[1]Vipin Kumar Jaiswal

[2]Jameel Ahmad

[3]Kiran Deep Singh

Leveraging Fog Computing for Virtual Machine Placement in the Cloud Computing Environment

Journal of Electrical Systems

Abstract: - In the era of cloud computing, the allocation of virtual machines to physical machines within a data center is a critical issue that involves the decision-making process of determining the optimal strategy. It involves optimizing resource usage of CPU and RAM, reducing latency, reducing energy consumption, and ensuring the completion of tasks. Fog computing addresses these challenges by extending processing and storage resources to the edge of the network. Similar to the cloud, it may employ virtual machines for efficient resource utilization. In this paper, the capability of fog computing is leveraged for efficient virtual machine placement. A comprehensive solution is presented for optimum resource utilization in the fog/cloud environment. iFogSim toolkit is used that address the challenges of reducing latency, increasing energy consumption, and increasing resource utilization in fog/cloud computing environments. Results show the effectiveness of the solution by indicating reduced latency, efficient CPU utilization at about 67.5% with other important metrices.

Keywords: Cloud computing, fog computing, iFogSim, quality of service, virtual machine placement.

I. Introduction

Cloud computing and fog computing are two different models for storing, processing, and accessing data. In cloud computing, data is centralized and stored in a remote data centre. Fog computing represents a model where data processing is closer to the devices on the edge. The big difference between fog computing and cloud computing is the place where data is processed [1] [2]. Fog computing brings virtual machines close to the users and, therefore, effectively solves the challenges of latency and network resource congestion. Furthermore, it also supports dynamic virtual machine placement, which acquires new data for better performance and improved resource utilization [3] [4].

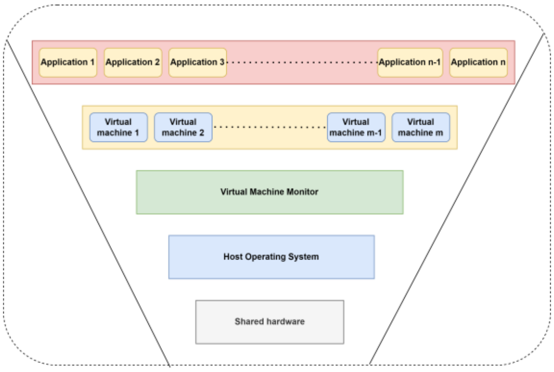

Fig. 1 Virtual Machine Placement in Cloud Computing Environment.

One of the fundamental technologies of the cloud computing environment is virtualization. A virtual machine (VM) shows the intended system, emulates a computer system, and facilitates the operation of an operating system. As shown in Figure 1, the software layer known as the hypervisor or VM Monitor (VMM) sits between the hard-ware platform and the virtual machines (VMs) within a server or physical machine. The VM monitor often facilitates the creation, migration, and termination of virtual machines (VMs). Figure 2 depicts that VM's classification may depend on how much of the targeted machine's functionality it implements [5].

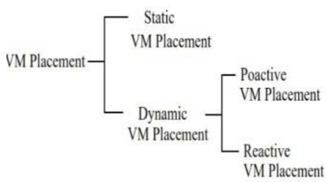

Fig. 2 Classification of Virtual Machines in the Cloud Environment.

[1] *Corresponding author: Department of Computer Science and Engineering, Integral University, Lucknow, Uttar Pradesh, India
[2]Department of Computer Science and Engineering, Integral University, Lucknow, Uttar Pradesh, India
[3]Department Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, Punjab, India

VM plays a very vital role in the improvement of resource utilization, the reduction of latency, the improvement of the quality of services, and the reduction of energy consumption. VM registration is a critical process that selects the most suitable jobs for hosting it to optimize energy consumption, improve resource utilization, and ensure that the support brings quality services. Fog integration reduces the load on data centres, increases latency by processing data closer to the centre, and enhances the workload for resources by management. Resource usage, latency, and power consumption must be balanced with traditional virtual machine deployment to view performance. It is also possible that system optimization might be a problem because of changes in operations for predicting future needs. Besides, fog computing enhances the time it takes to access data and make decisions and ensures security by reducing the chances that information can be interfered with while in transmission [6]. Tremendous development has gone on in process management, measures for protection, and integration with IoT devices [7] [8].

The problem of VM placement is a challenging issue that has a remarkable bearing on the rendering of cloud services. It involves finding the best placement of virtual machines in the data centres where the violation of service level agreements is minimal and energy consumption is low [9]. The services are required to provide a quick response time for critical applications; hence, it is necessary to place the virtual machines effectively, supporting the quality of service stringently. Many researchers have contributed towards utilization of resources in VM placement [10] [11]. Table 1 shows the summary of contributions by different authors.

II. VM Placement Strategies

There are different algorithms available for optimizing VM placement in cloud computing, by utilizing fog computing.

A. Genetic Algorithm (GA)

Genetic algorithms (GA) have proven to be highly effective in optimizing VM placement in cloud environments. Natural selection serves as the inspiration for the Genetic Algorithm (GA), a type of search heuristic. It is often employed to produce top notch solutions for optimization and search problems. They imitate the process of biological evolution to discover the most effective solutions. Assume that each VM placement scenario is a chromosome, with genes denoting which VM is placed on which physical machine. When evaluating these placements, the GA takes into account objectives such as resource utilization, power consumption, and latency.

Initialization is done by generating an initial population of possible VM placements. The Fitness Function evaluate the fitness of each individual in the population. The fitness function f can be defined in equation 1 below:

$$f(x) = \frac{1}{latency(x) + energy(x)}$$

where x represents a VM placement, and latency(x) and energy(x) are the latency and energy consumption for placement x, respectively.

Table 1 Summary of Contributions by authors.

Authors	**Methodology/ Technique**	**Key Findings/Contributions**	**Limitations**
Beloglazov and Buyya et al. [12]	Dynamic VM Consolidation	Proposed dynamic VM consolidation based on bin packing with custom modifications.	Limited comparison with existing algorithms
Basmadjian et al. [13]	Power Consumption Prediction Models	Presented prediction models for power consumption in data centres.	Lack of extensive empirical validation
Benbachir et al. [14]	Task Preemption Analysis	Studied task preemption across virtual machines and proposed trace synchronization method.	Complexity of trace synchronization
Dong et al. [15]	Task Scheduling Algorithm (MESF)	Proposed MESF algorithm for energy efficient task scheduling in cloud data centres.	Limited evaluation on real world datasets
Panigrahy et al. [16]	Virtual Machine Request Queue Optimization	Proposed geometric heuristics for VM request queue optimization.	Limited scalability
Kangkang et al. [17]	VM Placement based on	Emphasized on VM placement using multiple multidimensional knapsack problems.	Complexity of optimization

	Multidimensional Knapsack Problem		
Khanna et al. [18]	Dynamically Managed Algorithm	Proposed dynamically managed algorithm for VM consolidation to optimize various parameters.	Limited scalability in large scale environments
Wang et al. [19]	Gradient Search and Bin Packing	Utilized gradient search and bin packing for dynamic VM consolidation in multi tier web applications.	Limited applicability to single web application setups
Tripathi et al. [20]	Linear Programming Formulations	Used linear programming formulations for static and dynamic server consolidation problems.	Complexity of LP formulations
Prabha et al. [21]	Dynamic Consolidation with Migration Control	Heuristics and LP formulation for VM migration control are proposed.	Limited empirical validation
Gharehpasha et al. [22]	Whale Optimization Algorithm	In cloud data centers, the proposed method increases resource usage while consuming less energy.	Security and privacy concerns need to be resolved in order to take full advantage of virtualization's environment. SLAs, cloud work scheduling, and using several cloud platforms are important factors to consider.
Dubey et al. [23]	VM allocation algorithm based on water drop method	Better resource utilization, reduced energy usage, increased efficiency, and better overall performance	It is necessary to assess the quality of service offered by the suggested approach for various SLAs and virtual machine instances utilizing meta heuristic based optimization approaches.
Mejahed et al. [24]	Hybrid approach based on particle swarm optimization and flower pollination optimization with levy flight	The recommended method fared better in the simulated studies than the best fit bin packing strategy.	In addition to load balancing, live migration, and cost minimization, other considerations must be made while placing virtual machines.
Nikzad et al. [25]	Mulit-objective ant colony optimization	The proposed technique reduces energy consumption by 10.3%, SLA violation by 5.3%, and VM migration by 12.5%.	It has to be tested in a real world cloud context with open source tools like OpenStack.
Radi et al. [26]	VM consolidation method using modified	The analysis's findings indicate that the suggested strategy performs better in terms of energy use, SLA violations, and total number of virtual machine migrations.	It solely highlights CPU intensive workloads. The proposed technique has to be tested in a real world cloud setting.
Barthwal et al. [27]	Ant Colony Optimization	The recommended approach uses less energy while still adhering to SLAs than the power aware best fit solution.	By utilizing more resources, such memory, storage, and bandwidth, performance can be further enhanced.
Ibrahim et al. [28]	Efficient Adaptive Migration Algorithm	In terms of resource consumption, shut down hosts, migrations, and SLA breaches, the recommended method performs better.	A load balanced, resource aware virtual machine migration strategy is needed to maximize resource utilization and decrease SLA violations.

Mosa et al. [29]	Genetic Algorithm	Genetic algorithms reduce over and under utilization of CPU and memory.	More evaluations based on real workload traces ought to be conducted.
Tchana et al. [30]	Dynamic Software Consolidation	Around 40% less energy was used in private clouds, and roughly 40.5% less amount was spent on virtual machines on Amazon EC2.	To further maximize power gains, coordination is required between the consolidation of software on VMs and the consolidation of VMs on PM.
Shafiq et al. [31]	Load Balancing Algorithm	Implemented the least loaded algorithm, load balancing, and FFD comparison.	Low performance on servers that are heavily used

Only the most exceptional placements are carefully chosen for breeding. During this process of ”breeding” VMs are exchanged between placements, similar to how genes undergo crossover, to generate fresh possibilities for solutions. Finally, there is a small chance of random mutation that allows for the exploration of diverse VM configurations. Through this iterative process, the genetic algorithm gradually improves virtual machine placements to achieve the desired optimization goals.

B. Particle Swarm Optimization (PSO)

It is a highly effective technique for VM placement in cloud computing. It is a computational method that iteratively improves a candidate solution to optimize a problem based on a given measure of quality. It emulates the behavior of a flock of birds in search of sustenance. A swarm of particles with random positions and velocities is initiated and velocity is updated in equation 2 as:

$$vi(t+1) = wvi(t) + c1r1(pi * -xi(t)) + c2r2(g * -xi(t))$$

Here, $vi(t)$ is the velocity of particle i at time t, w is the inertia weight, c1 and c2 are cognitive and social coefficients, r1 and r2 are random numbers between 0 and 1, $pi *$ is the personal best position of particle i and g∗ is the global best position.

$$xi(t+1) = xi(t) + vi(t+1)$$

Each virtual machine acts as a “thread” in the PSO algorithm as above in equation 3. The algorithm takes resources, such as the virtual machine’s CPU, memory, and storage, according to the capabilities of the machine. Objects point to the search space, obtaining information from their best position (pbest) and the overall best position found by the entire population (gbest). Through the iterative process, the optimal placement of virtual machines, balance of performance metrics such as resource utilization, and reduced latency can be determined.

C. Simulated Annealing (SA)

It predicts the best global performance. It has proven to be very useful when dealing with various search engines. Firstly, start with an initial solution x and an initial temperature T and generate a neighbor solution x′ by making a small change to x. Now, the change in fitness is computed as Δf = f (x′) − f (x) with the new accepted solution with a probability in equation 4 below:

$$P(\triangle f, T) = \begin{cases} 1 & if \ \triangle f < 0 \\ \exp(-\frac{\triangle f}{T} & if \ \triangle f \geq 0 \end{cases}$$

Starting with an initial VM placement, SA then iteratively explores various configurations. It tolerates certain ”suboptimal” moves (such as swapping VM locations) in the early stages, resembling the initial high temperature. As the process cools down and decreases the acceptance rate for worse solutions, SA becomes more focused on finding better placements. It eventually converges on a good solution, although not necessarily perfect, for optimizing resource utilization, power consumption, or other goals.

D. Ant Colony Optimization (ACO)

A probabilistic approach called Ant Colony Optimization (ACO) is used to solve computational issues that boil down to choosing optimal routes across graphs. ACO simulates the behaviour of ants leaving pheromone trails to find the most efficient paths, drawing inspiration from real ant colonies. Firstly, initialize pheromone levels on all edges where each ant constructs a solution based on pheromone concentration and a heuristic value. The probability Pij of choosing path (i, j) is given in equation 5 as:

$$Pij = \frac{[\tau ij]^{\alpha}[\eta ij]^{\beta}}{\sum_{k\varepsilon allowed} [\tau ik]^{\alpha}[\eta ik]^{\beta}}$$

Here, τij is the pheromone level on edge (i, j) and ηij is the heuristic value (e.g., the inverse of distance or cost). Also, α and β are parameters controlling the influence of pheromone and heuristic values, respectively. Now, the pheromone levels on the edges are updated in equation 6 as:

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \sum_{ants} \triangle \tau_{ij}$$

where ρ is the evaporation rate, and Δτij is the amount of pheromone deposited by ants. VM placement involves the evaluation of resource suitability on physical machines through the use of ”trails”. VMs act as explorers, with more efficient placements leaving behind more potent pheromones. Over time, ACO converges on efficient VM placement that achieves a balance between CPU, memory, and network usage. Table 2 provides a detailed comparison of utilizing fog computing for effective VM placement in cloud computing environments.

Table 2 Comparison of VM Placement Algorithms

Algorithm	Approach	Complexity	Strengths	Weaknesses
Genetic Algorithm (GA)	Evolutionary	O(N · G)	Finds near-optimal solutions, parallelism	Can converge prematurely
Particle Swarm Optimization (PSO)	Swarm Intelligence Stochastic Optimization	$O(N \cdot T)$	Simple, few parameters	May get stuck in local minima
Simulated Annealing (SA)	Probabilistic, Heuristic, and Sequential Stochastic Search	$O(K)$	Avoids local minima, flexible	Slow convergence, parameter tuning
Ant Colony Optimization (ACO)	Nature Inspired Probabilistic Search	$O(N^2 \cdot T)$	Effective for discrete optimization	Computationally expensive

III. Leveraging Fog Computing for VM Placement

The proposed VM manager aims to integrate and optimize virtual machines by leveraging a three tier architecture (cloud, cloud, edge) to optimize latency, energy efficiency, and resource utilization. Integrating resource monitoring, intelligent placement algorithms, load balancing, and migration management, Fog Manager delivers optimal performance, reduces latency, and improves energy usage. It improves overall service quality and resource utilization, making it efficient for optimum deployment.

A. Resource Monitor

Resource Monitor gathers live data from every node and constantly monitors the utilization of resources such as CPU, memory, bandwidth, and energy consumption. It utilizes lightweight agents that are deployed on every node to send metrics back to a central monitoring server. Additionally, it monitors the accessibility and utilization of computational resources throughout the cloud, fog, and edge layers.

B. Placement Engine

It is responsible for determining the optimal placement of VMs based on current resource availability and application requirements. It uses heuristic or machine learning based algorithms to find the best placement for each VM for optimizing latency sensitivity, computational demand, data size, and network conditions. The optimization problem can be formulated in equation 7 as follows:

$$min \sum_{i\varepsilon V} \sum_{j\varepsilon N} (\alpha.latency_{ij} + \beta.energy_{ij} + \gamma.cost_{ij}).x_{ij}$$

Subject to:

$$\sum_{j\varepsilon N} x_{ij} = 1, \forall_i \varepsilon V$$

$$\sum_{i\varepsilon V} x_{ij}.resource\_demand_i \leq resource\_capacity_{j,} \forall_j \varepsilon N$$

Where:

- V is the set of VMs.
- N is the set of nodes (cloud, fog, edge).
- xij is a binary variable indicating if VM i is placed on node j.
- α, β, γ are weights for latency, energy, and cost.

C. Load Balancer

The load balancer ensures that workloads are evenly distributed across all tiers by adjusting resource allocation based on current load and estimated demand. It also balances traffic to avoid conflict and prevents someone from being overloaded. Finally, it optimizes performance by maximizing throughput and minimizing response time.

D. Policy Manager

The Policy Manager accommodates changes in the environment while enforcing regulations at all tiers. It establishes guidelines for security, resource allocation, and prioritization and ensures that all activities follow the established guidelines. Additionally, it enforces rules at all levels while adjusting for environmental changes.

E. Migration Manager

The migration manager manages the migration of virtual machines between nodes to maintain performance and resource usage. It performs three important tasks. First, it deals with the prerequisites that are responsible for starting the virtual machine migration. Second, it enables instant migration of virtual machines with minimal impact on services. Finally, it performs a cost effectiveness analysis to evaluate the costs and benefits before the switch is made.

IV. Simulation Setup

iFogSim is an open source toolkit developed for modeling and simulating fog computing environments. It extends CloudSim, a widely used library for cloud computing simulation and resource management. It enables users to simulate various scenarios involving fog computing, edge computing, and IoT applications and supports the evaluation of resource management and scheduling policies across both edge and cloud resources. During the experiment, a Java based Integrated Development Environment (IDE) such as Eclipse is setup, where the toolkit comprises several packages containing Java code for different implementations related to fog computing, IoT, and edge computing. It has some key classes such as Fog Device, Sensor, Actuator, Tuple, Application, Monitoring Edge, and Resource Management Service that allow users to import cloud and fog components into the simulation workspace to assess parameters such as energy consumption, cost, and latency. Users can define the network topology, create edges between application modules to depict data dependencies and apply scheduling schemes for resource allocation.

Deploying virtual machines in the cloud is important for improving resource allocation, reducing latency, and improving overall performance. It can also simulate many ideas for placing virtual machines in a cloud environment that help to evaluate the impact on QoS parameters such as latency, power consumption, and resource usage. It helps to evaluate various algorithms and rules for virtual machine deployment and helps develop better and more efficient operations. Hence, by simulating different scenarios, iFogSim helps researchers and practitioners analyze the effects of virtual machine placement decisions on latency, energy consumption, and resource utilization in fog computing environments.

A. Experimental setup for Simulation

The simulation provides the effectiveness and efficiency of the proposed solution in addressing the computing challenges such as latency reduction, resource optimization, and task distribution.

Table 3 Diverse configuration of Fog Nodes.

Fog Nodes Configuration		
Fog Node	**CPU(Cores)**	**RAM(GB)**
Fog Node 1	4	16
Fog Node 2	8	32
Fog Node 3	2	8

A.1 Fog Nodes Configuration

Table 3 provides the configuration used for fog nodes that represent distributed computing resources, meaning these resources find themselves at the closer edge of the network where effective data processing is done and latencies are minimized. Here, three fog nodes are used for diversified settings that serve different categories of workloads in such a way as to achieve effective resource determination based on the requirements of tasks and closeness to final users. Fog Node 1 is just an essential computing resource to handle the different workloads, so it is configured with 4 CPUs, 16GB of RAM, and 500GB storage. Fog Node 2 is an advanced computing resource that is configured with 8 CPUs, 32GB of RAM, and 1TB of storage. Fog Node 3 is an accessible computing resource to execute the low-resources-demanding task or as a backup node with 2 CPUs, 8GB RAM, and 250GB storage.

A.2 VMs Configuration

Similarly, virtual machines emulate physical computers and provide a flexible and scalable platform for deploying applications and services. Table 4 shows different VM configurations that make good use of fog node resources and meet the needs of many different applications. This makes the fog computing environment more flexible and scalable. VM 1 and VM 4 have been configured to work with moderate workloads with moderate resource requirements. The number of CPUs allocated to them is 2, and the RAM allocated is 4GB with 50GB storage. Importantly, VM 3 is geared for resources to be used in computationally intensive workloads or requiring higher system processing power. Computation resources were increased to 4 CPUs, 8GB RAM, and 100GB storage. Finally, VM 2 and VM 5 are light VMs, which should be recommended for light loads or when resource savings is a goal; these are set with low resource settings: 1 CPU, 2 GB RAM, and 20 GB storage.

Table 4 Diverse Configuration of VMs

VMs Configuration		
VM	**CPU (Cores)**	**RAM (GB)**
VM 1	2	4
VM 2	1	2
VM 3	4	8
VM 4	2	4
VM 5	1	2

A.3 Tasks Configuration

Table 5 represents computational workloads or processes that are executed within the fog computing environment for various tasks Task 1 and Task 4: These tasks require low latency, and the data is of small size, so they are apt for real time analytics and video streaming; Task 3: These represent cumbersome computation applications such that some flexibility in latency requirements can still be compromised; Task 2 and Task 5: These are medium latency applications with medium sized data. These two may correspond to processing and aggregation applications for data from IoT gadgets.

Table 5 Diverse Configuration of Tasks.

Tasks Configuration			
Task	**Length**	**File Size**	**Output Size**
Task 1	40000	300	300
Task 2	20000	200	200
Task 3	80000	400	400
Task 4	10000	100	100
Task 5	50000	250	250

V. Results and Findings

During the tests, the simulator acts like a real-life deployment environment where fog nodes, VMs, and tasks are very different from one another. This lets us see how well and quickly fog computing solutions work at solving modern computing problems like lowering latency, making the best use of resources, and distributing tasks evenly.

Results show that fog computing outperforms the conventional cloud computing paradigm in reducing latencies, energy efficiencies, and resource utilization, and improving availabilities and task completion rates. However, fog computing maintains similar ideologies while allowing for better performance, reliability, and responsiveness by using distributed infrastructures close to the end devices, which makes it a bright paradigm for a wide range of applications in edge computing.

A. Average Latency

Figure 3(a) describes the average latency of the tasks. A lower average latency in fog computing indicates faster processing of tasks because the tasks processed at the proximity of edge devices are supposed to have a lower latency. Thus, fog nodes reduces latency compared to the traditional cloud.

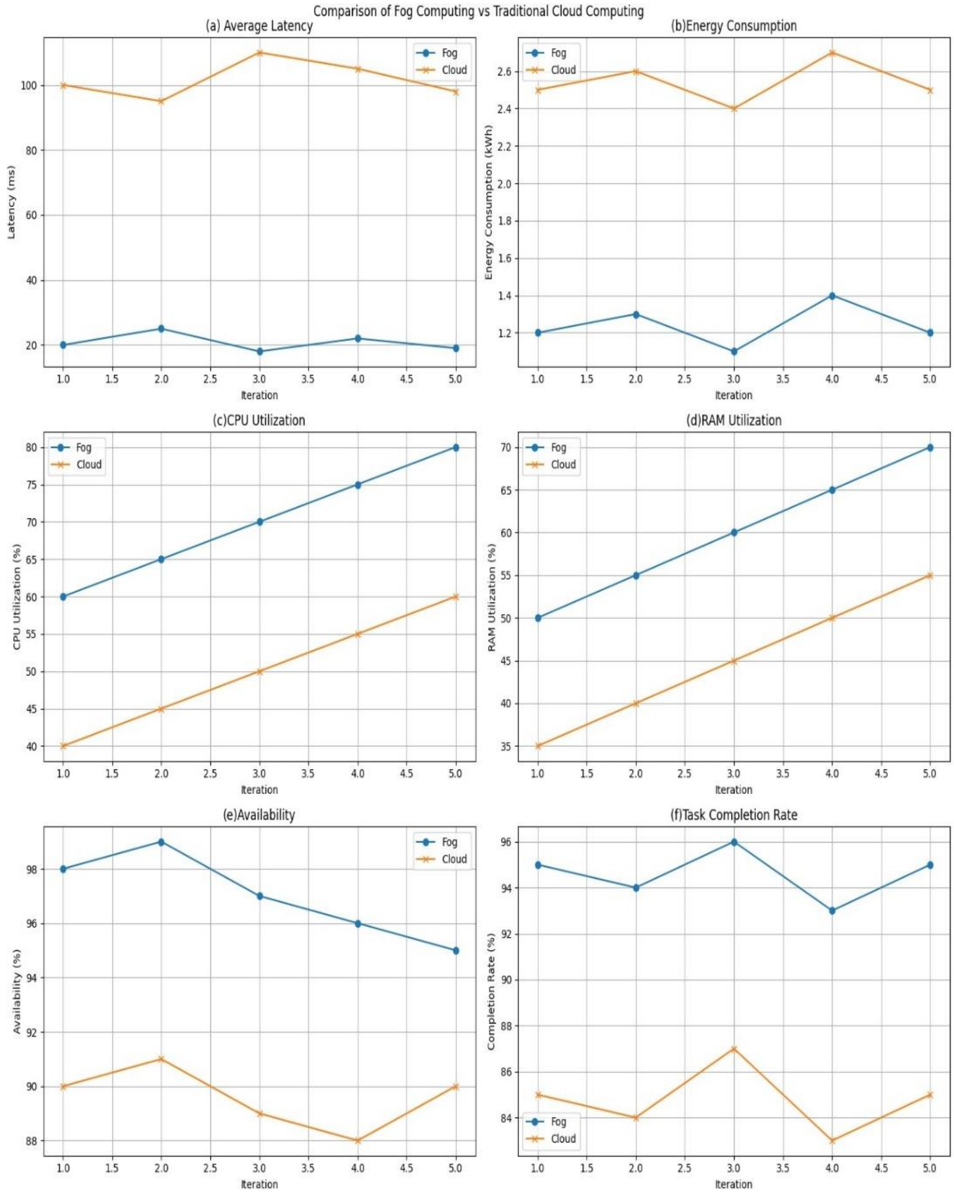

Fig. 3 Comparing results of Fog Computing vs Traditional Cloud Computing

B. Energy Consumption

Figure 3(b) indicates the level of consumption in terms of energy. These local processing capabilities decrease the need to move data to cloud data centres far away from major population centres energy is saved.

C. CPU Utilization

Figure 3(c) represents CPU utilization by fog nodes. High CPU utilization indicates high resource use and, by extension, the efficiency of task processing. Results shows high CPU utilization at about 67.5%, indicating that the fog nodes get utilized better to process tasks near the edge. This means that the fog nodes are effectively managing computational workloads, which is outperforming cloud computing, in which the data centre may be underused or overused.

D. RAM Utilization

Figure 3(d) shows that the higher RAM utilization indicates effective memory management and optimal execution of tasks. Here fog computing has more RAM usage than traditional cloud computing, thus showing that quite a good utilization of available memory resources to process the tasks has taken place in fog nodes.

E. RAM Utilization

Figure 3(e) shows the availability of VMs in both computing environments against various iterations. Higher availability implies better reliability and uptime of services. Here, fog computing presents higher availability, which may insinuate that VMs hosted at fog nodes are more reliable and accessible when compared to VMs hosted at centralized cloud data centres. Thus, locating computing resources near the edge device will ensure there is no interruption of service, and therefore, network failures or disruptions will not matter.

F. Completion Rate

Figure 3(f) depicts the completion rate of tasks executed in different computing environments. The higher the completion rate, the better the efficiency of task execution and allocation of resources. It is clear that fog computing presents a higher task completion rate, which is a good indicator that tasks are being performed efficiently and reliably compared to traditional cloud computing. It proves how fog computing sustains stringent latency requirements for the timely execution of tasks, improving user satisfaction and system performance.

VI. CONCLUSION

Cloud computing is an emerging computing platform that presents a virtual resource pool by centralizing computing resources connected to a network. Virtualization is one of the necessary technologies that make cloud computing possible. In this paper, fog computing is used to improve the VM placement strategies that efficiently reduce the VM traffic in the data centre and optimize resource utilization. The experimental setup utilized ifogSim to examine performance and resource utilization. The results validated the simulation of a real world deployment scenario and showed the efficiencies of solving modern computing problems by optimizing latency, energy consumption, resources, and performance.

REFERENCES

[1] Fatima, S., & Ahmad, S. (2019). An exhaustive review on security issues in cloud computing. KSII Transactions on Internet and Information Systems (TIIS), 13(6), 3219-3237, doi: 10.3837/tiis.2019.06.025.

[2] Khan, A. M., Ahmad, S., & Haroon, M. (2015, April). A comparative study of trends in security in cloud computing. In 2015 Fifth International Conference on Communication Systems and Network Technologies (pp. 586-590). IEEE.

[3] Ahmad, S., Beg, M. R., Ahmad, J., & Barua, N. (2010). Meet In The Middle Attack: A Cryptanalysis Approach. International Journal of Computer Applications, 975(1), 1-7, doi: 10.5120/467-772.

[4] Daman, R., & Tripathi, M. M. (2015). Encryption tools for secured health data in public cloud. International Journal of Innovative Science, Engineering & Technology, 2(11), 843-848.

[5] Sood, S. K., & Mahajan, I. (2018). Fog-cloud based cyber-physical system for distinguishing, detecting and preventing mosquito borne diseases. Future Generation Computer Systems, 88, 764-775, doi: https://doi.org/10.1016/j.future.2018.01.008.

[6] Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012, August). Fog computing and its role in the internet of things. In Proceedings of the first edition of the MCC workshop on Mobile cloud computing (pp. 13-16), doi:https://doi.org/10.1145/2342509.2342513.

[7] Khattar, N., Sidhu, J., & Singh, J. (2019). Toward energy-efficient cloud computing: a survey of dynamic power management and heuristics-based optimization techniques. The Journal of Supercomputing, 75, 4750-4810, doi: https://doi.org/10.1007/s11227-019-02764-2.

[8] Rani, S., Koundal, D., Kavita, F., Ijaz, M. F., Elhoseny, M., & Alghamdi, M. I. (2021). An optimized framework for WSN routing in the context of industry 4.0. Sensors, 21(19), 6474, doi: https://doi.org/10.3390/s21196474.

[9] Raman, C. J., & James, V. (2019). FCC: Fast congestion control scheme for wireless sensor networks using hybrid optimal routing algorithm. Cluster Computing, 22(Suppl 5), 12701-12711, doi: 10.1007/s10586-018-1744-8.

[10] Deep Singh, K., & Sood, S. K. (2020). 5G ready optical fog-assisted cyber-physical system for IoT applications. IET Cyber-Physical Systems: Theory & Applications, 5(2), 137-144, doi: https://doi.org/10.1049/iet-cps.2019.0037.

[11] Sood, S. K., & Deep Singh, K. (2023). Hmm-based secure framework for optical fog devices in the optical fog/cloud network. Journal of Optical Communications, 44(4), 475-483, doi: https://doi.org/10.1515/joc-2019-0155.

[12] Beloglazov, A., & Buyya, R. (2015). OpenStack Neat: a framework for dynamic and energy-efficient consolidation of virtual machines in OpenStack clouds. Concurrency and Computation: Practice and Experience, 27(5), 1310-1333, doi: https://doi.org/10.1002/cpe.3314.

[13] Jin, C., Bai, X., Yang, C., Mao, W., & Xu, X. (2020). A review of power consumption models of servers in data centers. applied energy, 265, 114806, doi: 10.1016/j.apenergy.2020.114806.

[14] Benbachir, A., & Dagenais, M. (2018). Hypertracing: Tracing through virtualization layers. IEEE Transactions on Cloud Computing, 9(2), 654-669, doi: 10.1109/TCC.2018.2874641.

[15] Dong, T., Xue, F., Xiao, C., & Li, J. (2020). Task scheduling based on deep reinforcement learning in a cloud manufacturing environment. Concurrency and Computation: Practice and Experience, 32(11), e5654, https://doi.org/10.1002/cpe.5654.

[16] Parida, S., Pati, B., Nayak, S. C., & Panigrahi, C. R. (2020). Offer based auction mechanism for virtual machine allocation in cloud environment. In Advanced Computing and Intelligent Engineering: Proceedings of ICACIE 2018, Volume 2 (pp. 339-351). Springer Singapore, doi: 10.1007/978-981-15-1483-8_29.

[17] Chowdhury, M. R., Mahmud, M. R., & Rahman, R. M. (2015). Implementation and performance analysis of various VM placement strategies in CloudSim. Journal of Cloud Computing, 4, 1-21, doi: https://doi.org/10.1186/s13677-015-0045-5.

[18] Khanna, G., Soh, S., Chaturvedi, S. K., & Chin, K. W. (2020). On enumeration of spanning arborescences and reliability for network broadcast in fixed-schedule dynamic networks. IEEE Transactions on Network Science and Engineering, 7(4), 2980-2996, doi: 10.1109/TNSE.2020.3008678.

[19] Wang, P., Nagrecha, K., & Vasconcelos, N. (2021). Gradient-based algorithms for machine teaching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 1387-1396).

[20] Tripathi, A., Pathak, I., & Vidyarthi, D. P. (2020). Modified dragonfly algorithm for optimal virtual machine placement in cloud computing. Journal of Network and Systems Management, 28, 1316-1342, doi: https://doi.org/10.1007/s10922-020-09538-9.

[21] Prabha, B., Ramesh, K., & Renjith, P. N. (2021). A review on dynamic virtual machine consolidation approaches for energy-efficient cloud data centers. Data Intelligence and Cognitive Informatics: Proceedings of ICDICI 2020, 761-780, doi: 10.1007/978-981-15-8530-2_60.

[22] Gharehpasha, S., Masdari, M., & Jafarian, A. (2021). Virtual machine placement in cloud data centers using a hybrid multi-verse optimization algorithm. Artificial Intelligence Review, 54(3), 2221-2257 doi: 10.1007/s10462-020-09903-9.

[23] Dubey, K., & Sharma, S. C. (2022). An extended intelligent water drop approach for efficient VM allocation in secure cloud computing framework. Journal of King Saud University-Computer and Information Sciences, 34(7), 3948-3958, doi: 10.1016/j.jksuci.2020.11.001.

[24] Mejahed, S., & Elshrkawey, M. (2022). A multi-objective algorithm for virtual machine placement in cloud environments using a hybrid of particle swarm optimization and flower pollination optimization. PeerJ Computer Science, 8, e834, doi: 10.7717/PEERJ-CS.834.

[25] Nikzad, B., Barzegar, B., & Motameni, H. (2022). Sla-aware and energy-efficient virtual machine placement and consolidation in heterogeneous DVFS enabled cloud datacenter. IEEE Access, 10, 81787-81804, doi: 10.1109/ACCESS.2022.3196240.

[26] Radi, M., Alwan, A. A., & Gulzar, Y. (2023). Genetic-based virtual machines consolidation strategy with efficient energy consumption in cloud environment. IEEE Access, 11, 48022-48032, doi:10.1109/ACCESS.2023.3276292.

[27] Barthwal, V., & Rauthan, M. M. S. (2021). AntPu: a meta-heuristic approach for energy-efficient and SLA aware management of virtual machines in cloud computing. Memetic Computing, 13, 91-110, doi: 10.1007/s12293-020-00320-7.

[28] Ibrahim, M., Imran, M., Jamil, F., Lee, Y. J., & Kim, D. H. (2021). EAMA: Efficient adaptive migration algorithm for cloud data centers (CDCs). Symmetry, 13(4), 690, doi: 10.3390/sym13040690.

[29] Mosa, A., & Sakellariou, R. (2019, July). Dynamic virtual machine placement considering CPU and memory resource requirements. In 2019 IEEE 12th international conference on cloud computing (CLOUD) (pp. 196-198). IEEE, doi: 10.1109/CLOUD.2019.00042.

[30] Tchana, A., De Palma, N., Safieddine, I., & Hagimont, D. (2016). Software consolidation as an efficient energy and cost saving solution. Future Generation Computer Systems, 58, 1-12, doi:10.1016/j.future.2015.11.027.

[31] Shafiq, D. A., Jhanjhi, N. Z., & Abdullah, A. (2022). Load balancing techniques in cloud computing environment: A review. Journal of King Saud University-Computer and Information Sciences, 34(7), 3910-3933, doi:10.1016/j.jksuci.2021.02.007.