Clustered Task Scheduling with Fuzzy C Clustering for Load Balancing in Cloud Data centers

Syed.Karimunnisa[1],

Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation Vaddesvaram, Guntur, AP, India-522302, karimun1.syed@gmail.com

Supriya Menon M[2]

Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation Vaddesvaram, Guntur, AP, India-522302,supriyamenon05@gmail.com

Abstract:

The varied mix of resources in cloud data centres presents substantial issues for scheduling in distributed computing environments, especially in the context of cloud computing. The dynamic fluctuations in the number of users and their demands add to the complexity. In order to meet these issues, a scheduling technique that can adjust to the inherent heterogeneity in resource distribution must be developed, with the ultimate goal of minimising request completion times. This paper presents a new scheduling method that groups tasks according to how similar their lengths are by using the Fuzzy C clustering algorithm. This method, which is well-known for its effectiveness and simplicity, makes task grouping more productive. Tasks in these clusters are then intelligently scheduled onto appropriate virtual machines (VMs) according to their capacity. Experimental research results confirm that the suggested scheduling strategy improves data centre performance by a significant amount by lowering makespan time. This study adds to the continuing endeavours to create efficient scheduling plans in the ever-changing cloud computing environment.

Keywords: Virtual machine, Fuzzy clustering, Scheduling, SLA.

1. Introduction

Cloud computing stands as a contemporary computer technology leveraging virtualized infrastructure to deliver secure and dependable services to end-users within a complex environment. Given its capacity to offer crucial information technology (IT) services, notably computing resources through virtual machines (VMs), cloud computing has garnered significant attention as a prevailing computing model [1], [2]. It has gained widespread popularity in the realm of leasing or renting cloud resources. The pay-as-you-go model of the cloud is a compelling draw for organizations and individual users looking to deploy their applications in the cloud. Applications exhibit diversity in size, resource requirements (compute/memory/storage), and execution time (long/short). At the cloud end, these applications are segmented into distinct executables known as jobs, tasks, and instances.

Cloud computing encompasses a broad spectrum of applications characterized by diverse workloads, resulting in varied demands for each application. In the cloud, workloads represent sets of inputs forwarded to the infrastructure cloud for processing, and the efficient handling of these workloads serves as a metric for performance[3]. Cloud computing offers shared computing and storage resources, where applications are subdivided into jobs or tasks and distributed across the cloud for parallel processing to achieve scalability.

Cloud service providers (CSP) must, based on the attributes of the task, be adept at managing diverse client requests by creating virtual machines (VM) in the cloud. The primary objective of the CSP is to minimize costs, prevent Service Level Agreement (SLA) violations, maximize resource utilization, and ultimately uphold a high level of Quality of Service (QoS)[4].

Tasks are differentiated by attributes such as start and end times, task ID, CPU rate, memory usage, cache memory usage, disk I/O time, priorities, bandwidth, and more. This information aids in describing the unique characteristics of each task. Tasks with different workloads seek resources from the cloud to complete their execution. A wide range of tasks is deployed in cloud environments, with some exhibiting regular periodicity, while others are unpredictable .

Scheduling, as outlined in references [5,6,7] involves the allocation of resources to tasks submitted by cloud clients within a designated timeframe. The primary goals of scheduling are to minimize makespan and response time, maximize resource utilization, and maintain a balanced load across all machines. A proficient scheduling algorithm contributes to favorable system performance. Cloud datacenters, as discussed in references [8,9,10] encompass a multitude of heterogeneous resources. The cost and completion time of a task in the cloud are contingent upon the nature of the resource to which it has been assigned.

Various scheduling policies have been implemented to address task failures and strive to minimize execution time, energy consumption, cost, SLA violation rates, and more . Numerous researchers have proposed multi-objective task scheduling using nature-inspired algorithms, and some have devised novel approaches for task scheduling and optimal resource allocation [11]. Another effective method of task scheduling involves categorizing tasks based on historical data and creating different types of virtual machines (VMs) to meet task demands.

A significant challenge in scheduling and allocation lies in assigning tasks to VMs. Allocating larger tasks to VMs with lower processing capabilities can result in extended processing times, potentially surpassing task deadlines. Similarly, short-range tasks may experience delays while waiting for the completion of previously allocated tasks, thereby diminishing overall cloud performance. One solution to address this imbalance is to create and group VMs of varying sizes, assigning tasks to the appropriate VMs with suitable resources [12]. This approach helps reduce virtual machine creation and task waiting times, ultimately preventing task failures.

2. Related Work

Tasks were grouped by user-assigned priorities and virtual machines (VMs) depending on processing speeds by Alworafi et al. [13]. The writers did not, however, address the issue of tasks having equal priority. Furthermore, task grouping may be considered unfair because it is based only on priorities set by the user. An alternate method that replaces user-assigned priority is to order jobs according to their duration and bandwidth needs.

In order to serve numerous organizations inside the community cloud, Dubey et al. [14] developed a management system that combines the Ideal Distribution Algorithm (IDA) and Enhanced IDA (EIDA). Reducing the execution cost and makespan related to processing workflow applications received by different organizations is the main goal of IDA. This is accomplished by implementing a prudent virtual machine (VM) allocation policy within IDA, which takes into account costs and deadlines for applications when allocating resources. The authors made an effort to ensure a balanced workload among the virtual machines (VMs) in addition to minimizing makespan and execution expenses.

Extended Max-Min Scheduling, which uses Petrinet for load balancing, is an improved version of the Max-Min method that was introduced by El-Kenawy et al. [15]. Instead of selecting tasks based on completion times, this method selects tasks based on expected execution times. One unique aspect of their approach is the use of Petrinets, which are excellent for simulating concurrent behaviors in distributed systems. The results show that this approach, as opposed to the previous Max-Min algorithm, generates schedules with a shorter makespan.

An improved First-In-First-Out (FIFO) task scheduling system using fuzzy clustering approaches was presented by Li et al. [16]. Virtual machines (VMs) are grouped according to computational power, bandwidth, and storage capacity in the suggested method. Both the original and enhanced FIFO methods were used in comparative testing. When compared to other approaches, the analytical results show a 40% increase in resource utilization.

A modified cloud resource provisioning algorithm (MCRP) was used in a different study [17] to allocate resources as efficiently as possible. The fuzzy C-Means approach, which is based on kernels, was utilized to group resources. The suggested approach used little memory while achieving the lowest possible execution time and cost. Notable is the evaluation's exclusive focus on memory utilization for resource provisioning—other resources were not taken into account.

A load balancing approach that considers categorization levels while accounting for virtual machine loads and the pre-estimation of task execution time before allocation was put forth by Fahim et al. [18]. Using worst-case execution time (WCET), the method divides tasks into stages according to the resources that are needed. Nonetheless, a lot of the current literature uses task time as the main classification criterion. The model's goal is to reduce load imbalance across virtual machines (VMs) while maintaining a high level of service; however, it is limited in that it does not account for actual workload demands, which frequently involve varying resource requirements for job completion. In conclusion, the method improves load balancing through job classification, streamlines the task scheduling process, and lowers overall energy usage.

The authors of a related study [19] created a model that addresses task workload by automatically managing and placing tasks in virtual machines (VMs) in the best possible way. They used a classification technique to organize virtual machines (VMs) into distinct classes according to CPU and memory utilization, and to classify jobs according to their size (low, medium, and high). The model maximized resource use while achieving a high degree of Quality of Service.

3. Methodology

When a client submits tasks to the cloud provider, the task manager receives and queues the tasks. Within a data center, multiple physical servers offer numerous virtual machines (VMs) as computational resources to execute these tasks. Each server is equipped with a local resource manager responsible for maintaining VM information, including the number of active VMs and their processing, storage, and bandwidth capacities. Additionally, a global resource manager compiles information from local managers, maintaining a comprehensive database of VMs across all physical servers.

A mathematical model for the research problem is proposed under the following assumptions:

- There are n independent and heterogeneous tasks intended for scheduling and execution on m heterogeneous machines.
- Each task is characterized by a length of L, measured in terms of the number of instructions (Million Instructions - MI).
- Each machine possesses a processing capacity expressed in million instructions per second (MIPS).

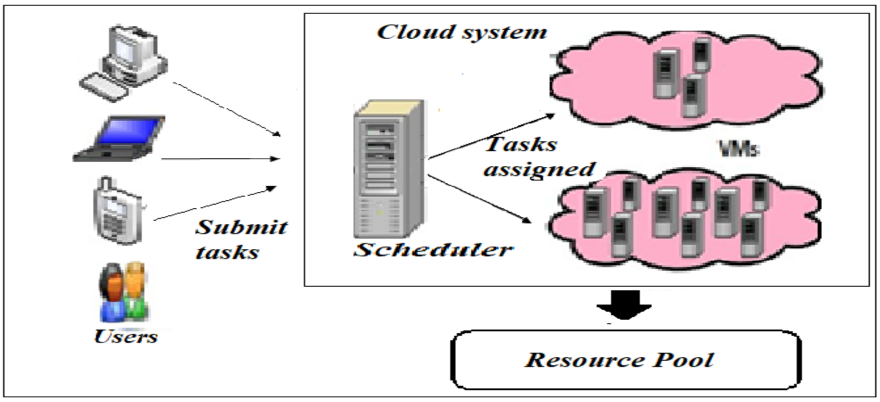


Figure 1 Process of Task scheduling

The scheduling problem is formulated with the objective of scheduling n tasks onto m machines in a manner that ensures a balanced workload on the machines or, alternatively, equal workload distribution according to their individual capacities. Additionally, the aim is to minimize the total completion time of the tasks.

Data clustering [38-41] stands as a crucial and extensively applied technique in the realms of data analysis and data mining. The fundamental goal of clustering is to partition the elements within a dataset into subsets, where elements within the same group exhibit greater similarity to each other than those in different groups. In the domain of data mining, numerous clustering techniques exist, and among them, Fuzzy clustering emerges as a straightforward and efficient approach.

The dataset under consideration is the meta-task set, comprising a batch of tasks designated for scheduling to available resources. The tasks are grouped into clusters based on the proximity of their lengths. The algorithm commences by randomly selecting a centroid value for each cluster. In the proposed system, where tasks are to be clustered, task lengths serve as the data points. The tasks are then clustered and scheduled onto suitable virtual machines (VMs), taking into account the processing capacity of each VM. This approach ensures a balanced workload across all VMs.

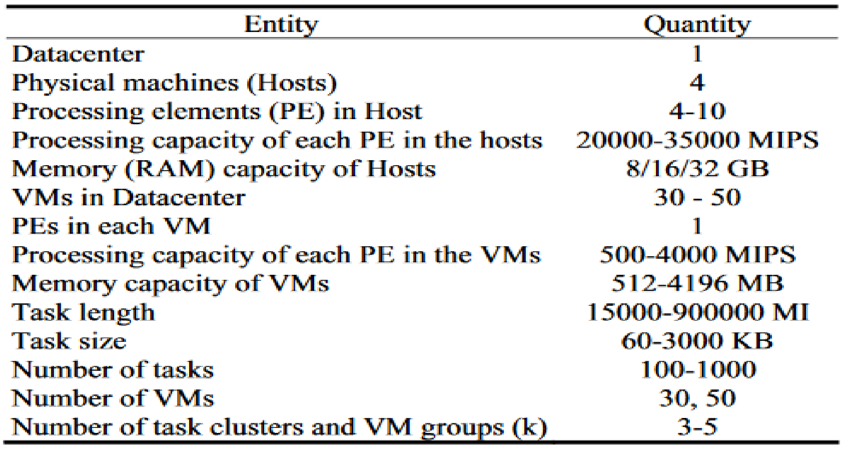
Proposed Task-Mapping Algorithm

```
(1) For each task T in TBQ = {LCMI, CI, MI, HCMI}
(2) For each virtual machine V in VM = {Type-1, Type-2, Type-3, Type-4}
(3) If task T → LCMI, then assign T to Type-1
Else if task T → CI, then assign T to Type-2
 Else if task T → MI, then assign T to Type-3
 Else assign T to Type-4
(4) End if
(5) End for
(6) Update TBQ
(7) Update the available VMs
```

4. Results and Discussion

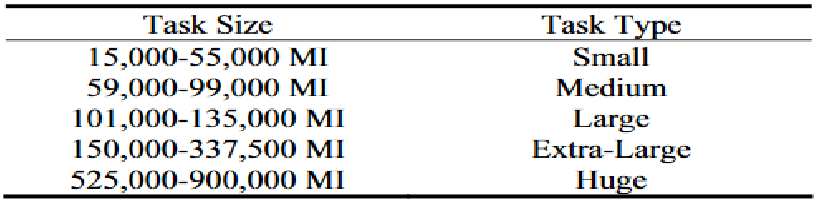
The experimental setup used for the demonstration is shown in Table 1. The simulation environment runs on a 32-bit version of Windows 7 with an 8 GB RAM and a core i5 processor. The Google cluster workload is used in the studies. The task size is classified as follows in Table 2 and spans from 15,000 MI to 900,000 MI.

Table 1. Experimental Setup



Entity	Quantity
Datacenter	1
Physical machines (Hosts)	4
Processing elements (PE) in Host	4-10
Processing capacity of each PE in the hosts	20000-35000 MIPS
Memory (RAM) capacity of Hosts	8/16/32 GB
VMs in Datacenter	30 - 50
PEs in each VM	1
Processing capacity of each PE in the VMs	500-4000 MIPS
Memory capacity of VMs	512-4196 MB
Task length	15000-900000 MI
Task size	60-3000 KB
Number of tasks	100-1000
Number of VMs	30, 50
Number of task clusters and VM groups (k)	3-5

Table 2. Task Classification from the Generated Workload



Task Size	Task Type
15,000-55,000 MI	Small
59,000-99,000 MI	Medium
101,000-135,000 MI	Large
150,000-337,500 MI	Extra-Large
525,000-900,000 MI	Huge

A performance metric is a well-established definition of a quantifiable parameter that reflects a specific facet of performance. It must be measurable and align with the performance objectives inherent in the scheduling problem. In the context of the scheduling problem, any solution aims to reduce both the makespan and task execution time. Consequently, the evaluation of the proposed method is conducted using these metrics.

Makespan is characterized as the maximum time required by a virtual machine (VM) to finish the tasks in the task queue. It is represented as the highest completion time among all tasks, as expressed by equation (1).

MCT = max{ }1, 2, 3, .., , 1, 2, ...,jCTi i m j n∈ ∈ (1)

The performance measures covered in the part before are used to analyse the findings. We run eight different experiments by changing the number of tasks and virtual machines (VMs). VM counts of 30 and 50 are used in the studies, while task counts of 500, 1000, 1500, and 2000 are used. Notably, these studies display varied characteristics in both tasks and resources.

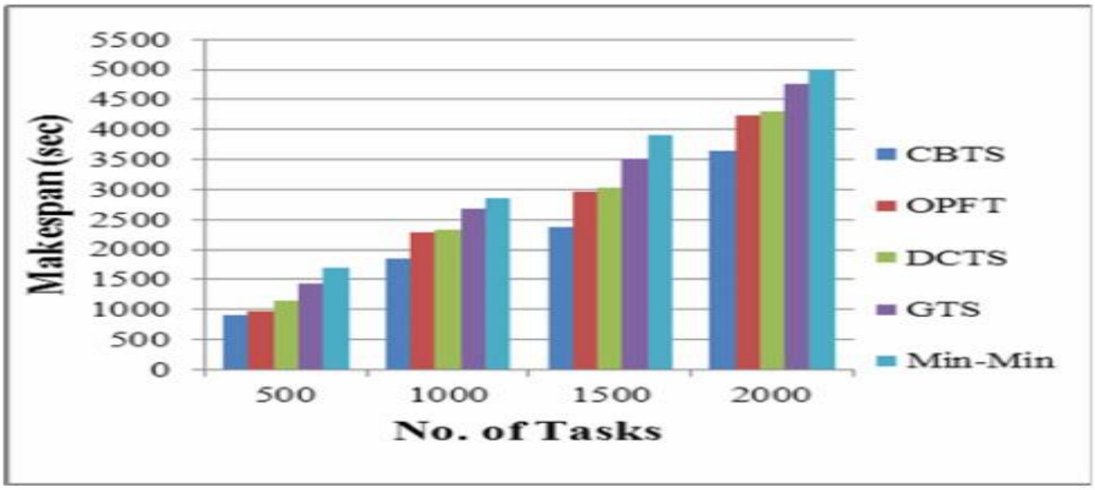


Figure2 Makespan Analysis

5. Conclusions

Scheduling within a distributed computing environment, such as cloud computing, remains a formidable challenge due to the heterogeneous nature of resources found in cloud data centers. The dynamic variation in the number of users and their requests further complicates the scheduling task. Consequently, there arises a demand for a scheduling strategy that accommodates this heterogeneity in the allocation of requests to appropriate resources, aiming to minimize the completion time of these requests. The suggested scheduling mechanism presents a technique that makes use of the Fuzzy C clustering algorithm to create task clusters according to how similar their lengths are. This approach is used for efficient task grouping and is well-known for its simplicity and effectiveness. Tasks in these clusters are then scheduled onto appropriate virtual machines (VMs) according to their capacity. The results of the experimental research show that the suggested approach improves data centre performance by cutting down makespan time.

References

[1] H. G. E. D. H. Ali, I. A. Saroit, and A. M. Kotb, Grouped Tasks Scheduling Algorithm Based on QoS in Cloud Computing Network,Egyptian Informatics Journal, Vol. 18, No. 1, pp. 11-19, March, 2017.

[2] J. Srinivas et al., ''Cloud computing basics,'' Creating Smart Enterprises, vol. 1, pp. 141–171, Jun. 2017, doi: 10.1201/9781315152455-6.

[3] D. Pooja, ''Cloud computing—Overview and its challenges,'' Int. J. Multidisciplinary, vol. 3085, no. 3, pp. 499–501, 2019.

[4] M. Sajid and Z. Raza, ''Cloud computing: Issues & challenges,'' in Proc. Int. Conf. Cloud, Big Data Trust, Jun. 2015.

[5] Q. Jiang, Y. C. Lee, M. Arenaz, L. M. Leslie, and A. Y. Zomaya, ''Optimizing scientific workflows in the cloud: A montage example,'' in Proc. IEEE/ACM 7th Int. Conf. Utility Cloud Comput., Dec. 2014, pp. 517–522, doi: 10.1109/UCC.2014.77.

[6] Manvi, S.S., Shyam, G.K. (2014). Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey. Journal of Network and Computer Applications, 41: 424-440. https://doi.org/10.1016/j.jnca.2013.10.004.

[7] Yousafzai, A., Gani, A., Noor, R.M., Sookhak, M.,Talebian, H., Shiraz, M., Khan, M.K. (2017). Cloud resource allocation schemes: Review, taxonomy, and opportunities. Knowledge and Information Systems, 50(2): 347-381. https://doi.org/10.1007/s10115-016-0951-y.

[8] Masdari, M., Gharehpasha, S., Ghobaei-Arani, M., Ghasemi, V. (2020). Bio-inspired virtual machine placement schemes in cloud computing environment: taxonomy, review, and future research directions. Cluster Computing, 23(4): 2533-2563. https://doi.org/10.1007/s10586-019-03026-9.

9] Anupama, K.C., Nagaraja, R., Jaiganesh, M. (2019). A perspective view of resource-based capacity planning in cloud computing. In 2019 1st International Conference on Advances in Information Technology (ICAIT), pp. 358-363.

[10] S. Anousha, M. Ahmadi, An improved min-min task scheduling algorithm in grid computing,in: J. J. H. Park, H. R. Arabnia, C. Kim, W. Shi, J. M. Gil (Eds.), International Conference on Grid and Pervasive Computing, Springer, Berlin, Heidelberg, 2013, pp. 103-113.

[11] G. B. H. Bindu, K. Ramani, C. S. Bindu, Energy aware multi objective genetic algorithm for task scheduling in cloud computing, International Journal of Internet Protocol Technology, Vol. 11, No. 4, pp. 242-249, October, 2018.

[12] M. A. Alworafi, A. Dhari, A. A. Al-Hashmi, Suresha, A. B. Darem, Cost-Aware Task Scheduling in Cloud Computing Environment, International Journal of Computer Network and Information Security, Vol. 9, No. 5, pp. 52-59, May, 2017.

[13] M. A. Alworafi, A. Al-Hashmi, A. Dhari, Suresha, A. B. Darem, Task-Scheduling in Cloud Computing Environment: Cost Priority Approach, International Conference on Cognition and Recognition, Lecture Notes in Networks and Systems, Springer, Singapore, 2018, pp. 99-108.

[14] K. Dubey, M. Y. Shams, S. C. Sharma, A. Alarifi, M. Amoon, A. A. Nasr, A Management System for Servicing Multi-Organizations on Community Cloud Model in Secure Cloud Environment, IEEE Access, Vol. 7, pp. 159535-159546, October, 2019.

[15] E. S. T. El-kenawy, A. I. El-Desoky, M. F. Al-rahamawy, Extended Max-Min scheduling using Petri Net and load balancing, International Journal of Soft Computing and Engineering, Vol. 2, No. 4, pp. 198-203, September, 2012.

[16] Li, Jian, Tinghuai Ma, Meili Tang, Wenhai Shen, and Yuanfeng Jin. "Improved FIFO scheduling algorithm based on fuzzy clustering in cloud computing." Information 8, no. 1 (2017).

[17] Suresh, Annamalai, and R. Varatharajan. "Competent resource provisioning and distribution techniques for cloud computing environment." Cluster Computing 22.5 (2019): 11039-11046.