Task Scheduling using Ant Colony Optimization and comparing the results with SJF and FCFS

Introduction:

CloudSim is an open-source simulation toolkit designed to model and simulate a cloud computing environment. It provides a modeling and simulation platform for researchers, practitioners, and educators to study and experiment with cloud computing environments and algorithms. CloudSim enables the modeling of various cloud computing entities, such as data centers, virtual machines, and applications, and the simulation of various cloud computing activities, such as resource provisioning, load balancing, resource allocation, and workload management.

One of the key features of CloudSim is its ability to model and simulate the cloud computing environment, including the underlying hardware and software infrastructure, and to evaluate the performance and scalability of various cloud computing algorithms.

Problem Statement :

To reduce the makespan time of cloudlets by using and comparing different task scheduling algorithms for mapping cloudlets over VMs.

Objective :

Apply the Shortest Job First Algorithm and First Come First Serve task scheduling algorithms on CloudSim and compare results with that of Ant Colony Optimization implemented.

Ant Colony Optimization:

Ant Colony Optimization (ACO) is a meta-heuristic optimization algorithm inspired by the behavior of ant colonies. Ants are known for their ability to efficiently find the shortest path from their colony to a food source, which they achieve through collective decision-making and communication.

In ACO, a population of artificial ants is used to solve optimization problems by searching for the optimal solution. Each ant constructs a solution sequentially by choosing the next move based on a combination of heuristics and pheromone(chemical compound released by ants) trail information left by previous ants. The pheromone trail information guides the ants toward the most promising solutions, while the heuristics ensure that the ants explore new areas in search of better solutions.

ACO has been applied to various optimization problems, including the traveling salesman problem, the knapsack problem, and the task scheduling problems. One of the key advantages of ACO over other optimization algorithms is its ability to handle problems with incomplete information. Ants make decisions based on the information available to them, allowing ACO to find near-optimal solutions even when some information is missing or uncertain. ACO has been used in a variety of industries, including telecommunications, logistics, and finance, to solve complex optimization problems. Its ability to adapt to changing environments, handle complex constraints, and find near-optimal solutions make it a valuable tool for optimization professionals.

In conclusion, Ant Colony Optimization is a powerful optimization algorithm that has been successfully applied to a variety of complex optimization problems. Its ability to handle problems with incomplete information adapt to changing environments and find near-optimal solutions make it an attractive solution for professionals in many industries.

Algorithm :

Initialization: Define the resource allocation problem and create a population of artificial ants. Initialize the pheromone trail information and heuristics used by each ant.

Construction: For each ant, allow it to construct a solution by choosing the next move based on the combination of heuristics and pheromone trail information. Repeat this process until each ant has constructed a complete solution.

Evaluation: Evaluate the solutions constructed by each ant and determine their quality based on metrics such as resource utilization, response time, and energy consumption.

Pheromone Update: Based on the quality of each solution, update the pheromone trail information for the corresponding moves. The amount of pheromone deposited is proportional to the quality of the solution of task scheduling.

Termination: Repeat the construction and evaluation steps until a satisfactory solution is found or a predetermined number of iterations have been completed.

Result: The best solution found by the ants is considered the result of the ACO algorithm. This solution is then used to allocate resources in CloudSim.

Monitoring: Continuously monitor the resource utilization, response time, and energy consumption of the allocated resources and adjust the pheromone trail information accordingly.

Shortest Job First :

Shortest Job First (SJF) is a scheduling algorithm used in computer operating systems to manage the allocation of CPU resources to processes. The main idea behind SJF is that the process with the shortest execution time should be executed first to minimize the waiting time of other processes. The SJF algorithm is known to be an optimal algorithm, meaning that it provides the minimum average waiting time for processes. However, it has some drawbacks, such as the difficulty in accurately predicting the execution time of a process, which can result in suboptimal performance. Additionally, processes with longer execution times may suffer from starvation if new processes with shorter execution times continuously arrive.

In conclusion, SJF is a simple and effective scheduling algorithm that is widely used in computer operating systems. Despite its limitations, its simplicity and optimality make it a popular choice for resource allocation in computing environments. However, its performance can be improved by incorporating other scheduling algorithms and techniques to better handle the complexities of modern computing systems.

First Come First Serve :

First Come First Serve (FCFS) is a scheduling algorithm used in computer operating systems to manage the allocation of CPU resources to processes. The main idea behind FCFS is straightforward: processes are executed in the order in which they arrive at the ready queue. The first process to arrive is executed first, and the next process is executed only after the previous process has been completed.

FCFS is a simple and easy-to-implement algorithm, making it a popular choice for resource allocation in computing environments. However, it has some drawbacks. The biggest disadvantage of FCFS is that it can result in longer waiting times for processes if a process with a long execution time arrives early in the queue. This can result in suboptimal performance, as other processes are forced to wait for the long-running process to complete.

In conclusion, FCFS is a simple and straightforward scheduling algorithm that is widely used in computer operating systems. Despite its limitations, its simplicity and ease of implementation make it a popular choice for resource allocation in computing environments.

CloudSim and its components :

CloudSim is an open-source simulation toolkit for modeling and evaluating cloud computing systems and data centers. It enables researchers and practitioners to evaluate and compare the performance of cloud computing systems, policies, and algorithms, without the need for expensive hardware resources.

CloudSim consists of several components that work together to simulate the behavior of cloud computing systems:

Datacenters: A datacenter represents the physical infrastructure of a cloud computing system, including servers, storage systems, and networking components. CloudSim enables the modeling of datacenters with different configurations, such as the number of servers, the type of processors, and the amount of memory.

Hosts: A host represents a physical machine in a datacenter, and it can be used to run virtual machines. In CloudSim, a host can have multiple processing elements (PEs), which can be used to run virtual machines in parallel.

Virtual Machines (VMs): A virtual machine represents a logical instance of an operating system and applications running on a host. In CloudSim, a VM can be dynamically created, destroyed, and migrated from one host to another.

Tasks: A task represents a unit of work that needs to be executed in a cloud computing system. In CloudSim, tasks can be submitted to the system, and they are assigned to VMs for execution.

Resource Provisioners: Resource provisioners are responsible for allocating physical resources, such as hosts and storage, to virtual machines. CloudSim provides several resource provisioners, such as the Simple Provisioner, that allocate resources based on simple policies.

Brokers: Brokers are responsible for submitting tasks to the cloud computing system and managing the allocation of resources to virtual machines. CloudSim provides several brokers, such as the DatacenterBroker, that implement different scheduling policies, such as First Come First Serve (FCFS) and Shortest Job First (SJF).

Cloudlet Scheduler: The cloudlet scheduler is responsible for scheduling tasks on virtual machines. It determines which tasks are executed on which virtual machines, based on various factors, such as the processing capabilities of virtual machines, the execution time of tasks, and the state of the system.

Experiment Environment :

Task Length: The length of each task represents the amount of processing required to complete the task. In CloudSim, the length of a task is specified in terms of MI (million instructions).

length = 10000

MIPS (Million Instructions Per Second): MIPS represents the processing power of a virtual machine, and it is specified in terms of the number of million of instructions that can be executed per second.

mips = 250

Bandwidth (BW): Bandwidth represents the amount of data that can be transmitted per second between a virtual machine and a host or between two virtual machines.

bw = 1000

VM Image Size: 10000

RAM: 512 MB

The number of Virtual Machines: The number of virtual machines that will be used to execute the tasks needs to be defined.

vms = different every time

Processing Element (PE): A processing element (PE) is a logical processor that can execute a task. In CloudSim, a host can have multiple PEs, and the number of PEs needs to be specified for each host.

pes = 1

File Size: 300

Output Size: 300

Allocation Policy: The allocation policy defines how virtual machines are assigned to physical hosts. In CloudSim, several allocation policies, such as the Simple Allocation Policy, can be used.

Scheduling Policy: The scheduling policy defines how tasks are assigned to virtual machines for execution. In CloudSim, several scheduling policies, such as First Come First Serve (FCFS) and Shortest Job First (SJF), can be used.

By defining these parameters, you can perform experiments to evaluate the performance of different task-scheduling algorithms in CloudSim.

Parameters for Ant Colony Optimization Algorithm :

Number of Ants: The number of ants that will be used to search for solutions.

m = number of VMs

Evaporation Rate: The evaporation rate determines the rate at which the pheromone trail evaporates.

a = 0.3

Initial Pheromone Value: The initial pheromone value represents the amount of pheromone that will be deposited on each edge of the solution graph.

c = 5

Pheromone Importance: The pheromone importance determines the weight that is given to the pheromone trail when searching for solutions.

Q = 100

Results :

SNo.	Cloudlets	VMs	Makespan of ACO	Makespan of FCFS	Makespan of SJF
1	20	8	23.799	24.599	24.299
2	50	8	85.2	113.4	112.699
3	100	8	314.4	366.599	365.3
4	120	8	422.799	482.99	481.5
5	150	8	615.4	763.8111	761.900
6	200	4	4935.8111	5105.0	5105.00
7	200	5	3195.999	3287.9999	3283.999
8	200	6	2298.4	2386.8	2383.399
9	200	7	1649.5001	1745.8004	1742.899
10	200	8	1108.6001	1305.0002	1302.5

Summary :

The Ant Colony Optimization (ACO) algorithm, Shortest Job First (SJF), and First Come First Serve (FCFS) are popular scheduling algorithms used in CloudSim to optimize the allocation of tasks to virtual machines. In several experiments, ACO has been found to outperform both SJF and FCFS.

ACO uses a metaheuristic approach to solve the task scheduling problem by simulating the behavior of ants searching for food. The algorithm uses pheromone trails to guide the search for solutions, and it adapts to changes in the environment over time, leading to more efficient task scheduling.

SJF, on the other hand, assigns tasks to virtual machines based on the length of the task, with shorter tasks being executed first. This approach can lead to improved utilization of resources, but it can also result in longer waiting times for longer tasks.

FCFS assigns tasks to virtual machines in the order they arrive, without considering the length of the task. This approach can result in poor utilization of resources and long completion times for tasks.

In several experiments, ACO has been found to outperform both SJF and FCFS in terms of resource utilization, task completion times, and energy consumption. This is due to the adaptive and flexible nature of the ACO algorithm, which allows it to respond to changes in the environment and optimize the allocation of tasks to virtual machines.

In conclusion, ACO is a powerful task-scheduling algorithm in CloudSim that outperforms both SJF and FCFS. Its ability to adapt to changes in the environment and its ability to optimize the allocation of tasks to virtual machines make it an attractive choice for organizations looking to optimize their cloud computing resources.