

Deadline-Aware Virtual Machine Optimization and Task Scheduling in Cloud Computing

Surojit Roy
Assistant Professor,
Department of Computer Application
Techno Main Saltlake
Kolkata, India

Dr. Hemanta Dey
Assistant Professor, MCA Dept.
Techno International New Town
Kolkata, India

Abstract: Cloud computing infrastructures in contemporary times were very dependent upon effective task scheduling, smart workload management, and proper VM allocation for offering reliable, high-performance computing services and at the same time achieving optimal operational cost, energy footprint, and physical data centre utilization. Since multi-processor job scheduling among heterogeneous distributed processing nodes is an NP-hard problem of combinatorial optimization without any polynomial-time solution, cloud management systems will be required to use heuristic, meta-heuristic, and hybrid approximation approaches to find nearly optimal mappings of tasks onto the available resources within finite time. In this study, the underlying cloud scheduling structures and their taxonomies (differently analysing compile-time static allocation in contrast with run-time dynamic allocation, independent batch of tasks vs. dependency-based scientific workflows) are explored and multidimensional performance assessment parameters are analysed. In particular, the emphasis is put on QoS provisions, Service Level Agreements, execution cost, makespan, and resource utilization in the cluster. Besides that, operational issues in distributed virtualization platforms are also considered. Among others, these are the complex latency and throughput overheads resulting from runtime VM live migration such as the iterative dirty-memory pre-copying process, secondary storage disk I/O overheads, and uninterrupted network connectivity, along with the critical challenges in multi-tenancy environment, that is, the threat of unapproved VM hopping, noisy neighbour VM's resource poaching behavior, and divergence in cloning configurations on shared network infrastructure. This paper attempts to solve the ongoing conflict between end-user's requirement of deadline guarantee and the provider's requirement of resource optimization through introducing the theoretical and mathematical formulation of a new framework referred to as Deadline-Based VM Optimization (DBVO). DBVO combines dynamic slack-time analysis with earliest completion time metrics to emphasize time-critical tasks, achieve system-wide load balancing, avoid schedule bottlenecks, and SLA violation fines. Moreover, an experimental roadmap to test performance of this new method using CloudSim simulator is outlined.

Keywords: Cloud Computing, Task Scheduling, VM Optimization, Quality of Service (QoS), Service Level Agreement (SLA).

1. INTRODUCTION

Cloud computing has become an essential computing paradigm that entails the delivery of computing resources and services through the Internet in such a way that both organizations and individuals are able to obtain computing resources without owning extensive physical infrastructure. Cloud computing involves the delivery of flexible and scalable computing resources including computational power, memory, storage, networking and software services. One important technology behind modern cloud computing is hardware virtualization that makes it possible to abstract physical computing resources and partition them into various virtual machines (VMs). The virtualized resources are capable of being provisioned, monitored, migrated and decommissioned depending on the dynamic workload conditions.

Task scheduling, that is the heart of this computing paradigm, is concerned with deciding how to allocate incoming computational workloads to physical or virtual resources based on certain criteria. Task scheduling can be described as the allocation of a certain number of computational tasks to virtual or physical machines or processing units in accordance with certain algorithms and policies. The efficiency of the chosen scheduling policy determines the performance, reliability, scalability, and cost of operation of the cloud computing system. There are a few aspects a good task scheduling algorithm should take into consideration, such as task execution time, resource availability, workload properties, communication cost, VMs capacity, energy consumption, QoS, etc.

The goal of cloud task scheduling is to find a compromise between a few conflicting performance metrics, such as throughput, resource utilization, task execution time, response time, makespan, cost of operation, energy

consumption, and SLA compliance. Higher values of throughput and resource utilization allow cloud providers to support more users on their infrastructure, while reducing execution and response time improves the performance of the cloud computing application. On the other hand, energy savings become even more relevant for large data centres, since the computing infrastructure could contain thousands of physical servers running all the time. Thus, the scheduling algorithms should consider not only the computational aspects, but economic and environmental ones too.

Task scheduling becomes considerably more difficult when considering heterogeneous and distributed cloud environments. In particular, the job scheduling problem on multiple processors and the problem of resource allocation are known to be NP-hard problems, i.e., there is no efficient algorithm known that would give an optimal solution in polynomial time for sufficiently large problem instances. With increasing numbers of tasks, virtual machines, hosts, and constraints, the problem space grows exponentially. Hence, deterministic exhaustive approaches cannot be used for large-scale cloud infrastructures due to computational complexity.

Hence, modern cloud infrastructures use heuristics and meta-heuristic optimization techniques for addressing scheduling problems. In such algorithms, one tries to find nearly optimal resource allocation policies by avoiding the process of searching throughout the whole solution space. Classical heuristics normally rely on problem-specific knowledge to produce solutions rapidly, while meta-heuristic techniques offer broad search techniques, which can be used for exploring difficult solution spaces and avoiding falling into bad local optima. Genetic Algorithms, Particle Swarm Optimization, Ant Colony Optimization, Simulated Annealing, Differential Evolution, and other optimization approaches based on population or trajectory have thus received much attention in the field of cloud scheduling. These techniques can also be adopted for dealing with single objective and multi-objective scheduling problems.

An important issue related to cloud scheduling is the conflict between objectives of various stakeholders. CSPs try to utilize their computing infrastructure in an optimal way to minimize operating costs and energy consumption, and at the same time maximize their profits. Increased resource utilization can help CSPs to increase the number of tasks processed with the available resources. However, increased resource utilization can also lead to resource contention and poor performance. On the contrary, cloud consumers are interested in the efficient and reliable execution of their applications, performance predictability, acceptable cost, and conformance to the prescribed Quality of Service levels. Thus, there should be a certain trade-off in the design of scheduling policies that would consider the interests of the provider and the user at once.

This task gets even harder when there are deadlines associated with jobs and Service Level Agreement (SLA).

SLA usually indicates the desired level of service between the provider and the client and includes such aspects as performance, availability, response time, etc. Non-compliance with deadlines can violate the SLA, incur penalties, reduce customer satisfaction and the quality of the provided service. Consequently, the challenge in deadline-aware scheduling is the optimization of resource usage and cost together with meeting deadlines.

Virtualization also poses new issues with respect to security and operation in distributed cloud infrastructures. The existence of numerous virtual machines running simultaneously on the same physical server makes the issues related to VM isolation, hypervisors, communication between VMs, resource contention, VM migration, and multi-tenant environment a factor influencing the security and reliability of scheduling processes. While VM migration is a handy tool for dynamic load balancing and resource management, it can be costly from the perspective of computational and network overhead. In addition, an erroneous resource allocation process will inevitably lead to resource contention and inefficient use of infrastructure. Thus, a proper scheduling framework must consider both performance and operational issues of virtualized environment.

In this sense, mathematical optimization models allow developing a systematic representation of relationships between tasks, virtual machines, deadlines, execution cost, and resource. An appropriate mathematical model can provide a basis for defining objective function and constraints and, hence, finding feasible task-to-VM assignment. Moreover, a well-developed mathematical model can become the background for assessing the effectiveness of heuristic and meta-heuristic scheduling approaches.

Thus, this paper seeks to examine the basic architectural design and principles behind cloud task scheduling, as well as the current scheduling schemes in use. A taxonomy of static and dynamic scheduling is developed with emphasis on their features, benefits, drawbacks, and applicability. The paper also assesses some of the important operational factors such as makespan, execution time, resource usage, throughput, cost, energy usage, deadline compliance, and SLA enforcement. In addition, this paper explores some of the security risks and operation issues related to distributed virtualization environment and their possible impacts on scheduling performance.

This paper also develops a mathematical model for Deadline-Based VM Optimization (DBVO). This model aims at solving the problem of effective allocation of tasks with deadlines to virtual machines considering resource availability and optimization requirements. The model intends to provide a systematic way of minimizing task execution time and resource expenses while maintaining good deadline compliance and efficient VM utilization.

2. LITERATURE REVIEW

Various researchers have studied task and resource scheduling in cloud computing from diverse viewpoints. Previous studies mainly focused on developing cloud resource management models and examining how virtualization can be employed for dynamic resource provisioning in cloud computing. Buyya et al. [1] provided a broad vision of cloud computing as a utility-oriented computing approach and explored virtual machines and market-based approaches to resource management in cloud computing. Their study laid down a strong foundation for future research in resource provisioning, SLA management and cloud resource management.

Calheiros et al. [3] developed CloudSim, an extendible cloud simulation platform that allows researchers to model and experiment on cloud computing platforms and resource provisioning policies. CloudSim supports the representation of data centres, physical hosts, virtual machines, applications and resource provisioning policies, thus allowing researchers to conduct scheduling experiments under controlled settings [3]. Such simulation platforms have contributed immensely to cloud computing scheduling because conducting direct experiments using commercial cloud platforms may be costly and challenging to replicate.

Research in the area of energy-efficient resource management is also of interest. Beloglazov, Abawajy, and Buyya [4] studied energy-aware resource management and suggested heuristics to enhance the energy efficiency of cloud data centres. In this context, the dependence between VM allocation, physical machine utilization, energy usage, and costs was highlighted. It was shown that resource consolidation and allocation optimization may reduce energy usage without significant impact on application performance [4].

Another promising research direction involves deadline-aware scheduling. In this respect, Abrishami, Naghibzadeh, and Epema [5] studied deadline-aware workflow scheduling in IaaS clouds. Specifically, they suggested IC-PCP and IC-PCPD2 algorithms to reduce workflow execution cost subject to user-defined deadlines. It was pointed out that the specifics of cloud computing (on-demand resource allocation and pay-per-use pricing) should be considered in such scheduling policies [5].

The combined consideration of energy consumption and deadlines has been another research area. Ding et al. [6] introduced energy-saving scheduling of VMs with consideration of deadlines and heterogeneous physical machines. In their research, they considered DVFS and proposed an optimal performance to power ratio for choosing physical machines. As per the results obtained from their study, the inclusion of energy considerations in VM scheduling improves energy efficiency while meeting deadline requirements [6].

Optimization using metaheuristics has been another important area of research due to the difficulty involved in solving cloud scheduling problems. Masdari et al. [7] discussed different metaheuristic techniques for cloud scheduling and analysed GA, ACO, PSO, and other such optimization approaches. They focused on the fact that most cloud scheduling problems generally need near-optimal solutions in a feasible amount of time [7].

Subsequent surveys have been able to classify cloud scheduling approaches on the basis of heuristic, meta-heuristic, and hybrid scheduling techniques. Kumar and others [8] conducted a thorough survey that considered QoS parameters such as response time, makespan, reliability, availability, energy consumption, cost, and resource utilization in the context of scheduling methods. It was also pointed out that the deadline constraint, priority constraint, and SLA constraint were also important considerations in cloud scheduling [8].

The work of Houssein et al. [9] is a systematic review and taxonomy of meta-heuristic approaches in cloud task scheduling. This work classified scheduling approaches depending on optimization criterion, task-resource mapping approaches, scheduling constraints, and single/multi-objective formulations. This paper highlighted that the optimization of resource utilization, makespan, cost, energy consumption, reliability, and other QoS parameters remained very important [9].

Workflow scheduling has also attracted considerable attention because workflows contain task dependencies and therefore create more constraints than independent task scheduling. Masdari et al. [10] conducted an extensive survey on workflow scheduling approaches in cloud environments and focused on budget constraint, deadline constraint, resource utilization, load balancing, and availability. Subsequent surveys have also highlighted the following issues in cloud

workflow scheduling; virtualization, VM migration, resource heterogeneity, energy consumption, deadlines, and QoS constraints [10], [11].

While these works have added immensely to the existing literature, there remain certain areas in the research. Current solutions for scheduling are usually focused on a single aspect or set of aspects from the list which includes the minimization of makespan, energy, costs, and deadline fulfilment. Cloud systems need more than just the above-mentioned issues since there is a need for the consideration of several different issues simultaneously and sometimes conflicting each other. Minimizing the execution time can result in energy consumption and cost growth, while aggressive consolidation of resources can make it more energy efficient but negatively influence the response time and SLA constraints.

As such, there is a constant need for scheduling mechanisms that consider deadlines, VM usage, run time, resources, cost, and energy as integral parts of a single optimization approach. One of the most interesting directions of study related to cloud task scheduling is the interaction between deadlines and VM selection. Scheduling technique that can select the right VM based on the task requirements, computing capacity, estimated run time, and deadlines could be used to enhance resource utilization and decrease the likelihood of missing deadlines.

Thus, this paper aims to analyse the architectural background and fundamentals of cloud task scheduling, provides a full taxonomy of static and dynamic scheduling algorithms, examines the key parameters and challenges of cloud task scheduling, and suggests the mathematical foundation of the proposed method known as Deadline-Based VM Optimization (DBVO). The suggested framework is designed to create the relationship between the tasks' characteristics, VMs' features, scheduling requirements, and deadlines. The purpose of the method is to serve as a foundation for optimizing VM selection.

3. RESEARCH GAP

Despite many works being carried out around task scheduling [14] and optimization in cloud computing, some limitations have not yet been overcome by the researchers. Several works have suggested optimization methods including heuristic, meta-heuristic and hybrid methods to minimize makespan, resource utilization, energy consumption, execution cost and Quality of Service (QoS).

In many cases, most of these algorithms are only concerned with optimizing one or few objectives and fail to consider the impact of task deadlines, VM choice, execution time, resource utilization and scheduling cost all at once.

The first limitation of many existing scheduling algorithms is that they have been developed to optimize the execution time of jobs, also referred to as makespan. Even though optimization of makespan is beneficial to the performance of the system, it may not necessarily imply that all tasks complete within their deadline period. In other words, a scheduling algorithm may produce less makespan while some deadline sensitive tasks have already missed their deadline period.

Secondly, existing meta-heuristic algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and many others can successfully be used in solving problems of scheduling in large search space [7], [9]. However, these algorithms may involve quite a significant amount of computational overhead due to iterations, evaluation of the population and fitness functions. Additionally, their efficiency depends heavily on parameter choice, initial population, convergence properties and nature of the work load. Thus, the challenge arises of designing a scheduling algorithm that would be able to balance the trade-off between quality and overhead.

Thirdly, VM selection and allocation are usually seen as auxiliary parts of the whole scheduling process. In a heterogeneous cloud environment, VMs can greatly differ in terms of computational capacity, available memory, cost and execution time. The VM selection based only on computational capabilities or current availability of the VMs will not necessarily result in the optimal allocation for a deadline-sensitive task. Therefore, the need arises to develop a more elaborate VM selection mechanism considering the characteristics of the task and the VM.

Fourth, there is a clear need to solve the problem of balancing resource usage and meeting deadlines. Using aggressive resource usage can optimize the infrastructure's efficiency; however, the over-usage of resources or the competition for the resources can also cause delays in the completion of jobs. On the other hand, more usage of VMs to fulfil deadlines can cause high costs and high energy consumption. Previous studies have analysed these factors separately or using multi-objective models, but there is still room for developing a practical model which considers all these factors [4], [6], [8].

Fifth, several previous studies evaluate their proposed scheduling algorithms using a few performance measures. For instance, an algorithm may be found more efficient in terms of makespan while ignoring factors such as deadline violations, VM usage, costs, and resource usage efficiency. There should be a way to evaluate algorithms using multiple performance measures such as makespan, average execution

time, deadline violation ratio, VM usage, throughput, cost, and energy consumption.

The other problem area relates to the dynamic nature of the cloud workloads. The cloud applications may come up dynamically, and their computation demands can change greatly over time. In static scheduling, it is always assumed that the knowledge about the tasks and resources is known before scheduling starts, while there are frequently changing workload scenarios in cloud computing. Hence, the scheduling policy must be able to adapt VM allocation to changing task and resource demands.

Also, the introduction of virtualization technology poses additional problems regarding the multi-tenancy, resource contention, VM migration, isolation, and heterogeneity. The existing techniques usually concentrate mostly on the optimization goal, and only little attention is paid to these operation issues. Including these virtualization constraints into the scheduling models would increase the usefulness of the developed algorithms.

4. CLOUD SCHEDULING ARCHITECTURE

Cloud scheduling is done through a hierarchical abstraction layer where the requests coming from different tenants are processed with the help of physical hardware pools. The main architectural components which take care of job lifecycle management include:

- **End User Layer:** Here, the client creates and sends individual computing jobs or complex workflow jobs with their respective job execution time, dependencies, arrival time, and deadline.
- **Data Centre Broker:** This acts as an intermediary middleware component. The broker accepts requests from users, keeps track of the global resources' status, decides the scheduling technique, and distributes jobs among allocated virtual machines.
- **Cloud Controller:** This component interacts with the broker bi-directionally and controls the hypervisor, manages virtual machine allocation, measures the physical load, and ensures virtual machine availability.

5.3 ALGORITHMIC PARADIGMS

The search mechanisms governing scheduler decision-making fall into three primary algorithmic classes:

Paradigm	Mechanism & Scope	Strengths	Limitations
Heuristic	Rule-based greedy dispatch (e.g., Min-Min, Max-Min, FCFS, Round Robin).	Extremely fast runtime $O(n \log n)$; minimal computational overhead.	Problem-specific; easily trapped in local optima; lacks global perspective.

- **Pool of Virtual Machines:** VM instances running on physical data center hosts with specific computation, memory, storage, and networking capacity.

5. CLASSIFICATION AND TAXONOMY OF SCHEDULING

Task scheduling techniques in cloud computing can be classified based on task dependencies, allocation timing, and algorithms used to optimize tasks.

5.1 TASK DEPENDENCY

- **Independent Scheduling:** There is no data dependency among tasks, nor any synchronization barrier. Task order is irrelevant to computation correctness, making task execution flexible and embarrassingly parallel.
- **Dependent (Workflow) Scheduling:** Tasks have strict dependencies among them, and can be represented as Directed Acyclic Graphs (DAGs). Children are dependent on parents, and a child task can only execute when all parent tasks have finished executing and intermediate data transfers are done.

5.2 ALLOCATION TIMING: STATIC AND DYNAMIC

- **Static Scheduling:** Decision about workload allocation is decided a priori, either during compilation time or at submission time. All characteristics of tasks (such as number of execution cycles, communication barriers, dependencies), as well as those of the underlying platform, need to be known beforehand. Although predictable from computational standpoint, static scheduling cannot adapt to runtime variability.
- **Dynamic (Online) Scheduling:** Decisions about workload allocation are made during runtime upon arrival of the job. They include current host states, network bandwidth, and queueing delay.

Meta-Heuristic	Iterative population-based exploration (e.g., Genetic Algorithms, PSO, ACO).	Broad search space exploration; delivers near-optimal solutions for NP-hard spaces.	High convergence time and tuning sensitivity; computationally intensive.
Hybrid	Synergistic combination of heuristic initialization and meta-heuristic exploration.	Counteracts heuristic traps while accelerating meta-heuristic convergence.	Increased architectural complexity; requires balanced multi-parameter tuning.

6. PERFORMANCE PARAMETERS AND QUALITY OF SERVICE (QOS)

The following set of system-centric and user-centric measures have been defined to analyze and compare various scheduling algorithms:

- **Makespan (Completion Time):** The overall time period required from the submission of the first batch of tasks until the last task is executed. A smaller makespan value indicates better scheduling efficiency.
- **Response / Reaction Time:** The time period taken between dispatching the user request until receiving the first service output. Essential for interactive web transactions and real-time database accesses.
- **Throughput:** The overall efficiency rate at which the set of tasks is being executed within a certain period of time.
- **Resource Utilization:** The amount of time that the allocated virtual and physical resources spend executing the task instead of remaining idle.
- **Load Balancing:** The balanced distribution of the computation load among the existing nodes.
- **Fault Tolerance:** The stability of the scheduling process to hardware and network failures without stopping operations and task corruption.
- **Energy Consumption:** The physical energy required by the computing and cooling infrastructure in the data center. The main target for sustainable cloud computing.
- **Service Level Agreement (SLA):** An agreement defining the limits on cost of execution, deadlines, availability, loss of packets and jitter.

7. DISTRIBUTED VIRTUALIZATION CHALLENGES AND OPEN ISSUES

Deploying scheduling frameworks in real-world hypervisor environments exposes several critical operational hurdles:

7.1 VIRTUAL MACHINE LIVE MIGRATION OVERHEADS

Dynamically allocated workload involves the migration of active VM instances from one physical server to another. The process presents three specific problems:

- **Memory State Consistency:** Maintaining consistency of operation entails repetitive copying of the modified

memory pages, CPU registers and buffers resulting in network traffic.

- **Disk Migration Latency:** Without any shared storage arrays, the process of disk migration results in increased I/O latencies leading to SLA breaches.
- **Network Persistence:** It is necessary to keep client connections alive during the migration process in order to avoid disconnections.

7.2 MULTI-TENANT HYPERVISOR SECURITY RISKS

A shared hardware infrastructure results in some security threats which the scheduler should take into consideration:

- **VM Poaching:** A “noisy neighbor” VM is consuming excessive shared cache, memory bandwidth, and/or bus resources.
- **VM Hopping:** Isolation flaws in the hypervisor being used to break out of guest OS limits to attack co-hosted VMs.
- **VM Cloning Variability:** Cloning virtual images without proper control leads to configuration conflicts, expired IP/MAC addresses and synchronized credential vulnerability.

8. PROPOSED FRAMEWORK: DEADLINE-BASED VM OPTIMIZATION (DBVO)

To resolve the dual challenge of user deadline compliance and provider resource efficiency, we propose the Deadline-Based VM Optimization (DBVO) framework [12].



8.1 MATHEMATICAL FORMULATION

Consider an incoming set of independent tasks $T = \{t_1, t_2, \dots, t_n\}$ and a pool of active virtual machines $V = \{v_1, v_2, \dots, v_m\}$. Each task t_j has length L_j (Million Instructions) and a firm deadline D_j . Each VM v_i provides compute capacity P_i (MIPS).

1. Expected Execution Time (EET):

$$EET(i, j) = L_j / P_i$$

2. Earliest Completion Time (ECT):

$$ECT(i, j) = Ready_i + EET(i, j)$$

where $Ready_i$ represents the timestamp at which VM v_i finishes all currently scheduled tasks.

3. Slack Time Evaluation (S):

$$S(i, j) = D_j - ECT(i, j)$$

A mapping is feasible if and only if $S(i, j) \geq 0$. DBVO prioritizes tasks exhibiting minimum positive slack (highest urgency) and maps them to VMs that satisfy the deadline while preserving capacity for subsequent tasks.

8.2 DBVO ALGORITHMIC FLOW

Step 1: Ingest batch of submitted user tasks and calculate execution lengths and deadline parameters.

Step 2: Query Cloud Controller for real-time VM capacities, current ready times, and active queue depths.

Step 3: Construct the EET and ECT matrices across all (task, VM) permutations.

Step 4: Compute Slack Time $S(i, j)$. If $S(i, j) < 0$ across all available VMs, flag task for priority preemption or auto-scaling.

Step 5: Sort tasks in ascending order of minimum slack time (Earliest Deadline / Urgency First).

Step 6: Assign each sorted task to the VM that achieves minimum completion time while maintaining load variance below threshold DI_{max} .

Step 7: Update VM ready times and iterate until queue saturation reaches zero.

9. CONCLUSION

Task scheduling is the foundational pillar governing cloud computational throughput, cost efficiency, and QoS fulfillment. This paper presented a systematic review of cloud scheduling models, contrasted heuristic and meta-heuristic paradigms, evaluated multi-dimensional performance metrics, and identified virtualization challenges including migration overhead and isolation breaches. The proposed Deadline-Based VM Optimization (DBVO) formulation establishes a rigorous foundation for deadline-aware task mapping.

Future work will focus on implementing and benchmarking DBVO within the CloudSim simulation framework against baseline Min-Min, Max-Min, and Genetic Algorithm schedulers, evaluating its efficacy under synthetic and real-world scientific workflow workloads.

10. REFERENCES

- [1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009, doi: 10.1016/j.future.2008.12.001.
- [2] M. Masdari, S. ValiKardan, Z. Shahi, and S. I. Azar, "Towards workflow scheduling in cloud computing: A comprehensive analysis," *Journal of Network and Computer Applications*, vol. 66, pp. 64–82, 2016, doi: 10.1016/j.jnca.2016.01.018.
- [3] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011, doi: 10.1002/spe.995.
- [4] A. Beloglazov, J. H. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management

of data centers for Cloud computing,” *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755–768, 2012, doi: 10.1016/j.future.2011.04.017.

[5] S. Abrishami, M. Naghibzadeh, and D. H. J. Epema, “Deadline-constrained workflow scheduling algorithms for Infrastructure as a Service Clouds,” *Future Generation Computer Systems*, vol. 29, no. 1, pp. 158–169, 2013, doi: 10.1016/j.future.2012.05.004.

[6] Y. Ding, X. Qin, L. Liu, and T. Wang, “Energy efficient scheduling of virtual machines in cloud with deadline constraint,” *Future Generation Computer Systems*, vol. 50, pp. 62–74, 2015, doi: 10.1016/j.future.2015.02.001.

[7] M. Masdari, S. ValiKardan, Z. Shahi, and S. I. Azar, “A review of metaheuristic scheduling techniques in cloud computing,” *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 275–295, 2015, doi: 10.1016/j.eij.2015.07.001.

[8] M. Kumar *et al.*, “A comprehensive survey for scheduling techniques in cloud computing,” *Journal of Network and Computer Applications*, 2019.

[9] E. H. Houssein, A. G. Gad, Y. M. Wazery, and P. N. Suganthan, “Task scheduling in cloud computing based on meta-heuristics: Review, taxonomy, open challenges, and future trends,” *Swarm and Evolutionary Computation*, vol. 62, p. 100841, 2021, doi: 10.1016/j.swevo.2021.100841.

[10] S. Kaur, P. Bagga, R. Hans, and H. Kaur, “Quality of Service (QoS) aware workflow scheduling (WFS) in cloud computing: A systematic review,” *Arabian Journal for Science and Engineering*, vol. 44, pp. 2867–2897, 2019.

[11] “Workflow scheduling in cloud environment – Challenges, tools, limitations & methodologies: A review,” *Measurement: Sensors*, vol. 24, p. 100436, 2022.

[12] “A Quantum-Inspired Evolutionary Framework for Multi-Objective Task Scheduling in Sustainable Cloud Computing Environments”, Dr. Hemanta Dey, S. Rana, *International Journal of Computer Applications Technology and Research*, Volume 15, Issue 03, Page No 109 – 113, 2026, ISSN: 2319–8656, DOI:10.7753/IJCATR1503.1016.

[13] H. Dey, A. Datta, “A Comprehensive Study on Blockchain Technology Architecture, Applications, and Future Research Directions” in the “The International Journal of Engineering Trends and Applied Science (IJETAS)”, ISSN: 2454-6194, Vol.-1, Issue-2, November 2025 pp: 24-31.

[14] H. Dey, Dr. A. Saxena, “Improving Efficiency and User Satisfaction in Cloud Computing Through Task Scheduling: A Comparison Among Different Scheduling Algorithms” in the “Journal of Namibian Studies”, 35 (2023): 293-307 ISSN: 2197-5523 (online) in a Special Issue on Engineering, Technology and Sciences.