HYBRID JOB SCHEDULING METHOD BASED ON CLOUD COMPUTING USING FMPSO STRATEGY AND CUCKOO SEARCH ALGORITHM

Rajeshwari.B
Assistant Professor
Department of Computer Science and Engineering
Sri Indu Institute of Engineering & Technology
Sheriguda, Telangana 501510
Email id: bele.rajic22@gmail.com

Donthagani Uma
Assistant Professor
Department of Computer Science and Engineering
Sri Indu Institute of Engineering & Technology
Sheriguda, Telangana 501510
Email id: umadonthagani@gmail.com

ABSTRACT: As the world pushes toward faster and more efficient computing, cloud computing has become a popular computing technique to satisfy these expanding needs. In order to guarantee efficient and economical executions, a cloud environment need a good task scheduling technique. This study proposes a hybrid task scheduling algorithm, FMPSO, based on fuzzy system and modified particle swarm optimization technique, to enhance load balancing and cloud performance. Four modified velocity updating techniques and a roulette wheel selection mechanism are first considered by the FMPSO approach, both of which increase the global search's efficiency. Then, to avoid local optima and other problems plaguing PSO, it uses crossover and mutation operators. Finally, fitness in this schema is determined by a fuzzy inference method. Task duration, central processing unit speed, memory size, and total execution time are all input parameters for the proposed fuzzy system. By incorporating these fuzzy systems, the FMPSO technique achieves its goal of reducing execution time and resource utilisation. The FMPSO algorithm is analysed using the CloudSim toolkit, and simulation results demonstrate its superiority over competing approaches in terms of makespan, improvement ratio, imbalance degree, efficiency, and total execution time. With the proposed FMPSO algorithm, the local optima issue is sidestepped by always choosing the first available VM if the load on all of them is constant. However, if the primary VM's processing speed is slow, the whole processing will be time-consuming. This work applies a modification to the CUCKOO search algorithm to solve this problem. Cuckoo search techniques pool all virtual machines under continuous load and choose the quickest ones to expedite processing. It is possible to further reduce the time required for production.

***Keywords** – FMPSO strategy, Fuzzy theory, CUCKOO search algorithm.*

1. INTRODUCTION

Algorithms for scheduling tasks are described as the process for choosing the resources to carry out activities with the least amount of waiting and execution time.

Scheduling levels

Two layers of scheduling algorithms exist in the cloud computing environment:

• First level: a set of distribution rules for VMs are found at the host level.

• Second level: A collection of rules to assign tasks to VMs are found at the VM level.

We chose task scheduling algorithms as a research topic because it is the biggest challenge in cloud computing and the primary factor that controls the performance criteria, such as (execution time, response time, waiting time, network, bandwidth, and services cost) for all tasks, as well as controlling other factors that can affect performance, such as power consumption, availability, scalability, storage capacity, and buffer cap.

Tasks scheduling algorithms definition and advantages

A collection of rules and regulations known as tasks scheduling algorithms are used to allocate tasks to the appropriate resources (CPU, memory, and bandwidth) in order to achieve the best degree of performance and resource utilisation.

Task scheduling algorithms advantages

- Control QoS and performance in the cloud.
- Control the CPU and RAM.
- Effective scheduling algorithms reduce overall task execution time while optimising resource use.
- Improving equity across all tasks.
- Increasing the quantity of jobs that are successfully finished.
- Task planning using a real-time system.
- Improving system throughput.
- Improving load distribution.

Tasks scheduling algorithms classifications

Tasks scheduling algorithms classified as in Figure 1.

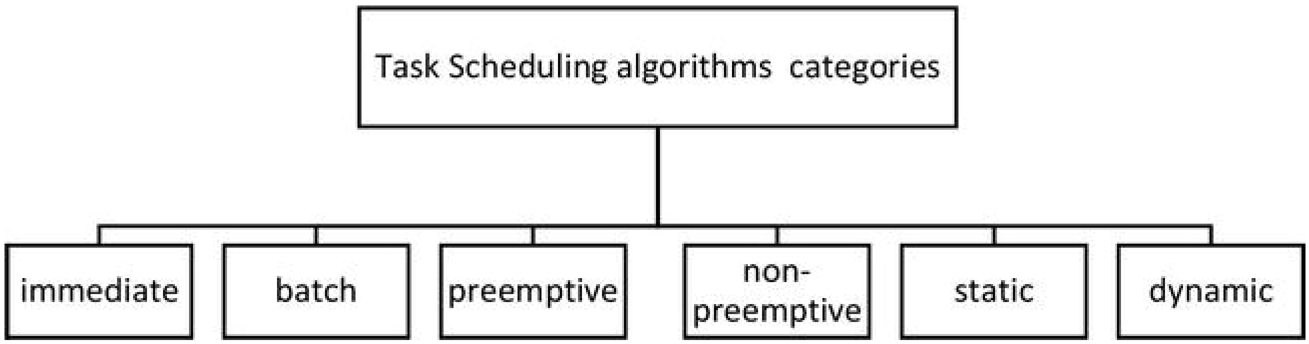

Figure 1. Tasks scheduling classes.

Tasks scheduling algorithms can be classified as follows

- Immediate scheduling: when new tasks arrive, they are scheduled to VMs directly.
- Batch scheduling: tasks are grouped into a batch before being sent; this type is also called mapping events.
- Static scheduling: is considered very simple compared to dynamic scheduling; it is based on prior information of the global state of the system. It does not take into account the current state of VMs and then divides all traffic equivalently among all VMs in a similar manner such as round robin (RR) and random scheduling algorithms.
- Dynamic scheduling: takes into account the current state of VMs and does not require prior information of the global state of the system and distribute the tasks according to the capacity of all available VMs [4, 5, 6].
- Preemptive scheduling: each task is interrupted during execution and can be moved to another resource to complete execution [6].
- Non-preemptive scheduling: VMs are not re-allocated to new tasks until finishing execution of the scheduled task [6].

In this research, we focus on the static scheduling algorithms. Static scheduling algorithm such as first come first service (FCFS), shortest job first (SJF), and MAX-MAX scheduling algorithms in complexity and cost within a small or medium scale.

Task scheduling system in cloud computing

The task scheduling system in cloud computing passes through three levels [7].

- The first task level: is a set of tasks (Cloudlets) that is sent by cloud users, which are required for execution.
- The second scheduling level: is responsible for mapping tasks to suitable resources to get highest resource utilization with minimum makespan. The makespan is the overall completion time for all tasks from the beginning to the end [7].
- The third VMs level: is a set of (VMs) which are used to execute the tasks as in Figure 2.

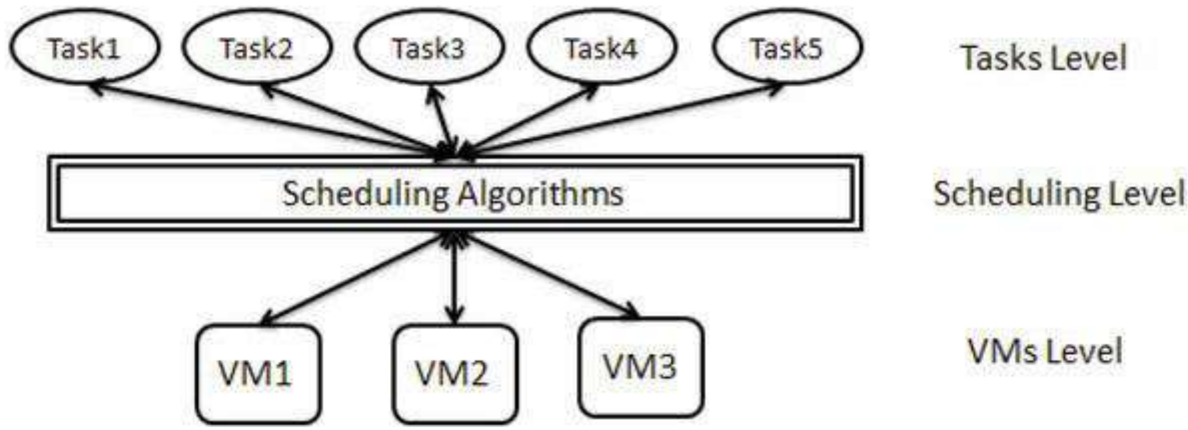

Figure 2. Task scheduling system.

This level passes through two steps

- The first step is discovering and filtering all the VMs that are presented in the system and collecting status information related to them by using a datacenter broker [8].
- In the second step a suitable VM is selected based on task properties [8].

Heuristic algorithms minimise the number of assumptions made before scheduling a job by considering factors like load and job length. Approximate and optimal scheduling strategies both use the same underlying formal computational model and input data. However, since the solution space for optimal schedulers is NP-complete, approximate scheduling techniques are used instead. Without a doubt, the more data we have, the more problems we face when trying to put it to use. Therefore, heuristic strategies are widespread and useful in the actual world. Distributed scheduling is an area where swarm intelligence-based methods have shown to be both popular and effective. The prior advantage is that they can tackle optimization problems without needing in-depth expertise in that area.

2. LITERATURE REVIEW

In-depth analysis of swarm optimization techniques:

Many different swarm optimization methods, beginning with Evolutionary Programming and ending with the most current method, Grey Wolf Optimization, have been developed since the 1960s. Each of these algorithms has shown success in tackling a different optimization challenge. The paper provides a comprehensive review of well-known optimization techniques. Selected methods are briefly described and contrasted using tests using 30 widely-used benchmark functions. The advantages and disadvantages are also discussed. A number of statistical methods are then used to identify the notable performances. It turns out that Differential Evolution (DE) is the best method out there, with Particle Swarm Optimization (PSO) coming in second.

Scheduling tasks in a cloud computing task using a variant of the PSO algorithm.

Fixing the task of scheduling tasks is essential for fully realising the benefits of a cloud computing infrastructure. Although research has shown that obtaining the best solution is very difficult, it is optimal to discover a sub-optimal one by using heuristic algorithms. In this study, three heuristic approaches to cloud-based task scheduling are evaluated and contrasted. The PSO algorithm, genetic algorithms, and modified PSO algorithms are all examples of such techniques for efficient job scheduling. All three of these algorithms are designed to maximise efficiency by generating a schedule that allows each task to be completed in the least amount of time possible.

Scheduling tasks in the cloud using an optimization task based on symbiotic organism search

Appropriate task scheduling is a crucial component in realising the full potential of cloud computing. Scheduling several tasks over many virtual machines may help cut down on wait times and increase system utilisation. Developing a precise solution for large job sizes is challenging since the task scheduling issue is NP-complete. Here, we offer the Discrete Symbiotic Organism Search (DSOS) technique for optimal cloud-based task scheduling. To address these kind of difficulties in numerical optimization, researchers have developed a new metaheuristic optimization approach called Symbiotic Organism Search (SOS). In this sense, SOS mimics the mutualism, commensalism, and parasitism that naturally occur amongst organisms in an environment. In a head-to-head comparison based on simulation results, DSOS is shown to be superior than Particle Swarm Optimization (PSO), a popular heuristic optimization strategy for task scheduling problems. Because it converges more fast as the search size grows, DSOS is well-suited for problems of this magnitude that include scheduling. Results from a T-test comparing the proposed approach to PSO showed that, in particular for large search areas, DSOS performed far better.

Examination of the relative impacts of various membership functions on the effectiveness of fuzzy logic controllers:

In fuzzy set theory, membership functions (MFs) are the basic building blocks that determine how fuzzy a set is. For a particular task, MF shapes are essential because of the effect they have on a fuzzy inference system. Many other forms are possible, including triangular, trapezoidal, gaussian, and others. The only real requirement for an MF is that its value falls inside the interval [0,1]. This paper describes a fuzzy strategy for creating a controller based on fuzzy logic, with the goal of analysing the effect of the membership function in a fuzzy logic controller and contrasting the results obtained by using the controller with three distinct membership functions. As such, work has been done to develop a fuzzy-based antenna azimuth control system.

Strategies for scheduling scientific workflows on the cloud and on grids that minimise costs: Analyses, categorizations, and unanswered questions:

Process scheduling is a challenging issue in scientific computing systems because it must balance the conflicting problems of minimising the cost of workflow execution while still satisfying users' quality of service criteria. There have been a variety of cost optimization approaches proposed to improve the cost-effectiveness of Scientific Workflow Scheduling (SWFS) in cloud and grid computing. There has not been a published study of the literature about techniques for supporting cost optimization in the context of SWFS in cloud and grid computing. Furthermore, there is a gap in the present literature when it comes to exploring helpful recommendations and analyses to grasp cost optimization of SWFS procedures. With a comprehensive analysis of existing SWFS approaches in cloud and grid computing, this study aims to categorise cost optimization problems and SWFS features and investigate the subject of cost optimization in SWFS. In addition, it provides a breakdown of cost metrics into monetary and temporal cost parameters based on distinct scheduling epochs. We believe our results will help practitioners and researchers choose the most appropriate cost optimization technique in light of the highlighted features and characteristics. We also provide directions for how this dynamic area of study may develop in the future.

3. METHODOLOGY

The phrase "cloud computing" is used here to refer to a concept wherein customers are given access to shared pools of computational resources on a pay-as-you-go basis. Effective job scheduling is crucial for maximising productivity in the cloud. Typical scheduling methods fail in distributed systems because task scheduling is an NP-complete problem.

This study proposes a hybrid task scheduling algorithm, FMPSO, based on fuzzy system and modified particle swarm optimization technique, to enhance load balancing and cloud performance. Four modified velocity updating techniques and a roulette wheel selection mechanism are first considered by the FMPSO approach, both of which increase the global search's efficiency. Then, it uses the crossover and mutation operators to circumvent local optimum and other PSO restrictions. Finally, fitness in this schema is determined by a fuzzy inference method.

Task duration, central processing unit speed, memory size, and total execution time are all input parameters for the proposed fuzzy system. By incorporating these fuzzy systems, the FMPSO technique achieves its goal of reducing execution time and resource utilisation. The FMPSO algorithm is analysed using the CloudSim toolkit, and simulation results demonstrate its superiority over competing approaches in terms of makespan, improvement ratio, imbalance degree, efficiency, and total execution time.

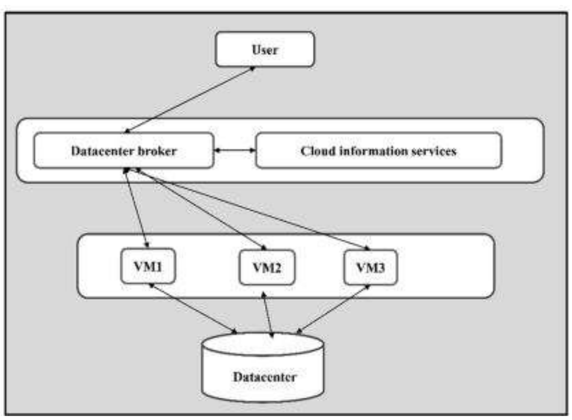

Fig.2: System architecture

Task scheduling is an essential solution for any endeavour that necessitates the use of many resources. For instance, if tasks are divided up evenly throughout a team, everyone's efforts will be maximised and the quality of the final product will increase. Network applications often only need a few of specially designed servers to process requests from users all over the globe. Requests may be handled more effectively by servers if they are planned in advance; otherwise, response times will be much slower than they already are.

When compared to the expense of constructing a company's own infrastructure—consisting of personal computers and networking hardware—moving programmes to the cloud is a no-brainer.

Request scheduling in cloud systems is becoming more important as the number of apps available grows. New algorithms like Round Robin and Shortest Job First were developed to optimise scheduling, however they underperformed. Eventually, algorithms like Genetic Algorithms (GA) and Particle Swam Optimization (PSO) were created that mimicked natural processes. Despite its improved performance, this algorithm still suffers from the local optima issue on occasion.

For PSO algorithms, it's like a flock of birds all taking off at once in a search for sustenance. If one bird is near the food, its fitness value increases, and the others will likely follow. If a group of birds' fitness values are all the same, they risk being caught in the local optimum trap. In this task scheduling scenario, the user's request acts as the source of food, and the birds' main objective is to choose the VM that can execute the request in the shortest amount of time.

For this study's purpose of resolving the local optimum issue, the author devised a hybrid fuzzy-based PSO task scheduling approach. FMPSO is the acronym for this strategy.

Fuzzy: We'll use a roulette wheel selection approach to determine fitness, and then allocate resources accordingly. (Modified PSO) While the current PSO often gets stuck in local optimum solutions, the suggested approach employs the SJFP technique to prevent this from happening (shortest job first processed). Using its job-sorting capabilities, SJFP selects the FIRST VM to process a request through crossover and mutation if PSO maintains a constant fitness value.

The approach of scheduling that is proposed relies on three fundamental principles. To begin, no single task is the same. Second, users submit n tasks, which should ideally be distributed over m virtual machines to make the most use of available resources and minimise the time it takes to complete the execution. Tasks are scheduled according to the first-come, first-served concept if there are more jobs than virtual machines. In any case, the suggested scheduling method is utilised to delegate tasks to virtual machines (FMPSO).

After the initial population has been constructed as described above, its fitness is evaluated using fuzzy theory. In the suggested system, we used a fuzzy hierarchy. When compared to a comparable monolithic fuzzy system, the most distinguishing feature of a hierarchical fuzzy system is that it has fewer rules.

In this research, the author compares his suggested FMPSO against a number of preexisting algorithms. These algorithms include the SGA (Standard GA), MGA (Standard GA with the combination of SJFP), SPSO (Standard PSO), and MPSO (Standard PSO with the combination of SJFP). We have built MPSO and compared its execution time to that of the suggested FMPSO since it is difficult to develop all existing algorithms.

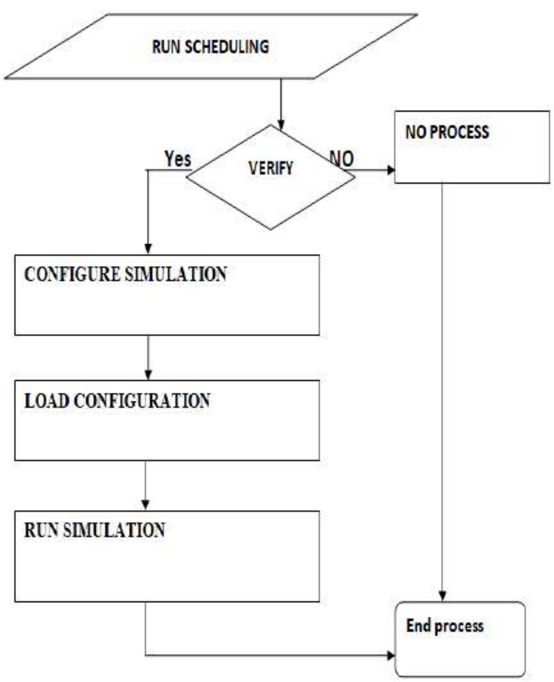

Fig.3: Dataflow diagram

5. EXPERIMENTAL RESULTS

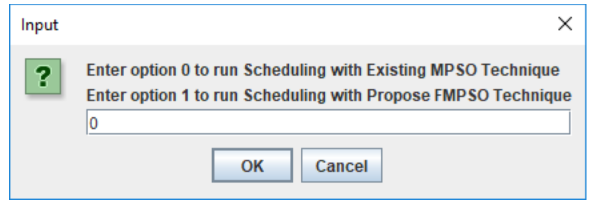

Fig.4: Input screen

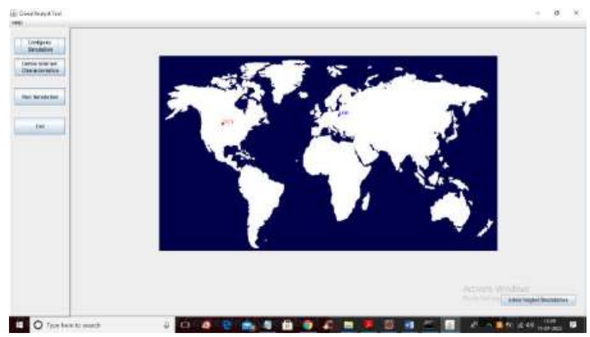

Fig.5: Data center configuration

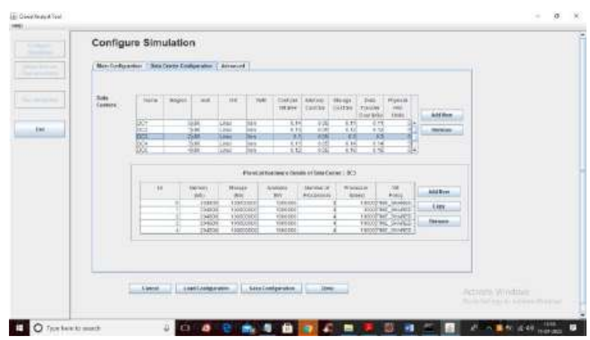

Fig.6: Preprocess dataset

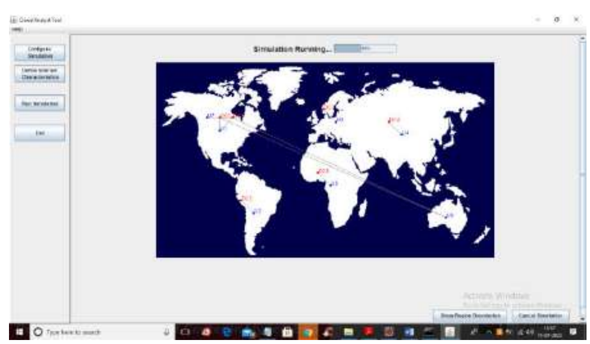

Fig.7: Run simulation

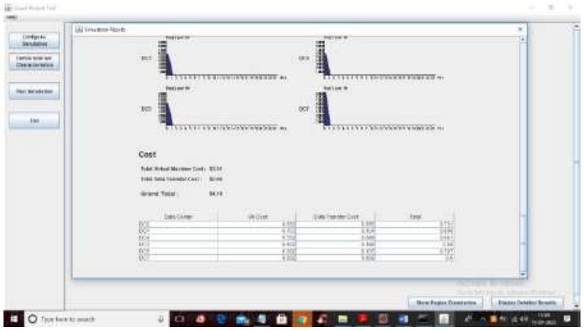

Fig.8: Existing MPSO technique

In above screen with MPSO technique user has to pay $4.18

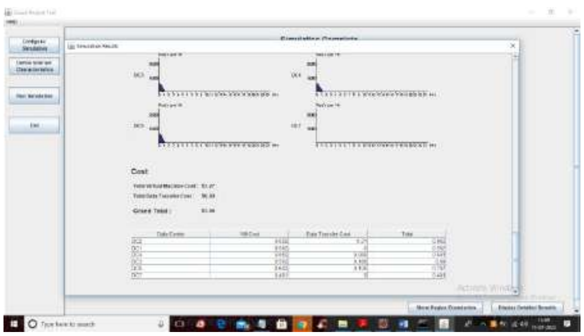

Fig.9: Proposed FMPSO technique

In the preceding screen, efficient use of system resources resulted in a drop in cloud fees to $3.96, which, when added to the existing MPSO, totaled $4.16. This means that the suggested FMPSO might be used for effective task management and preparation.

EXTENSION:

With the proposed FMPSO algorithm, the local optima issue is sidestepped by always choosing the first available VM if the load on all of them is constant. However, if the initial VM's processing speed is slow, the whole processing will be time-consuming. The CUCKOO search algorithm is being employed to find a solution to this problem. Cuckoo search techniques pool all virtual machines under continuous load and choose the quickest ones to expedite processing. There's time for an even quicker maketime.

Taking cues from the cuckoo's innate propensity to lay eggs in the nests of other birds, the cuckoo search algorithm ensures that the host cuckoo bird's egg has a high chance of hatching in the chosen nest. We will also choose data centres or virtual computers with the fastest processing speeds to cut down on Makespan time and speed up processing.

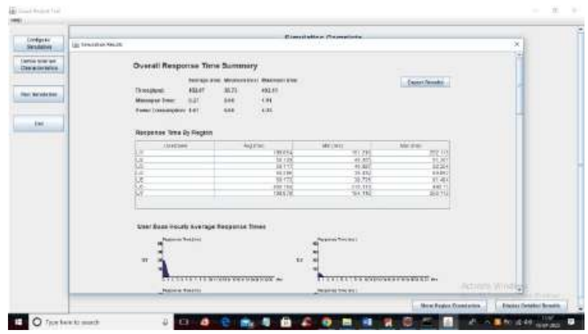

Fig.10: Extension CUCKOO search algorithm

Throughput speed was 162 KPBS, makespan time was reduced to 0.27 seconds, and power consumption was cut in half to 1.07 watts using the cuckoo search extension shown above. Consequently, the efficiency of the scheduling approach may be greatly enhanced by integrating FMPSO and cuckoo search.

6. CONCLUSION

Task scheduling is a crucial topic in cloud systems to improve service quality and system performance. However, task scheduling in distributed systems is NP-complete, leading to the introduction of several strategies. The scheduling method suggested in this paper uses fuzzy theory and the PSO technique to minimise wait time and maximise resource utilisation. The suggested technique (FMPSO) uses a fuzzy system to calculate fitness using significant inputs like task duration, CPU speed, RAM size, and overall execution time. In order to scan a larger search space, four modified velocity updating algorithms are used. Additionally, the POS algorithm is paired with crossover and mutation operators to enhance optimization performance. We have implemented the SPSO, SGA, MPSO, and FUGE scheduling algorithms to assess the effectiveness of various scheduling algorithms with CloudSim. According to the experimental findings, the suggested method offers glaring gains in terms of imbalance severity, makespan, execution time, efficiency, and improvement ratio. This method has proven to be quicker and more highly convergent than the present scheduling algorithms, which are already one of the most used optimization techniques. If all VMs have a constant load, the proposed FMPSO algorithm avoids the local optima problem by selecting the first one. However, if the processing speed of the first VM is low, processing will take a long time. To solve this issue, we are using the CUCKOO search algorithm as an extension work in this paper.

REFERENCES

1. Kumar Garg S, Buyya R. Green Cloud Computing and Environmental Sustainability, Australia: Cloud Computing and Distributed Systems (CLOUDS) Laboratory Department of Computer Science and Software Engineering, The University of Melbourne; 2012
2. Al-maamari A, Omara F. Task scheduling using PSO algorithm in cloud computing environments. International Journal of Grid Distribution Computing. 2015;8(5):245-256
3. Anantharam, Banoth, and Gurnadha Guptha. "Security Issues in Various Cloud Computing: Solution and Occur-Rent Solutions." International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN (2016): 2456-3307.Ramotra A, Bala A. Task-Aware Priority Based Scheduling in Cloud Computing [master thesis]. Thapar University; 2013
4. http://www.techinmind.com/what-is-cloud-computing-what-are-its-advantages-and-disadvantages/ [Accessed: 01 October 2017]
5. https://siliconangle.com/blog/2016/04/29/survey-sees-rapid-growth-in-enterprise-cloud-adoption/ [Accessed: 01 October 2017]
6. Basney, J., Livny, M., & Mazzanti, P. (2000). Harnessing the capacity of computational grids for high energy physics. Conference on computing in high energy and nuclear physics, Italy (pp. 10–13). .
7. Bittencourt, L. F., Goldman, A., Madeira, E. R. M., Fonseca, N. L. S., & Sakellariou, R. (2018). Scheduling in distributed systems: A cloud computing perspective. Computer Science Review, 30, 31–54.
8. Kurikala, Geetha, K. Gurnadha Gupta, and A. Swapna. "Fog computing: Implementation of security and privacy to comprehensive approach for avoiding knowledge thieving attack exploitation decoy technology." International Journal of Scientific Research in Computer Science, Engineering and Information Technology 2.4 (2017): 176-181.