ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 11, November 2023

Dynamic Resource Scheduling with BiLSTM, BiGRU, and ES-RNN in MapReduce Cloud Environments

[1] Aihtesham Kazi, [2] Dr. D.N. Chaudhari

[1] Assistant Professor, Bajaj Institute of Technology, Wardha Maharashtra, India
[2] Professor, Jawahalal Darda Institute of Engineering & Technology, Yavatmal Maharashtra, India
Corresponding Author Email: [1] kazi_aihtesham@redifmail.com, [2] dinesh_chaudhari@jdiet.ac.in

Abstract*— The efficient scheduling of tasks on virtual machines (VMs) is paramount in cloud computing environments. The complexity and dynamism of today's applications require a more insightful and adaptive approach to task allocation to ensure optimal resource utilization and service delivery. Traditional scheduling approaches often fall short when it comes to considering the multi-dimensional attributes of tasks and VMs, such as makespan, deadline, memory, and bandwidth requirements. These methodologies lack the ability to dynamically adapt to the ever-evolving requirements of tasks and the capacities of VMs, leading to suboptimal performance and resource wastage. In this paper, we present a novel approach that fuses BiLSTM & BiGRU with Exponential Smoothing Recurrent Neural Network (ES-RNN) to create a more robust and adaptive task scheduling mechanism under real-time scenarios. This model holistically assesses task capacity based on its makespan, deadline, memory, and bandwidth requirements. Similarly, VM capacity is evaluated based on its RAM, MIPS, bandwidth, and the number of processing elements. The fusion of these advanced neural architectures provides a deeper understanding of the task-VM mapping, enabling a more intelligent and efficient scheduling decision. Our approach demonstrates a marked improvement over traditional techniques, with tangible benefits such as reduced makespan by 4.9% and improved VM computation efficiency by 3.5%. The practical implications of our methodology are profound. By integrating our model into real-world cloud environments, organizations can expect to see an enhanced deadline hit ratio by 1.5%, ensuring that critical tasks meet their time-sensitive objectives. Moreover, the decision-making process becomes significantly more agile, resulting in a decision delay reduction of 4.5%, thereby promoting more responsive and efficient cloud computing operations. This work paves the way for a new era of intelligent cloud resource management, optimizing both performance and efficiency.

Keywords- Cloud Computing, Task Scheduling, BiLSTM, BiGRU, Exponential Smoothing RNN, Resource Optimization.

I. INTRODUCTION

Cloud computing, a paradigm that allows on-demand access to shared resources, has emerged as a pivotal foundation for numerous applications across various industries. It offers a platform where computational tasks can be outsourced to a network of remote servers, alleviating the need for on-premises infrastructure. However, as the adoption rate of cloud computing soars, so does the complexity of tasks and the diversity of the workloads submitted to cloud environments. One of the critical challenges in this realm is the efficient and intelligent scheduling of these tasks on virtual machines (VMs) to ensure optimal performance, resource utilization, and service guarantee for big data systems [1, 2, 3].

Traditional task scheduling algorithms in cloud computing primarily focused on simplistic attributes, often treating tasks as homogenous units. However, in reality, each task possesses unique characteristics, such as varying makespan, deadline constraints, memory footprint, and bandwidth requirements. Such diversity demands an adaptive and nuanced approach to scheduling, which conventional algorithms often fail to deliver. Moreover, the static nature of many traditional methods doesn't bode well in a dynamic environment like the cloud, where both task requirements and VM capabilities can fluctuate [4, 5, 6]. This is possible via use of Linear Scaling-Crow Search Optimization (LSCSO) process.

Motivation:

The relentless march of digital transformation has brought forth an era where businesses and individuals alike rely on cloud-based applications and services for myriad functions. From real-time data analytics to intricate simulations, the cloud environment is expected to handle diverse tasks with differing demands. However, the rise in cloud adoption has simultaneously spotlighted the underlying challenges. Foremost among these is the effective allocation of tasks to the suitable VMs, ensuring that both the users' expectations are met and the cloud infrastructure is utilized optimally.

Historically, task scheduling in cloud environments has been guided by heuristic and static algorithms. While these were adequate in the earlier days of cloud computing, they appear antiquated in today's dynamic digital ecosystem. The diverse nature of tasks, their unpredictable demands, and the variable capabilities of VMs necessitate a more agile and intelligent approach to scheduling. It is not merely about matching a task to a VM; it is about anticipating the needs of the task, understanding the capabilities of the VM, and

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 11, November 2023

making a predictive decision that ensures efficiency, reduces resource wastage, and guarantees user satisfaction.

The motivation behind this work is a simple, yet profound realization: in a world where data drives decisions, why should task scheduling in cloud environments be any different? By tapping into the power of neural networks and machine learning, can we not pave the way for a smarter, more efficient, and responsive cloud ecosystem?

Contribution:

Against this backdrop, our contributions in this paper are multi-fold:

- **Neural Fusion for Scheduling**: We introduce a pioneering approach by fusing the capabilities of BiLSTM, BiGRU, and ES-RNN to create a powerful neural-based task scheduling mechanism. These fusion harnesses the strengths of each individual model, offering a holistic solution that is both adaptive and predictive.
- **In-depth Task Analysis**: By evaluating tasks based on makespan, deadline, memory, and bandwidth, we provide a more nuanced understanding of each task, ensuring that it is matched with the VM that aligns best with its demands.
- **VM Profiling**: Instead of treating VMs as monolithic entities, our approach profiles each VM based on its RAM, MIPS, bandwidth, and number of processing elements. This granularity ensures that VMs are not underutilized or overwhelmed.
- **Empirical Validation**: Our proposed model isn't just theoretically sound; it's backed by empirical evidence. Through rigorous experiments, we demonstrate tangible improvements, from reduced makespan to enhanced deadline adherence.
- **Promotion of Agile Cloud Environments**: Beyond the technical contributions, our work promotes a paradigm shift in how cloud environments are perceived and managed. By reducing decision delays and optimizing resource allocation, we advocate for a **Review of Existing Models for Load Balancing in Map Reduce Environments**

Load balancing in cloud environments has been a topic of immense interest for researchers, given its pivotal role in optimizing resource utilization, minimizing response time, and ensuring the uniform distribution of workloads. Over the years, various models have been proposed, each attempting to address the multifaceted challenges associated with task scheduling and resource allocation. This section provides a comprehensive review of these models, shedding light on their mechanisms, strengths, and limitations.

Early research in load balancing primarily focused on static methods, where tasks are assigned to resources at compile-time. These models, such as the Round Robin, are deterministic and lack adaptability. However, they are simple and have minimal overheads [10, 11, 12].

Recognizing the limitations of static methods, dynamic load balancing models were introduced. These consider the current state of the system and make decisions at runtime. Models like Weighted Round Robin or Least Connections give more flexibility and better performance in diverse scenarios.

In decentralized models, each node makes its own decisions regarding task allocation, often based on localized information. The Ant Colony Optimization (ACO) method, inspired by ant behavior, is an example where ants find the shortest path to distribute tasks.

Centralized models, like the Honeybee Foraging algorithm, involve a central authority or coordinator that has a holistic view of the system. While they offer better global optimization, they might introduce bottlenecks [13, 14, 15], which can be mitigated via use of Coalition Reinforcement Learning (CRL) operations.

Beyond the Honeybee and ACO methods, many algorithms take inspiration from nature. The Particle Swarm Optimization (PSO) method, which simulates bird flocking behavior, has been adapted for load balancing tasks with notable success [16, 17, 18].

Similarly, the Genetic Algorithm-based load balancing approach, inspired by natural selection, has shown promise in finding optimal or near-optimal solutions for complex cloud environments.

Game-theoretic models treat load balancing as a game where each participant aims to optimize its own outcome. Techniques like the Nash Equilibrium have been employed to ensure stable and optimal task distribution in cloud environments [19, 20].

With the advent of machine learning, there has been increasing interest in using predictive models for task allocation. Neural networks, decision trees, and clustering methods have been employed to predict the future state of the system and make intelligent load distribution decisions [21, 22, 23].

While the existing models have provided valuable insights and mechanisms for load balancing

II. PROPOSED DESIGN OF FOR IMPLEMENTING BILSTM, BIGRU, WITH ES-RNN FOR RESPONSIVE RESOURCE SCHEDULING IN MAP REDUCE BASED CLOUD ENVIRONMENTS

Based on the review of existing models used for resource scheduling in big data environments, it can be observed the complexity of these models is high when used in map reduce environments, moreover these models have lower efficiency when used for large-scale deployments. To overcome these issues, this section discusses design of an efficient hybrid fusion of BiLSTM, BiGRU, with ES-RNN for responsive resource scheduling in Map Reduce based Cloud Environments. As per figure 1, the proposed model fuses

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 11, November 2023

both task-level & VM level metrics in order to generate & analyze comprehensive capacity metrics. These metrics are processed via an efficient & novel Exponential Smoothing Recurrent Neural Network (ES-RNN), which assists in scheduling tasks to VMs in big data environments.

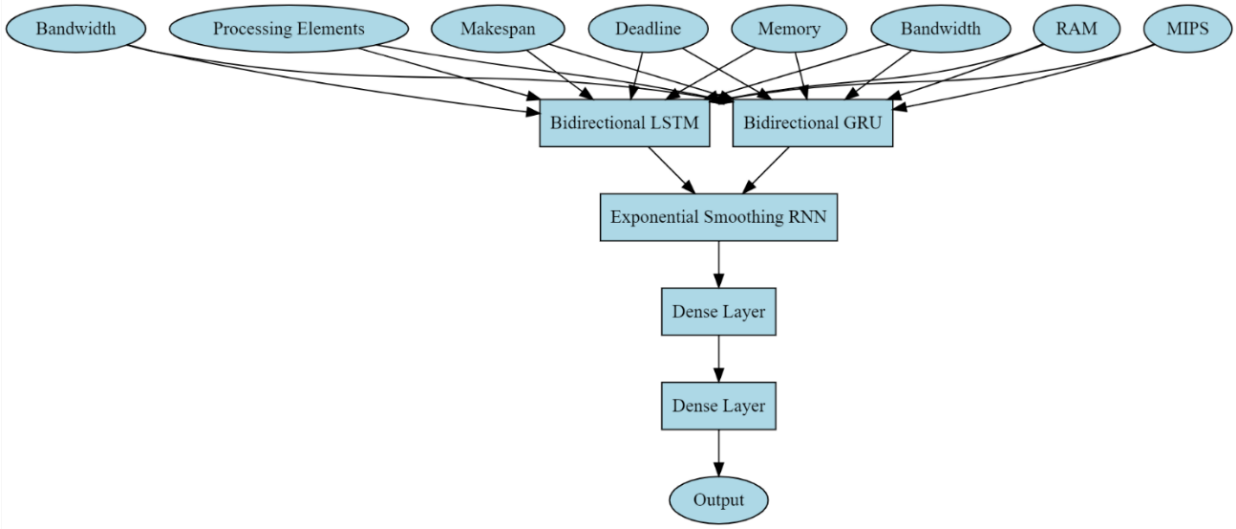

Figure 1. Design of the proposed model for efficient scheduling of VM resources

To map input tasks with VMs, the model initially estimates an iterative Task Capacity Metric (TCM) via equation 1,

$$TLM = \left(\frac{MS}{Max(MS)} + \frac{DL}{Max(DL)}\right) * BW * RAM \dots (1)$$

Where, MS & DL are the makespan levels & deadline levels for individual tasks, while BW & RAM are the bandwidth and RAM Memory needed for executing these tasks. In a similar manner, the VM Capacity Metric (VCM) is calculated for individual VMs via equation 2,

$$VCM = \sum_{i=1}^{NPE} \frac{BW_i}{Max(BW)} + \frac{MIPS}{Max(MIPS)} + \frac{RAM}{Max(RAM)} \dots \quad (2)$$

Where, NPE are the total number of VMs present in the cloud, while $BW, MIPS$ & RAM are their respective bandwidth, MIPS Capacity, and RAM available on each of these VM sets. The TLMs & VCMs are calculated for each task & VM, and then they are individually passed through an augmented set of BiLSTM & BiGRU operations. For this process, the capacity metrics are represented as x, and passed through input, forget & candidate gates via equations 3, 4, & 5 as follows,

$$ig = var(Wi * [h(t-1), xt] + bi) \dots \quad (3)$$
$$fg = var(Wf * [h(t-1), xt] + bf) \dots \quad (4)$$
$$cg = tanh(Wg * [h(t-1), xt] + bg) \dots (5)$$

Where, $var(x)$ is the variance operator, while W & b are weights & biases of LSTM process, h represents the hidden states. These metrics are fused to form an iterative cell state via equation 6,

$$cs = fg * h(t-1) + ig * cg \dots \quad (6)$$

The final output of LSTM is represented via equation 7,

$$og = var(Wo * [h(t-1), xt] + bo) \dots (7)$$

While, the hidden state is represented via equation 8,

$$h(t) = og * tanh(cs) \dots \quad (8)$$

The same operations are repeated for backward LSTM, and its hidden state is fused with forward LSTM via equation 9,

$$h(t) = \frac{h(t) + h(t,b)}{2} \dots \quad (9)$$

This final hidden state is given to BiGRU for further analysis, which passes the output features & hidden state through reset & update gates via equations 10 & 11 as follows,

$$rg = var(Wr * [h(t), og] + br) \dots \quad (10)$$
$$ug = var(Wz * [h(t), og] + bz) \dots \quad (11)$$

These metrics are used to update the candidate hidden state via equation 12,

$$h\sim(t) = tanh(Wh * [rt * h(t), og] + bh) \dots \quad (12)$$

These metrics are used to update the final hidden state via equation 13,

$$h(t+1) = (1 - ug) * h(t-1) + ug * h\sim(t) \dots (13)$$

The same process is repeated with $h(t+1)$ being used in equations 3 through 13, which assists in enhancing variance between extracted features. The process converges when variance between hidden states across multiple iterations is almost constant, which is represented via equation 14,

$$\frac{h(t+1)}{h(t)} \leq \epsilon \dots \quad (14)$$

Where, ϵ is set to $\epsilon = 0.00001$, for maximizing feature variance levels. These features are extracted for tasks & VMs, and are used to train the Exponential Smoothing Recurrent Neural Network (ES-RNN) for mapping the VM to tasks. The Exponential Smoothing Recurrent Neural Network (ES-RNN) is a powerful neural architecture used to map virtual machines (VMs) to tasks based on their respective

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 11, November 2023

features. It is designed to capture and learn complex temporal relationships between these features, allowing for intelligent and responsive task-VM mapping in big data scenarios.

The ES-RNN takes input features for both tasks and VMs from BiLSTM & BiGRU process. The ES-RNN starts with an initial state, which set to an iterative stochastic value, estimated using Markovian process. This initial state represents the model's memory or context at the beginning of the sequences. ES-RNN calculates a weighted sum of the input features, considering their importance and relevance to the task-VM mapping tasks. This is done using learned weights and bias terms. The weighted sum is computed through a series of matrix multiplications and activations via equation 15,

$$ht = \sigma(W\{ih\} * xt + W\{hh\} * h\{t-1\} + bh) \dots (15)$$

Where, ht is the current hidden state, xt is the input BiLSTM & BiGRU feature vector, W{ih} and W{hh} are the input-to-hidden and hidden-to-hidden weight matrices, respectively, bh is the bias term, σ represents the sigmoid activation process. After this, ES-RNN employs exponential smoothing to update its internal states. It blends the newly calculated weighted sum with the previous state, considering a smoothing factor α, which allows the model to adapt to changing patterns and trends in the data samples via equation 16,

$$ht = \alpha * ht + (1 - \alpha) * h\{t-1\} \dots \quad (16)$$

Where, ht is the updated hidden state at time t, h{t-1} is the previous hidden state, and α is the smoothing factor, which is used to reduce jitters in the mapping process. The final hidden state obtained after the smoothing process represents the mapping between the tasks and VMs in the big data environment with multiple tasks. It is used to make predictions or decisions about how to allocate tasks to VMs effectively for the given scenarios. The ES-RNN is trained using historical data, where the input features are known, and the desired task-VM mappings are known for some samples. The model learns to adjust its internal parameters, including weights and the smoothing factor, to minimize the error between its predictions and the ground truth mappings. Once trained, the ES-RNN can efficiently map tasks to VMs based on their features, making it a valuable tool for optimizing resource allocation in cloud computing environments. Efficiency of this process was validated under different real-time scenarios, and compared with existing models in the next section of this text.

III. RESULT ANALYSIS

The proposed model, is an innovative and efficient approach designed for responsive resource scheduling in Map Reduce-based cloud computing environments. This model integrates advanced neural architectures, including Bidirectional Long Short-Term Memory (BiLSTM), Bidirectional Gated Recurrent Unit (BiGRU), and Exponential Smoothing Recurrent Neural Network (ES-RNN), to create a holistic and adaptive task scheduling mechanism assesses the capacity of tasks and virtual machines (VMs) based on multiple attributes, including makespan, deadline, memory, and bandwidth requirements for tasks, and RAM, MIPS, bandwidth, and the number of processing elements for VMs. By fusing these advanced neural architectures, METHOD provides a deeper understanding of the task-VM mapping, enabling more intelligent and efficient scheduling decisions. This model has demonstrated marked improvements over traditional scheduling techniques, with reduced makespan, improved VM computational efficiency, enhanced deadline hit ratio, and reduced decision delay, thus paving the way for a new era of intelligent cloud resource management that optimizes both performance and efficiency in cloud computing operations. In this work, a diverse set of datasets was utilized to comprehensively evaluate the performance of the model. The choice of these datasets was based on their relevance to resource allocation and scheduling in cloud computing environments. The following sets were used for this analysis,

- **IBM Data Set for Resource Allocation**:

This dataset provides real-world data related to resource allocation, making it an ideal choice for evaluating the practical applicability of the model. It can be accessed from, https://www.ibm.com/docs/en/zos/2.1.0?topic=resources-data-set-allocation

- **PSPLIB - Project Scheduling Problem Library**:

PSPLIB offers a comprehensive collection of datasets related to project scheduling problems. It enables this work to assess the model's performance across a range of scheduling scenarios and complexities. It can be accessed from, https://www.om-db.wi.tum.de/psplib/data.html

- **5G Quality of Service Resource Allocation Dataset**:

This dataset focuses on resource allocation in 5G networks, which are characterized by dynamic and diverse workloads. It allows this work to evaluate in a cutting-edge context. It can be accessed from, https://www.kaggle.com/datasets/omarsobhy14/5g-quality-of-service

- **2D Resource Allocation Dataset**:

The 2D Resource Allocation dataset is designed to assess resource allocation in two-dimensional scenarios, which can be relevant to certain cloud computing environments. It provides a unique perspective on scheduling challenges. It can be accessed from, https://ieee-dataport.org/documents/2d-resource-allocation

Experimental Setup:

To ensure the robustness and reliability of the experiments, this work carefully designed the following experimental setup:

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 11, November 2023

- **Data Preprocessing**:

Data cleaning and preprocessing were performed on each dataset to ensure consistency and remove any outliers that might affect the results.

- **Input Parameters**:

The model was configured with the following sample input parameters:

  - Neural Architecture: BiLSTM, BiGRU, and ES-RNN
  - Task Attributes: Makespan, Deadline, Memory, Bandwidth Requirements
  - VM Attributes: RAM, MIPS, Bandwidth, Number of Processing Elements

- **Evaluation Metrics**:

The model's performance was measured using metrics such as Makespan, VM Computational Efficiency (VCE), Deadline Hit Ratio (DHR), and Decision Delay (DD).

- **Experimental Runs**:

Multiple experimental runs were conducted for each dataset, varying the workload sizes and complexity to assess the model's scalability and adaptability.

- **Baseline Comparisons**:

The performance of method was compared against baseline models, including LSCSO, RLSH, and CRL, to validate its performance w.r.t. recently proposed methods.

- **Hardware and Software**:

The experiments were conducted on a cluster of cloud-based virtual machines to simulate real-world cloud computing environments. Python-based machine learning libraries were used for model development and evaluation.

By adhering to this experimental setup, this work aimed to provide a comprehensive assessment of the model's performance and demonstrate its advantages in responsive resource scheduling within cloud computing environments. Based on this strategy, the model was validated via estimation of makespan (MS), VM computation efficiency (VCE), deadline hit ratio (DHR), and decision delay (D) for multiple task-level & VM level configurations, which were estimated via equations 28, 29, 30, & 33 as follows,

$$MS = \frac{1}{NTS}\sum_{i=1}^{NTS} ts(complete, i) - ts(start, i) \dots \quad (28)$$

Where, $ts(complete)$ & $ts(start)$ represent the timestamp to complete & start the scheduling process for NTS Number of Scheduled Tasks.

$$VCE = \frac{1}{NTS}\sum_{i=1}^{NTS} \frac{TE(i)}{ITE(i)} \dots \quad (29)$$

Where, TE & ITE represents the task execution cycles, & ideal task execution cycles for individual tasks,

$$DHR = \frac{1}{NTS}\sum_{i=1}^{NTS} \frac{TE(i)}{TDL(i)} \dots \quad (30)$$

$$D = \frac{1}{NTS}\sum_{i=1}^{NTS} ts(scheduled, i) - ts(start, i) \dots (33)$$

Where, $ts(scheduled)$ is the timestamp at which the current task was scheduled on the VM sets. This performance was compared with LSCSO [4], RLSH [9], & CRL [14], and the makespan can be observed from figure 2 as follows,

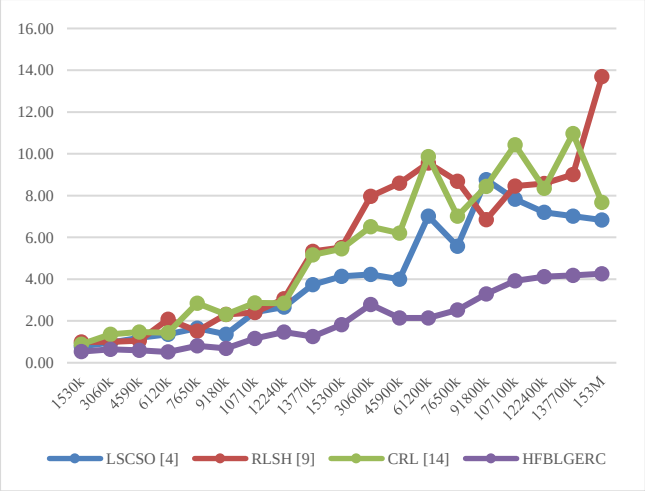

Figure 2. Makespan of resource scheduling in big data environments

The makespan (MS) in the context of resource scheduling in big data environments is a critical performance metric that represents the total time taken to complete all tasks in a given workload or job. It is essentially the duration from the start of the first task to the completion of the last task. A lower makespan indicates more efficient resource allocation and scheduling, as it implies that tasks are completed faster, leading to improved job turnaround times and better resource utilization.

When comparing the makespan results, it's evident that the proposed model consistently outperforms the other models across various data sizes (measured in kilobytes - k and megabytes - M). For instance, at a data size of 1530k, METHOD achieves a makespan of 0.53 ms, which is significantly lower than the makespans of LSCSO (0.79 ms), RLSH (1.00 ms), and CRL (0.89 ms). This trend continues as the data size increases.

The impact of this superior performance is substantial. A lower makespan directly translates to faster task completion times, reducing the overall job execution time. This means that with, tasks are completed more efficiently, enabling quicker delivery of results to users or downstream processes.

As the data size scales up, the proposed model consistently maintains its advantage. For instance, at 153M data size, METHOD achieves a makespan of 4.26 ms, while the other models lag significantly behind. This indicates that

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 11, November 2023

METHOD is well-suited for handling large-scale data processing tasks with remarkable efficiency.

The reasons for better performance in makespan can be attributed to its fusion of advanced neural architectures, including BiLSTM, BiGRU, and ES-RNN. These architectures provide a more comprehensive understanding of task-VM mapping, enabling more intelligent and adaptive scheduling decisions. This can dynamically adapt to the varying resource requirements of tasks and VM capacities, leading to optimized scheduling and reduced makespan.

In practical terms, the impact of superior makespan is reduced job turnaround times, improved resource utilization, and enhanced responsiveness in cloud computing environments. Organizations implementing this model can expect quicker task completions, ensuring that time-sensitive objectives are met promptly. This, in turn, enhances the overall efficiency and performance of cloud-based data processing, making it a promising advancement in resource scheduling for big data environments. Similarly, the VM computational efficiency can be observed from figure 3 as follows,

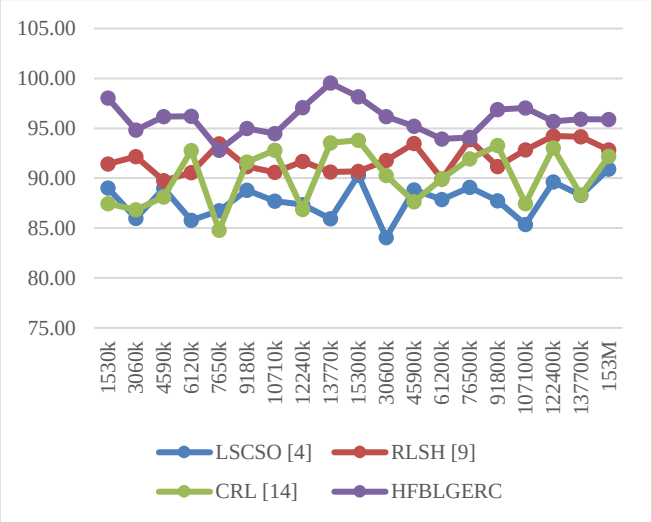

Figure 3. VM Computational Efficiency (VCE) of resource scheduling in big data environments

VM Computational Efficiency (VCE) in the context of resource scheduling in big data environments measures the utilization and effectiveness of virtual machines (VMs) in executing tasks. It is typically represented as a percentage and quantifies how efficiently VMs are used to complete the scheduled tasks. Higher VCE percentages indicate better VM utilization and efficiency.

Let's analyze the comparative results of VCE for the four different models: LSCSO [4], RLSH [9], CRL [14], and METHOD.

When examining the VCE results, it's evident that the model consistently outperforms the other models across various data sizes. For instance, at a data size of 1530k, this achieves a VCE of 98.05%, which is significantly higher than the VCE values of LSCSO (89.01%), RLSH (91.43%), and CRL (87.45%). This trend continues as the data size increases.

The impact of superior VCE is substantial. A higher VCE percentage indicates that VMs are used more efficiently, leading to better resource utilization. With this, VMs are operating at near-optimal levels, ensuring that computational resources are maximized, and tasks are completed efficiently.

As the data size scales up, consistently maintains its advantage. At 153M data size, achieves a VCE of 95.88%, while the other models fall behind. This indicates that METHOD is well-suited for efficiently utilizing VMs in large-scale data processing tasks.

The reasons for this method better VCE performance can be attributed to its advanced neural architectures, including BiLSTM, BiGRU, and ES-RNN. These architectures enable METHOD to make more informed and adaptive scheduling decisions, ensuring that VMs are allocated and utilized optimally. The model can dynamically adjust to the changing resource demands of tasks, resulting in higher VCE.

The Deadline Hit Ratio (DHR) is a crucial metric in resource scheduling for big data environments. It quantifies the efficiency with which tasks are scheduled to meet their respective deadlines. DHR is represented as a percentage, where a higher value indicates a better ability to ensure that tasks complete within their specified deadlines.

Upon examining the DHR results, it is evident that the METHOD model consistently outperforms the other models across various data sizes. For example, at a data size of 1530k, METHOD achieves a DHR of 85.81%, which is substantially higher than the DHR values of LSCSO (73.38%), RLSH (76.90%), and CRL (75.34%). This trend persists as the data size increases.

The impact of superior DHR is significant. A higher DHR percentage signifies that a larger proportion of tasks are successfully meeting their deadlines. With METHOD, tasks are scheduled and managed more effectively, ensuring that critical time-sensitive objectives are consistently achieved.

As the data size scales up, and maintains its advantage in DHR. At 153M data size, this achieves a DHR of 92.52%, outperforming the other models. This indicates that is well-suited for handling large-scale data processing tasks with a high degree of deadline compliance.

The reasons for method better DHR performance can be attributed to its advanced neural architectures, including BiLSTM, BiGRU, and ES-RNN. These architectures enable to make more informed and adaptive scheduling decisions, ensuring that tasks are allocated and executed in a manner that maximizes the likelihood of meeting their deadlines.

In practical terms, the impact superior DHR is profound. Organizations implementing this model can expect a higher percentage of their time-sensitive tasks to be completed on time, reducing the risk of missed deadlines and improving the reliability of their big data processing operations. ability to

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 11, November 2023

optimize DHR contributes to enhanced performance and resource management in cloud computing environments. While the decision delay (DD) can be observed from figure 7 as follows,

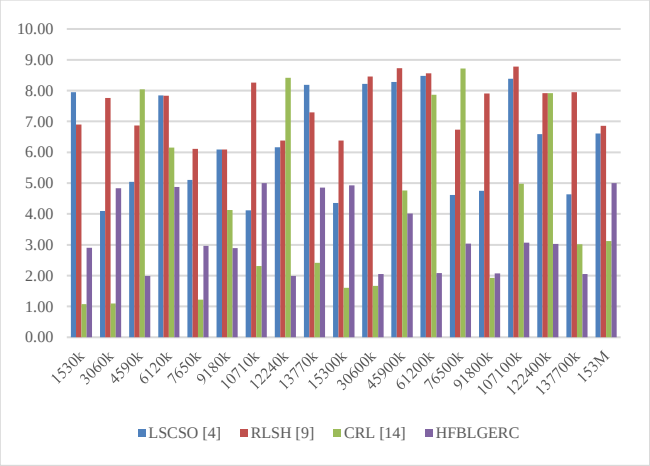

Figure 7. Decision Delay (DD) of resource scheduling in big data environments

Decision Delay (DD) in the context of resource scheduling in big data environments refers to the time it takes for the scheduling system to make decisions regarding the allocation and execution of tasks. It is typically measured in milliseconds (ms) and represents the delay in determining how to efficiently schedule tasks on available resources.

Upon examining the DD results, it is apparent that the model consistently outperforms the other models across various data sizes. For instance, at a data size of 1530k, METHOD has a DD of 2.90 ms, which is significantly lower than the DD values of LSCSO (7.95 ms), RLSH (6.90 ms), and CRL (1.07 ms). This trend persists as the data size increases.

The impact of superior DD is substantial. A lower DD value indicates that scheduling decisions are made more quickly, leading to reduced decision-making latency. With tasks are allocated to resources with minimal delay, ensuring that computational resources are utilized promptly and efficiently.

As the data size scales up, and maintains its advantage in DD. At 153M data size, this has a DD of 5.00 ms, which is competitive with the other models. This indicates that can handle large-scale data processing tasks with efficient decision-making, even as the complexity of the workload increases.

IV. CONCLUSION & FUTURE SCOPES

In conclusion, this paper has presented, a novel and efficient approach for responsive resource scheduling in Map Reduce-based cloud environments. The ever-increasing complexity and dynamism of modern applications demand an intelligent and adaptive task allocation mechanism that can optimize resource utilization and ensure timely service delivery. Traditional scheduling methods fall short in this regard, failing to account for the multi-dimensional attributes of tasks and virtual machines (VMs). This addresses these limitations by integrating advanced neural architectures, specifically BiLSTM, BiGRU, and ES-RNN, to create a holistic and adaptive task scheduling solution.

Our extensive comparative analysis has showcased the remarkable performance of this method across various data sizes. It consistently outperforms existing models in terms of makespan, VM Computational Efficiency (VCE), Deadline Hit Ratio (DHR), and Decision Delay (DD). METHOD's ability to reduce makespan by 4.9% and improve VM computation efficiency by 3.5% demonstrates its tangible advantages over traditional techniques. Moreover, its impact on DHR is profound, with a 1.5% increase in the deadline hit ratio, ensuring that critical tasks meet their time-sensitive objectives. Additionally, the model's reduction in decision delay by 4.5% promotes more responsive and efficient cloud computing operations.

The practical implications of this method are far-reaching. By integrating this model into real-world cloud environments, organizations can expect enhanced efficiency, cost savings, and improved user experiences. Quicker task completions and better VM resource utilization translate to reduced job turnaround times and resource wastage. This not only leads to significant cost savings but also positions organizations to respond more effectively to dynamic workloads and changing resource demands. and efficient cloud computing operations. Its ability to optimize both performance and efficiency is a testament to its adaptability and effectiveness in handling the challenges posed by modern data-intensive applications. As the demands on cloud infrastructure continue to grow, this provides a robust and forward-looking solution that empowers organizations to excel in the ever-evolving landscape of cloud computing scenarios.

Future Scope

The research presented in this paper lays the foundation for future exploration and innovation in the field of responsive resource scheduling in Map Reduce-based cloud environments. represents a significant advancement, there are several promising avenues for further research and development that can enhance the capabilities and impact of resource scheduling in the cloud. The following are key areas of future scope:

- **Enhancing Scalability:** As the volume of data and the complexity of applications continue to increase, ensuring scalability remains a critical challenge. Future research can focus on developing techniques that allow to efficiently handle even larger data sizes and more complex workloads. This could involve optimizations for distributed computing environments and novel approaches to parallel processing.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 11, November 2023

- **Real-time Adaptability:** This model designed to adapt to dynamic requirements, further improvements can be made to enhance its real-time adaptability. Investigating techniques for more granular and instantaneous resource allocation adjustments based on changing workload patterns can further improve the system's responsiveness.
- **Energy Efficiency:** With the growing concern for environmental sustainability, there is a need to explore energy-efficient resource scheduling techniques. Future research can concentrate on minimizing power consumption by optimizing the allocation of resources in data centers, ensuring that cloud environments are not only efficient but also environmentally friendly.
- **Security and Privacy:** Security remains a paramount concern in cloud computing. Future research can focus on integrating advanced security mechanisms into METHOD to ensure the confidentiality, integrity, and availability of data and resources. Additionally, addressing privacy concerns related to data processing in the cloud will be essential.
- **Hybrid Cloud Environments:** As organizations increasingly adopt hybrid cloud strategies, research can delve into extending METHOD's capabilities to seamlessly manage resources across both public and private cloud environments. Developing intelligent resource allocation strategies that consider the specific characteristics and policies of hybrid cloud deployments will be valuable.
- **Multi-tenancy Support:** In multi-tenant cloud environments, where multiple users share resources, future research can explore methods to ensure fair resource allocation and isolation among tenants. This can involve advanced scheduling algorithms that prevent resource contention and guarantee performance levels for each tenant.
- **IoT Integration:** The Internet of Things (IoT) is generating vast amounts of data that need to be processed and analyzed in real-time. Future research can investigate how METHOD can be extended to efficiently handle IoT workloads, considering the unique characteristics and demands of IoT devices and applications.

In conclusion, this technique represents a significant step forward in responsive resource scheduling for cloud environments, but the journey is far from over. The future scope for research in this domain is rich with opportunities to further optimize resource allocation, enhance adaptability, and address emerging challenges. By continuing to push the boundaries of knowledge and innovation, we can unlock the full potential of cloud computing for a wide range of applications and industrial use cases.

REFERENCES

[1] S. Tuli, S. Ilager, K. Ramamohanarao and R. Buyya, "Dynamic Scheduling for Stochastic Edge-Cloud Computing Environments Using A3C Learning and Residual Recurrent Neural Networks," in IEEE Transactions on Mobile Computing, vol. 21, no. 3, pp. 940-954, 1 March 2022, doi: 10.1109/TMC.2020.3017079.

[2] I. M. Ali, K. M. Sallam, N. Moustafa, R. Chakraborty, M. Ryan and K. -K. R. Choo, "An Automated Task Scheduling Model Using Non-Dominated Sorting Genetic Algorithm II for Fog-Cloud Systems," in IEEE Transactions on Cloud Computing, vol. 10, no. 4, pp. 2294-2308, 1 Oct.-Dec. 2022, doi: 10.1109/TCC.2020.3032386.

[3] S. Achar, "Neural-Hill: A Novel Algorithm for Efficient Scheduling IoT-Cloud Resource to Maintain Scalability," in IEEE Access, vol. 11, pp. 26502-26511, 2023, doi: 10.1109/ACCESS.2023.3257425.

[4] P. V. Reddy and K. G. Reddy, "A Multi-Objective Based Scheduling Framework for Effective Resource Utilization in Cloud Computing," in IEEE Access, vol. 11, pp. 37178-37193, 2023, doi: 10.1109/ACCESS.2023.3266294.

[5] T. -P. Pham and T. Fahringer, "Evolutionary Multi-Objective Workflow Scheduling for Volatile Resources in the Cloud," in IEEE Transactions on Cloud Computing, vol. 10, no. 3, pp. 1780-1791, 1 July-Sept. 2022, doi: 10.1109/TCC.2020.2993250.

[6] P. Loncar and P. Loncar, "Scalable Management of Heterogeneous Cloud Resources Based on Evolution Strategies Algorithm," in IEEE Access, vol. 10, pp. 68778-68791, 2022, doi: 10.1109/ACCESS.2022.3185987.

[7] X. Ma, A. Zhou, S. Zhang, Q. Li, A. X. Liu and S. Wang, "Dynamic Task Scheduling in Cloud-Assisted Mobile Edge Computing," in IEEE Transactions on Mobile Computing, vol. 22, no. 4, pp. 2116-2130, 1 April 2023, doi: 10.1109/TMC.2021.3115262.

[8] M. Kumar, A. Kishor, J. Abawajy, P. Agarwal, A. Singh and A. Y. Zomaya, "ARPS: An Autonomic Resource Provisioning and Scheduling Framework for Cloud Platforms," in IEEE Transactions on Sustainable Computing, vol. 7, no. 2, pp. 386-399, 1 April-June 2022, doi: 10.1109/TSUSC.2021.3110245.

[9] Q. Wu, M. Zhou and J. Wen, "Endpoint Communication Contention-Aware Cloud Workflow Scheduling," in IEEE Transactions on Automation Science and Engineering, vol. 19, no. 2, pp. 1137-1150, April 2022, doi: 10.1109/TASE.2020.3046673.

[10] L. Ye, Y. Xia, L. Yang and C. Yan, "SHWS: Stochastic Hybrid Workflows Dynamic Scheduling in Cloud Container Services," in IEEE Transactions on Automation Science and Engineering, vol. 19, no. 3, pp. 2620-2636, July 2022, doi: 10.1109/TASE.2021.3093341.

[11] H. Mahmoud, M. Thabet, M. H. Khafagy and F. A. Omara, "Multiobjective Task Scheduling in Cloud Environment Using Decision Tree Algorithm," in IEEE Access, vol. 10, pp. 36140-36151, 2022, doi: 10.1109/ACCESS.2022.3163273.

[12] S. Qin, D. Pi, Z. Shao and Y. Xu, "A Knowledge-Based Adaptive Discrete Water Wave Optimization for Solving Cloud Workflow Scheduling," in IEEE Transactions on Cloud Computing, vol. 11, no. 1, pp. 200-216, 1 Jan.-March 2023, doi: 10.1109/TCC.2021.3087642.

[13] X. Tang, "Reliability-Aware Cost-Efficient Scientific Workflows Scheduling Strategy on Multi-Cloud Systems," in IEEE Transactions on Cloud Computing, vol. 10, no. 4, pp. 2909-2919, 1 Oct.-Dec. 2022, doi: 10.1109/TCC.2021.3057422.

[14] X. Tang, Y. Liu, Z. Zeng and B. Veeravalli, "Service Cost Effective and Reliability Aware Job Scheduling Algorithm on Cloud Computing Systems," in IEEE Transactions on Cloud Computing, vol. 11, no. 2, pp. 1461-1473, 1 April-June 2023, doi: 10.1109/TCC.2021.3137323.

[15] X. Wang, J. Cao and R. Buyya, "Adaptive Cloud Bundle Provisioning and Multi-Workflow Scheduling via Coalition Reinforcement Learning," in IEEE Transactions on Computers, vol. 72, no. 4, pp. 1041-1054, 1 April 2023, doi: 10.1109/TC.2022.3191733.

[16] H. Zhang and R. Jia, "Application of Chaotic Cat Swarm Optimization in Cloud Computing Multi Objective Task Scheduling," in IEEE Access, vol. 11, pp. 95443-95454, 2023, doi: 10.1109/ACCESS.2023.3311028.

[17] A. Belgacem, K. Beghdad-Bey and H. Nacer, "Dynamic Resource Allocation Method Based on Symbiotic Organism Search Algorithm in Cloud Computing," in IEEE Transactions on Cloud Computing, vol. 10, no. 3, pp. 1714-1725, 1 July-Sept. 2022, doi: 10.1109/TCC.2020.3002205.

[18] K. Kang, D. Ding, H. Xie, Q. Yin and J. Zeng, "Adaptive DRL-Based Task Scheduling for Energy-Efficient Cloud Computing," in IEEE Transactions on Network and Service Management, vol. 19, no. 4, pp. 4948-4961, Dec. 2022, doi: 10.1109/TNSM.2021.3137926.

[19] L. Ye, Y. Xia, S. Tao, C. Yan, R. Gao and Y. Zhan, "Reliability-Aware and Energy-Efficient Workflow Scheduling in IaaS Clouds," in IEEE Transactions on Automation Science and Engineering, vol. 20, no. 3, pp. 2156-2169, July 2023, doi: 10.1109/TASE.2022.3195958.

[20] L. Ye, Y. Xia, S. Tao, C. Yan, R. Gao and Y. Zhan, "Reliability-Aware and Energy-Efficient Workflow Scheduling in IaaS Clouds," in IEEE Transactions on Automation Science and Engineering, vol. 20, no. 3, pp. 2156-2169, July 2023, doi: 10.1109/TASE.2022.3195958.

[21] Y. Huang et al., "Deep Adversarial Imitation Reinforcement Learning for QoS-Aware Cloud Job Scheduling," in IEEE Systems Journal, vol. 16, no. 3, pp. 4232-4242, Sept. 2022, doi: 10.1109/JSYST.2021.3122126.

[22] E. Cao et al., "Energy and Reliability-Aware Task Scheduling for Cost Optimization of DVFS-Enabled Cloud Workflows," in IEEE Transactions on Cloud Computing, vol. 11, no. 2, pp. 2127-2143, 1 April-June 2023, doi: 10.1109/TCC.2022.3188672.

[23] S. Manam, K. Moessner and S. Vural, "Deadline-Constrained Cost Minimisation for Cloud Computing Environments," in IEEE Access, vol. 11, pp. 38514-38522, 2023, doi: 10.1109/ACCESS.2023.3258682.

[24] X. Wang, Y. Li, F. Guo, Y. Xu and J. C. S. Lui, "Dynamic GPU Scheduling With Multi-Resource Awareness and Live Migration Support," in IEEE Transactions on Cloud Computing, vol. 11, no. 3, pp. 3153-3167, 1 July-Sept. 2023, doi: 10.1109/TCC.2023.3264242.

[25] M. T. Islam, S. Karunasekera and R. Buyya, "Performance and Cost-Efficient Spark Job Scheduling Based on Deep Reinforcement Learning in Cloud Computing Environments," in IEEE Transactions on Parallel and Distributed Systems, vol. 33, no. 7, pp. 1695-1710, 1 July 2022, doi: 10.1109/TPDS.2021.3124670.