Task Scheduling Using Enhanced Sunflower Optimization in Cloud Environment

R. Ilakiya[1], B. Parkavi[2],R. Anushiya[3], Dr. M. Suresh Kumar[4]
[1, 2, 3] Dept of Information Technology
[4]Professor, Dept of Information Technology
[1, 2, 3, 4] Sri Rama Krishna Institute of Technology, Coimbatore, Tamil Nadu, India

Abstract- *The abstract commences by introducing the importance of task scheduling and the obstacles associated with it. It emphasizes the necessity for effective optimization methods to tackle the complexities arising from elements such as task interdependencies, resource limitations, and ever-changing environments. Particle Swarm Optimization (PSO) has emerged as a promising algorithm for addressing the intricacies of task scheduling in cloud computing settings. This abstract provides an overview of PSO's application in cloud task scheduling, underscoring its capability to enhance resource allocation efficiency, adjust to dynamic workloads, and boost overall system performance. By emulating the social interactions of particles within a swarm, PSO dynamically navigates through solution spaces, maintaining a balance between exploration and exploitation to move towards optimal task distributions. This abstract explores the process of PSO in cloud task scheduling, highlighting its adaptability to shifting conditions and its flexibility in optimizing various objectives. Additionally, it underscores the importance of PSO in enhancing resource usage, reducing make-span, and handling the issues related to scalability and effectiveness in cloud computing. Through its iterative refinement procedure, PSO offers a reliable and scalable solution for task scheduling in cloud environments, leading to improved performance and service quality for cloud-based applications and services.*

Keywords- Task scheduling, completion time, task scheduler, resource information server, virtual machine, meta-heuristic scheduling, enhanced sunflower optimization.

I. INTRODUCTION

Cloud computing is a recent method that offers easy access to shared computer resources like networks, servers, storage, software, and services over the network when needed. It minimizes management efforts and communication with providers while allowing quick provisioning and release. Currently, cloud services are mainly utilized in business settings, focusing on providing IT infrastructure on demand. Cloud computing has the potential to significantly impact various sectors, including search engines, social networks, e-business, virtual worlds, and innovations. Nevertheless, it is still in its early stages and requires further exploration. Issues such as the absence of a common programming paradigm, open interfaces, adequate service level agreements, and application portability limit the effectiveness of cloud computing solutions. Neglecting these challenges may lead to consumer reliance on closed, proprietary systems, hindering the mobility of applications that are being transitioned to the cloud. Furthermore, users often provide commercial providers access to their data and applications without a clear quality of service agreement. The primary objective is to establish a framework for evaluating the benefits of cloud computing over traditional IT infrastructure. The emergence of cloud computing providers and the question of portability are driving factors in the development of this field. Many businesses have already integrated cloud computing services into their IT structures.

- The Enhanced Sunflower Optimization (ESFO) algorithm has exhibited significant enhancements in energy consumption and make-span measures compared to other algorithms in task scheduling benchmarks. The ESFO algorithm aims to enhance both exploitation and exploration in solving task scheduling challenges within a cloud environment.
- Evaluation of the ESFO algorithm using various task scheduling benchmarks has showcased its superiority over comparable algorithms, particularly in terms of energy consumption and make-span measures.

II. RELATED WORK

Researchers have made significant progress in recent years in the field of task scheduling within a cloud environment, utilizing various methods and algorithms to improve accuracy resilience.

The research paper by G. Natesan,A. Chokkalingam et al.[1] task scheduling's main goals are to minimize the schedule's objective and schedule the task based on available resources. The grey wolf algorithm was stated in this study to improve system performance and reduce scheduling problems. This method's primary objective is to reduce energy usage as

well as make span. The simulation's conclusion indicates that, in contrast to alternative algorithms currently in use, the suggested Mean GWO algorithms produces relatively ample results.

The research led by N. Mansouri, B.M.H. Zade, M.M. Javidi et al.[2] an appropriate arranging tasks strategy is required in cloud settings to offer economical implementations. In order to enhance cloud throughput and load balancing, this paper presents the modified particle swarm optimization (PSO) approach and a fuzzy system as the foundation of the FMPSO hybrid task scheduling algorithm. The FMPSO strategy first considers a roulette wheel selection technique along with, Four modified velocity updating methods have been developed to improve the global search capability. Next, in order to get around some of PSO's shortcomings, like local optima, it makes use of crossover and mutation operators. Lastly, the system of fuzzy inference is employed.

In their study, J. Meshkati, F. Safi-Esfahani [3] a massive electrical Cloud data center energy consumption raises their operating expenses. This demonstrates the value of spending money on energy-saving measures. One way to lower energy consumption is to use meta-heuristic algorithms to move virtual machines around dynamically the right reality nodes. To discover superior options in a search area, meta-heuristic algorithms should discover a middle ground between exploitation and exploration. Exploration is the process of searching a larger area for a solution, whereas exploitation is the process of creating new solutions from pre- existing ones. In biological, The One type of meta- heuristic algorithm is the artificial bee colony (ABC) algorithm. Optimization a sign-oriented method, its exploitation power is comparatively weaker than its strong exploration ability.

J. Yang, et al. [4] compared to a traditional distributed system, the task scheduling is more intricate. Based on a cloud computing analysis task scheduling system, created a simplified form, cloud computing in related literature. In contrast to earlier research on task scheduling algorithms for cloud computing as a mathematical tool, the simplified model in this work has as its foundation game theory. The algorithm for task scheduling that takes into take into consideration the stability of the balanced task is suggested, and it is based on the theory of games. The model of task scheduling proposed here is for nodes that compute and is built upon the algorithm for balanced scheduling.

X. Wei,et al.'s research [5] presents a method of task scheduling optimization employing ant colony optimization algorithm in cloud computing to address problems with inconsistent load, sluggish convergence speed, and low virtual machine resource utilization present in earlier methods for optimizing task scheduling. First, An enhanced An algorithm for ant colonies scheduling model to be suggested for preventing The enhancement approach avoiding getting caught up in local optimization. In cloud computing, task scheduling is the foundation for the task scheduling satisfaction function is then created By integrating the three goals of the least amount the length of waiting, the level of resource load balance, and the task completion cost in order find the ideal task scheduling conclusion.

III. METHODOLOGY

The Enhanced Sunflower Optimization algorithm is a sophisticated meta-heuristic method that draws inspiration from the innate behavior of sunflowers. This approach proves highly efficient in tackling optimization challenges across diverse fields.

Problem Formulation

Clearly outlining the task scheduling issue involves identifying the tasks, resources, constraints, and goals for optimization. It is essential to determine if the problem is static or dynamic and if it requires single or multiple criteria optimization.

Encoding

Develop a suitable encoding scheme to represent the solution space. This could involve representing schedules as permutations, binary strings, or other suitable representations depending on the problem characteristics.

Objective Function

The objective function(s) to be optimized, a common objectives in task scheduling include minimizing make-span, minimizing total completion time, maximizing resource utilization, or balancing workload distribution.

Initialization

Initialize the population of sunflowers representing candidate solutions. Ensure that the initial solutions adhere to problem constraints and are diverse to promote exploration.

Enhanced Sunflower Optimization

- **Germination Phase:** Update the positions of sunflowers based on the traditional sunflower

optimization rules, considering factors such as attraction towards the sun and avoidance of collision with other sunflowers.

- **Pollination Phase:** Enhance the pollination mechanism by introducing probabilistic pollination strategies, such as Levy flight or Gaussian distributions, to explore the solution space more effectively.
- **Competition Phase:** Incorporate competition mechanisms to encourage diversity and prevent premature convergence. This could involve introducing crowding or tournament selection mechanisms.
- **Adaptation Mechanisms:** Implement adaptive mechanisms to dynamically adjust algorithm parameters or strategies based on the problem landscape. This could involve adaptive step sizes, mutation rates, or population sizes.
- **Local Search:** Optionally incorporate local search techniques to exploit promising regions of the solution space. This could involve hill-climbing, simulated annealing, or genetic operators.
- **Termination Criteria:** Define stopping criteria for the optimization process, such as a maximum number of iterations, reaching a certain fitness threshold, or stagnation in the search process.

Solution Extraction

Obtain the optimal solution(s) achieved after the completion of the optimization process. Assess the quality of the solution(s) by evaluating them based on the specified objective function(s).

Performance Evaluation

The effectiveness of the enhanced sunflower optimization algorithm can be assessed by comparing the solutions it produces with those of other metaheuristic algorithms or exact methods. Assess metrics such as solution quality, convergence speed, and scalability.

Parameter Tuning and Sensitivity Analysis

Conducting experiments to adjust parameters is essential in optimizing algorithm settings for a particular task scheduling issue. Sensitivity analysis should be conducted to grasp how alterations in parameters affect the overall performance of the algorithm.

Validation and Testing

Validate the algorithm on benchmark instances and real-world datasets to assess its applicability and effectiveness in practical scenarios.

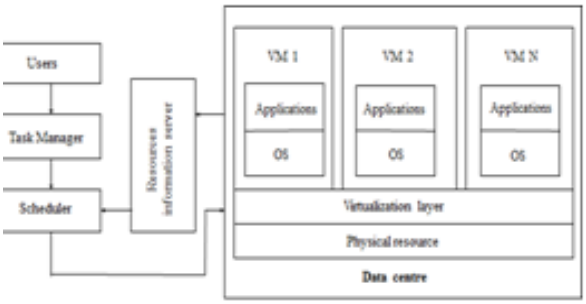
Documentation and Dissemination

Document the methodology, implementation details, experimental results, and insights gained from applying enhanced sunflower optimization to task scheduling. Share findings through research papers, presentations, or open-access repositories to contribute to the research community.

Feedback

Continuously refine and enhance the algorithm based on feedback, emerging techniques, and new insights gained from further research and experimentation.

Architecture diagram



IV. CURRENT SYSTEM

Numerous scholars have put forth solutions to address the scheduling and resource allocation issue. Tsai put forth an improved differential evolution algorithm-based multi-object approach. The current approach offers a cloud computing time and cost model. An algorithm for load balancing and scheduling that ignores job sizes was proposed by Magukuri. When processing requests, the writers took the server's refresh times into account.

V. PROPOSED SYSTEM

Introducing the ESFO algorithms to resolve the cloud environment's task scheduling problem. To strengthen the exploitation and exploration capacities in ESFO, a new pollination strategy will be implemented. Assessing the suggested ESFO algorithm against several organizing tasks benchmarks to be able to find its advantages and disadvantages. The most crucial step in locating an optimization problem's global optimum is careful solution

space exploration. The ESFO algorithm is contrasted with other algorithms, such as Round Robin Optimization and First Come First Serve. An improved version of SFO, known as the ESFO algorithm, is to be established to deal with this issue

VI. EXPERIMENTAL SETUP

I. First Come First Serve (FCFS) Scheduling

First come, first served implies that we should handle the job that was received first, regardless of other properties. We can use our real- time scenario to map this particular situation. The person who joined the line first will receive the ticket when we are waiting in line for movie tickets. Only the second person will be severed. The same approach will be used for scheduling as well. In this context, the issue of work scheduling is being discussed. One CPU can be loaded with multiple jobs, as is common knowledge. Scheduling will be handled by the scheduler. If the scheduler employs the FCFS strategy, the job will be scheduled on the processor to be processed based on which process arrived first. Once a processor has been assigned to a task, it cannot be stopped until the task is finished. We refer to this as non-preemptive scheduling. Pre- emptive scheduling is what we do if we are able to stop. Algorithms with optimal scheduling will reduce average waiting times and turnaround times.

II. Round Robin (RR) Scheduling

One of the most commonly utilized algorithms is undoubtedly the round robin (RR) scheduling algorithm. Each process in this algorithm is allocated a fixed time quantum. Nonetheless, it encounters certain issues, most of which are associated with the time quantum's size. Process response and waiting times tend to increase as the time quantum grows. Conversely, when the time quantum is too small, the CPU is burdened with greater context switches, leading to heightened CPU overhead. This delves into optimization methods for the Round Robin algorithm. Rather than employing a static time quantum, several algorithms have suggested using a dynamic one. Processes in the ready queue have their time quantum determined based on factors like mean, median, and dispersion in relation to their remaining burst time. Additionally, a strategy based on multiple time quanta is recommended. Ultimately, the implementation and results have proven that these algorithms effectively address the issues associated with the traditional Round Robin algorithm, enhancing wait times, response times, and turnaround times.

III. Shortest Job First (SJF)

Scheduling the processes in FCFS scheduling based on the time of their arrival. On the other hand, the SJF scheduling algorithm arranges the processes based on their peak time. The process that has the shortest burst time on the list of processes that are available in the ready queue will be scheduled first in SJF scheduling. However, this algorithm is very hard to implement in the system because it is very hard to predict the burst time required for a process.

IV. Particle Swarm Optimization (PSO)

By iteratively updating the positions and velocities of particles based on their own and neighbours experiences, PSO explore the solution space efficiently and can converge to good solutions for task scheduling problems, PSO's ability to balance exploration(searching the solution space broadly) and exploitation(focusing on promising regions) makes it effective for optimization tasks like task scheduling.

VII. SOFTWARE ENVIRONMENT

CLOUD-SIM

A simulation library called CloudSim provides basic cases for managing different system components like scheduling and provisioning. It also describes users, virtual machines, data centers, and computational resources. It is an open-source, free framework for duplicating the infrastructure and services in cloud computing. Well, it is fully written in Java and is developed by the Clouds Lab group. In order to test a hypothesis before developing software and replicate tests and outcomes, it is utilized to simulate and model an environment for cloud computing.

No money is being invested: There's not setup or upkeep costs when using a simulation tool such as CloudSim.

Easy to use and scalable: can modify specifications by altering a few lines of code, adding or removing resources.

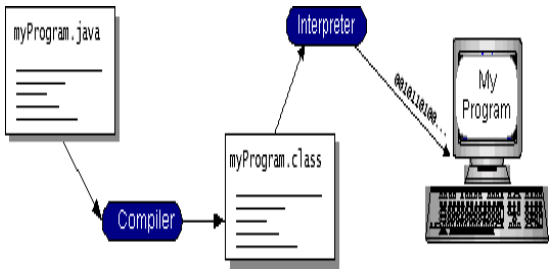
It is possible to assess risks earlier: The utilization of actual test environments within cloud computing confines the ability to experiment to the size of the test environment, posing significant challenges when attempting to replicate results. Through simulation, you can evaluate your product against various test scenarios, addressing issues without any limitations before proceeding to the final deployment.

JDK

The Java programming language stands out for its ability to be compiled and interpreted. Initially, a program

translated into Java byte codes using a compiler, creating codes understood by the Java platform's interpreter. The interpreter parses and executes each Java byte code command on the computer. While interpretation occurs every time the program is run, compilation is a one-time process.

All Java interpreters, whether in Web browsers for applets or development tools, essentially function as the Java Virtual Machine. Java byte codes enable a "write once, run anywhere" capability, allowing programs to be compiled on any platform with a Java compiler into byte codes that can then run on any Java virtual machine.



VIII. RESULTS

Output of Proposed System

I. First Come First Serve (FCFS)

Is a simple scheduling algorithm that schedules tasks based on their arrival times. In context of task scheduling, accuracy is not typically a metric used to evaluate FCFS, as FCFS does not prioritize tasks based on any criteria other than their arrival order.

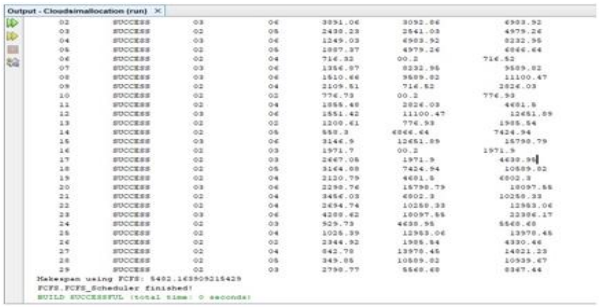


Fig.8.1: FCFS Scheduling

II. Round Robin (RR) Scheduling

It can access the performance of the Round Robin algorithm using various criteria such as efficiency, fairness, response time, etc.

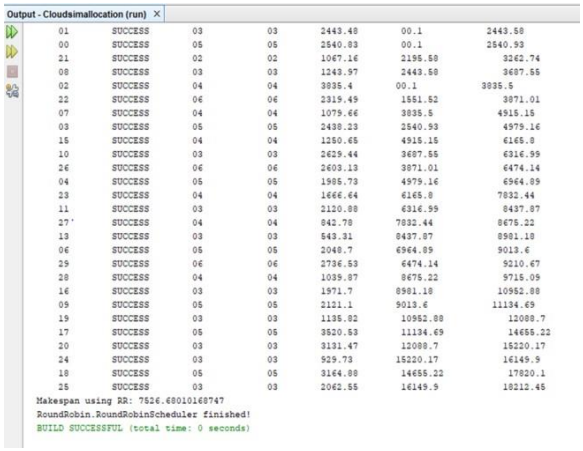


Fig.8.2: RR Scheduling

III. Shortest Job First (SJF)

To calculate accuracy of Shortest Job First (SJF) scheduling algorithm, by

Accuracy= (Total Waiting Time of SJF)/(Total Waiting Time of Other Algorithm)*100%

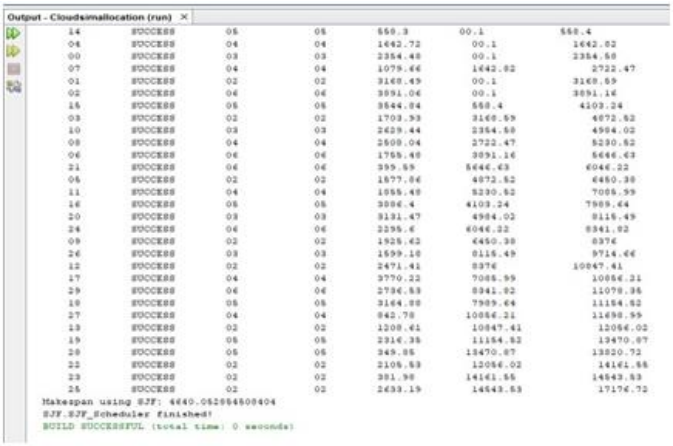


Fig.8.3: SJF Scheduling

IV. Sunflower Optimization (SFO)

Analyze the accuracy of the Sunflower Optimization algorithm based on the comparison with benchmarks and any statistical analysis performed with factors like algorithm runtime and scalability in your interpretation.

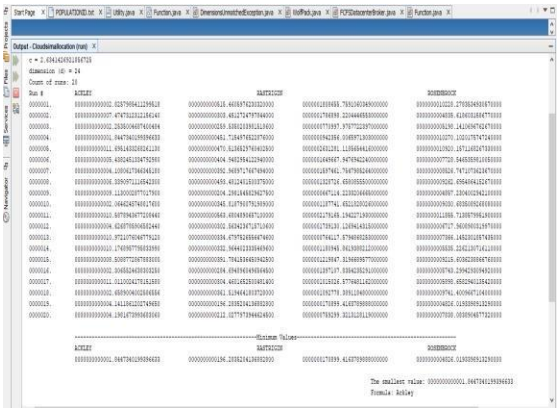


Fig.8.4: SFO Scheduling

V. Particle Swarm Optimization (PSO)

Creates a set of test instances representing different task scheduling scenarios. Each test case includes information such as task duration, resource requirements, and constraints.

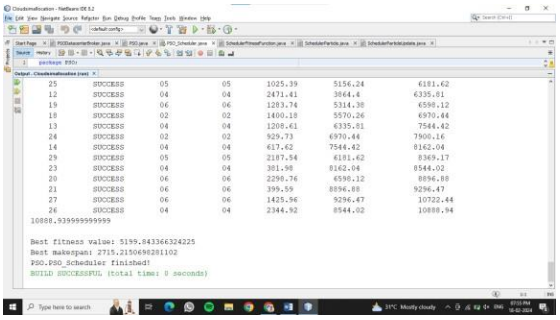


Fig.8.5: PSO scheduling

Scheduling	**Make span**
First Come First Serve(FCFS)	5129.625758533817
Shortest Job First(SJF)	4217.043341627036
Round Robin (RR)	4006.570314680345
Sunflower Optimization(SFO)	3557.046400653260
Particle Swarm Optimization(PSO)	2622.879784299157

Model Evaluation result

IX. CONCLUSION

The cloud has gained increasing popularity in business and research circles for its numerous advantages, such as instant self-service, quality service, pay-as-you-go pricing, virtualization, and scalability. Recent studies have introduced an innovative method for reducing energy consumption in task scheduling. Given that task scheduling is a crucial challenge in cloud computing, this paper has presented an ESFO algorithm that enhances the pollination process of the standard SFO algorithm to efficiently explore a solution space. The ESFO algorithm is proficient at determining an optimal task scheduling solution due to its use of polynomial time complexity. Through comparing the proposed algorithm with others, the outcomes demonstrate a significant enhancement with the ESFO algorithm.

REFERENCES

[1] G. Natesan, A. Chokkalingam, Task scheduling in heterogeneous cloud environment using mean grey wolf optimization algorithm, ICT Express 5 (2019)

[2] N. Mansouri, B.M.H. Zade, M.M. Javidi, Hybrid task scheduling strategy for cloud computing by modified particle swarm optimization and fuzzy theory, Comput. Ind. Eng. 130 (4) (2019)

[3] J. Meshkati, F. Safi-Esfahani, Energy-aware resource utilization based on particle swarm optimization and artificial bee colony algorithms in cloud computing, J. Super Comput. 75 (5) (2019).

[4] J. Yang, et al., A task scheduling algorithm considering game theory designed for energy management in cloud computing, Future Gener.Comput. Syst. 105 (2020).

[5] X. Wei, Task scheduling optimization strategy using improved ant colony optimization algorithm in cloud computing, J. Ambient Intell. Humaniz. Comput. (2020).

[6] Shukla, D. K., Kumar, D., & Kushwaha, D. S. (2021). Task scheduling to reduce energy consumption and makespan of cloud computing using NSGA-II. Materials Today: Proceedings.

[7] Azizi, S., Shojafar, M., Abawajy, J., & Buyya, R. (2022). Deadline-aware and energy-efficient IoT task scheduling in fog computing systems: A semi- greedy approach. Journal of network and computer applications, 201, 103333.

[8] Mangalampalli, S., Swain, S. K., & Mangalampalli, V. K. (2022). Multi Objective Task Scheduling in Cloud Computing Using Cat Swarm Optimization Algorithm. Arabian Journal for Science and Engineering, 47(2), 1821-1830.

[9] Ibrahim, I. M. (2021). Task scheduling algorithms in cloud computing: A review. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(4), 1041-1053.

[10] Mangalampalli, S., Swain, S. K., & Mangalampalli, V. K. (2021). Prioritized Energy Efficient Task Scheduling Algorithm in Cloud Computing Using Whale

Optimization Algorithm. Wireless Personal Communications, 1-17.

[11] Mukherjee, D., Nandy, S., Mohan, S., Al- Otaibi, Y. D., & Alnumay, W. S. (2021). Sustainable task scheduling strategy in cloudlets. Sustainable Computing: Informatics and Systems, 30, 100513.

[12] Emami, H. (2022). Cloud task scheduling using enhanced sunflower optimization algorithm. ICT Express, 8(1), 97-100.

[13] Ijaz, S., Munir, E. U., Ahmad, S. G., Rafique,M. M., & Rana, O. F. (2021). Energy-makespan optimization of workflow scheduling in fog–cloud computing. Computing, 103(9), 2033-2059.

[14] D. Ding, et al., Q-learning based dynamic task scheduling for energy efficient cloud computing, Future Gener. Comput. Syst. 108 (2020) 361–371.

[15] J. Xiao, et al., Game theory–based multi-task scheduling in cloud manufacturing using an extended biogeography-based optimization algorithm, Concurr. Eng. 27 (4) (2019) 314–330.