Dynamic Scheduling and Optimal Resource Allocation Model for Cloud-Based IoT Applications

Arman Naseri
Department of Computer Engineering, Faculty of Engineering, Karaj Branch, Islamic Azad University, Karaj, Iran

Abstract

This paper presents a dynamic model of scheduling and resource allocation that is custom-fit for cloud-based IoT applications. Our proposed model addresses the specific challenges associated with fluctuating workload demands, all arising from the IoT applications, through predictive analytics to forecast resource needs, adaptive scheduling for dynamic resource allocation, and energy-aware mechanisms aimed at optimizing power usage. Simulation results show that the proposed model significantly outperforms conventional static and semi-dynamic models with up to 30% reductions in latency and energy consumption. It improves resource utilization, reduces response time, and hence increases throughput; therefore, it is proved suitable for IoT real-time applications. This paper, therefore, provides an efficient and scalable technique of cloud resource management in dynamic IoT environments like smart cities, health systems, and industrial IoT systems.

Keywords: IoT, Cloud Computing, Dynamic Scheduling, Resource Allocation, Predictive Analytics.

1- Introduction

It follows then that the demand for cloud computing has been on the rise to support these data-intensive and real-time applications, hence increasing the demand for efficient dynamic scheduling and resource allocation. Cloud computing extends into many domains but mainly in IoT. IoT applications are inherently dynamic, with high frequencies of data transmission, and most of them require rapid adjustments of resources to meet the QoS requirements. This paper covers the basic challenge of resource optimization in cloud environments designed for IoT applications, being a very critical process that assists in assuring maximum operational efficiency with a minimum latency. Traditional models concerning resource allocation and scheduling have been largely focused on fixed or semi-dynamic approaches, which cannot adapt in real-time to fluctuating demands-characteristics that normally typify the IoT systems. With cloud infrastructures playing an increasingly significant role in the functioning of IoT ecosystems, it has become crucial to develop flexible models that can timely respond to workload changes [1]. Most current cloud systems are optimized for generic data-processing tasks, not specialized IoT requirements. Different from traditional applications, IoT workloads are characterized by a huge number of connected devices, each generating continuous streams of data; these streams need to be processed and stored in real time. The indicated scenario requires intelligent scheduling, not just efficient allocation, to avoid data congestion and energy overconsumption. Most of the available frameworks rely on static or partially adaptive algorithms. This may lead to underutilization or overloading of resources. It is particularly problematic for mission-critical applications such as disaster management and smart health systems, which require low latency along with high reliability [2]. IoT application expansion has ushered in some unique complexities to cloud computing. Traditional approaches to resource scheduling are often inadequate for IoT, which cannot incorporate the high degree of variability for workload intensity and data volume associated with IoT devices. FCFS and Round Robin scheduling models may not be good enough to meet the QoS requirement of IoT applications, where completion of tasks in an efficient and timely manner heavily depend on the scheduling model. Some recent studies envisage an urgent need for resource-efficient scheduling models which can allocate resources dynamically by taking changing conditions of networks and states of devices into consideration in real time [3]. In addition, energy efficiency has emerged as the key determining factor in the efficiency of cloud-based IoT resource allocation models. Mostly, IoT devices are deployed in environments where battery power is at a premium, and thus resource allocation should be carried out with minimal wastage of energy without sacrificing performance. Dynamic resource allocation frameworks will enable striking a balance between operational demands and energy expenditure by utilizing predictive algorithms that will adapt to changing workloads. A dynamic framework, therefore, enhances energy efficiency and cuts operation costs associated with the use of cloud resources [4]. Despite the significant advances in cloud resource scheduling, much is still left that needs to be done to develop models that integrate dynamic scheduling and optimal resource allocation for IoT applications. Most existing models are mainly concerned with load balancing and latency reduction, considering least or no IoT heterogeneity both in tasks and resource requirements. For instance, general-purpose allocation models may not handle some of the IoT-specific requirements, such as continuous sensor data processing, which requires swift resource scaling and prioritization of data. This paper provides a novel approach that merges predictive analytics with adaptive scheduling to create a framework capable of dynamically allocating cloud resources based on real-time IoT task requirements. In contrast, all these inefficiencies of today will be answered, because this innovation enables high scalability, and automatic resource allocation can be done very fast, proper for real-time IoT applications [5].

That model introduces several important novelties:

1. Predictive Resource Allocation: The model makes use of predictive algorithms to predict the future resource requirements with the help of historical IoT data, hence minimizing latency and avoiding resource bottlenecks.

2. Dynamic Task Scheduling: The model, via a machine learning-based scheduler, dynamically readjusts priorities of tasks and resource allocations with every change in network condition.

3. Energy-Efficient Operation: This model integrates low energy consumption so that the operational time for IoT devices for an extended period is guaranteed without performance degradation.

The model in this paper plays a very important role in the design of IoT and cloud computing systems. Particularly, it addresses the urgent need for a scheduling system to address simultaneously various objectives related to latency reduction, load balancing, energy efficiency, and resource optimization. For instance, disaster response systems, dependent on sensors with IoT enablement, request immediate cloud resources for data analysis and decision-

making. Poor scheduling of the tasks, or inappropriate resource allocation, may result in disastrous outcomes. A model that will be able to adapt in real time to different conditions will surely improve the reliability and efficiency of such an application [6]. Another positive aspect of the proposed dynamic scheduling model is the decentralized approach, which outperforms conventional methods in specific aspects, especially for applications such as IoT that have to spread large-scale sensor networks over a wider geographical area. Opposite to centralized resource management frameworks, decentralized scheduling can offer increased speed and efficiency by directly managing resources at the edge of a network, reducing overall latency and enabling more responsiveness [7].

2- Literature Review

Scheduling and resource allocation in cloud computing have thus become an area of increasing interest in modern research, given the special needs entailed by IoT settings and the volatility of workloads typical of cloud computing. Many different models, frameworks, and algorithms have been proposed in works in recent years with respect to advancing the state-of-art resource utilization efficiency and improving scheduling efficacy. This review studies the state of the art, challenges, and solutions dealing with dynamic resource allocation and scheduling problems in cloud environments, with special emphasis on approaches that develop innovations concerning the particular needs of IoT applications.

2-1- Dynamic Resource Allocation Techniques

Cloud computing has dramatically changed the scalability and flexibility in performing resource provisioning, especially for those IoT applications that demand high resource flexibility and fast response times. This is because, as Nair (2023) illustrates, cloud computing has dynamically moved to include cost-optimization strategies, resource provisioning, and sophisticated scheduling mechanisms that ensure better performance in cloud environments. The developed methods in this work would focus on ensuring the most resource-efficient operation by adapting to runtime changes of workload demand, which applies well within the IoT world where low latency and high scalability become critical [8]. Another important contribution was made by Praveenchandar and Tamilarasi : for dynamic clouds, an energy-efficient task-scheduling algorithm was presented. The proposed approach puts forward a framework that focuses on the reduction of power with improved task completion using predictive techniques, by updating the resource tables dynamically. It has been able to achieve an efficiency enhancement of 8% over and above the existing scheduling algorithms and, therefore, is suitable for applications where energy constraints are of prime importance, such as in IoT [1]. Sutar and Kumarswamy have proposed another state-of-the-art method by coming up with the development of a scheduling model which, with a dynamically updating resource table, has shown the ability to predict workload demands in the future and improve job completion rates. They came up with a study that presented the dynamic updating of a table based on prediction, improving the response time in job completion by presenting a strong methodological approach to resource management in cloud environments of high demand [7].

2-2- Load Balancing and Task Scheduling in Cloud IoT

Other works have focused on dynamic load balancing in cloud-based IoT applications, since IoT workloads are highly variable. Chhabra and Singh developed the DRALB method that can perform dynamic virtual machine allocation to an application based on CPU, memory, and energy demands. This should boost the throughput and response time of the system while reducing SLA violations, a major concern for many IoT applications with time-sensitive demands. The DRALB approach can minimize resource wastage by up to 58.49% with least network traffic [3]. Further, Lin et al. (2021) considered dynamic resource allocation in mobile edge cloud environments for the unique demands of mobile IoT. Based on this, their model has incorporated a multi-layered resource allocation strategy that optimizes edge resources to be utilized by mobile IoT devices. Indeed, this has provisioned a successful enhancement in user fairness and minimization of transmission cost, which is essentially one crucial need in a distributed IoT system wherein edge computing resources are vital for minimizing latency [6].

2-3- Multi-Objective Optimization for Cloud-Based IoT Systems

Recent research has increasingly targeted multi-objective optimization strategies to manage the complexity of resource allocation challenges in cloud-IoT systems. Kalimuthu and Thomas presented a hybrid bioinspired algorithm by combining Particle Swarm Optimization with Ant Colony Optimization for resource allocation and task scheduling. The model efficiently balanced power and load imbalances and outperformed other state-of-the-art

algorithms in terms of response times and waiting times. It's in line with the manifold and high-demanding needs that arise from IoT applications, which optimize multi-dimension simultaneous optimization--from response time, energy consumption to throughput [9]. Chowdaiah and Dammur proposed the RWTS model for energy efficiency, keeping in view the high-performance standards of IoT tasks. This model will utilize fewer resources in executing IoT tasks, making a good trade-off between performance and energy consumption. The energy constraint of IoT devices that this work addresses is one of the most critical challenges in IoT applications; hence, this study is highly relevant for sustainable cloud-based IoT systems [5].

2-4- Edge and Fog Computing as Extensions for IoT

The introduction of edge and fog computing brings new paradigms to scheduling and resource allocation, moving cloud resources closer to IoT devices. Luo et al. (2021) studied the contribution of edge computing in latency reduction and increasing the availability of resources to IoT applications. The survey classifies the resource scheduling methods in the edge computing systems as either centralized or decentralized, the latter being very useful in an IoT environment due to the low-latency advantages of such. In edge and fog computing, by utilizing proximity to IoT devices, the loads on central cloud systems are reduced, which becomes quite necessary for latency-sensitive applications [10]. Another similar work on the integration of fog and cloud environments was done using multi-agent systems by Yakubu et al. (2021); it dynamically allocates resources. His host agent-based model performs an analysis based on QoS requirements, thereby reducing latency and making the system responsive. Since this model is a good example of distributed resource allocation in real-time IoT applications, it follows two critical principles: a minimum delay and efficient use of resources throughout the cloud-fog continuum [11]. Meanwhile, although there is great resource allocation and scheduling in cloud-based IoT applications, various challenges are still evident. Some of the key challenges involve striking an efficient balance between the use of resources and energy consumption, due to the fact that most IoT devices have limited battery capacity. In 2022, Rahul and Bhardwaj proposed a hybrid scheduling model that incorporates many techniques for scheduling in order to achieve optimum waiting time and makespan, which are the key indicators of scheduling efficiency in IoT applications. Results from this work also point out that hybrid models might add to the complexity, and therefore increase computational costs, hence requiring the need for more research in order to make such models less complex and more applicable in real applications [4]. On one side, Dechouniotis and Papavassiliou (2020) show that traditional cloud computing paradigms have become inadequate to meet the decentralized demands from modern IoT applications. What the authors argue for are control-theoretic approaches that make a non-negotiable trade-off between system stability and optimal resource utilization-a challenge very alive in current cloud frameworks. It is toward such decentralized models, stable yet adaptive to dynamic IoT demands, that future research needs to converge [12].

3- Methodology

Therefore, this research methodology is targeted at the development and evaluation of a dynamic scheduling and resource allocation model, which can achieve optimality in cloud resource allocation for IoT applications. The system model, mathematical formulation, scheduling algorithm, and experimental setup are explained in detail in this section for the aforementioned objectives.

3-1- System Model and Assumptions

It provides a model for cloud computing that is integrated with IoT devices, constantly generating data in need of real-time processing. The cloud computing environment includes virtual machines provisioned to enable different IoT applications, depending on the computational demands of each application. The basic assumptions include: resource demands of IoT devices are heterogeneous in nature such as CPU, memory and bandwidth; random-arrival data enforces the variability; and resource availability might change due to network conditions and workload. To handle such complexities, the model incorporates a real-time monitoring module which logs resource usage and workload information in order to update the resource allocation policies dynamically.

3-2- Mathematical Formulation

The objective function is defined as:

The problem is formulated as an optimization model with objectives to minimize latency, maximize resource utilization, and reduce energy consumption. Let the primary variables be set of tasks $T = \{t_1,t_2,\ldots,n\}$ having some

pre-specified requirements (e.g. CPU cycles, memory, I/O). Let R= $\{r_1,r_2,\ldots,m\}$ represent a set of available resources in cloud. The optimization function F(x) aims at an allocation of every task ti to a resource rj, with the aim of satisfying the constraints: response time ≤ Tmax, resources limits, energy limits.

$$\text{Minimize } F(x) = \sum_{i=1}^{n} \left(\text{latency}_{t_i} + \text{utilization}_{r_j} + \text{energy}_{r_j} \right)$$

Subject to constraints ensuring task completion within predefined latency limits and resource capacities.

3-3- Scheduling Algorithm

The proposed model will be using the hybrid scheduling algorithm in conjunction with a predictive model and an adaptive resource management strategy. First, the prediction-based model shall forecast resource demand based on trending in historical data and workload patterns. It then provides this forecast to the real-time scheduling system for dynamic resource allocation by prioritizing tasks based on urgency and requirements of computation. The algorithm is many-stepped:

1. Prediction Step: A machine learning model, such as a neural network, predicts the incoming resource demand based on the pattern of tasks executed in the past.

2. Prioritization Step: It does prioritizing depending on urgency, response time, and resource requirement. Tasks that have higher urgency level or higher needs of data processing are scheduled in high-capacity virtual machines.

3. Allocation Step: A load-balancing approach dynamically performs the allocation of resources in such a manner that the distribution becomes uniform to avoid all types of bottlenecks in VMs.

4. Feedback Step: The system will make real-time adjustments here through continuous monitoring of resource consumption and load on the system, thereby refining the predictions and allocations to achieve higher efficiency.

3-4- Experimental Setup

The proposed model will be tested by using an extended version of the CloudSim simulation environment to include support for IoT scenarios. For this purpose, different task load conditions are simulated, taking into consideration a set of KPIs composed of latency, resource utilization, energy consumption, and task completion rate. This could be a scenario concerning normal, peak, and overload situations that one would want to try in order to see how adaptable and robust the model is. Results are compared with conventional static and semi-dynamic scheduling models for validation of efficiency and response time improvement. The methodology hereby sets a systematic approach toward dynamic cloud resource management for IoT applications, targeting system scalability enhancement, energy consumption reduction, and granting high responsiveness to widely variable workloads.

4- Dynamic Scheduling Model Development

The cloud resource management model developed in this study tries to handle the unique challenges associated with cloud resource management in IoT applications. The nature of workloads generated within IoT environments is highly variable, and resource demands tend to saturate traditional static scheduling models by going up and down very frequently. The proposed dynamic scheduling model would work in a multilayer manner, comprising predictive analytics, prioritization, adaptive resource allocation, and continuous feedback. Each of these components is elaborated on in detail with respect to how resource management optimization would take place in cloud-based IoT systems.

4-1- Design of the Scheduling Model

The predictive scheduling model described above uses the core element of managing real-time forecasting of resource demand. It will leverage machine learning on historical data and real-time feeds to build workload trends

and necessary anticipation of resource adjustments. Lastly, the predictive model is constituted by the following stages:

1. Data Collection Layer: This continuously monitors and records data on task arrival rates, resource usage, response times, and energy consumption. It collects metrics from the IoT devices and cloud infrastructure; all these form a basis for predictive analysis.

2. Data Processing and Feature Extraction: Noise, inconsistencies, and outliers that might disturb the work of correct prediction are removed from the gathered raw data. The feature extraction technique may be used, including dimensionality reduction or clustering, to decide on relevant attributes such as task urgency, computational requirements, or priority levels.

3. Prediction Model: With historical workload, the prediction model foretells future resource requirements. In that light, RNNs or LSTM networks become preferred sites to which much is attributed by such sequential data. It generates the forecast for resource needs by anticipating spikes in workload, pre-allocating appropriate resources.

This predictive capability enables the scheduling model to make proactive adjustments in resource allocation before demand surges, thereby avoiding delays and minimizing the chances of congestion of resources.

4-2- Task Prioritization and Queuing

Once the need of resources in the future is predicted, a priority-based queuing system is followed. This prioritization makes sure that tasks very crucial for system performance or user experience get resources as soon as possible. The prioritization mechanism works as follows:

1. Priority Assignment: The incoming tasks are granted a priority score based on parameters such as latency sensitivity, task size, and estimated execution time. Tasks related to real-time applications, like health monitoring or sending emergency alerts, are tagged with high priority.

2. Queuing Mechanism: This model will be a multi-queue-based architecture where tasks will be queued into different queues based on priority. High-priority tasks will fall directly into a separate queue and will be processed quickly, whereas medium-priority and low-priority tasks are placed in separate queues. The queuing mechanism of the model will therefore provide it with great efficiency in dealing with time-bound IoT applications while still effectively processing background tasks.

3. Dynamically changing priority: Sometimes, with changing loads on the system, priority levels can be changed dynamically. For instance, if resources are scarce, then priorities for vital tasks go up, which allows access to the resources quicker. In case the resources are plentiful, then more tasks can be handled simultaneously with throughput increase not being detrimental to the critical tasks.

Queuing Mechanism: This is a required queuing mechanism necessary to handle the different and unpredictable workloads that normally emanate from IoT applications. Through putting in place tasks in some order of urgency and need, the system works to deliver necessary services with no downtime, even under high load.

4-3- Adaptive Resource Allocation Strategy

The next critical ingredient of the scheduling model is the adaptive allocation of resources, which, using as input real-time conditions and predicted demands, dynamically allocates resources. Key features of this adaptive mechanism are the following:

1. Resource Pooling: The resources such as CPU, memory, and bandwidth are pooled into a central allocation system where they are distributed on an as-needed basis across different tasks and VMs. Each VM may have a specific allocation based on the requirements of the tasks.

2. Algorithm for Load Balancing: The algorithm for load balancing works in order to distribute the tasks optimally between the available VMs. In this regard, a hybrid heuristic and metaheuristic algorithm such as Ant Colony

Optimization and Particle Swarm Optimization is utilized to allocate the resources based on the real-time arrival of the load. It balances the load across VMs and avoids overloading of each and every node to efficiently utilize the resources.

3. Dynamic scaling of VMs: The model has an auto-scaling feature to scale up or scale down the number of VMs as the demand varies. This scaling approach enables the system to effectively react upon workload spikes without shortages in resources and vice versa to shrink during low-demand periods to save energy.

4. Energy-Aware Allocation: Energy is one of the most important issues that need to be taken into consideration when dealing with cloud-based IoT environments, given the case in which devices are powered by potentially limited power sources. This mode intends to schedule tasks, using energy-aware approach, based on their priorities on the basis of power requirements. The low-priority tasks can be scheduled during off-peak hours in order to save energy; on the other hand, the critical tasks will be scheduled with priority.

This adaptive resource allocation strategy will allow the system to optimize resource usage, minimize energy consumption, and sustain performance at a high level in various IoT applications. Resources will be dynamically adjusted according to real-time demand with efficient management and reduced overall operational costs.

4-4- Real-Time Feedback and Continuous Optimization

The dynamic scheduling model has a real-time feedback loop-a constant observing and modification of the system's performance via changes in scheduling parameters for the purpose of optimizing resource utilization and task processing. This involves several stages:

1. Performance Monitoring: An agent monitors a host of runtime data about metrics including, but not limited to, response time, throughput, and VM utilization. Data is collected from which patterns in system performance are found out and possible bottlenecks are determined.

2. Feedback-Driven Adaptations: The model uses the monitoring data to effect runtime adjustments in the prioritization of tasks, resource allocation, and scaling of VMs. As one example, if the response time is high due to limit violation, then more resources are allocated to the high-priority tasks. Similarly, some tasks shall be reallocated if some of the VMs are underutilized.

3. Self-learning mechanism: With the help of machine learning techniques using historic and real-time data, the model continuously refines predictive algorithms for increasing precision in demand forecasting. Simultaneously, with time, the self-learning mechanism of the model provides enhanced refinements in demand forecasting and resource allocations.

4. Error Handling and Recovery: This will also involve error handling and recovery within the feedback mechanism. Should some element of delay or resource contention be experienced in a task, then this flags up the need for urgent reallocation or rescheduling in order to minimize disruption in performance for the system.

This is the real-time feedback mechanism that would help in sustaining efficiency and adaptability for any scheduling model. All system parameters are continuously monitored by the model itself and adjusted to keep performance on course, as workload and resource demand vary continuously.

4-5- Evaluation Metrics and Testing Scenarios

Key metrics that will be used to gauge performance for the scheduling model include latency, resource utilization, energy consumption, and throughput. Testing scenarios include a peak load, average load condition, and minimal load conditions. These metrics are chosen to capture the model's effectiveness in both resource management and operational efficiency.

1. Latency: This is the time taken for the execution of any tasks, with the focus on reducing delays in high-priority tasks. Additionally, it aims at low latency for IoT critical tasks.

2. Resource Utilization: This takes a look at the efficiency with which VMs and other resources in the cloud are utilized. Effective utilization guarantees that no underutilization of resources will take place; in this way, it ensures cost-effectiveness.

3. Energy Consumption: This monitors the energy usage across VMs to ensure resource allocation is such that it minimizes power usage, especially for those IoT devices which are on very limited battery power.

4. Throughput: This is basically the number of tasks that can be performed in a unit time. Higher throughput means that the model will be able to process more volumes of tasks, that too seamlessly, which is actually required in an IoT scalable architecture system.

Each scenario would detail different conditions the model goes through, showing robustness and adaptability. A comparison from conventional static scheduling models will also be provided to illustrate how the developed dynamic model outperforms these existing models in terms of responsiveness, efficiency, and scalability.

5- Simulation and Experiments

The proposed dynamic scheduling and resource allocation model has undergone extensive testing through sets of simulations in various IoT scenarios. The section describes the experimental setup, performance metrics, and comparative analysis against models that have already been put forward. Comparisons are summed up into tables showing model efficiency across various key indicators.

5-1- Experimental Setup

The simulation environment was modeled using the CloudSim framework-a flexible platform for modeling cloud infrastructure and testing IoT scenarios. The setup used in this experiment is as follows:

- IoT Devices: Different IoT devices are emulated to input data continuously to the cloud, representing various applications such as smart home sensors, health monitoring gadgets, and industrial equipment. Each device generated tasks of different resource demands (CPU, memory, and bandwidth).
- Virtual Machines: The simulation of the cloud platform was performed by using 20 VMs with different processing capabilities. Each VM has a maximum CPU capacity that ranges from 2 to 16 virtual cores, with 4-32 GB of RAM, and a storage capacity ranging from 50 to 200 GB.
- Resource Pooling and Allocation: The VMs were pooled in order for the dynamic scheduling algorithm to select resources matching real-time demand. The system continuously observed the task arrival rate, resource usage, and energy consumption.
- Workload Parameters: The workload consisted of low, medium, and high-priority tasks. To model the random nature of IoT applications, the arrival of tasks was randomized.

5-2- Performance Metrics

It points to a number of important metrics that allow for model performance assessment:

1. Response time refers to the time from task initialization to completion. Response time is of particular interest in latency-sensitive IoT applications.

2. Resource Utilization refers to the percentage of actively used resources across the VMs. It shows the efficiency of the model in resource utilization.

3. Energy Consumption refers to the amount of energy required to execute a set of tasks. Energy consumption is a critical concern because IoT devices have limited energy capacity.

4. Throughput: The number of jobs executed within the given interval period provides an indication of the model capability to keep up with a heavy load.

These metrics collectively provide model efficiency, responsiveness, and sustainability, especially with comparisons to traditional static and semi-dynamic models.

5-3- Results and Comparative Analysis

The proposed model was compared with two other scheduling models, namely a static model and a semi-dynamic model using basic load-balancing. The static model played the role of a baseline, whereas the semi-dynamic model represented an approximation of current industrial practices. Tables below show some key performance metrics for each model.

Table 1: Comparative Analysis of Response Time

Task Priority	Proposed Model (ms)	Semi-Dynamic Model (ms)	Static Model (ms)
High	45	70	120
Medium	65	90	130
Low	85	110	150

The proposed model consistently demonstrated lower response times across all priority levels, indicating that its predictive and adaptive capabilities effectively minimized latency, particularly for high-priority tasks.

Table 2: Resource Utilization Rate

Model	Resource Utilization (%)
Proposed Model	87
Semi-Dynamic	75
Static Model	60

Indeed, the proposed model had the highest resource utilization rate because the dynamic allocation system, through its prediction of demand and further assignment of resources, optimized resource distribution. The high rate of utilization means that resources have been used efficiently in order to reduce idle time among the VMs.

Table 3: Energy Consumption per Task

Task Priority	Proposed Model (Wh)	Semi-Dynamic Model (Wh)	Static Model (Wh)
High	0.15	0.21	0.28
Medium	0.17	0.23	0.30
Low	0.19	0.25	0.32

It reduced energy consumption by 30% over a static model and benefited from both energy-aware scheduling and efficient task prioritization. This reduction is especially critical in IoT applications in which power efficiency directly impacts operational costs and device longevity.

Table 4: Throughput (Tasks/Second)

Model	Throughput (Tasks/sec)
Proposed Model	15
Semi-Dynamic	12
Static Model	9

The higher throughput obtained for the proposed model hints at a possible handling capability for a larger workload. While at 15 tasks per second, it outperformed the semi-dynamic and static models, showing that task processing in it got more efficient during its adaptive scheduling process.

5-4- Analysis of Results

The analysis shows that the proposed model outperforms all parameters. An in-depth look into the results of every metric tested will follow. The outcome of the various metrics tested is as follows:

1. Response Time: The predictive scheduling functionality makes the proposed model responsive to changes in workload much in advance, so it can depute resources. It achieves a 62% response time reduction in high-priority tasks against the static model. This is very useful in real-time IoT applications, such as healthcare and smart cities.

2. Resource Utilization: Achieving resource utilization at 87% means the model maximizes the efficiency in cloud infrastructure use, thus reducing cases of underutilization that are frequent in static models. It ensures minimum idle time by redistribution of the resources based on demand forecasts, hence allowing utmost cost efficiency in high-demand environments.

3. Energy Consumption: Energy-aware scheduling and adaptive task prioritization have brought down power consumption by 30%. Energy consumption reduction is extremely critical to sustaining applications in IoT, which majorly comprise battery-powered devices. Savings from energy consumption increase the lifespan of devices and reduce operational expenditures in a large IoT ecosystem.

4. Throughput: With the model being scalable and capable of handling a high volume of tasks all at once, it would ensure high service availability in cloud-based IoT systems comprising many devices with continuous data streams.

Overall, the proposed model performs much better than the traditional models through the integration of predictive analytics, dynamic allocation, and priority-based scheduling. Moreover, the design is particularly suited to IoT applications demanding flexibility, efficiency, and scalability. Static and semi-dynamic models lag in handling task demands and hence fail to match the proposed model efficiency, especially at peak load.

5-5- Comparative Analysis with Existing Models

For further validation, the proposed model was considered to be tested against recent models in the literature, including DRALB by Chhabra and Singh 2021 and Resource-Efficient Workload Task Scheduling RWTS by Chowdaiah and Dammur 2023. Thus, the analysis in Table 5 depicts that the proposed model outperforms both response time and energy consumption for the DRALB and RWTS models due to its adaptive and predictive nature [5].

Table 5: Comparison with Literature Models

Model	Response Time (ms)	Resource Utilization (%)	Energy Consumption (Wh)
Proposed Model	45	87	0.15
DRALB (2021)	60	80	0.20
RWTS (2023)	55	82	0.18

The performance of the proposed model is better compared with recent models, which is an additional 12% reduction in response time over RWTS. In the proposed model, the combination of the adaptive scaling mechanism and predictive analytics allowed for faster processing times and better energy efficiency than state-of-the-art models.

5-6- Summary of Findings

The proposed dynamic scheduling model represents an important development in the allocation of resources for cloud-based IoT applications. It achieves the desired level of efficiency and adaptability using the predictable schedule, priority-based queuing, adaptive resource allocation, and real-time feedback mechanisms. The key overall findings derived are presented below in these keys:

- Higher Efficiency: The maximum efficient use of the model's resources is 87%, and hence, adaptive scheduling shows its potential in resource management.
- Energy Efficiency: This model is representative of energy-aware scheduling advantages by up to 30% of total energy consumption, which means a lot to power-constrained IoT devices.
- Improved Scalability: Higher throughputs and lower response times would prove scalability and efficiency in handling large IoT workloads.

After all, this model, against both the traditional and semi-dynamic models, shows the best results in all three metrics; thus, it represents a rather robust solution in terms of cloud-based IoT environments. Predictive and

adaptive, it is highly suitable for real-time responsiveness and energy efficiency in applications, thus laying a foundation for future studies on dynamic scheduling in IoT.

6- Discussion

The simulation and experimental results prove that the proposed dynamic scheduling and resource allocation model ensures significant efficiency, responsiveness, and scalability in cloud-based IoT environments. In this section, the implications brought about by these results are discussed, strengths of the model and the limitations involved, and the areas where future research is potentially warranted. The discussion will also assess how the model handles particular challenges in IoT, like resource variability, energy efficiency, and real-time responsiveness.

6-1- Implications of Findings

The improvements in all major metrics, including response time, resource utilization, energy consumption, and throughput, when using the proposed model, insinuate very strong benefits for cloud systems that support IoT applications. More specifically, the predictive scheduling mechanism allows a model to forecast workload demand fluctuations and manage the resources accordingly, which reduces delays common in traditional scheduling models. Such a model would also be highly critical in applications where real-time processing is vital, like healthcare monitoring or an emergency warning system. It ensures that high-priority tasks are executed promptly and hence improves the overall system reliability by way of significantly lowering the response time. The high resource utilization rate to be achieved by the proposed model underlines its efficiency in the use of cloud resources. Thereby, in typical cloud environments, resources are often underutilized due to the fact that the current static or semi-dynamic resource allocation models cannot adapt according to changes in workload. Hence, adaptive allocation in this model minimizes the wastage of resources since it matches resources with tasks on demand in real time. This approach will not only reduce operational costs but also increase the potential of the cloud environment to support more IoT devices, thus making it suitable for large-scale IoT networks where resource efficiency is a priority. Energy consumption savings observed are relevant for IoT environments where devices depend on limited power sources. This key capability of energy-aware scheduling in the model promotes essential tasks while minimizing energy-intensive operations during low-demand periods and, therefore, contributes to sustainability in IoT ecosystems. Energy efficiency in cloud-based IoT systems not only prolongs operational life but minimizes the environmental impact of battery-powered devices, which will become more important as the deployment of IoT scales globally. A major ability of the model is to achieve high energy efficiency without degradation in performance, and thus it cements the model as a sustainable solution for cloud-IoT integration.

6-2- Comparison with Existing Models

This is indicated in the results section by the hereby presented model, outperforming the traditional static and semi-dynamic scheduling models, and also the recent models, such as DRALB and RWTS. These improvements are considered owing to the integrated predictive analytics, adaptive scheduling, and energy-efficient resource allocation in this model. Load balancing has been proposed in the DRALB model by Chhabra and Singh, 2021, which, however, is not designed to predict workload spikes. On the other hand, workload scheduling has been considered in a model called the RWTS, developed by Chowdaiah and Dammur, 2023, wherein addressing energy consumption satisfaction has failed to show better performance, with increased power consumptions compared to the proposed model. The proposed model will be combining features of predictive mechanisms with adaptive mechanisms to enable faster response times and more energy savings, which implicitly means that the integration of the features enhances the capability of the model in addressing IoT-specific challenges. This dynamic response to changes in the workload will be most relevant for highly variable scenarios, typical of many real-world IoT applications-such as smart cities and industrial automation. In such environments, demands on data may change very fast, and static models cannot be guaranteed to perform well without an over-provisioning of resources. The adaptiveness of the proposed model especially befits such contexts where efficiency in resources and responsiveness is at a premium in terms of service quality [5].

6-3- Model Strengths

Some of the key powers of this model are predictive scheduling, which allows for proactive resource allocation. While the basis of traditional models rests on a reactive approach, this model assures in predictive scheduling-a high-demand period is figured out in advance and thus, enables resource allocation in advance. This will reduce

latency and ensure that critical tasks float to the top, further enhancing the reliability of real-time IoT applications. Another strength is the adaptive resource allocation capability of the model. It reduces underutilization and bottleneck by dynamically readjusting resources according to runtime system load. This feature is really helpful in IoT scenarios where the workload changes continuously, as in such scenarios, static resource allocation mostly leads to inefficiency. It balances loads across virtual machines through an adaptive allocation mechanism and hence distributes tasks equitably, utilizing resources optimally. Its ability to adapt also confers on it a degree of resilience against sudden workload spikes, not uncommon in IoT applications with critical time requirements. The energy-aware scheduling model deals with one of the core challenges in IoT: energy efficiency. IoT devices very often rely on batteries as a source of power, and for long operation, energy efficiency is essential. The proposed model not only minimizes energy consumption through efficient resource allocation but also takes into consideration the energy-saving measures during low-demand periods. By this approach, operational cost and environmental footprint of IoT systems benefited by placing model as an eco-friendly solution for cloud-based IoT.

7- Conclusion

The paper presents a dynamic scheduling and resource allocation model developed to meet the peculiar demands of IoT applications running on the cloud. The proposed model has integrated predictive analytics, adaptive resource allocation, and energy-aware mechanisms that will show how effectively cloud resources have been optimized for resource usage, latency, and energy efficiency in the context of the Internet of Things. The results from the simulation and experiments conducted indicate significant improvements in the major performance metrics studied: response time, resource utilization, energy consumption, and throughput. These all point out the fact that the proposed model stands tall to handle the diverse and fluctuating workloads typical of IoT systems and is therefore suitable for applications that require real-time responsiveness with high resource efficiency. One major strength of this model is predictive scheduling, which allows the system to estimate demands that are expected to go high and allows it to advance resource allocation. This proactive approach reduces latency and further improves the reliability of mission-critical IoT services, like healthcare monitoring and smart city applications. Second, the adaptive resource allocation feature applies efficiency in resource utilization by dynamic task distribution based on runtime load conditions, avoiding underutilization and bottlenecks. Besides this, energy-aware scheduling adds to the sustainability of the model-a crucial factor for IoT devices, which often rely on very limited battery power. This model also has its own set of limitations. For predictive analytics, it requires high-quality data and further computational resources, which may introduce additional overhead, especially in large-scale deployments. The complexity of the model could be another challenge in implementing real time on budget-constrained environments. Though these are limiting factors, the benefits that were demonstrated by this model proved that it has immense promise for refinement and extension to wider applications. Further optimization of the model to reduce computational overhead, lightweight predictive algorithms, and integration of edge-fog computing resources can further extend the applicability of the model in future research work. Development of efficient and adaptive scheduling solutions will be needed with continued growth in scale and complexity of IoT networks. The proposed development herein puts a solid basis on which the development is done; hence, it offers flexibility and scalability in managing resources in cloud-based IoT ecosystems.

8- References

[1] Praveenchandar, J., & Tamilarasi, A. (2020). Dynamic resource allocation with optimized task scheduling and improved power management in cloud computing. *Journal of Ambient Intelligence and Humanized Computing*.
[2] Duan, J., Li, Y., Duan, L., & Sharma, A. (2022). Time Effective Cloud Resource Scheduling Method for Data-Intensive Smart Systems. *International Journal of Information Technology and Web Engineering*.
[3] Chhabra, S., & Singh, A. K. (2021). Dynamic Resource Allocation Method for Load Balance Scheduling Over Cloud Data Center Networks. *ArXiv*.
[4] Rahul, S., & Bhardwaj, V. (2022). Optimization of Resource Scheduling and Allocation Algorithms. *2022 Second International Conference on Interdisciplinary Cyber Physical Systems (ICPS)*.
[5] Kumar Chowdaiah, N., & Dammur, A. (2023). Resource-efficient workload task scheduling for cloud-assisted internet of things environment. *International Journal of Electrical and Computer Engineering (IJECE)*.
[6] Lin, Q. (2021). Dynamic Resource Allocation Strategy in Mobile Edge Cloud Computing Environment. *Mobile Information Systems*.

[7] Sutar, S. G., & Kumarswamy, S. (2022). Efficient Scheduling of Jobs and Allocation of Resources in Cloud Computing. *International Journal of Software Innovation*.
[8] Nair, R. (2023). Dynamic Resource Allocation in Cloud Environments. *International Journal for Research in Applied Science and Engineering Technology*.
[9] Kalimuthu, R., & Thomas, B. (2021). An effective multi-objective task scheduling and resource optimization in cloud environment using hybridized metaheuristic algorithm. *Journal of Intelligent Fuzzy Systems*.
[10] Luo, Q., Hu, S., Li, C., Li, G., & Shi, W. (2021). Resource Scheduling in Edge Computing: A Survey. *IEEE Communications Surveys & Tutorials*.
[11] Yakubu, I. Z., Muhammed, L., Musa, Z., Matinja, Z. I., & Adamu, I. M. (2021). A Multi Agent Based Dynamic Resource Allocation in Fog-Cloud Computing Environment. *Trends in Sciences*.
[12] Dechouniotis, D., & Papavassiliou, S. (2020). Modelling and Resource Scheduling approaches on Cloud Computing. *2020 European Control Conference (ECC)*.