Journal of Theoretical and Applied Information Technology
15th August 2023. Vol.101. No 15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

TASK SCHEDULING IN CLOUD COMPUTING: PRIORITY-BASED ALGORITHMS AND FUTURE DIRECTIONS IN DATA SCIENCE

RASHA AL BASHAIREH[1]

[1]Assistant Professor, Tafila Technical University, Department of Computer Science, Tafila, Jordan

E-mail: [1]ralbashaireh@ttu.edu.jo

ABSTRACT

In the era of distributed computing, Cloud Computing (CC) has been considered an emerging technology that delivers on-demand services using the worldwide online medium. One of this technology's main challenges is scheduling tasks and allocating resources to achieve the best performance with minimal optimized execution time, and less resource time and usage. Thus, the importance of task priority has been recognized by many researchers since tasks submitted to the cloud can have various sizes, resource utilization, and execution times. Therefore, existing task scheduling algorithms prioritize tasks, where the task with the highest priority is allocated to the available resource. This paper investigated the task priority problem in cloud-based Task Scheduling (TS) algorithms via three different directions centered around the following contributions; first, a fruitful discussion and comparison of some selected cloud priority-based TS algorithms regarding their scheduling method and parameters, algorithm performance, and limitations reflecting how the priority issue is handled in the cloud environment. In the Data Science and Artificial Intelligence age, the interaction and intervention between the Big Data sector and CC are gradually advancing due to mutual development. Cloud environments have become a prominent platform for big data applications, forcing the integration between Data Science and CC techniques to evolve. On this basis, secondly, the paper addresses the directions of potential future research of cloud priority-based TS schemes in Data Science. Finally, as a result, a conceptual framework incorporating a task content-type component is proposed to support the processing and scheduling of data-related tasks submitted to the cloud.

Keywords: *Cloud Computing, Task Scheduling, Priority Algorithms, Data Science, Artificial Intelligence*

1. INTRODUCTION

Over the recent few years, Cloud Computing (CC) has evolved as one of the most prominent and sophisticated research fields. As declared by the National Institute of Standards and Technology (NIST) in 2011 [1], CC is officially defined as *"a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."* CC's primary objective is to effectively enable access to geographically dispersed and remote resources [2]. In the last decade, popular online social networking platforms, email, document sharing, and online gaming websites have been hosted on the cloud [3]. In addition, famous enterprises such as IBM, Google, Microsoft, Apple, and Amazon are very successful and active in this field. CC is an internet-based computing paradigm that consists of three layers; an infrastructure layer known as *Infrastructure as a Service (IaaS),* a platform-based layer known as *Platform as a Service (PaaS),* and a software-based layer called *Software as Services (SaaS)* [1].

In IaaS, the associated data and computational infrastructure are supplied to the end user as a service via a Virtual Machine (VM). IaaS deals with VM, networks, and servers. Since VM is an operating system with a software implementation that can easily be created, used, and managed. In PaaS, the cloud vendors provision a software development computing platform as a service to the developer. PaaS offers developers key development tools such as storage for databases, a web server, operating systems, and an environment for programming language execution [1]. It enables application developers the freedom to design and build software and applications that operate on a cloud platform without being concerned about maintaining the necessary hardware and software.

In SaaS, the end user receives software applications and operating systems from the cloud provider. Since the SaaS model allows users to access software applications through services that run on infrastructure governed by the SaaS vendor [4].

The newly emerged CC technology is continuously developing day by day, facing various challenges, one of them being scheduling the various received tasks. Task scheduling is one of the main problems that result in reducing system performance. Tasks are scheduled based on parameters such as task size, VM size, task priority, and so on, where the VM and tasks are offered in numerous sizes. Optimal task scheduling algorithms are measured using some metrics such as response time, resource utilization, and execution time. The priority of tasks is an important key issue for task scheduling in cloud computing environments because some tasks must be served by the system earlier than others since these tasks cannot stay for a long time in a system [5]. Thus, an optimal task scheduling algorithm in cloud computing must consider the priority of tasks. Briefly, this paper investigated the problem of task priority in cloud-based Task Scheduling algorithms. The study was conducted in more than one direction, summarized in the following research objectives: First, Introduce the term Task Scheduling in the cloud profoundly and investigate superior priority-based task scheduling algorithms and how the priority issue is handled in the cloud environment. Second, cloud environments serve as a vital platform for big data applications, forcing the integration between Data Science and CC techniques to evolve in the modern Artificial intelligence (AI) age. The paper aims to address potential future research directions for big data cloud priority-based task scheduling schemes in Data Science. Third, to support the processing and scheduling of big data-related tasks in cloud environments. The paper also aims to propose a conceptual framework for task scheduling incorporating a content-type component that can enhance prioritizing such big data-related tasks according to their content data types in cloud settings.

The task priority problem is allocating a set of tasks received by the Datacenter broker to the available list of VMs, focusing on the priority of tasks to achieve the goal of minimal optimized execution time, less response time, and resource usage. The execution time of a task in the cloud environment depends on the performance of the machines which implement it, which differs from device to device. Hence, various priority-based cloud task scheduling algorithms have been proposed by several researchers. To achieve the first objective, three different Priority-based task scheduling algorithms with distinct factors and parameters have been chosen and discussed in this paper to compare the task scheduling method based on priority in the cloud environment. The paper described each method and how the priority issue is handled in each algorithm. Accordingly, these three selected algorithms are compared using various criteria such as the scheduling method, scheduling parameter, algorithm performance, and algorithm limitations.

In the era of big data industries and with the growth of cloud computing usage, and continuous in-depth research on current task scheduling problems and techniques, some new research directions emerged and may become a hot field of future studies in data science. In this research, to achieve the second objective, the integration between cloud Task scheduling techniques and the Data Science (DS) and AI-based techniques have been discussed in section 5 by addressing primary potential future directions highlighting the significance of cloud-based data processing and analysis in real-time including the integration with Machine Learning (ML) techniques as well as the implications for data security and privacy in the cloud.

In the cloud, the tremendous magnitude and variety of data raise the demand for sophisticated computations to make it accessible. Hence, performance efficiency is a crucial concern and can be enhanced by looking at vital elements like the task content type submitted by the cloud user. To address such concerns and to achieve the third objective, in section 6, we proposed a conceptual framework that incorporates a content-type component that can be used to classify users' tasks according to their content data types into text, image, audio, and video. This framework is an integrated one that also incorporates a classification method using AI approaches. Since the majority of cloud task scheduling strategies described in the literature do not consider the content type of the user-submitted tasks [2][3][4][5], which can enhance and balance workload distribution among various servers in the cloud environment.

In summary, the significant contributions of this paper are as follows:

1. A comprehensive comparison between the three discussed priority-based Task scheduling algorithms regarding the scheduling method and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

parameters, their algorithm performance, and limitations.

2. The potential future research directions of CC in data science, including vital suggestions are discussed. This might help in leading the roadmap for research on current trends regarding the integration between DS, AI, and CC techniques.
3. A conceptual framework incorporating a content-type component is proposed to support processing and scheduling big data-related tasks submitted to the cloud.

The remainder of this paper is neatly organized as follows. Section 2 describes how task scheduling is tackled in cloud-based settings and its three phases. Section 3 presents an overview of three chosen priority-based task scheduling algorithms; the Dynamic Optimization Algorithm, the Priority Based Job Scheduling Algorithm, and the Efficient Optimal Algorithm. A deep discussion and comparison of these algorithms are also provided in Section 4. The recommendations and future directions of the integration between CC task scheduling and data science techniques are outlined in section 5. The proposed conceptual framework incorporating the task content-type component is presented in section 6. At last, the conclusion and future work are briefly summarized in Section 7.

2. TASK SCHEDULING

In terms of the cloud environment, a clear definition of Task Scheduling (TS) terminology can be understood when the terms task and scheduling are clearly defined separately. Here, the term task represents a user request made to the cloud [6]. The term schedule refers to a set of policies to control the order of work performed by a computer system [2]. TS is a crucial activity in the cloud environment that maps tasks evenly to the appropriate resources based on task characteristics and requirements, where tasks are allocated to the VMs to be executed. A successful TS procedure improves the request's response time for these tasks and achieves a high system performance with the best throughput and reduced cost. There are three general phases to the scheduling process in a cloud environment [2]:

-**Resource Discovering and Filtering**: The Datacenter Broker here discloses the available resources (virtual memory, processors, storage, etc.) in the cloud computing network system and gathers respective status information.

-**Resource Selection:** This phase is a deciding phase, where the required resource is selected based on specific parameters of task and resource.

-**Task Submission**: The task is submitted and assigned to the selected resource from the previous phase. The simplified TS phases are illustrated by Choudhary et al. [3] in Figure 1.

The Cloud Information Services (CIS) is an entity responsible for discovering the resources and registering the datacenter entity. The Datacenter Broker is an entity in charge of facilitating the negotiations amongst the cloud providers and the SaaS. While the Datacenter is responsible for modeling the primary infrastructure-level services (hardware) provided by cloud suppliers, where a set of homogeneous and heterogeneous hosts is encapsulated. The VM entity is responsible for modeling a VM that runs on a cloud host to deal with a specific task [4].

The cloud in a CC environment consists of a collection of data centers, where each has several hosts or servers [4]. The host is seen as a VM in the PaaS layer to conceal the environment dependencies. The utilization of data center resources is improved through efficient scheduling. As mentioned above, data centers hold several virtual servers running at any time and receive many tasks request simultaneously [4]. An optimal task schedule has two primary goals, one from the cloud users' perspective and the other from the cloud service provider's perspective. From a cloud user's point of view, it’s vital to accomplish all their tasks in less response time. Whereas cloud service providers' main concentration is on how to use the maximum resources available in a cloud environment to accommodate the most tasks while making the most profit, where their performance affects task scheduling performance [4].

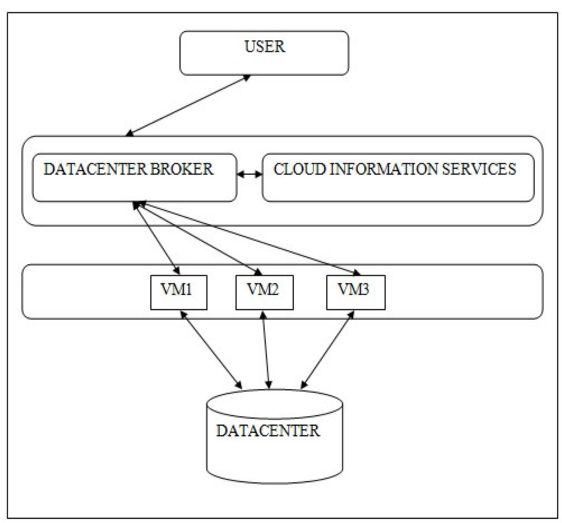

Figure 1: Task Scheduling Phases in Cloud Environment

Journal of Theoretical and Applied Information Technology
15th August 2023. Vol.101. No 15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3. PRIORITY-BASED TASK SCHEDULING ALGORITHMS

In Cloud Computing, Tasks submitted to the cloud can have various computational times, deadlines, resource utilization, execution times, and different file size [7]. Thus, it's essential to schedule the tasks based on priority wise because some tasks need to be scheduled first while others can wait. On the other hand, many tasks should be executed with the available resources to achieve the best performance, minimal completion time, less response time [8], resource usage, etc. Because of these different objectives and the high performance of the computing ambiance, many priority-based cloud task scheduling algorithms have been proposed to outweigh the appropriate allocation map of tasks due to various factors [7].

This section provides an overview of three selected priority-based task scheduling algorithms, along with a description of how the priority issue is handled in each of the algorithms.

3.1 A Dynamic Optimization Algorithm for Task Scheduling in Cloud Environment:

Monika Choudhary and Sateesh Peddoju proposed a Dynamic Optimization Algorithm to solve the problem of Task Scheduling in the cloud environment. In their paper [3], the concepts of three different approaches are merged to create and develop a new scheduling algorithm. The three approaches are Task Grouping, Prioritization, and Greedy Allocation. Task Grouping, which generally means a set of components grouped based on some attributes. In terms of the cloud notion, it is a collection of similar tasks grouped and scheduled collectively [9], aiming to reduce the cost-communication ratio. Moreover, prioritization determines the prominence of a task based on certain parameters where the order of tasks is built accordingly; this is in terms of task scheduling [7]. A Greedy Allocation refers to the work of the Greedy algorithm, which is appropriate for a dynamic heterogeneous resource environment. It is a well-known approach used to choose the optimal choice in a specific moment, and this serves as one of the solutions for the task scheduling problem [10].

The authors proposed a technique to group tasks based on two constraint groups, the first is deadline-constrained tasks, and the second is cost-based tasks. The main idea of their proposed algorithm is: First, incoming received tasks in the broker are grouped according to their constraint categories, which are deadline constrained and the low-cost requirement. Second, prioritizing parameter is based on the constraint categories of tasks, reflecting a shorter deadline and more profit. Accordingly, the task with a shorter deadline is given a higher priority and scheduled to execute first. In addition, tasks with more profit must be scheduled on lost-cost machines. Third, for each prioritized task in the deadline constrained group, Turnaround Time is calculated at each resource by this Equation (1) as follows:

Turnaround Time = Resource Waiting Time + (Task Length/ Processing Power of Resource) (1)

Then the VM with minimum turnaround time according to Equation (1) is selected. Moreover, for each prioritized task in the cost-based group, the selection of VM is based on the machine processing power and its cost, where the resource with the least cost is selected. Accordingly, the Tasks with minimum cost are scheduled to the VM. However, the cost of each task is computed by the Equation (2) as follows:

Cost of Task = (Task length / Processing Power of Resource) * Resource Cost. (2)

Finally, the resource capacity and waiting time are updated accordingly to the third step and Equation (1), Equation (2). Figure 2 illustrates their proposed algorithm.

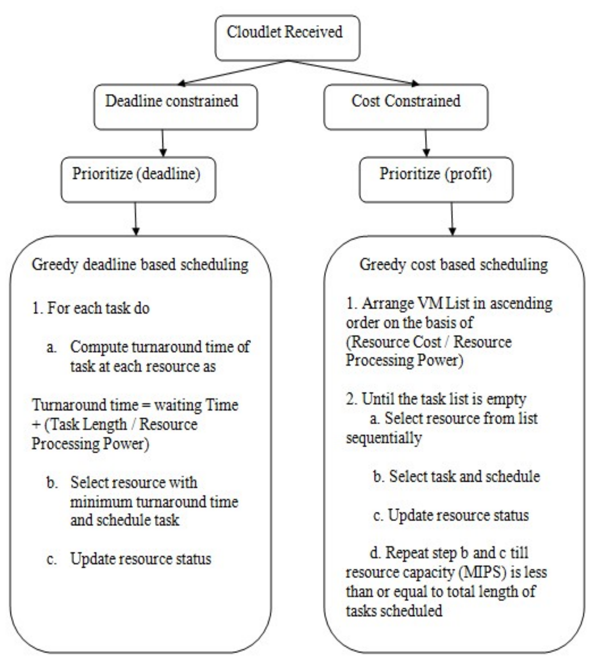

Figure 2: Choudhary and Peddoju Proposed Algorithm.

The authors used the CloudSim toolkit [11] to justify the correctness of their algorithm, where the jobs are dynamically distributed, and the experiments are accomplished compared with a sequential approach which is the default in the CloudSim simulator. The results show improved cost and superior task completion time, as shown in Figure 3.

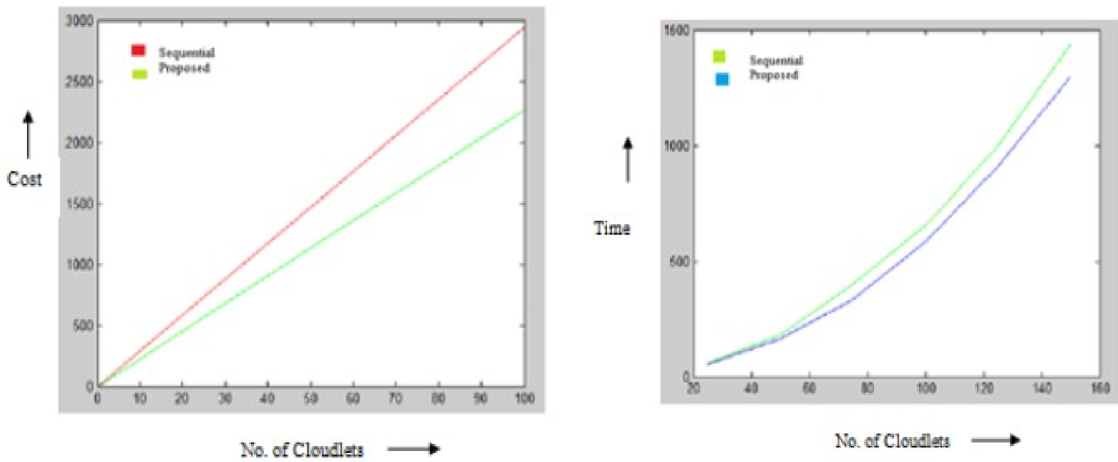

Figure 3: Analysis of Execution Cost and Task Completion Time.

3.2 A Priority-Based Job Scheduling Algorithm in Cloud Computing:

Shamsollah Ghanbari and Mohamed Othman proposed a Priority-based Job Scheduling Algorithm in the cloud environment. In their paper [5], the authors focus on the wide variety of attributes that face cloud environments. From their point of view, TS algorithms should pay attention to multi-attribute and multi-criteria properties of jobs in cloud environments. A Priority-based Job Scheduling algorithm (PJSC) was proposed based on a mathematical model. The algorithm adapts the theory of the Analytical Hierarchy Process (AHP), which is a Multi-Criteria Decision-Making (MCDM) and Multi-Attribute Decision-Making (MCDM) model. AHP architecture comprises three levels: the objective level, attributes level, and alternatives level, respectively [5]. The foundation of AHP is a comparison matrix. Therefore, AHP is suitable for priority-based problems such as scheduling with various attributes and alternatives [12].

Basically, the foundation of PJSC is a consistent comparison matrix where the resource with specific priority is requested by the job [5]. The comparison matrixes related to each job are computed according to the resource accessibilities since each job's priority is compared to others. In addition, the comparison matrix of resources is also computed. Then a priority vector (a vector of weights) is computed for each comparison matrixes.

Their Scheduling algorithm consists of three different levels of priorities, which are the scheduling level (objective level), resources level (attribute level), and task or job level (alternative level) [5]. The main idea of The PJSC algorithm is summarized in the following steps [5]:

1) Enter J=J1,..., Jm a set of jobs requesting a recourse.

2) Enter C=C1,..., Cd a set of the available resources.

3) For J, compute a consistent comparison matrix based on the priority of resources accessibilities (d matrix), and compute the priority vector for all the d matrixes.

4) Make a matrix with priority vectors for the matrix in the previous step and give it a name.

5) For C, calculate a consistent comparison matrix that determines which resource has higher priority than others according to the decision maker (s), then compute the priority vector for the matrix, then apply step 4 and name it.

6) Compute Value of Priority of jobs (PVS) vector.

7) select the job with the maximum PVS priority value and allocate a suitable resource.

8) Update the jobs list.

9) End.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Despite the paper's primary goal of providing a new priority-based job scheduling algorithm, the authors also analyzed the proposed model and discussed some aspects of the algorithm, such as complexity and finish time. Their proposed algorithm complexity is based on computing the priority vectors of comparison matrixes depending on the number of jobs and resources [5]. Moreover, regard the finish time; the authors suggested obtaining the average value of the finish time as a possible way of calculating it for the PJSC algorithm. Although this algorithm mainly focuses on priority, it's worth mentioning that the finish time is one of its main limitations, an improvement to gain less finish time is recommended.

3.3 Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment:

Agarwal Amit Agarwal and Saloni Jain proposed an Efficient Optimal Algorithm for Task Scheduling in Cloud Computing Environment. In their work [2], a priority scheduling algorithm called a Generalized Priority Algorithm (GPA) is developed where the task's priority is based on the user demand and specific parameters such as task size and VM model, and speed [4]. In addition, the authors discussed two algorithms to help develop a new generalized priority-based algorithm: the First Come First Serve (FCFS) and the Round Robin (RR) scheduling algorithms.

Briefly, in FCFS scheduling, the process is fast and straightforward, where the tasks that come first are allocated to the available VM. The disadvantage of FCFS is that it is non-preemptive [2], where the shortest tasks at the back of the queue must wait for the long task at the front to finish. In RR Scheduling, more fairness is considered; each task will execute in turn without waiting for the preceding tasks to finish [2]. The drawback of the RR algorithm is that large-size tasks take a long time to complete [4]. The solutions that may overcome these drawbacks were employed and integrated to form a new generalized priority-based algorithm by the authors in [2]. Thus, the proposed algorithm prioritizing key factors is the size for prioritizing tasks and the Million Instructions Per Second (MIPS) for prioritizing the VMs. The tasks are prioritized according to their size; the task with the largest size has the highest rank. However, the VMs are prioritized according to their MIPS value; the VM with the highest MIPS has the highest rank or priority [2]. Accordingly, the task with the highest rank is assigned to the VM.

This policy performs better than FCFS and RR scheduling resulting in less execution time [2]. The main idea of their proposed algorithm is: first, based on the computational power, VMs are created and allocated to different Datacenters, and task length is assigned. Second, a VM load balancer will maintain an index table of VMs. Third, bound tasks according to the length and respective MIPS. Finally, the tasks with the largest size get the VM with the highest MIPS [2]. The Authors compared their proposed algorithm with the RR scheduling and FCFS algorithms through experiments conducted on several VMs and workload traces. The results show that their proposed GPA algorithm was more efficient and performed better than FCFS and RR.

4. DISCUSSION

The main problem this paper discusses is allocating a set of tasks received by the Datacenter broker to the available list of VMs, focusing on the priority of tasks, to achieve the goal of minimal optimized execution time. Since the performance of machines implementing each task differs from machine to machine, the execution time of a task in a real cloud computing environment depends on such performance. Hence, various task-scheduling algorithms have been introduced and developed by several researchers, where the performance of the system relies on the adaptation of the appropriately selected algorithm. Three different Priority-based task scheduling algorithms [3] [5] [2] with various factors and parameters have been selected and investigated in this paper to compare the task scheduling method based on priority in the cloud environment. Each method of such algorithms is briefly described in the previous section. Accordingly, I have created Table 1 to highlight the result of the comparison of these three algorithms using the following criteria: the scheduling method, scheduling parameter, algorithm performance, and algorithm limitations. In the Dynamic Optimization Algorithm [3], the priorities of tasks are based on task minimum execution time or minimum cost after deploying a task grouping and prioritization method using a greedy approach specifying the resource allocation [3]. Using the CloudSim tool, the authors compared the algorithm to a sequential approach. The results show the proposed algorithm's enhanced cost and better task completion time. However, a main limitation of this algorithm is the communication overhead issue, which needs to be minimized. Hence, the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

communication overhead can be reduced in future work by grouping the cost-based tasks before resource allocation based on the resource capacity. In addition, it is recommended to consider the type and length factors of tasks in the future for a convenient task-scheduling approach [6].

The Priority-based Job Scheduling Algorithm (PJSC) [5] is based on a multiple-criteria decision-making model. The result of their paper meets the main objective, which is the priority since the proposed algorithm has reasonable complexity based on computing the priority vectors of the comparison matrixes. Mention that the worst-case complexity of their algorithm is calculated as the Equation (3) as follows:

$$\Omega = \alpha 2.81 + d * m2.81 \tag{3}$$

where d and m are the numbers of resources and jobs, respectively.

The main limitation of this algorithm is handling the inconsistency issue of the comparison matrix, where investigating the consistency is a critical issue to indicate if the comparison matrixes have a reasonable value or not. Moreover, another limitation is the finish time (makespan). Since this algorithm mainly focuses on the priority of jobs, the algorithm is not expected to give an optimal finish time. Therefore, the proposed algorithm needs to be improved for future work to gain less finish time.

The priority is given based on specific criteria in the Generalized Priority Algorithm [2]. Thus, tasks are prioritized based on the size criteria, where the task with the largest size has the highest priority, and the VM are prioritized based on their MIPS value, where the VM with the highest MIPS has the highest priority [2]. Hence, allocate to the VM the task with the highest priority. The algorithm is tested in a CloudSim toolkit, and the experiment conducted its ability to give better performance compared with the FCFS and RR algorithms. Thus, the results show that the algorithm is more efficient than FCFS and RR algorithms [2]. One of the main limitations of this algorithm is the use of limited tasks. More tasks can be taken with the possibility of minimizing the execution time in the future.

Moreover, the algorithm can be developed for Grid environments, and the difference in time between the cloud and grid can be observed [2]. Not considering the completion time and load balancing as a criterion for prioritizing can also be another limitation of this algorithm [13]. As observed from the comparison in Table 1 below, the best performance is given by the Dynamic Optimization Algorithm with more overhead. Meanwhile, the PJSC Algorithm focuses more on priority with less finish time as a limitation. The Generalized Priority Algorithm considers execution and response time parameters to find the optimal task scheduling algorithm.

Finally, to conclude this discussion, we found that an efficient Task Scheduling procedure is crucial for cloud applications to handle massive amounts of tasks driven by various types of applications, especially in a worldwide disaster such as Corona, where it profited from the cloud-based e-learning settings as described by [14]. Moreover, Task Scheduling is a complex endeavour in big data processing clusters installed on clouds. The tremendous amount of data raises the demand for sophisticated computations in the cloud. Users' task content type is a vital concern in cloud computing. Many cloud TS strategies described in the literature do not consider the content type of the user-submitted tasks [2][3][4][5], which can enhance and balance workload distribution among various servers in the cloud environment. Task Scheduling techniques can be improved if equipped with related DS and AI-based techniques. Hence, there is a need to study and recommend potential future directions that can lead to novel, superior scheduling algorithms and adjustments of present solutions to fulfill the requirements of the data science ecosystem as briefly described in the next section.

Additionally, new solutions and frameworks are needed to efficiently handle, and schedule data tasks loaded to the cloud. Due to the summarized findings in Table 1 of the compared algorithms, some of the mentioned limitations include communication overhead issues. To process such issues, it was recommended to consider the type and length factors of submitted tasks [6]. Meanwhile, other limitations were the inconsistency issue [5] and the ignorance of vital criteria for prioritizing, such as the completion time and load balancing [2]. Therefore, in this paper, to overcome such limitations, a conceptual framework incorporating a task content-type component is proposed to support the processing and scheduling of data-related tasks submitted to the cloud, as explained in detail in section 6.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: Comparison Table for some Priority-based Task Scheduling Algorithms in Cloud Environment

Scheduling Algorithm	Scheduling Method	Scheduling Parameter	Algorithm Performance	Limitations
A Dynamic Optimization Algorithm for Task Scheduling in Cloud Environment [3]	Task grouping, Prioritization, and Greedy allocation	Priorities to tasks are employed based on group type (deadline, cost)	Improved cost and better completion time of tasks	Communication overhead
A Priority-Based Job Scheduling Algorithm (PJSC) in Cloud Computing [5]	Priority using Analytical Hierarchy Process (AHP), a Multi-Criteria and Multi-Attribute Decision-Making (model	PVS (value of priority of jobs): A priority vector (vector of weights) is calculated for each task comparison matrixes.	Reasonable complexity based on computing the priority vectors of the comparison matrixes	Inconsistency. Optimal finish time not provided
A Generalized Priority Algorithm (GPA) in Cloud Computing Environment [2]	Priority using task size and VM MIPS value parameters	Execution time, response time	better performance than the FCFS and RR algorithms (less execution time)	limited number of tasks

5. RECOMMENDATIONS AND FUTURE DIRECTIONS IN DATA SCIENCE

According to the above discussion in section 4, despite some state-of-the-art basic task scheduling algorithms described in this paper, it's highly recommended that Task Scheduling techniques might be better improved if equipped with related Data Science and Artificial intelligence-based techniques. Hence, Task scheduling techniques are vital for current and future data science applications in cloud settings. Therefore, several potential future directions, especially for the priority case, are recommended to focus on the configuration and integration of AI and big data, especially in the era of real-time settings. These advances will lead to the need for novel, superior scheduling algorithms and adjustments of present solutions to fulfill the requirements of the data science ecosystem. For instance, it's time to look inside the content of the tasks submitted to the cloud and what datatypes it emerged from and prioritize accordingly. In the next section, we proposed a conceptual framework regarding this issue.

The following are some of the related AI-DS suggested potential future directions shaped around the concept of cloud-based data processing and analysis in Real-time Processing, including the need for the idea of Multi-Cloud, and the integration with AI and Machine Learning, as well as the Security and Privacy concerns:

5.1 Integration With Data Analytics Platforms and Big Data Volumes:

Due to the continuous expansion of big data use in various fields, and simultaneously most enterprises ' data-related applications moved to cloud environments. As a result, cloud-based task scheduling techniques nowadays become required to handle and allocate target resources in data analytics platforms (e.g., Spark and Hadoop) efficiently, to manage complex data sets. With such circumstances, priority-based task scheduling techniques must also be improved to work in such environments and handle big data. This includes creating novel solutions considering massive data volumes processing and storage recent challenges while maintaining general cloud computing characteristics like network latency and resource allocation.

5.2 Real-Time Processing Settings:

With the continuous increase of cloud-based data science solutions that adopt real-time processing, priority-based task scheduling techniques and algorithms must be developed to adapt and satisfy the demands of these scenarios. Such improvement might entail optimizing the priority algorithms to adjust to dynamic changing circumstances in real-time and accordingly prioritize the required tasks effectively. Moreover, this will play a significant role in the process of Auto-scaling and dynamic resource allocation in the cloud environment, especially in the big data era and the vast amount of data being processed and created in real-time applications.

5.3 Integration Of Cloud Computing With Artificial Intelligence And Machine Learning:

Since the cooperation and interference between AI and data-related tasks are rapidly increasing, the need for intelligent priority-based task scheduling algorithms is continuously arising in cloud-based environments. For example, such approaches can help in predicting the next task priorities using machine learning approaches as well as predicting the required resource requirements. Moreover, ML approaches can also have a significant role in the task of dynamically allocating optimal resources. In addition to its classification-related roles to classify tasks based on specific features inside the cloud to satisfy a particular objective. This intelligent intervention can reduce cost and improve the efficiency, scalability, and resource utilization of cloud-based data analysis and processing, reflecting the main goals of using cloud-based environments.

5.4 Multi-Cloud Utilization:

Data analysis professional workers and scientists sometimes work with data acquired from multiple resources and hence multiple cloud platforms. Accordingly, scheduling data work-related tasks through multi-cloud platforms will become increasingly crucial since small to mid-size enterprises now aim to optimize their cloud-based systems as well as their general computerized infrastructure to process their big data cloud-based tasks efficiently and timely manner.

5.5 Data Security and Privacy Issues:

The conjunction of the use of big data through cloud platforms is expanding. This means aspects of data privacy and security concerns are extremely important, especially with the rapid integration between cyber security and data science techniques. Therefore, Task scheduling techniques with sufficient security and privacy measures are now essential and will find more significant usage. For example, Data Science AI techniques with such measures can help predict tasks with a cloud-driven attack.

6. CONCEPTUAL FRAMEWORK FOR TASK SCHEDULING IN CLOUD COMPUTING INCORPORATING A TASK CONTENT-TYPE COMPONENT

In the age of big data application processing, the cloud user's submitted tasks content type is a valuable concern in the cloud computing setting. Additionally, various factors contributing to an efficient task scheduling process can be preferably addressed in a broad context that can be implemented into big data-related applications. Research for task scheduling in cloud settings is continuously progressing and has yet to be further revealed.

Based on the above reasons, in this section, we proposed a conceptual framework in Figure 4. Which is an integrated framework incorporating AI and DS components with task scheduling techniques in the cloud environment to support the procedure of processing and scheduling big data-related tasks submitted to the cloud. The suggested conceptual framework has been articulated through three primary stages: resource allocation, content-type classification component, and priority component.

6.1 Resource Allocation:

The resource manager in cloud computing online settings is responsible for distributing the available resources to guarantee the best utilization of resources for the cloud-based submitted tasks. In the modern age of data, numerous data as tasks are submitted to the cloud from several data sources such as Social Media, Multimedia, Mobile Apps, Financial, and IoT Data [15]. Therefore, to handle this immense quantity of data, the cloud service providers allocate the required resources based on resource availability for such tasks before transferring them to the next component.

6.2 Content-Type Component:

The proposed "Content-Type Component" is an AI-integrated component capable of processing submitted user cloud tasks by classifying them into text, image, audio, and video based on their various types of content. State-of-the-art machine learning techniques, such as supervised machine learning algorithms, can be employed for the task of classification, as outlined in the proposed framework.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

This classification component generates a list of classified tasks based on the content type. For instance, one of the well-known characteristics for classifying such tasks according to content type is based on the file fragment type using the FFT-75 dataset in [16]. This recently available diverse and balanced dataset includes sampled file fragments from 75 known file types, including content types such as text, image, audio, and video, randomly. The dataset is labeled with class IDs and fine-tuned for classification, specifically using supervised machine-learning models.

6.3 Priority Component:

The Priority-based Algorithm component is responsible for the task-scheduling processes. According to the suggested conceptual framework, this component is responsible for mapping a suitable VM group to each task type list returned from the previous content-type component, resulting in a list of scheduled tasks ready to be processed by the chosen group of VMs in the Datacenter. Whereas, here in this proposed framework, it's suggested that the VMs are divided into four types of groups according to the type of content into text, image, audio, and video groups. Each VM type has unique computation, storage power, and network configurations. Even more clearly, each VM has been assigned a task based on the cloud task's content type. For example, video tasks need more than 1000 MIPS, a large memory of more than 16 Gigabytes (GB), and a storage capacity of around 360 GB. In contrast, text-based tasks need about 500 MIPS of processing power and 4GB of memory, respectively.

Accordingly, the submitted tasks are mapped to the group of VMs of the same content type. Moreover, suggested future work on this framework is to propose an optimized hybrid scheduling priority-based approach to map the content type-based classified list of tasks to the suitable available VM to be accordingly processed in the Datacenter sub-component. Finally, mention that hosts include processing resources and storage components for providing data despite time and place conditions. However, VMs share specific host machine resources, such as storage, processing power, operating systems, and network configurations and interfaces, which are all regarded as virtual computers.

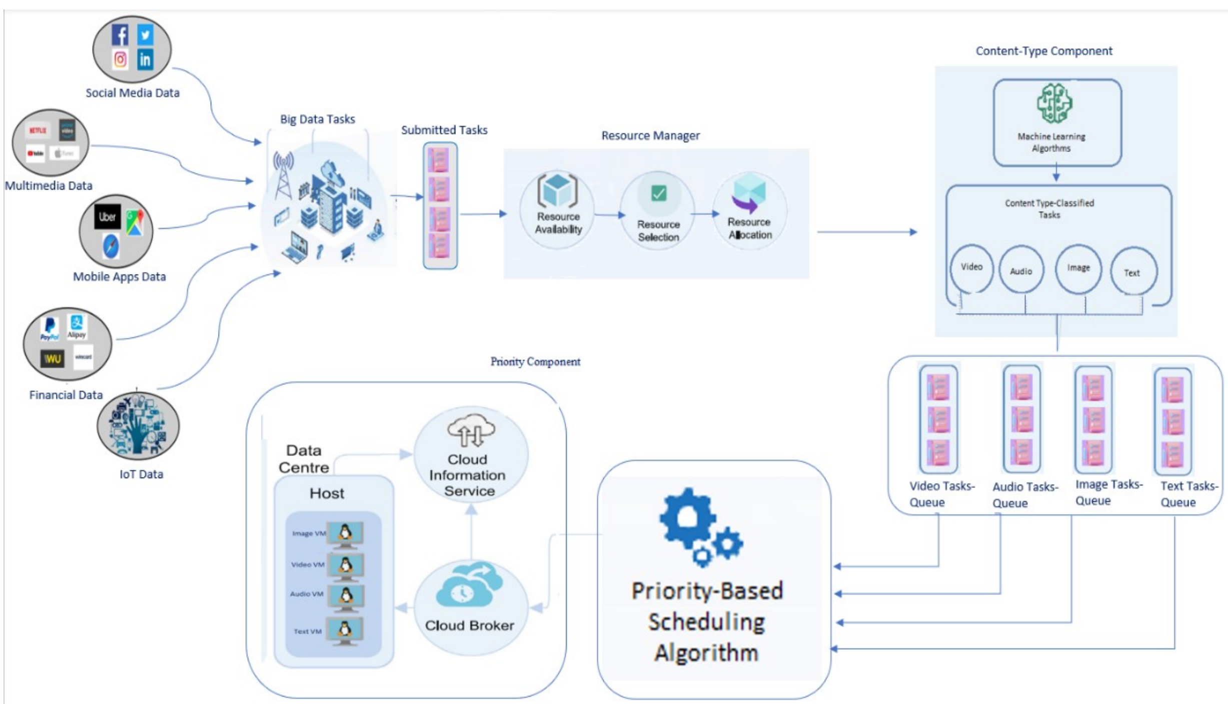

Figure 4: Proposed Conceptual Framework for Task Scheduling in Cloud Computing Incorporating a Task Content-Type Component

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

7. CONCLUSION AND FUTURE WORK

In the Cloud Computing environment, one of the most critical emerging challenges is Task Scheduling, the job of allocating a set of tasks received by the Datacenter broker to the available list of Virtual Machines. Since tasks submitted to the cloud can have various sizes, resource utilization, and execution times. Thus, scheduling the tasks based on priority wise is essential to achieve minimal optimized execution time, less response time, and resource usage, leading to better machine performance. Hence, in this paper, three different Priority-based Task Scheduling algorithms with special factors and criteria have been selected and argued to compare the Task Scheduling method based on priority in the cloud environment. The three chosen algorithms are the Dynamic Optimization Algorithm, the Priority-based Job Scheduling Algorithm, and the Efficient Optimal Algorithm. Each of the three algorithms considered the task's priority as an essential issue dealing with it according to specific criteria, leading to their advantages and limitations. The paper described each algorithm task scheduling method and how the priority issue is handled. Accordingly, these three algorithms were compared using various criteria such as the scheduling method, scheduling parameter, algorithm performance, and algorithm limitations. From the comparison, the Dynamic Optimization Algorithm gives better performance regarding completion time but has more overhead. In contrast, the Priority-based Job Scheduling Algorithm focuses more on priority and does not give optimal completion time. Observed that the Efficient Optimal Algorithm considers more parameters, such as execution and response time parameters, to find the optimal Task Scheduling method in a cloud environment.

Moreover, this paper discussed the integration of CC and Data Science techniques due to their continuous development in big data industries. Accordingly, vital suggestions for potential future directions of data-related cloud tasks have been outlined in this paper. They focus on the significance of cloud-based data processing and analysis in real-time, including the integration with AI techniques and the implications for data security and privacy in the cloud. However, Task Scheduling is a rigorous job in big data processing clusters installed on clouds. The tremendous data volumes and types are rising in the cloud due to the emerging use of big data applications.

Consequently, to facilitate the process of handling and scheduling data-related tasks loaded to the cloud. We proposed an integrated conceptual framework incorporating a cloud task content-type component that employs AI techniques to classify cloud users' tasks based on their content data types into text, image, audio, and video. Since in the literature, there is a lack of investigation of the content type of tasks received to the cloud, which has a significant role in workload distribution via many servers in cloud settings.

In conclusion, this paper mainly dealt with the problem of task priority in cloud-based task scheduling algorithms. The study was conducted in more than one direction, summarized in the following contributions:

1. Investigated three of the superior priority-based task scheduling algorithms and compared them regarding their scheduling method and parameters, algorithm performance, and limitations. The findings of the comparison are presented in Table 1.
2. Several potential future research directions, especially for the priority case, are discussed and recommended regarding the integration between AI and big data cloud-based tasks.
3. A conceptual framework for task scheduling in CC incorporating a content-type component is proposed to support processing and scheduling big data-related tasks submitted to the cloud.

However, as a researcher, I believe that priority is a crucial issue factor for scheduling various cloud tasks with distinct characteristics, especially in the age of data, to support the processing and scheduling of big data-related tasks in cloud environments. It is strongly recommended to improve Task Scheduling procedures to profit from being combined with relevant Data Science and AI-based methods. Task scheduling strategies are essential for current and future data science applications in cloud settings. Hence, the suggested future work for this paper is outlined as follows:

- Browse how priority-based task scheduling algorithms can be developed to adjust to real-time dynamically changing conditions in cloud-based data-related applications such as real-time data mining techniques.
- Browse the state-of-the-art priority-based task scheduling algorithms based on Machine Learning techniques.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

- Explore more about the challenges and techniques of task data privacy and security in cloud platforms.
- Developing an implementation strategy for the proposed conceptual framework using optimized hybrid scheduling approaches to map the content type-based classified list of tasks to a suitable available VM. Also, later, the implementation will be evaluated and explicitly compared to some manifested load-balancing techniques. Many parameters can be tested at that time, such as cost and energy consumption, overhead time, and optimization time.

REFERENCES:

[1] B. Sosinsky, *Cloud Computing Bible*, Hoboken, NJ, USA: Wiley, pp. 1–22, 2011.

[2] D. Agarwal, S. Jain, "Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment," *International Journal of Computer Trends and Technology*, vol. 9, no. 7, pp. 344–349, 2014.

[3] M. Choudhary, S. Peddoju, " A dynamic optimization algorithm for task scheduling in cloud environment," *International Journal of Engineering Research and Applications (IJERA)*, vol. 2, no. 3, pp. 2564–2568, 2012.

[4] V. Geetha, R. A. Devi, T. Ilavenil, S. M. Begum and S. Revathi, "Performance comparison of cloudlet scheduling policies," *Proceedings of IEEE International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS)*, pp. 1–7, 2016.

[5] S. Ghanbari, M. Othman, "A priority based job scheduling algorithm in cloud computing," *In Procedia Engineering*, vol. 50, no. 0, pp. 778–785, 2012.

[6] M. Awan, M. Shah, "A Survey on Task Scheduling Algorithms in Cloud Computing Environment," *International Journal of Computer and Information Technology*, vol. 4, no. 2, pp. 441–448, 2015.

[7] S. A. Murad, A. J. M. Muzahid, Z. R. M. Azmi, M. I. Hoque, and M. Kowsher, "A review on job scheduling technique in cloud computing and priority rule based intelligent framework," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 6, pp. 2309–2331, 2022.

[8] D. Mukherjee, S. Ghosh, S. Pal, AA. Aly, DN. Le "Adaptive Scheduling Algorithm Based Task Loading in Cloud Data Centers," *IEEE Access*, vol. 10, pp. 49412–49421, 2022.

[9] P. J. Wild, P. Johnson, H. Johnson, "Understanding task grouping strategies," *People and Computers XVII—Designing for Society*, pp. 3–20, 2004.

[10] S. Singh, K. Kant, "Greedy grid scheduling algorithm in dynamic job submission environment," *International Conference on Emerging Trends in Electrical and Computer Technology (ICETECT)*, pp. 933–936, 2011.

[11] 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

[12] M. Madhooshi, "developing an integrated model for calculating the customer lifetime value," *The 4th International Management Conference*, 2007.

[13] A. Mahmoud, M. Zarina, W. Nik, F. Ahmad, "Multi-criteria strategy for job scheduling and resource load balancing in cloud computing environment," *Indian Journal of Science and Technology*, vol. 8, no. 30, pp. 1–5, 2015.

[14] R. Al Bashaireh, "Cloud-Based E-learning: Concepts, and its Beneficial Role During the COVID-19 Dilemma," *In Proceedings of Seventh International Congress on Information and Communication Technology*, pp. 491–505. Springer, Singapore, 2023.

[15] M. Chen, S. Mao, Y. Liu, "Big data: A survey," *Mobile networks and applications*, vol. 19, no. 2, pp. 171–209, 2014. DOI 10.1007/s11036-013-0489-0.

[16] G. Mittal, P. Korus, N. Memon, "File Fragment Type (FFT) - 75 Dataset". IEEE Dataport, 2019, doi: 10.21227/kfxw-8084.