

Article

Web Technology Grounded Effects of Task Scheduling in Distributed and Cloud Systems

Halbast Rasheed Ismael¹, Lozan M. Abdulrahman², Zryan Najat Rashid³, Riyadh Qashi⁴

¹IT Dept., Technical College of Informatics, Akre University for Applied Sciences, Duhok, Iraq

²ITM Dept., Technical College of Administration, Duhok Polytechnic University, Duhok, Iraq

³Network Dept., Technical College of Informatics, Sulaimani Polytechnic University, Sulaimani, Iraq

⁴Vocational School Center 7, Electrical Engineering of the City of Leipzig, Leipzig, Germany

*Corresponding author: Halbast Rasheed Ismael (email: rasheed.ismael@gmail.com)

Received 01 September 2022; Accepted 07 November 2022; Issue Published 15 December 2023

Abstract: One definition of the word "distributed system" describes it as "a set of entities that collaborate in order to find a solution to a problem that cannot be solved by a single entity using their own resources." This description of a distributed system is an example of a distributed system. As the number of algorithms that are mathematically complicated continues to increase, distributed computing systems have emerged as a direct result of this trend. The optimization of a distributed computing system has been accomplished via the development of methods for the distribution of work and the scheduling of jobs. Because of this, the system has been able to be used in a more efficient manner. Task scheduling refers to the process of selecting the order in which actions are carried out in response to a given set of circumstances. On the other hand, task allocation is the process of allocating tasks to the processors in a system that are the most fit for taking on those tasks. This procedure determines which processors are assigned the jobs. Within the context of distributed systems, the objective of this article is to provide a detailed review of the several approaches to task scheduling that have been used by researchers.

Keywords: Web Technology; Task Scheduling; Distributed System; Cloud Computing.

Copyright © 2023 Journal of Smart Internet of Things (JSIoT) published by Future Science for Digital Publishing and Sciendo. This is an open access article license CC BY (<https://creativecommons.org/licenses/by/4.0/>)

1. Introduction

These capabilities are made possible by a programming framework that is referred to as a distributed system. This framework makes it possible for a collection of workstations to operate as a single system when they are used in conjunction with one another. The primary objective of this endeavor is to make an effort to disguise the unattractiveness of restricted resources that are distributed over a big number of diverse hosts who are all different from one another. The phrase "distributed system" is used in the field

of information technology to describe to a collection of storage devices, databases, and autonomous processors that collaborate with one another to achieve a shared objective. This is the material that the user has entered into the system, which is as follows: [1] The communication network is accountable for a variety of tasks, some of which include, but are not limited to, the transmission of information, the facilitation of communication, and the synchronization of the many processes. These are only some of the functions that the network is responsible for. There are certain systems that are composed of a set of processes that are intended to make it possible for hosts to communicate with one another. These systems are referred to as communication systems. The hosts are able to interact with one another via the use of these mechanisms. There are also additional systems that are designed to closely connect a large number of hosts together and offer the application with a single interface. These systems are different from the ones described above. The purpose of these systems is to be used in cooperation with one another simultaneously. Different forms of distributed systems are referred to as loosely linked distributed systems and tightly connected distributed systems, respectively. In general, these terms are used to describe the two types of distributed systems. The occurrence of this is rather common. The following is a representation of the text that was input by the user: [2]

Computer systems make decisions on the sequence in which tasks should be completed as well as the processor that should be used to carry them out . These decisions are made via the use of methods from the fields of task scheduling and task allocation. These judgements are made with relation to the order in which the activities should be completed when they are completed. The process of task scheduling is carried out in order to determine the sequence in which tasks are finished and the allocation of processors in line with the responsibilities that they are allocated. This is an important step in the process. When specific requirements are satisfied, the process of mission scheduling is carried out, which includes carrying out both the allocation and the selection of the execution order from among the available options. Additionally, it is accountable for concurrently carrying out both of these responsibilities[3]. This is the text that the user has supplied, which the user has decided to refer to as. From a scheduling perspective, the most basic units that may be planned are tasks, which are composed of threads or processes. Tasks can be scheduled in a variety of ways. The building blocks of tasks are the most essential components. In order to schedule work, there are a variety of various approaches that may be used. Examples of activities that are thought to be NP-hard problems include the scheduling of jobs in a heterogeneous distributed system that includes a high number of processors. This is an example of an activity. Specifically, this is because the process of task scheduling is an activity that involves looking for the most effective potential answer. This is the reason why this is the case. Specifically, this is the text that the user has input, which is denoted by the number . What differentiates an NP-Hard problem from other kinds of problems is the fact that the answer to an NP-Hard problem can only be computed in exponential time. It is possible, however, to solve other kinds of problems in an exponential amount of time. The complexity class that is referred to as NP-Hard is one that is of a difficulty level that is either equal to or greater than that of NP. This class is characterized by the fact that it is tough. The fact that this class is challenging is one of its defining characteristics. The phrase "non-deterministic polynomial time" is abbreviated as "non-deterministic polynomial time," which denotes that problems of this complexity cannot be solved within a time period that is polynomial[4]. Non-deterministic polynomial time signifies that this is the case. One way to abbreviate this statement is to use the term "non-deterministic polynomial time." The use of approximation strategies is a common technique that is utilized in order to rapidly seek a solution inside a time range that is regarded as acceptable. Genetic algorithms and heuristics are two examples of the tactics that fall under this category. It is possible that the text that the user has input will not be located. This is a possibility. Task scheduling is a process that is used in the context of distributed systems. Its purpose is to optimised the utilization of the system's hardware resources in order to get the highest possible level of performance from the system. Distributed systems are a set of components that are linked to one another over a network and work together to solve a specific issue. The name "distributed system" refers to this organisations of components. It is possible that this is the definition of a distributed system when it comes to the context of handling problems of

this kind. Distributed systems are distinguished from other types of organisations by a number of primary characteristics that distinguish them from other types of organisations. Some of the variables that contribute to these obstacles include the geographical dispersion of distributed systems, the absence of a synchronized physical clock, and the frequent use of distributed systems to handle complex problems that would be impossible to address inside a single organisations. These are only some of the elements that contribute to these challenges. This is the input that was provided by the user, which is as follows:[5].

The remainder of this paper organized as follows. Section 2 provides background theory including: distributed systems, Scheduling in Distributed Systems. The literature review is presented in section 3, Section 4 provided Comparison and Discussion, and the conclusion is in Section 5.

2. Background Theory

2.1 Distributed System

A distributed system is a network of autonomous processors or entities that work together to discover answers to issues that cannot be handled by a single entity operating in isolation. When we talk about a distributed system, we are referring to this network. Both a shared memory and a physical timekeeping device are absent from the units that are able to interact with one another across a network. These units are able to communicate with one another. Furthermore, these units are often situated in a variety of geographical regions, and they also have their own individual arrangements that are distinct from one another. This is the number that the user has entered, which is [6]. Wide-area networks and computer systems are two examples of distributed systems that may emerge in a variety of different forms. These systems can be found in a variety of instances. They are both examples of distributed systems. Both of these systems are examples. Peer-to-peer networks and client-server networks are both instances of distributed systems; yet, they are distinct from one another and may be described in their own right. It is possible to distinguish between the two types of networks. In a great number of instances, the client-server architecture is used as the primary basis for distributed systems. Presented in Figure 1 is a graphical illustration of the essential architecture that serves as the basis for a distributed-based system.

"Dependable systems." The fact that this is the case is shown in Figure 2, which demonstrates that dependability is a crucial concept in distributed systems, which spans a wide range of fundamental requirements. In their presentation, which can be found in [7], the authors highlight some of the most essential qualities of dependability. These characteristics include how reliable something is. The following categories of attributes are included here:

- It is possible to use the word "availability" to refer to the rate at which a system may be used within the framework of a system[8].
- On the other hand, when we speak about dependability, we are referring regarding the capacity of the system to perform in a consistent method and without any issues.
- Safety is the assurance that there will be no disastrous outcomes, even in the event that the system fails to function appropriately. Assurance against catastrophic impacts is the term that is used to describe this protection.
- Maintainability: In the event that the system has a malfunction, it is possible to fix it in a short amount of time and with a little amount of work, and the users are often unaware of the issue with the system. An important benefit is that this is the case[9].

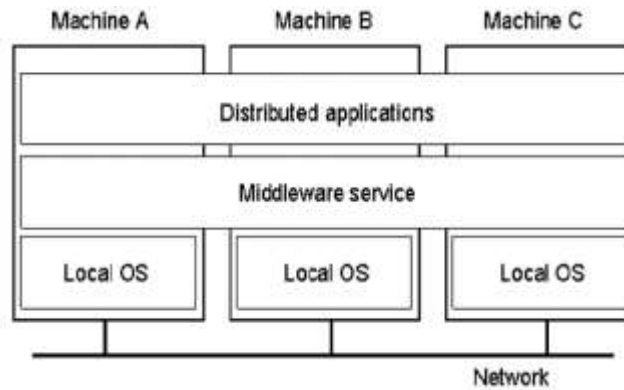


Figure 1: Architecture of a distributed system[7]

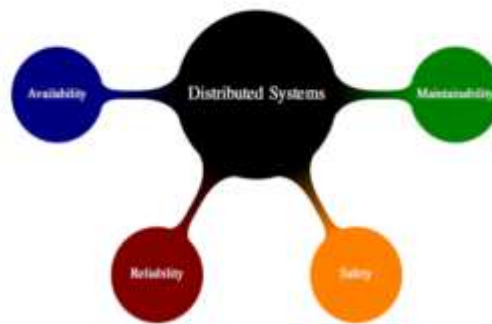


Figure 2: Dependable distributed systems[12]

2.2. Scheduling in Distributed Systems

Undertaking assignment scheduling is the process of deciding the sequence in which tasks will be completed. This procedure is referred to as "putting together a schedule[10]." The term "process scheduling" is also used to refer to this concept on occasion. Despite the fact that the hardware had a limited capacity, task schedulers were utilized in distributed systems to guarantee that the hardware was utilized to the greatest degree feasible inside the system. To frame it another way, the purpose of these schedulers was to devise a procedure that would ensure that no processors would be left idle while tasks were being done. This was the goal all along[11].

A. Task Scheduling

The algorithms that are used for task allocation and scheduling in a distributed computing system have the primary goal of optimizing the overall performance of the system. This is the case because the algorithms are designed to maximize the efficiency of the system. This is the case in order to ensure that the system is able to perform to the best of its abilities. The major purpose of the algorithms is to do this kind of thing[12]. The number in question is going to be . Throughout the process of determining the order in which each task should be completed, task scheduling algorithms are used in order to accomplish the goal of assigning tasks to the worker or processor who is the most suited for the job[13]. This is done in order to achieve the purpose of allocating tasks to the worker or processor who. To phrase this another way, one might say that a schedule is a list of tasks that are allocated to processors in a certain period of time during the course of the day. This would be an alternative way of putting this. To phrase it another way, this is yet another strategy that may be used. Work schedules are responsible for

assigning tasks to processors and ensuring that those activities are carried out in the sequence that the scheduler has specified [14]. In addition, they are responsible for ensuring that the tasks are carried out in appropriate order.

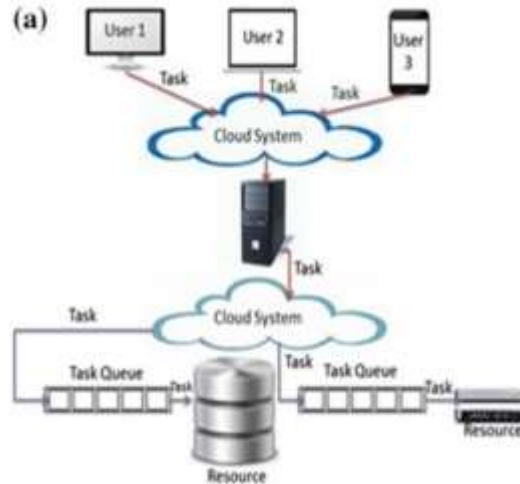


Figure 3: Task-scheduling architecture.

B. Scheduling Classification

The process of work scheduling may be broken down into two distinct components: static scheduling and dynamic scheduling. Both of these components can be isolated from one another throughout the process. In contrast to one another, these two components are distinct from one another. Static scheduling refers to the scheduling that occurs at the build time, whilst dynamic scheduling refers to the scheduling that occurs during runtime. Both types of scheduling are referred to as "scheduling." When a static scheduling approach is used, there is a limitation placed on the whole amount of tasks that may be planned. As a result, we have arrived at the fourth point. When a dynamic scheduling model is used, on the other hand, new tasks are continually added to the system while the scheduling process is being carried out simultaneously. Throughout the whole of the scheduling process, this is something that takes place[15]. is the number in question. In circumstances in which static scheduling is used, it is very essential for all of the information about the jobs to be accessible in advance whenever it is feasible to do so. The purpose of offering these particulars is to serve as an example, bringing attention to the many conditions that need to be achieved in order to successfully perform the work. On the other hand, dynamic scheduling is aware of a great number of information on the tasks that are going to be completed in the near future, in contrast to dynamic scheduling, which is merely aware of a little amount of information or none at all about the jobs themselves. On the other hand, dynamic scheduling is aware of a great lot of information about the tasks that are going to be carried out in the not too distant future[16]e. This approach, which is more difficult to implement and which I will discuss in more depth in the following paragraphs, results in an increase in throughput while simultaneously lowering the amount of time that is necessary to do the work. During the process of task scheduling, algorithms are categorized as either preemptive or non-preemptive, based on the conduct that they exhibit. When determining the classification of this kind of labor, the question of whether or not a job may be placed on hold and then picked up again at a later point in time is taken into consideration[17]. When it comes to the manner in which actions are carried out, the use of preemptive algorithms is highly advantageous. Stopping the operations that are now being carried out on the processing unit is something that the system is able to

do in order to guarantee that another activity is carried out on the unit. Policies that have been defined are taken into consideration while making the decision to preempt. As opposed to what non-preemption algorithms do, preemptive algorithms often carry out tasks even when the activities have not yet begun in their entirety. This is the opposite of what non-preemptive algorithms also perform. On the other hand, this is in contrast to the conduct of algorithms that are not preemptive. Additionally, in order for the system to continue operating in its present condition while concurrently using preemptive strategies, extra resources are required [18].

C. Scheduling Schemes and Policies

It is possible to carry out a plan or another organized activity by using a scheme, which is a method for carrying out the plan or activity. Schemes may be used to carry out activities. This section's objective is to offer an explanation of two unique scheduling approaches that are often used in the workplace. FIFO, which is an acronym that stands for "first-in, first-out," and EDF, which is an acronym that stands for "earliest deadlines first," are two time management tactics that are used often. The concept that the task that is finished on the computer will be the one that is finished first is what is meant by the expression "first-in, first-out" (FIFO) when it is used in the context of computer work. Because it would leave the processing units with a long length of idle time, which would result in a big amount of lost time, using FIFO would be a huge loss of resources. This would be the case because it would leave the processing units doing nothing. The time is now [19, 20]. It is possible to carry out a plan or another organized activity by using a scheme, which is a method for carrying out the plan or activity. Schemes may be used to carry out activities. This section's objective is to offer an explanation of two unique scheduling approaches that are often used in the workplace. FIFO, which is an acronym that stands for "first-in, first-out," and EDF, which is an acronym that stands for "earliest deadlines first," are two time management tactics that are used often. The concept that the task that is finished on the computer will be the one that is finished first is what is meant by the expression "first-in, first-out" (FIFO) when it is used in the context of computer work. In the event that you were to implement the first-in, first-out (FIFO) approach, the processing units would have a significant amount of downtime, which would lead to a significant loss of resources. The number is twenty-eight[21].

3. Literature Review

According to the results of the research that Zhao et al. conducted in the year 2020 [22], a unique strategy to resource allocation that is presented is referred to as HPS+. This technique was shown to be effective. For the purpose of achieving the objective of reducing the quantity of traffic that is sent across wide area networks (WAN), HPS+ makes use of an improved hyper-graph by using an enhanced hyper-graph partition algorithm. The simulation of the linkages between combined task-data and data-datacenter is accomplished with the assistance of this technique, which ultimately leads to a reduction in the amount of traffic that is present in the wide area network. In addition, it makes use of a coordination mechanism to reallocate network resources in accordance with the needs of the task, so reducing the total amount of time required for the creation process to the greatest extent that is practically possible. A comparison was made between the Google datacenter model and the real China-Astronomy-Cloud model. The results of the comparison showed that HPS+ greatly reduces the number of data transfers by as much as 53 percent. In addition to this, as compared to more conventional processes, it reduces the amount of time required to make a product by a total of 39 percent.

Khorasani and his colleagues [23] provided a system that gives prosumers the chance to participate in both power trading and scheduling throughout the course of the research that they carried out in 2021. This arrangement was made possible by the system. Prosumers are able to actively participate in a variety of marketplaces and negotiate their involvement in those marketplaces when they make use of

this framework. The framework is comprised of a peer-to-peer market, which is also referred to as a P-2-P market. This market gives prosumers the opportunity to engage in energy trading with one another inside the community. In addition, there is a resilient market that enables the network operator to make advantage of the adaptability of prosumers in order to deal with constraints that are placed on the network. Because of the robust market, this skill is now within achievable reach. A system for quick distributed market adjustment has been proposed as a viable solution. This system has been proposed as a means for effectively resolving various markets while simultaneously protecting the interests of prosumers. According to the results of the simulation, which demonstrated that it is beneficial for prosumers, the proposed framework is economically advantageous for prosumers. This is due to the fact that it enables prosumers to efficiently minimise the amount of money that they spend on electricity. As an additional point of interest, in order to effectively manage grid constraints via the formation of a flexibility market, the grid operator may concurrently make use of the adaptability of prosumers. Prosumers are responsible for scheduling their own resources, as opposed to having those resources controlled from a central location. This obligation falls on each individual prosumer. The framework that was provided was able to show its computational efficiency and has the capability of being easily applied in communities that had a number of various dimensions. This was a consequence of the aforementioned.

Following the findings of their research, Yao et al. [24] (2020) recommended that a Hybrid Fault Tolerant Scheduling Algorithm (HFTSA) be used in virtualized cloud systems for the purpose of scheduling separate workloads that are subject to deadlines. Given the results of their investigation, this suggestion was made in light of those findings. The use of this strategy would include the utilization of a variety of different types of procedures. The High-Frequency Trading System Authority (HFTSA) has, in a few cases, chosen to use fault-tolerant mechanisms. Resubmission and replication are two concepts that come to mind. The specific characteristics of the task as well as the resources that are accessible via the cloud are taken into account throughout the process of picking these solutions. Following the completion of this selection, the resources that are pertinent to the job are reserved at a later time. This selection takes place throughout the process of task scheduling. An approach known as elastic resource provisioning is used by HFTSA for the purpose of scheduling. This approach is utilized in order to ensure that resources are allocated appropriately and that the utilization of those resources is maximized. In the case of the HFTSA, there is a resubmission option that can be performed online, which permits the correction of any mistakes that may have been made. Additionally, there is a process that may be used for making modifications online, and this process can be utilized for works that do need resubmission. It is possible to update fault-tolerant ways using this system, which is something that can be done in the event that it is required to be done. It has been hypothesized by researchers that the Highly Fault-Tolerant Scheduling Algorithm (HFTSA) has the potential to improve resource utilization while also ensuring that jobs are completed in a timely manner. Regarding the idea of fault tolerance, this is relevant to your discussion. It is important to note that the unexpected performance fluctuations of cloud resources have a negative impact on the situation when it comes to time-sensitive jobs that have strict deadlines. The failures that have already occurred are compounded by this new set of circumstances.

Within the context of their presentation, Fargard et al., 2020 [25] presented an innovative approach to the scheduling of activities that are associated with workflow, as well as the provision of resources. The phrase "Greedy Resource Provisioning and modified HEFT" is what the abbreviation GRP-HEFT refers to together. It was because of the capabilities of this approach that it was given the appellation. Taking into account the restrictions of a budget that are based on the hourly pricing model of modern infrastructure as a service (IaaS) clouds, the objective of this technique is to reduce the amount of time that is necessary to finish a specific application while also taking into account the constraints of a budget. Among the several strategies that they presented, one of them was the use of a greedy algorithm as a method for the allocation of resources. Within the framework of this method, a priority is given to every instance type, and the priority is established based on the efficiency rate of that specific instance type.

They also made adjustments to the HEFT algorithm in order to include a budget constraint into the overall system. All of this was done in addition to the point that was made before. In the next step, the GRP-HEFT technique was evaluated in comparison to other complicated workflow scheduling systems, such as MOACS (Multi-Objective Ant Colony System), GA (Genetic method), and PSO (Particle Swarm Optimization). Following the completion of the preceding stage, this assessment was carried out. With an average improvement of 13.64 percent, 19.77 percent, and 11.69 percent, respectively, the data demonstrated that the performance of GRP-HEFT was superior to that of GA, PSO, and MOACS for a number of well-established scientific workflow applications over a wide range of problem sizes. This was the case for a number of different scientific workflow applications. As a consequence of this, the findings indicated that the performance of GRP-HEFT was superior to that of GA, PSO, and MOACS. Additionally, when it comes to the complexity of the temporal process, the (GRP-HEFT) algorithm operates more effectively than the (GA), (PSO), and (MOACS) algorithms. Each of these algorithms has its own advantages and disadvantages.

Throughout the course of their inquiry, Yuan and his colleagues [26] (2020) concentrated on enhancing the efficacy of servers that were a component of distributed green cloud systems (DGCs). They employed a task G/G/1 queuing system in order to achieve their aim of tackling a single-objective restricted optimization challenge. This approach allowed them to fulfil their goal. They used the Bees Algorithm, which is based on the idea of simulated annealing, in order to look for a solution that was very near to being the greatest alternative that might possibly be available. The fine-grained spatial task scheduling (FSTS) algorithm is created within the framework of this approach with the purpose of decreasing the amount of money that a supplier of distributed grid computing (DGC) have to pay for the cost of energy. This is done in order to reduce the amount of money that is required to be paid. The work from a broad variety of applications is successfully distributed over a large number of DGCs, and it does this by predicting the operating speed of each DGC and the number of servers that are switched on. Additionally, it achieves this by effectively spreading the work. A further advantage is that it guarantees that all application actions are completed within the specified amount of time for responses, which is a considerable advantage. Research that is based on actual data has shown that FSTS has a greater performance in terms of power utilization when compared to other benchmark scheduling systems. This was demonstrated by the findings of previous research.

For cloud workflow activities and virtualized resources, Cui et al. [27] introduced a technique for collaborative adaptive scheduling that is based on heterogeneous distributed deep learning. This approach was developed for use with cloud workflow tasks. This strategy necessitates the use of resources that are virtualized. When it comes to multi-queue, multi-cluster workflow job scheduling and resource allocation, this technology is able to successfully address the issues that arise. As a result of the fact that it incorporates a significant number of different kinds of deep neural networks into the scheduling model of the cloud system, this is made feasible. We designed the most efficient technique of scheduling in order to cut down on the amount of time that was lost waiting for workflow tasks to be finished and to make the most efficient use of energy that was feasible. The framework that was supplied is capable of quickly resolving the multi-objective optimization problem of resource allocation and cloud workflow task scheduling. As a consequence, a solution that is virtually optimum in terms of scheduling has been produced as a result of this framework's ability to resolve the problem. As a result of the findings of the exhaustive testing, which demonstrated that the framework was successfully implemented, this solution was developed as a consequence of the findings.

Using an environment that is used for clustering, Wu et al., 2020 [28] presented a scheduling and deployment technique that is suited for the management of large-scale tasks. This approach fits the requirements of the environment. Wu and his colleagues were the ones who thought up and developed this approach. Following the implementation of this strategy, a sub-system that corresponds to it was built, and a comprehensive explanation of the way by which the system operates was released. This solution is equipped with a strong method that simultaneously assures the efficiency of the operations

of the data center firm, enhances the utilization of resources, and minimizes the expenditures that are associated with infrastructure charges.

In their study, Starriness and Karatza [29] (2019a) investigated the challenges that are related with the scheduling of parallel work in distributed systems that make use of bag-of-task-chain applications. This activity was carried out as a component of the research that they conducted. Two separate scenarios for resource allocation are used in order to investigate two distinct ways to task-chain scheduling. These scenarios are utilized to evaluate the two different techniques. A broad variety of workload scenarios are simulated in order to assess the effectiveness of task-chain scheduling algorithms. This evaluation is carried out via the use of simulation. It has been concluded, on the basis of the findings of the investigations, that the Probabilistic Task-Chain approach is not as successful as the 2 Random Choices Task-Chain routing strategy on paper. Whatever kind of scheduling was used, this is the effect that will take place regardless of the methodology that was taken. It has been shown via the simulation that CLNTF is better than FCFSC in terms of performance when it comes to the probabilistic task-chain routing scenario. This was demonstrated after the simulation was carried out. On the other hand, when it comes to the circumstance of the two random choices task-chain routing, FCFSC performs much better than CLNTF could.

Ding and his colleagues [30] were responsible for the development of a task scheduling system for the Linux kernel, which was subsequently given the name RTSA. Because of the findings of their research, the use of this tactic was made public in the year 2019. In the context of the Linux development framework, this method takes into account the mapping connection that exists between ROS (Robot Operating System) and the significance that is essential to their relationship. The multi-core load task processing overhead and the communication overhead are two of the numerous characteristics that are taken into consideration by this system. There are many more aspects as well. There are a myriad of other considerations as well. Simulation was used in order to validate the implementation of the RTSA technique. Furthermore, comparative analyses were conducted between the RTSA methodology and the SSC approach, as well as the Round-Robin Scheduling method and the Serial execution method. There is a potential that RTSA will reduce the amount of time that is necessary for task scheduling, improve the overall performance of the system, and reduce the amount of communication overhead that is associated with it. All of these benefits are possible.

In their academic study, Xuan and colleagues [31] (2019) examined a bag-of-tasks (Bot) scheduling issue in a heterogeneous distributed system. This topic was specifically focused on time management. This research was carried out with the purpose of gaining a deeper comprehension of the circumstance. They constructed a mathematical model in order to handle this topic in an effective manner, and in addition to that, they offered two innovative ways. Both of these approaches were presented. An active genetic algorithm is the first technique that is being taken. This algorithm contains a recently discovered local search factor in addition to a modify operator that has been designed after careful research. A new scheduling technique is referred to as the second strategy, which is known by its acronym SAP, which stands for sorting-allocation pulling. SAP is designed to assign work to available time slots on compatible processors and then reallocate those tasks to time segments that were previously accessible in order to reduce the amount of time that is necessary to complete the activity. This is done in order to cut down on the amount of time that is required to complete the activity.

A strategy known as entropy weight clustering scheduling (EWCS) was presented by Huo et al., 2019 [32] on their website. This strategy was developed with the intention of being used for the purpose of selecting nodes utilized on the cloud platform. The dynamic heterogeneous redundancy (DHR) architecture of mimetic defence theory is included into this method, in addition to the K-Means clustering approach that is used in machine learning. This tactic incorporates both of these components into its overall design. The method is composed of four stages: a screening of risk levels, a calculation of entropy weight, load balancing, and clustering optimization. Each stage is performed in order. Each of these processes is essential to the broader plan that is being implemented. It is feasible to make the

conclusion, on the basis of the results of the simulation, that the strategy that is being evaluated is not only adequate, but also capable of accurately replicating the common operating environment (MCOE). This is a conclusion that may be drawn. In addition, it is a software that has shown its ability to successfully facilitate the supply of solutions to scheduling difficulties via the use of machine learning.

The Largest Gang First Served with Interruption (LGFS-I) and the Adapted First Come First Served with Interruption (AFCFS-I) were the two methods of gang scheduling that Starriness and Karatza [33] (2019c) investigated in their research. Both of these methods were presented as separate approaches to gang scheduling. These two methods were referred to as LGFS-I and AFCFS-I, respectively, throughout their respective implementations. Both of these strategies were implemented in a decentralised framework that allows local professions and gangs to work together to share resources. According to the results of the trials, the features of the workload components were a major influence in determining the success of the gang scheduling strategies. This conclusion was reached as a result of the findings of the trials. When compared to the AFCFS-I policy, the LGFS-I algorithm demonstrated much superior performance and performance. On the basis of the variability in the number of tasks assigned to each gang, the frequency of local work arrivals, and the amount of time it took for service changes to take place for both local jobs and gang responsibilities, it was determined that LGFS-I is superior than AFCFS-I. This conclusion was reached after analysing the data. After doing an analysis of the data, this result was arrived at.

Starriness and Karatza [34] (2019b) conducted a research in which they evaluated a range of different methods to structure a bag of chores. The results of this study are presented in the following manner. During the course of their investigation, they looked at a distribution of work bags that had two distinct job routing systems and two distinct scheduling techniques. In addition, they investigated the distribution of work bags. In addition, they investigated two other scheduling approaches that were available. Among the three different scheduling scenarios that were investigated, one of the situations that was investigated was the scheduling of static and dynamic probabilistic and periodic task group schedules. Furthermore, two other scheduling options were studied as well. These scenarios were as follows: (a) a static scheduling policy that used a probabilistic routing model and periodic task group scheduling, and (b) a dynamic routing strategy that featured periodic task group scheduling. Both of these situations were described below. Both of these scenarios are discussed in the following paragraphs. When reaction time and fairness were taken into consideration in each and every random arrival scenario, it was concluded that the 2 Random Choices (FCFS) scheduling scenario was preferable than the Probabilistic-Periodic scenario. This conclusion was reached after comparing the two scenarios. The Two Random Choices (FCFS) scenario was an extra scenario that performed much better than the Probabilistic-Periodic scenario. Before putting a decision into action, it is of the utmost importance to have a comprehensive grasp of the requirements for work service. This is because frequent scheduling is required. This is because there is a need for periodic scheduling.

The authors Susanto et al., 2019 [35] presented a coflow scheduler that is capable of lowering the typical amount of time that is required to finish activities. This scheduler was developed by the writers. One further advantage of using this scheduler is that it gives users some degree of control over the order in which actions are carried out. This is an extra advantage. The most current iteration of their project has as its primary objective the expansion of the number of occupations that are dependent on the intricate structure of activities that include a number of stages. The findings of the testing reveal that our solutions deliver, on average, twice as much performance as the alternatives that are now available. This is in contrast to the alternatives that are already available. On the other hand, in the event that there is an increase in the amount of traffic, our solutions could be able to give performance that is four times greater than what it was before.

During the course of their research on the subject of edge computing, Zhang and Xie [36] (2019) placed a significant amount of attention on the reduction of the amount of time that the task scheduling process takes on cellular devices. The reason for this was to fulfil the conditions that were set out by

their study. In this particular technique, which is an algorithm based on a delay model, Little's law serves as the foundation. Throughout, it serves as the basis for everything. However, the paradigm that is being discussed is not completely free of any flaws or deficiencies. The construction of the cellular edge computing system began with the use of a delay model that was more accurate than the one that was first utilized. Because of this, a Lyapunov function of delay was developed in order to make the recommended scheduling strategy more easily understandable. This was done in order to make the approach more comprehensive. It has been shown that the proposed scheduling methodology has the potential to reduce latency when compared to the method that is currently being utilised for the purpose of scheduling things. It was determined that this was the case when simulations were carried out.

A network distributed flight control system structure has been developed by Yuwei et al. [37] (2019). This structure produces an open system structure by using clustered computer resources and a distributed network. Yuwei and his colleagues were the ones who made this construction. This arrangement was suggested by the authors as a potential solution to the problem. An exhaustive investigation has been carried out in order to acquire a more in-depth comprehension of the problems that are associated with fault-tolerant control, system task modelling, and system job scheduling. In light of the findings of the simulation, it would seem that the method is capable of efficiently generating system task assignments that are in accordance with the limitations and the requirements of the system's operation in particular with regard to the amount of time that is required. In addition to this, it is also capable of managing technical obstacles, such as delegating system responsibilities and reorganising networks in the event that one of the control electronic devices has a malfunction. One of the most important advantages is this capacity. This eventually results in an improvement since it leads to an increase in the fault tolerance and dependability of the system, which is a consequence of this.

Hajoui and his colleagues [38] presented an innovative way to improving urban traffic in their study that was published in 2019. This technique aimed to reduce the amount of time spent commuting as well as the amount of congestion that would be encountered. The notion for this technique originated from the methods that are used in the process of resolving load balancing and work scheduling issues in distributed distributed systems. These methods served as a source of motivation and inspiration. A more particular illustration of the source of inspiration was the employment of optimisation models, such as Ant Colony optimisation (ACO), in order to shorten the amount of time it took to produce the product. When seen through the prism of this technique, the makespan is directly proportional to the amount of time that is spent in situations where there is a considerable degree of traffic congestion. Taking into consideration the findings, it would seem that smart cities need to concentrate a higher emphasis on the management of traffic congestion in order to efficiently manage overcrowding, as well as to limit pollution and energy inefficiency. This objective may be accomplished by actively managing the flow of traffic on major routes and by collecting information about drivers in real time. Both of these methods are conceivable.

Freitas and colleagues [39] highlighted in their research that was published in 2019 that distributed load balancers might possibly benefit from the use of the distributed graph model (DGM) and data structure in order to ease the process of constructing communication-aware heuristics (LBs). This was stated as part of their study. The Distributed General Manager (DGM), which was built for this purpose, has as its major aim the utilisation of task affinity information in order to improve data localization during work scheduling in a distributed manner. On the other hand, after all is said and done, this will ultimately result in a reduction in the expenses that are associated with centralization. When compared to centralised graph exemplification with the same goals, the data show that Distributed Graph Mining (DGM) provides the prospect for speed-ups of up to 50.4% when employing 40 cores that are practicable. This is the case when the objectives are the same. A considerable difference may be seen between those who use centralised graph exemplification and those who do not. Additionally, they provide a proof-of-concept distributed scheduler that they call Edge Migration. This is in addition to the fact that they supply this distributed scheduler. This particular scheduler makes use of DGM, which is

an abbreviation that implies distributed global memory. In addition to this, they provide an explanation of how they were able to successfully develop this scheduler by using the Charm++ parallel programming language. As a result of these observations, it is reasonable to arrive at the conclusion that the principal overhead in distributed load balancers is still rather substantial. Despite the fact that the communication analysis is much more efficient when utilising distributed graph management (DGM), this is the situation that has arisen regardless.

In the paper that they created in 2019, Ran et al. introduced a method that they called DeepEE. [40] This mechanism was described in the publication. This framework makes use of deep reinforcement learning, which is more often referred to as DRL, in order to achieve the best possible degree of energy efficiency in data centres. During the process of putting this method into action, both the information technology and the cooling systems are taken into account concurrently. Parameterized Action Space-based Deep Q-Network (PADQN) is the name that the creators of DeepEE have given to the approach that they have developed. In an effort to find a solution to the problem that is posed by hybrid action space, this approach was developed. The use of this approach not only allows for the modification of the airflow rate of the cooling system to take place concurrently, but it also helps to enhance the work schedule of the information technology system. In addition, in order to improve the precision and effectiveness of the cooling systems as well as the information technology systems, (PADQN) takes into account a control mechanism that functions on two separate time scales. This is done in order to get more accurate and efficient results. By comparing their approach to the conventional methods of solitary and collaborative optimisation, they demonstrate that their technique has the potential to cut energy usage by as much as fifteen percent and ten percent, respectively. This may be achieved via the use of sophisticated simulations that are constructed on the basis of real data. Additionally, their method makes use of a parameterized work area, which leads to a more constant improvement in the amount of force that is used. This is a significant advantage. Furthermore, it strikes a more suitable balance between the quality of service and the saving of energy, which is a significant benefit.

Authors of [41] offered a strategic strategy to solve the key obstacles that are connected with the Hadoop task allocation approach across a variety of computing environments, including popular clouds, in their work that was published in 2019. Deshai and his colleagues presented this plan in their work that was published in 2019. This method was provided in advance of the release of their work, which was about to take place. To calculate and assess the durations of completion for each job slot and the succeeding data block, this strategy makes use of the optimum least makespan technique. This technique is used to compute and evaluate the durations of completion. This action is taken in order to guarantee that the plan will be carried out in an efficient manner. The amount of time that is spent on carrying out the actions that are linked with the MR phase will be reduced to the fullest degree that is practicable with the assistance of this strategy. A complete simulation was carried out by us in order to evaluate the overall efficacy of the proposed scheme in comparison to the Hadoop scheme in two distinct computing scenarios that are often seen in cloud policy. Here is a description of these potential outcomes. It would seem that the method that has been shown has the capacity to effectively reduce the amount of time that is anticipated to be necessary for the completion of the map phase, while at the same time lowering the extent to which it is dependent on remote processing. Taking into consideration the results of the simulation that is now being carried out, this conclusion may be drawn. As a result of this, the influence that network and disc congestion has on the processing of global data is reduced as a consequence of this. This is because of the fact that this.

Cheng et al. proposed a complete cloud workload management system that they referred to as sCloud in their article that was published in 2019 [42]. In addition to taking into consideration heterogeneity, the objective of this strategy is to improve the performance of systems that are located in datacenters that are both self-sufficient and dispersed. Nonlinear programming is a technique that can be used to address the difficulty of imposing transactional workloads. This technique may be utilised to solve a constrained optimisation problem. Using nonlinear programming, it is possible to find a solution

to this issue. For the purpose of describing this occurrence, the word "constrained" has been used. A batch job migration approach is offered as a means of further enhancing system throughput in situations when the supply of green power shifts from one location to another. This is done in order to accommodate specific scenarios. The throughput of the system may be further improved by using this strategy, which is shown below. A flexible batch task manager is one of the features that they have included into sCloud as part of their attempts to improve it. It is the responsibility of this manager to ensure that deadlines are not missed while also facilitating the efficient administration of the flow of work. Based on the results of the tests, it has been determined that sCloud is capable of achieving system performance that is capable of being highly comparable to that of an ideal system, while also being able to adjust to variations in the amount of power that is available.

It is a well-known truth. Zhao et al. (2018) [43] did a study in which they studied whether or not it would be able to automate the stage of the design process that includes assigning periods and priorities to distributed hard real-time systems. Specifically, they were interested in determining whether or not this step could be automated. There are a great number of different applications for embedded systems that do have the potential to have problems of this kind. A few of examples of these applications are the automobile and aviation industries. We plan to implement a strategy that incorporates the development of a one-of-a-kind optimisation approach that makes use of the capabilities of the ILP solver, in addition to strategies that are relevant to the situation that is currently being dealt with. For the purpose of tackling a particular subproblem for the sole purpose of optimising period assignment, the method that has been suggested is not only much more effective than the cutting-edge technology that is now being used, but it also creates solutions that are significantly more efficient.

In their 2018 publication [44], the authors Li et al. offer a hybrid Pareto-based tabu search algorithm (HPTSA) with the intention of concurrently minimising four separate objectives. A few examples of these objectives include the maximum workload, the total workload, the makespan, and the earliness/tardiness (E/T) criterion. Additional examples include the makespan. This approach, which was developed in order to discover a solution to an actual issue, has a main focus on the subject of multiobjective distributed flexible job shop scheduling as its core topic of discussion. It has been claimed that the technique that has been given makes use of a range of different ways in order to commence the group of answers that have been supplied. The attributes of the issue as well as the characteristics of the target are taken into consideration by these strategies. In order to get the greatest possible outcomes with respect to the E/T criterion, it is highly advised that a reversal strategy that has been painstakingly created be used. This is because the aim is to achieve the best possible results. In order to generate a number of scenarios with varying degrees of issue severity, authentic production data from the steelmaking system is used so that the scenarios may be constructed. Every single one of these instances is produced in a totally random manner. Through the use of the HPTSA methodology that was developed, a selection of four effective multiobjective optimisation strategies was made with the intention of carrying out thorough comparisons. It is clear that the approach that was offered demonstrates superior performance when compared to the successful techniques that were chosen. This is the case when the facts are taken into consideration.

Within this article that was published in 2018, Qin et al. [45] investigate the use of static work scheduling approaches in distributed systems. 2018 was the year that the article was published. They provide the HCPTD methodology, which is a scheduling method that makes use of a variety of distinct scheduling principles, such as task replication scheduling and table scheduling, among others. With the help of this tactic, it is possible to determine with more precision the activities that are of the biggest significance. This approach may also be used to construct a scheduling sequence that is more effective, which is another potential application. In the course of the process of selecting resources, consideration was given to both the earliest potential completion time of the work as well as the shortest distance to the exit. This action was taken in order to guarantee that there would be sufficient resources accessible. Following that, the work was assigned to the resource that had the fewest resources available, and the

choice to assign the work was made based on the Processor Selection Priority (PSP). Based on the findings of the experiments, it has been shown that the (HCPTD) algorithm has extraordinary performance. These results have been proved to be true.

In 2018, Amin and Obermaisser [46] created a time-triggered static scheduler for a heterogeneous distributed system. This scheduler was designed to coordinate the execution of tasks. The specific scheduler in question makes use of a real-time database in combination with diagnostic queries in order to discover any possible issues that may become evident in the future. This approach is the one that is responsible for determining the timing of diagnostic queries, as well as the replication and deletion of data in the real-time database. It is also the one that is being given. There is a reduction in the length of time that is necessary to become aware of issues, and a diagnostic framework that may be used is constructed via the utilisation of this historical knowledge. It is possible to profit from each of these advantages concurrently. A preference-based technique is used by the project in order to accomplish the goals of merging the various systems and controlling diagnostic activities. When using this strategy, limitations on periodicity and prioritisation are taken into consideration. Both the use of empirical approximation and the implementation of prototypes are considered to be the fundamental components of this methodology. In the context of a diagnostic environment that includes a substantial number of questions, the objective of this presentation is to demonstrate how time-triggered scheduling may be used in an effective manner. Using this technique leads to significant improvements in the accuracy, integrity, and performance of distributed systems that display a broad variety of characteristics. These improvements are achieved by the use of this strategy.

Oikonomou et al., 2018 [47] were successful in resolving the problem of scheduling video transcoding operations by using a cloud-based system that is spread across a number of data centres. The cloud was used in order to attain this goal. The techniques that were provided were examined and evaluated utilising simulation testing, which was carried out on two real-world scenarios. The simulation tests were carried out in order to evaluate the methodologies. While the first configuration was designed for real-time streaming, the second configuration was meant for batch file transcoding on social networks. Both configurations were constructed from scratch. Each of the configurations was constructed. It has been discovered that the heuristic PEN, which makes use of a Pareto-optimal technique to fulfil the optimisation objectives of network efficiency, processing time, and energy, is the one that results in the greatest number of outcomes that are advantageous. In light of the findings, this conclusion may be drawn.

When it comes to scheduling jobs that make use of bag-of-tasks execution models on a cluster of distributed processing models, the authors of the paper that was published by Stavrinides and Karatza (2018) [48] investigate the challenges that are involved with this procedure. Their attention is specifically directed into the constraints that are connected to the use of this scheduling strategy. The evaluation of the effectiveness of scheduling algorithms in a variety of task routing situations begins with the use of a scheduling algorithm as the first step in the process. The design of a scheduling algorithm is the first step in this assessment process. In order to achieve the aim of anticipating how algorithms will be executed in circumstances where the demand placed on the system is variable, simulation is often used. Utilising simulation is the means by which this is done. The scheduling technique, the routing strategy, and the load that was imposed on the system each had varying degrees of effect on the performance of the system. These factors all contributed to the overall performance of the system.

4. Comparison And Discussion

Researchers have offered a description of the techniques and approaches that are used for work scheduling in distributed systems in prior studies that they have published. The purpose of this research is to come up with a complete analysis of the many different strategies and methods that are available.

Table 1, which can be seen below, contains a detailed comparison of the bulk of the research that has been investigated in this study. This comparison is shown in the table that can be viewed below. Throughout the history of distributed systems, there have been several issues that have arisen, and these issues have been resolved in a variety of different ways, from a varied range of views. One algorithm that serves as an excellent example of this is the Hybrid Fault-Tolerant Scheduling Algorithm (HFTSA), which is discussed in reference [21]. When it comes to cloud-based systems, the objective of this algorithm is to enhance the utilisation of resources and guarantee that activities are completed in a timely way. In addition, the Ant Colony Optimisation (ACO) approach was presented in [35] as a means of enhancing the conditions of urban traffic by reducing the amount of time spent as a consequence of congestion and the amount of time it takes to travel. This was done in order to promote the improvement of urban traffic conditions. After conducting a comparison between the ACO algorithms and the Road Traffic Algorithm, it was found that the findings provided by the comparison were satisfactory. There was a comparison in place. When the results of this study are taken into account, it seems that it is possible to offer drivers with information in real time on the routes that they are now driving. For the goal of scheduling gang jobs in a cluster of distributed processors, the author of the paper provided two methods, namely AFCFS-I and LGFS-I. These algorithms were designed to accomplish this objective. It is recommended that these respective methods be used in combination with one another. It was decided, on the basis of the results of the investigation, that the LGFS-I strategy performed much better than the AFCFS-I policy. The EWCS algorithm was presented by the author as a means for assessing the effectiveness of the system while they were doing their inquiry. The data have led to the conclusion that the EWCS method is capable of providing optimum heterogeneity while simultaneously ensuring availability. This conclusion was reached at the conclusion of the investigation. As an additional point of interest, a condensed synopsis of each of the publications that are related to task scheduling in distributed systems is presented in table1, which can be found further down on this page. You may find this summary in the table that is provided.

Table 1: Summary of reviewed papers related to the task scheduling in the distributed systems

Ref.	Methods /Algorithm	Tools	Objective	Achieved
[22]	trading algorithm Scheduling Algorithm	MATLAB	Propose a systematic structure for coordinating the scheduling and power trading activities of a group of prosumers inside a transactive energy market (TEM).	- The suggested framework shown computational efficiency and may be readily applied to communities of varying sizes. - financially advantageous - the grid operator may use the flexibility of prosumers to handle grid limitations
[23]	The technique used for scheduling tasks with a high level of	MATLAB	Introduce a G/G/1 queuing mechanism to evaluate the efficiency of	Experiments have shown that FSTS may reduce energy consumption and achieve the greatest

	precision in terms of geographical distribution is called the Fine-grained geographical Task Scheduling (FSTS) algorithm..		servers in DGCs. In order to reduce the energy expenses of a distributed generation and consumption (DGC) supplier..	data transfer rate when compared to several modern scheduling approaches.
[24]	The Hybrid Fault Tolerant Scheduling Algorithm (HFTSA)	MATLAB	The issue at hand is the scheduling of independent tasks with deadline constraints in cloud systems. To address this, a hybrid approach is proposed that combines resubmission and replication operators. This approach aims to improve resource utilisation and ensure that task implementation timeframes are met.	High-frequency trading system architecture (HFTSA) may improve the efficiency of resource utilisation and ensure timely completion of tasks while also providing fault tolerance.
[24]	based on heterogeneous distributed deep learning used adaptive scheduling algorithm	Python and Tensorflow	The challenge in cloud work is in effectively managing the income of the service provider while maintaining a high level of customer service. Optimal task scheduling and allocation of resources in a flow system	The proposed approach effectively addresses the multi-objective optimisation challenges of cloud workflow task scheduling and resource allocation, offering a nearly optimum scheduling technique.
[25]	(The PADQN (Parameterized Action Space	MATLAB	Create an innovative and very effective	The proposed PADQN has the potential to achieve

	based Deep Q- Network) method		framework, DeepEE, based on Deep Reinforcement Learning (DRL), to optimise job scheduling and cooling in energy-efficient data centres. controlling (adjusting air- flow rate).	energy savings of up to 15% and 10% when compared to the baseline optimisation methods in isolated and join scenarios, respectively. By using a parameterized action space, the technique achieves more stability in power use.
[26]	coflow scheduler of multi-stage jobs & (LBEF) Least Blocking Effect First algorithm	Simulation by programing language	A scheduling technique is proposed to coordinate coflows of a multi-stage task, taking into account the multi- dimensional properties of the job. The aim is to obtain a reduced average task Completion Time (JCT).	The experimental results demonstrate that the technique offers, on average, double the performance of existing systems, and quadruple the performance in scenarios with bursty traffic.
[27]	sCloud control, placement, Job Migration algorithm	MATLAB	The objective is to optimise the overall data transfer rate in scattered, self- sustaining data centres.	Experimental results demonstrate that sCloud is capable of achieving system performance that is close to ideal, even in the presence of fluctuating power availability.
[28]	RTSA (ROS Task Scheduling Algorithm)	Oracle VM VirtualBox&simulation	When evaluating the development framework of ROS and Linux, it is important to take into account many elements such the distribution of workload over different cores,	Optimise job scheduling duration, minimise communication overhead, and enhance system performance.

			the computational cost of tasks, and the overhead of communication.	
[29]	sorting-allocation-pulling (SAP), genetic algorithm	MATLAB	Examine a scheduling issue involving a bag-of-tasks (BoT) in a heterogeneous distributed system, taking into account offline time.	The proposed algorithms (SAP, genetic algorithm) with other two algorithms, which are CBS3M_EDF_ROFF and H2ACO was compared, where the proposed algorithms were more effectiveness.
[30]	task-chain scheduling algorithms	Simulation by programming language	Examine the complexities associated with scheduling parallel tasks that consist of bag-of-task chains in distributed systems.	The effectiveness of the task chain scheduling method relies on the use of resource allocation strategies. In the instance of routing in the 2 Random Choices Task-Chain, FCFSC demonstrated superior performance.
[31]	EWCS Algorithm	MATLAB	Evaluate the efficiency of the system and prioritise the business need to replicate a typical operating environment (MCOE).	The EWCS algorithm may maximise heterogeneity while ensuring availability.
[32]	Decentralised flight control system	Python-MATLAB	Description of Method, task modeling, and simulation of an algorithm for system tasks scheduling in a systems computing network.	- facilitate the implementation of fault-tolerant scheduling for the network distributed flight control system architecture -Address the issue of implementing fault-tolerant control in the system after the malfunction of the

				control electronic equipment.
[33]	(ACO) algorithm	MATLAB	Enhance urban transportation infrastructure to effectively alleviate congestion and minimise journey duration.	The ACO algorithms were compared with the Road Traffic Algorithm, and the results produced were good. They indicate the effectiveness of informing drivers in real-time about the related routes to combat congestion, energy waste, and therefore reduce pollution.
[34]	novel task scheduling algorithm Lyapunov Function	Simulation by programming language	Optimising task scheduling to minimise delays in cellular edge computing systems.	The scheduling system may achieve a maximum decrease in delay of 55% for customers with low traffic volume.
[35]	GangScheduling Algorithms (AFCFS-I)&(LGFS-I)	C Programming	Explore methodologies for organising gang tasks in a network of dispersed processors, where local tasks are prioritised for execution and gangs are not subject to negotiation.	The LGFS-I algorithm outperformed the AFCFS-I policy. The superiority of LGFS-I over AFCFS-I was assessed based on the variability in the number of tasks per gang, the frequency of local job arrivals, and the variability in service time for both local jobs and gang responsibilities.
[36]	Distributed Graph Model (DGM).	MATLAB & Charm++	In order to include a communications-aware heuristic into distributed load balancers (LBs), we propose the use of a Distributed Graph Model (DGM) and	The Distributed Graph Mining (DGM) may produce a significant increase in performance, up to 50.4 times faster, when using 40 virtual cores compared to centralised graph representations that

			accompanying data structure.	serve the same function.
[37]	2 Random Choices algorithm, (FCFS), (PSGF-x)	Simulation by programing language	The performance of scheduling techniques in different task routing algorithm scenarios, including fairness in the implementation order of tasks.	The Random Choices First-Come-First-Serve (FCFS) example outperforms the Probabilistic Periodic instance in terms of both BoT response time and fairness.
[38]	-task group scheduling algorithms - BoT scheduling tasks in a distributed systems.	Simulation by programing language	Evaluate the system's performance by measuring the task response time and the scheduling overhead by analysing the number of queue arrangements.	The system performance is influenced by the scheduling policies, which rely on the routing strategy and the system load.
[39]	HCPTD algorithm	MATLAB	To enhance the efficiency of activities with interdependencies in a distributed environment and address the limitations of current table scheduling techniques.	HCPTD significantly enhances the scheduling efficiency of distributed systems.
[40]	global scheduling algorithms PEN algorithm	Simulation via Amazon EC2 cloud computing	Examine the issue of organising video transcoding tasks over a Cloud service that covers numerous datacenters.	The heuristic known as PEN (Processing, Energy, and Network) addresses all three optimisation objectives, namely processing time, energy consumption, and network efficiency, by using a Pareto-optimal method, which offers the most advantages.

5. Conclusion

For the purpose of determining whether or not the algorithms that are used for task scheduling in

distributed systems are efficient, it is a challenging endeavour that necessitates the utilisation of suitable evaluation approaches via the utilisation of appropriate evaluation strategies. In order to accomplish the aim of this work, which is to provide a complete study of a broad range of newly created scheduling algorithms for distributed systems, the purpose of this paper is to satisfy that target. A further description of the key job allocation criteria that are used for each strategy is provided in addition to this. In addition to these techniques, there are a number of research studies that specifically target and aim to improve, divide the workload in an equitable manner, optimise, and so on. These studies are in addition to the tactics that have been discussed. The primary objective of these investigations is to find ways to improve the situation. In addition, the difficulty of using these recommended algorithms in the specific distributed systems environment that is necessary for practical evaluation is still an issue that has not been handled in the field of scheduling. This is a problem that has not been resolved. At this point, there has been no attempt made to solve this issue. In the present moment, there is no solution that has been discovered for this problem.

References

- [1] M. M. Sadeeq, N. M. Abdulkareem, S. R. Zeebaree, D. M. Ahmed, A. S. Sami, and R. R. Zebari, "IoT and Cloud computing issues, challenges and opportunities: A review," *Qubahan Academic Journal*, vol. 1, no. 2, pp. 1-7, 2021.
- [2] S. R. Zeebaree, H. M. Shukur, L. M. Haji, R. R. Zebari, K. Jacksi, and S. M. Abas, "Characteristics and analysis of hadoop distributed systems," *Technology Reports of Kansai University*, vol. 62, no. 4, pp. 1555-1564, 2020.
- [3] P. Y. Abdullah, S. Zeebaree, K. Jacksi, and R. R. Zeabri, "An hrn system for small and medium enterprises (sme) s based on cloud computing technology," *International Journal of Research-GRANTHAALAYAH*, vol. 8, no. 8, pp. 56-64, 2020.
- [4] J. Saeed and S. Zeebaree, "Skin lesion classification based on deep convolutional neural networks architectures," *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, pp. 41-51, 2021.
- [5] P. Y. Abdullah, S. Zeebaree, H. M. Shukur, and K. Jacksi, "HRM system using cloud computing for Small and Medium Enterprises (SMEs)," *Technology Reports of Kansai University*, vol. 62, no. 04, p. 04, 2020.
- [6] N. O. Salim, S. R. Zeebaree, M. A. Sadeeq, A. Radie, H. M. Shukur, and Z. N. Rashid, "Study for food recognition system using deep learning," in *Journal of Physics: Conference Series*, 2021, vol. 1963, no. 1: IOP Publishing, p. 012014.
- [7] K. Jacksi, R. K. Ibrahim, S. R. Zeebaree, R. R. Zebari, and M. A. Sadeeq, "Clustering documents based on semantic similarity using HAC and K-mean algorithms," in *2020 International Conference on Advanced Science and Engineering (ICOASE)*, 2020: IEEE, pp. 205-210.
- [8] H. S. Malallah, R. Qashi, L. M. Abdulrahman, M. A. Omer, and A. A. Yazdeen, "Performance Analysis of Enterprise Cloud Computing: A Review," *Journal of Applied Science and Technology Trends*, vol. 4, no. 01, pp. 01-12, 2023.
- [9] A. A. Yazdeen, R. Qashi, H. S. Malallah, L. M. Abdulrahman, and M. A. Omer, "Internet of Things Impact on Web Technology and Enterprise Systems," *Journal of Applied Science and Technology Trends*, vol. 4, no. 01, pp. 19-33, 2023.
- [10] L. M. Abdulrahman, S. H. Ahmed, Z. N. Rashid, Y. S. Jghef, T. M. Ghazi, and U. H. Jader, "Web Phishing Detection Using Web Crawling, Cloud Infrastructure and Deep Learning Framework," *Journal of Applied Science and Technology Trends*, vol. 4, no. 01, pp. 54-71, 2023.
- [11] R. R. Zebari, S. R. Zeebaree, Z. N. Rashid, H. M. Shukur, A. Alkhayyat, and M. A. Sadeeq, "A Review on Automation Artificial Neural Networks based on Evolutionary Algorithms," in *2021 14th International Conference on Developments in eSystems Engineering (DeSE)*, 2021: IEEE, pp. 235-240.
- [12] M. A. Omer, A. A. Yazdeen, H. S. Malallah, and L. M. Abdulrahman, "A Survey on Cloud Security: Concepts, Types, Limitations, and Challenges," *Journal of Applied Science and Technology Trends*, vol. 3, no. 02, pp. 47-57, 2022.
- [13] S. Mohsin, B. W. Salim, A. K. Mohamedsaeed, B. F. Ibrahim, and S. R. Zeebaree, "American Sign Language Recognition Based on Transfer Learning Algorithms," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 5s, pp. 390-399, 2024.

- [14] S. A. Mostafa et al., "Applying Trajectory Tracking and Positioning Techniques for Real-time Autonomous Flight Performance Assessment of UAV Systems," *Journal of Southwest Jiaotong University*, vol. 54, no. 3, 2019.
- [15] Z. S. Hammed, S. Y. Ameen, and S. R. Zeebaree, "Investigation of 5G wireless communication with dust and sand storms," *Journal of Communications*, vol. 18, no. 1, 2023.
- [16] H. M. Zangana and S. R. Zeebaree, "Distributed Systems for Artificial Intelligence in Cloud Computing: A Review of AI-Powered Applications and Services," *International Journal of Informatics, Information System and Computer Engineering (INJIISCOM)*, vol. 5, no. 1, pp. 1-20, 2024.
- [17] B. W. Salim and S. R. Zeebaree, "Kurdish Sign Language Recognition Based on Transfer Learning," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, no. 6s, pp. 232-245, 2023.
- [18] L. M. Abdulrahman, S. R. Zeebaree, and N. Omar, "State of Art Survey for Designing and Implementing Regional Tourism Web based Systems," *Academic Journal of Nawroz University*, vol. 11, no. 3, pp. 100-112, 2022.
- [19] R. K. Ibrahim et al., "Clustering Document based on Semantic Similarity Using Graph Base Spectral Algorithm," in *2022 5th International Conference on Engineering Technology and its Applications (IICETA)*, 2022: IEEE, pp. 254-259.
- [20] N. M. ABDULKAREEM and S. R. ZEEBAREE, "OPTIMIZATION OF LOAD BALANCING ALGORITHMS TO DEAL WITH DDOS ATTACKS USING WHALE OPTIMIZATION ALGORITHM," *Journal of Duhok University*, vol. 25, no. 2, pp. 65-85, 2022.
- [21] R. K. Ibrahim, S. R. Zeebaree, K. Jacksi, M. A. Sadeeq, H. M. Shukur, and A. Alkhayyat, "Clustering document based semantic similarity system using TFIDF and k-mean," in *2021 International Conference on Advanced Computer Applications (ACA)*, 2021: IEEE, pp. 28-33.
- [22] L. Zhao, Y. Yang, A. Munir, A. X. Liu, Y. Li, and W. Qu, "Optimizing Geo-Distributed Data Analytics with Coordinated Task Scheduling and Routing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 2, pp. 279–293, Feb. 2020, doi: 10.1109/TPDS.2019.2938164.
- [23] M. Khorasany, A. Najafi-Ghalelou, and R. Razzaghi, "A Framework for Joint Scheduling and Power Trading of Prosumers in Transactive Markets," *IEEE Trans. Sustain. Energy*, vol. 12, no. 2, pp. 955–965, Apr. 2021, doi: 10.1109/TSTE.2020.3026611.
- [24] G. Yao, Q. Ren, X. Li, S. Zhao, and R. Ruiz, "A hybrid fault-tolerant scheduling for deadline-constrained tasks in Cloud systems," *IEEE Trans. Serv. Comput.*, pp. 1–1, 2020, doi: 10.1109/TSC.2020.2992928.
- [25] H. R. Faragardi, M. R. Saleh Sedghpour, S. Fazliahmadi, T. Fahringer, and N. Rasouli, "GRP-HEFT: A Budget-Constrained Resource Provisioning Scheme for Workflow Scheduling in IaaS Clouds," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 6, pp. 1239–1254, Jun. 2020, doi: 10.1109/TPDS.2019.2961098.
- [26] H. Yuan, M. Zhou, Q. Liu, and A. Abusorrah, "Fine-grained and arbitrary task scheduling for heterogeneous applications in distributed green clouds," *IEEECAA J. Autom. Sin.*, pp. 1–13, 2020, doi: 10.1109/JAS.2020.1003177.
- [27] D. Cui et al., "Cloud Workflow Task and Virtualized Resource Collaborative Adaptive Scheduling Algorithm Based on Distributed Deep Learning," p. 4.
- [28] C. Wu, E. Haihong, and M. Song, "A Distributed Task Scheduling System Suitable for Massive Environments," in *2020 International Conference on Inventive Computation Technologies (ICICT)*, Coimbatore, India, Feb. 2020, pp. 750–755. doi: 10.1109/ICICT48043.2020.9112531.
- [29] G. L. Stavriniades and H. D. Karatza, "Scheduling Bag-of-Task-Chains in Distributed Systems," in *2019 IEEE 14th International Symposium on Autonomous Decentralized System (ISADS)*, Utrecht, Netherlands, Apr. 2019, pp. 1–6. doi: 10.1109/ISADS45777.2019.9156029.
- [30] N. Ding, X. Meng, Y. Zhao, and D. Wu, "ROS Task Scheduling Algorithm in Multi-Core System," in *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*, Tianjin, China, Dec. 2019, pp. 516–519. doi: 10.1109/ICPADS47876.2019.00078.
- [31] H. Xuan, S. Wei, Y. Li, and H. Guo, "Off-Line Time Aware Scheduling of Bag-of-Tasks on Heterogeneous Distributed System," *IEEE Access*, vol. 7, pp. 104777–104788, 2019, doi: 10.1109/ACCESS.2019.2899926.
- [32] L. Huo, P. Shao, F. Ying, and L. Luo, "The Research on Task Scheduling Algorithm for the Cloud Management Platform of Mimic Common Operating Environment," in *2019 18th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, Wuhan, China, Nov. 2019, pp. 167–171. doi: 10.1109/DCABES48411.2019.00049.
- [33] G. L. Stavriniades and H. D. Karatza, "Scheduling Different Types of Gang Jobs in Distributed Systems," in *2019 International Conference on Computer, Information and Telecommunication Systems (CITS)*, Beijing, China, Aug. 2019, pp. 1–5. doi: 10.1109/CITS.2019.8862091.

- [34] G. L. Stavrinides and H. D. Karatza, "Scheduling Different Types of Bag-of-Tasks Jobs in Distributed Systems," in 2019 10th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, Jun. 2019, pp. 13–18. doi: 10.1109/IACS.2019.8809138.
- [35] H. Susanto, A. M. Abdelmoniem, H. Zhang, B. Liu, and D. Towsley, "A Near Optimal Multi-Faced Job Scheduler for Datacenter Workloads," in 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, Jul. 2019, pp. 2026–2036. doi: 10.1109/ICDCS.2019.00200.
- [36] Y. Zhang and M. Xie, "A More Accurate Delay Model based Task Scheduling in Cellular Edge Computing Systems," in 2019 IEEE 5th International Conference on Computer and Communications (ICCC), Chengdu, China, Dec. 2019, pp. 72–76. doi: 10.1109/ICCC47050.2019.9064217.
- [37] C. Yuwei, Z. Zhengyong, L. Zhengren, and L. Aijun, "A task scheduling method for network distributed flight control system," in 2019 Chinese Automation Congress (CAC), Hangzhou, China, Nov. 2019, pp. 480–483. doi: 10.1109/CAC48633.2019.8996326.
- [38] Y. Hajoui, O. Bouattane, M. Youssfi, and E. Illoussamen, "Application model of tasks scheduling problem in distributed systems solving problems of road traffic planning in smart cities," in 2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS), Marrakech, Morocco, Oct. 2019, pp. 1–7. doi: 10.1109/ICDS47004.2019.8942310.
- [39] V. Freitas, A. de L. Santana, M. Castro, and L. L. Pilla, "Distributed Memory Graph Representation for Load Balancing Data: Accelerating Data Structure Generation for Decentralized Scheduling," in 2019 International Conference on High Performance Computing & Simulation (HPCS), Dublin, Ireland, Jul. 2019, pp. 787–794. doi: 10.1109/HPCS48598.2019.9188134.
- [40] Y. Ran, H. Hu, X. Zhou, and Y. Wen, "DeepEE: Joint Optimization of Job Scheduling and Cooling Control for Data Center Energy Efficiency Using Deep Reinforcement Learning," in 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, Jul. 2019, pp. 645–655. doi: 10.1109/ICDCS.2019.00070.
- [41] N. Deshai, R. S. Shankar, K. Sravani, and D. Ravibabu, "A Developed Task Allotments Policy for Apache Hadoop Executing in the Public Clouds," in 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), Pondicherry, India, Mar. 2019, pp. 1–4. doi: 10.1109/ICSCAN.2019.8878857.
- [42] D. Cheng, X. Zhou, Z. Ding, Y. Wang, and M. Ji, "Heterogeneity Aware Workload Management in Distributed Sustainable Datacenters," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 2, pp. 375–387, Feb. 2019, doi: 10.1109/TPDS.2018.2865927.
- [43] Y. Zhao, V. Gala, and H. Zeng, "A Unified Framework for Period and Priority Optimization in Distributed Hard Real-Time Systems," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 37, no. 11, pp. 2188–2199, Nov. 2018, doi: 10.1109/TCAD.2018.2857380.
- [44] J.-Q. Li, P. Duan, J. Cao, X.-P. Lin, and Y.-Y. Han, "A Hybrid Pareto-Based Tabu Search for the Distributed Flexible Job Shop Scheduling Problem With E/T Criteria," *IEEE Access*, vol. 6, pp. 58883–58897, 2018, doi: 10.1109/ACCESS.2018.2873401.
- [45] L. Qin, F. Ouyang, and G. Xiong, "Dependent task scheduling algorithm in distributed system," in 2018 4th International Conference on Computer and Technology Applications (ICCTA), Istanbul, May 2018, pp. 91–95. doi: 10.1109/CATA.2018.8398662.
- [46] S. Amin and R. Obermaisser, "A Time-Triggered Scheduling Algorithm for Active Diagnosis in Heterogeneous Distributed Systems," in 2018 IEEE International Conference on Computational Science and Engineering (CSE), Bucharest, Oct. 2018, pp. 44–55. doi: 10.1109/CSE.2018.00013.
- [47] P. Oikonomou et al., "Scheduling Video Transcoding Jobs in the Cloud," in 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada, Jul. 2018, pp. 442–449. doi: 10.1109/Cybermatics_2018.2018.00099.
- [48] G. L. Stavrinides and H. D. Karatza, "Task Group Scheduling in Distributed Systems," in 2018 International Conference on Computer, Information and Telecommunication Systems (CITS), Alsace, Colmar, France, Jul. 2018, pp. 1–5. doi: 10.1109/CITS.2018.8440181.