## Evaluation Parameters

After running the simulation, specific parameters were measured based on the objectives of the proposed model. The main parameters used, as indicators of the model's goals (see Section 1.3) were average waiting time, average execution time, throughput, resource utilization, cost, profit, and power consumption. Details on each factor are presented as follows:

* + - * **Average Waiting Time (AWT):** The average waiting time of a task is defined as the ratio of the sum of waiting times of all tasks to the total number of tasks. The average waiting time is measured by computing the difference between the time the task is submitted to the system and the time of starting the execution of all the tasks, as shown in Equation 3.1*.* The waiting time includes the time taken for negotiation, mapping tasks to VMs, data transfer time, and the time at which VMs are migrated.

(3.1)

where:

**AWT** is the average waiting time of all tasks in seconds

**m** denotes the number of tasks running in the system per unit of time

**ST** (i) denotes the start time of execution of the task i in seconds **SubT** (i) denotes the time for submission of the task i in econds

* + - * **Average Completed Time (ACT):** This is the total time taken by each task to finish execution. It is measured by computing the difference between the time of submission of each task and the time of ending execution of each task.

(3.2)

Where

**ACT** is the average completed time of all tasks in seconds

**m** denotes the number of tasks running in the system per unit of time

**Ext (i)** denotes the finish execution time of the task i in seconds

**Subt (i)** denotes the time of submission of the task i in seconds

* + - * **Throughput (TH):** The throughput measures the overall performance of the system. Throughput indicates the number of tasks that our model can execute in a specific time.

TH = (C/T) × 1𝟎𝟎𝟎𝟎 (3.3)

where:

**TH** denotes the throughput of the system **C** is the total number of completed tasks **T** denotes the simulation time in seconds

* + - * **Average VM Utilization (AVU)**: This represents the utilization of the VM in terms of the CPU, memory, storage and bandwidth used by all tasks to finish execution. The CPU utilization for each VM is defined as a percentage ratio of the CPU, memory, storage and bandwidth utilization divided by 4 (which is the number of factors including in computing VM utilization). The CPU utilization is the amount of CPU used for all tasks in a VM over the total CPU of the VM. Memory utilization, storage utilization, and bandwidth utilization are computed in the same way as CPU utilization, as shown in Equations 3.4, 3.5, 3.6 and 3.7.

(3.4)

(3.5)

(3.6)

(3.7)

(3.8)

where:

UVM (i) is the average utilization of VM i CVM (i) is the CPU utilization of VM i

UC (i) is the used CPU for all tasks executed in VM i AC (i) is the total CPU of VM i

MVM is the memory utilization of VM i

UM (i) is the used memory for all tasks executed in VM i AM (i) is the total memory of VM i

SVM (i) is the storage utilization of VM i

US (i) is the used storage for all tasks executed in VM i AS (i) is the total storage of VM i

BVM (i) is the bandwidth utilization of VM i

UB (i) is the used bandwidth for all tasks executed in VM i AB (i) is the total bandwidth of VM i

n denotes the number of VMs

* + - * **Average Resource Utilization (ARU):** Host utilization in our model represents the total utilization of all VMs running on the host. The average resource utilization is computed by summing the host utilization of all available hosts in the data center (Equations 3.9 and 3.10).

(3.9)

where:

HU is the average host utilization for host j

UVM (i) denotes the utilization of all VMs in the host j n denotes the number of VMs in host j

(3.10)

where:

ARU denotes the resource utilization

HU is the average host utilization for host j as shown in Equation 3.9 m denotes the number of hosts in the data center

* + - * **Execution Cost (EC)**: This factor denotes the cost consumers should pay to execute tasks in the providers' resources. It includes the cost of CPU processing, memory, using data storage, and transferring data (Equation 3.11). These costs are all defined in the data center specifications and differ based on the resources.

This is an important factor for consumers and one that most resource allocated algorithms aim to minimize.

(3.11)

where:

EC is the cost of execution of all tasks

n denotes the number of tasks executed in the data center T (i) is the execution time in seconds of task i

costCPU is the cost of processing CPU in $ / seconds R (i) is the size of RAM of task i in MB

costRAM is the cost of memory used in $ / MB S (i) is the size of storage of task I in MB costStorage is the cost of storing data in $ / MB D (i) is the size of the task I file in Kb

costB (i) is the cost of transferring data in $ / Mb

* + - * **Total Profit (TP):** The profit model is based on a pay-as-you-go policy that is applied in many cloud systems to address the highly variable demand for cloud resources and to calculate the cost of executing tasks in the cloud data center (Lee et al. (2012)). Thus, profit represents the total income the provider can gain from executing tasks in their resources. It is calculated depending on the total execution cost for all tasks, penalty costs, and power consumption costs, as shown in Equation 3.12. The penalty cost is computed based on the provider policy and is defined in the SLA; it represents the costs the provider can afford to pay if the SLA is not satisfied. In the proposed model, the penalty is paid for execution delay and is computed according to the delay constraints, as shown in Equation 3.13.

(3.12)

(3.13)

where

TP is the total profit the provider gains from execution of all tasks EC is the cost of execution of tasks

Pen (i) is the penalty the provider can afford to pay for delay in task i

Pcost is the cost of using power for executing tasks, as shown in Equation 3.14 reqt (i) is the time task i is required for execution

exet (i) is the time task i is taken for execution P is the penalty cost defined for the delay

T is the number of tasks

* **Power Consumption (PCost):** This is based on the power cost in kWh, which differs according to the specifications of the data center. It is computed by multiplying the power cost by the total execution time for all tasks, as shown in Equation 3.14.

(3.14)

where:

PCost is the total power cost of execution of all tasks PC is the total execution time for all tasks

Powercost is the cost of power defined in $/seconds

* + - * **Task Completed Rate (TCR)**: This is the number of completed tasks over the total number of submitted tasks.

(3.15)

where:

TCR is the rate of completed tasks, and ranges from [0-1]

Tc is the number of completed tasks

Ts is the number of submitted tasks

* **The SLA Violation (SLAV):** In this research, an SLA violation can occur in relation to deadline and migration time. A SLA violation of deadline constraints occurs when the task deadline is missed and is computed according to the rate of completed tasks computed as shown in Equation 3.16

(3.16)

where:

SLAD denotes the SLA violation based on deadline

TCR is the task completed rate computed by Equation 3.15

A SLA violation for migration time occurs when the consumer does not receive their requested resources. In technical terms, SLA violations occur when VMs cannot acquire the amount of MIPS that are requested. In this case, the SLA violation occurs when the requested CPU is greater than the available capacity of CPU. It is computed as the sum of unallocated MIPS to the sum of the requested MIPS as shown in Equation 3.17.

(3.17)

where:

SLAM denotes the SLA violation based on migration time

RMIPS (i) denotes the MIPS requested by the VM i for running the task AMIPS (i) denotes the actual MIPS that were allocated to the VM i.

The overall SLA violations are computed as shown in Equation 3.18.

(3.18)

where:

SLAD denotes the SLA violation based on deadline SLAM denotes the SLA violation based on migration time SLAV denotes the overall SLA violation

* + - * **The Imbalance Factor (IF):** This refers to the balance of the load among VMs. It is measured based on the time for executing tasks in VMs. A small value of IF indicates good load balancing. Equation 3.19 shows how to compute IF.

(3.19)

Where

IF is an imbalance factor

Tmax is the maximum time for execution of tasks Tmin is the minimum time for execution of tasks Tav is the average time for execution of tasks

* + - * **Average Fitness Values (AFV)**: The average fitness value for each solution is calculated by applying the PSO algorithm and evaluating selected objectives, as shown in Equation 4.20.

(3.20)

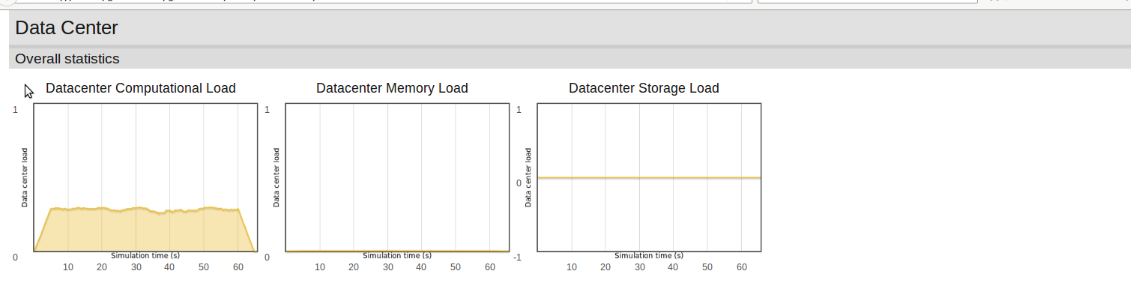
Where

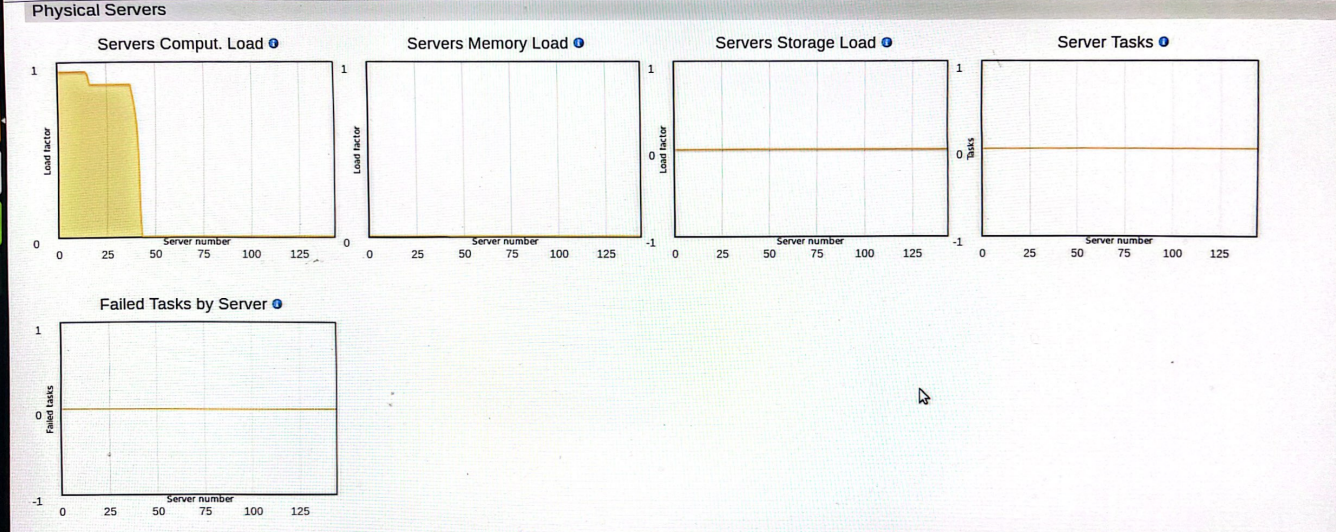
AFV is the average fitness value for all tasks i is the index of the task

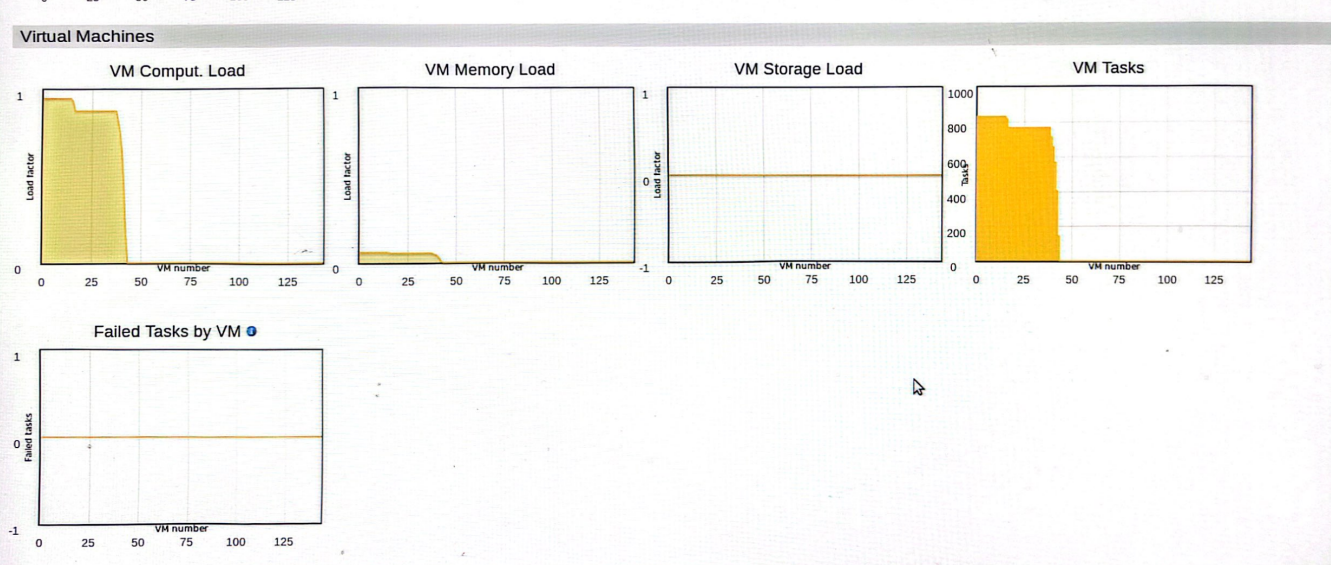
m is the total number of tasks in the simulation

F (i) is the value of fitness function for all solutions

* + - * **Processing Time (PT)**: the processing time is the time taken to run a specific algorithm. In the SLA negotiation, it is called negotiation time and in VM migration, it is called migration time.

****

****

****

****اي شي ممكن يرسم للتوضيح ياريت تظيفة استاذ