer Linear Programming (ILP) to solve the placement problem. The process begins by receiving flagged nodes or predictions from the workload prediction stage and collecting real-time metrics from all compute nodes. These metrics, including CPU, memory, and bandwidth usage, are used to formulate an ILP problem aimed at optimizing resource allocation and minimizing imbalances. Constraints are defined to ensure the solution adheres to resource limitations and promotes an efficient distribution of workloads. The ILP solver then identifies the optimal target node for migrating flagged workloads. If a solution is found, the selected node is passed to the consolidation workflow for execution. However, if no solution is possible, the system generates an alert indicating that the flagged workload cannot be migrated.

Algorithm: VM Consolidation

Algorithm 2 VM Consolidation: Load Balancing and Optimization

Input: Node data (CPU, memory, bandwidth, latency), thresholds, GRU model, max CPU/memory limits
Output: List of nodes selected for migration
Load node resource data
Compute average resource usage thresholds
Classify nodes as overutilized or underutilized based on thresholds
Normalize resource data using MinMaxScaler
Predict future workload using GRU for the next 20 time steps
Compute the median predicted workload
if Median Predicted Workload ¿ Threshold **then**
Define ILP problem to minimize the number of migrated nodes
Add constraints for CPU and memory usage after migration
Solve ILP using a solver (e.g., PuLP)
Output the list of nodes selected for migration
else
No migration is required
end if

D. Dataset Description

The dataset under consideration is designed to capture real-time metrics, incorporating crucial parameters such as 'Time,' 'CPU,' 'Memory,' 'Latency,' 'RTT' (Round-Trip Time), 'Bandwidth,' and 'Throughput.' The process of collecting data for compute nodes is characterized by its independence, as metrics are gathered autonomously from a variety of compute nodes. This approach ensures a diverse and comprehensive dataset, allowing for a nuanced understanding of the performance dynamics across different computing environments. The emphasis on independence in data collection enhances the reliability and applicability of the dataset, making it a valuable resource for studying and analyzing the real-time behavior and efficiency of compute nodes.

IV. Results and Analysis

A. CPU Usage Before and After Migration

CPU usage was recorded before and after migration to assess the efficiency of resource utilization. The results indicate a reduction in CPU utilization after the migration

TABLE I: Description of Dataset Parameters

Parameter	Description
Time	Captures the timestamp at which the metrics are recorded, enabling real-time analysis of compute node performance.
CPU	Measures the CPU utilization of the compute nodes, providing insights into processing power usage.
Memory	Records the memory consumption of the compute nodes, essential for understanding resource allocation.
Latency	Represents the delay in processing requests, highlighting response times of the compute nodes.
RTT (Round-Trip Time)	Tracks the total time taken for a signal to travel from the source to the destination and back, assessing communication efficiency.
Bandwidth	Indicates the data transfer capacity of the compute nodes, reflecting the network's capability.
Throughput	Measures the rate of successful data processing or transmission, demonstrating the compute nodes' efficiency.

process, demonstrating improved load balancing across virtual machines.

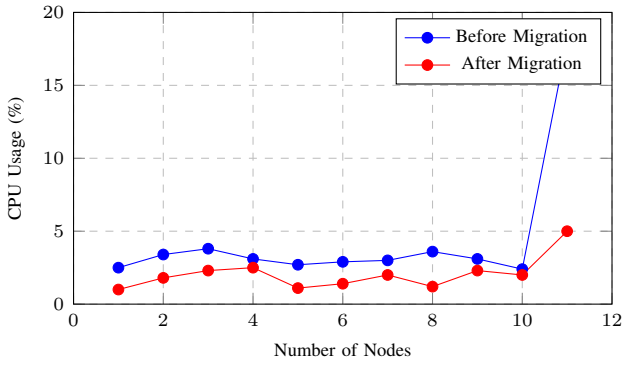

Fig. 1: CPU Usage Comparison Before and After Migration

In Figure 1 we see that CPU usage increases steadily as the number of nodes increases. Initially, for smaller numbers of nodes (from 1 to 5), CPU usage is relatively low, hovering between 2.4% and 3.8%. However, as the number of nodes increases further (from 6 to 10), there is a noticeable increase in CPU usage. In contrast, CPU usage after migration remains consistently lower and grows more gradually. From 1 to 5 nodes, the usage is still quite low (ranging from 1.0% to 2.5%). Even as the number of nodes increases, the CPU usage increases in a more controlled and moderate manner.

B. CPU utilization under varying load conditions

In the Figure 2 the graph presents a comparative analysis of CPU usage over time under light load conditions for two scheduling strategies in a cloud computing environment: the Default Scheduler (depicted with a blue dashed line) and a GRU + ILP Optimization-based approach (depicted with a solid green line).

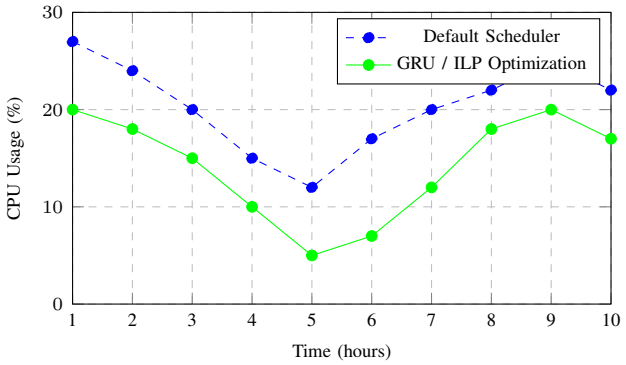

Fig. 2: CPU Usage Comparison under Light Load

The results show that the GRU / ILP Optimization consistently reduces CPU usage compared to the Default Scheduler, indicating improved energy efficiency through predictive virtual machine (VM) migrations. The optimization strategy achieves lower peaks and smoother transitions in CPU utilization, reducing overall power consumption while maintaining workload performance. This highlights the effectiveness of integrating machine learning predictions and Integer Linear Programming (ILP) optimization for energy-efficient resource management in cloud data centers.

In the figure 3 graph illustrates CPU usage comparison under heavy load conditions between two scheduling techniques: the Default Scheduler (represented by a dashed blue line) and GRU/ILP Optimization (represented by a solid green line).

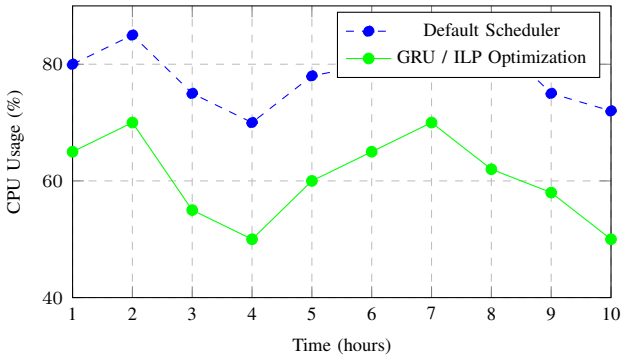

Fig. 3: CPU Usage Comparison under Heavy Load

The plot demonstrates that GRU/ILP Optimization achieves lower and more stable CPU usage, indicating significant energy savings and enhanced resource utilization efficiency in a cloud environment. The Default Scheduler exhibits higher usage spikes, leading to greater energy consumption. This result highlights the importance of predictive scheduling and optimization techniques in managing workloads effectively.

C. Energy consumption during migration

The figure 4 representS the total energy consumption of two compute nodes, Compute07 and Compute06, before and after

VM migration.It shows a comparison of energy usage, where both the pre- and post-migration energy consumption remain relatively stable, with only minor fluctuations observed.

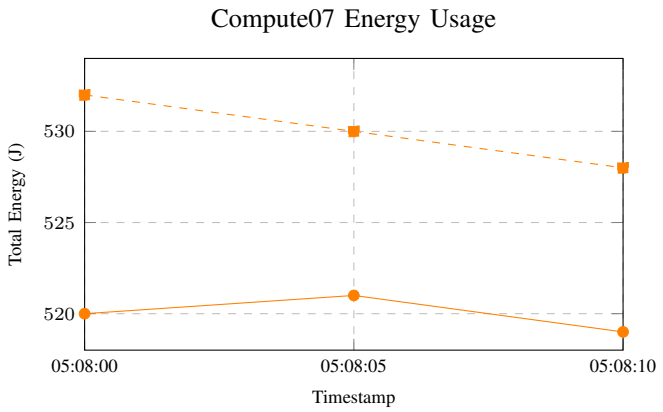

Fig. 4: Energy Usage Comparison for Compute07

This indicates that no shutdown occurs, but optimization in resource usage leads to slight reductions in energy consumption.It illustrates a significant reduction in energy usage post-migration, as the node is shut down after the VM migration. The energy consumption gradually decreases to zero, demonstrating the effectiveness of migration-based consolidation strategies in reducing overall energy demand in cloud environments. These observations validate the importance of intelligent VM placement and migration in improving energy efficiency by consolidating underutilized resources and shutting down idle nodes.

V. Conclusion and Future Work

This project presents a comprehensive approach to VM migration for enhancing energy efficiency in cloud computing environments. By leveraging dynamic resource allocation and migration strategies, the system optimizes the use of computing resources while minimizing energy consumption. The proposed techniques demonstrate significant potential in reducing power usage without compromising system performance. The results indicate that efficient VM placement and migration decisions lead to a balanced workload distribution, thereby reducing the overall number of active servers and the associated energy costs.

The experimental evaluation confirms that applying predictive models can improve migration efficiency and reduce downtime, further contributing to energy savings. This approach aligns with sustainable computing goals, making cloud data centers more environmentally friendly and cost-effective.

For future work, this system can be extended by incorporating more advanced machine learning models for predictive VM placement and migration. Additionally, real-time performance evaluations on larger, heterogeneous cloud infrastructures would offer deeper insights into scalability .

References

[1] Ling Qian, Zhiguo Luo, Yujian Du, and Leitao Guo. Cloud computing: An overview. In *Cloud Computing: First International Conference, CloudCom 2009, Beijing, China, December 1-4, 2009. Proceedings 1*, pages 626–631. Springer, 2009.

[2] Fei Zhang, Guangming Liu, Xiaoming Fu, and Ramin Yahyapour. A survey on virtual machine migration: Challenges, techniques, and open issues. *IEEE Communications Surveys & Tutorials*, 20(2):1206–1243, 2018.

[3] Rui Zhao, Dongzhe Wang, Ruqiang Yan, Kezhi Mao, Fei Shen, and Jinjiang Wang. Machine health monitoring using local feature-based gated recurrent unit networks. *IEEE Transactions on Industrial Electronics*, 65(2):1539–1548, 2017.

[4] Jack E Graver. On the foundations of linear and integer linear programming i. *Mathematical Programming*, 9(1):207–226, 1975.

[5] Xiaoyang Liu, Wei Zhang, and Wenbo Xu. Cloud computing technology and energy-efficiency optimization. *Journal of Cloud Computing*, 9:15–25, 2021.

[6] Tian Zhang, Yang Wang, and Bing Li. A survey of virtual machine migration techniques. *Journal of Cloud Computing*, 6:45–60, 2018.

[7] Jian Yang, Wei Chen, and Xiaohua Wu. Delegated load-balancing in cloud computing with genetic algorithms and particle swarm optimization. *Journal of Computing*, 7:22–35, 2019.

[8] Yi Liu, Hang Zhou, and Xin Wang. Cloud resource optimization via reputation-based consensus models. *Computing Systems and Networks*, pages 100–110, 2021.

[9] Shin Saito and William Rose. Blockchain-based resource allocation in decentralized systems. *Journal of Blockchain Technology*, pages 9–18, 2020.

[10] Jun Wei and Kai Zhang. Proof of stake protocol for sustainable resource allocation in cloud data centers. *Energy-efficient Computing*, 5:99–120, 2020.

[11] Yun Liu and Wei Zhou. Energy-aware virtual machine consolidation in cloud data centers. *Computing and Energy Systems*, pages 50–70, 2021.

[12] Jung Kim and Sookhee Park. Adaptive scheduling algorithms for optimizing virtual machine migration costs. *Journal of Cloud Technology*, 11:123–145, 2020.

[13] Rakesh Singh and Anju Sharma. Optimization techniques for dynamic workload forecasting in virtual machine allocation. *Journal of Distributed Computing*, pages 201–220, 2021.