RESEARCH ARTICLE

A Harris Hawk Optimisation system for energy and resource efficient virtual machine placement in cloud data centers

Madhusudhan H. S.[1], Satish Kumar T.[2], Punit Gupta[3]*, Gavin McArdle[3]

1 Department of Computer Science & Engineering, Vidyavardhaka College of Engineering, Mysuru, Karnataka, India, **2** Department of Computer Science & Engineering, BMS Institute of Technology & Management, Bengaluru, Karnataka, India, **3** School of Computer Science, University College Dublin, Dublin, Ireland

* punit.gupta@ucd.ie

Abstract

Virtualisation is a major technology in cloud computing for optimising the cloud data centre's power usage. In the current scenario, most of the services are migrated to the cloud, putting more load on the cloud data centres. As a result, the data center's size expands resulting in increased energy usage. To address this problem, a resource allocation optimisation method that is both efficient and effective is necessary. The optimal utilisation of cloud infrastructure and optimisation algorithms plays a vital role. The cloud resources rely on the allocation policy of the virtual machine on cloud resources. A virtual machine placement technique, based on the Harris Hawk Optimisation (HHO) model for the cloud data centre is presented in this paper. The proposed HHO model aims to find the best place for virtual machines on suitable hosts with the least load and power consumption. PlanetLab's real-time workload traces are used for performance evaluation with existing PSO (Particle Swarm Optimisation) and PABFD (Best Fit Decreasing). The performance evaluation of the proposed method is done using power consumption, SLA, CPU utilisation, RAM utilisation, Execution time (ms) and the number of VM migrations. The performance evaluation is done using two simulation scenarios with scaling workload in scenario 1 and increasing resources for the virtual machine to study the performance in underloaded and overloaded conditions. Experimental results show that the proposed HHO algorithm improved execution time(ms) by 4%, had a 27% reduction in power consumption, a 16% reduction in SLA violation and an increase in resource utilisation by 17%. The HHO algorithm is also effective in handling dynamic and uncertain environments, making it suitable for real-world cloud infrastructures.

Citation: H. S. M, T. SK, Gupta P, McArdle G (2023) A Harris Hawk Optimisation system for energy and resource efficient virtual machine placement in cloud data centers. PLoS ONE 18(8): e0289156. https://doi.org/10.1371/journal.pone.0289156

Editor: Ali Safaa Sadiq, Nottingham Trent University School of Science and Technology, UNITED KINGDOM

Received: November 17, 2022

Accepted: July 12, 2023

Published: August 11, 2023

Data Availability Statement: https://www.cs.huji.ac.il/labs/parallel/workload/ The complete simulation data and sample of dataset is shared in the supporting information section. Results: Energy Consumption (kwh) HHO PSO PABFD W1 101.99 110.26 128.94 W2 117.3 126.42 184.28 W3 118.73 130.32 213.4 W4 114.08 129.56 145.6 W5 105.75 117.26 121.28 CPU Utilization HHO PSO PABFD W1 74.13 71.45 62.23 W2 75.35 73.28 65.3 W3 77.25 74.21 66.78 W4 73.09 73.54 64.78 W5 74.61 72.87 64.21 VM Migrations HHO

Introduction

Cloud computing is a paradigm for providing on-demand computational services and resources through the internet, such as data storage and computing power [1]. Cloud computing offers customers on-demand resources in the form of virtual machines (VMs) and

PSO PABFD W1 18590 21457 22100 W2 24109 25632 26234 W3 25696 26543 28256 W4 24427 25387 26678 W5 24145 25286 25875 SLA violations HHO PSO PABFD W1 14.58 15.02 17.3 W2 15.24 17.21 18.25 W3 13.71 16.54 18.43 W4 15.03 16.85 17.76 W5 16.07 17.65 17.42 Sample dataset:; MaxJobs: 76872; MaxRecords: 76872; Preemption: No; UnixStartTime: 788722174; TimeZone: 0; TimeZoneString: US/Pacific; StartTime: Thu Dec 29 09:29:34 PST 1994; EndTime: Sat Dec 30 23:54:09 PST 1995; MaxNodes: 400 (48 interactive, 352 batch, 6 service, 10 I/O); MaxProcs: 400; Note: service and I/O partitions are not used to run jobs; MaxQueues: 37; Job Number – a counter field, starting from 1.; Submit Time – in seconds. The earliest time the log refers to is zero, and is usually the submittal time of the first job. The lines in the log are sorted by ascending submittal times. It makes sense for jobs to also be numbered in this order.; Wait Time – in seconds. The difference between the job's submit time and the time at which it actually began to run. Naturally, this is only relevant to real logs, not to models.; Run Time – in seconds. The wall clock time the job was running (end time minus start time).; Number of Allocated Processors – an integer. In most cases this is also the number of processors the job uses; if the job does not use all of them, we typically don't know about it.; Average CPU Time Used – both user and system, in seconds. This is the average over all processors of the CPU time used, and may therefore be smaller than the wall clock runtime. If a log contains the total CPU time used by all the processors, it is divided by the number of allocated processors to derive the average.; Used Memory – in kilobytes. This is again the average per processor.; User ID – a natural number, between one and the number of different users.; Group ID – a natural number, between one and the number of different groups. Some systems control resource usage by groups rather than by individual users.; Executable (Application) Number – a natural number, between one and the number of different applications appearing in the workload. in some logs, this might represent a script file used to run jobs rather than the executable directly; this should be noted in a header comment.; Queue Number – a natural number, between one and the number of different queues in the system. The nature of the system's queues should be explained in a header comment. This field is where batch and interactive jobs should be differentiated: we suggest the convention of denoting interactive jobs by 0.; Partition Number – a natural number, between one and the number of different partitions in the systems. The nature of the system's partitions

accomplishes their tasks while meeting Quality of Service (QoS) requirements. Each VM is designed to target a certain computing resource capability (e.g., the number of CPUs, I/O bandwidth and memory capacity). Using a physical machine (PM) or host to run several VMs, Virtualisation technology increases a data centre's energy efficiency by decreasing the amount of hardware in use and increasing the resource usage of physical machines. Cloud providers need to schedule the virtual machines to suitable physical machines so that both users' and the providers' objectives will be optimised.

The notion of cloud data centres comes from the fact that cloud computing makes use of data centre infrastructure to offer services. Cloud data centres will manage 94% of workloads by 2021 [2]. However, the operations of these data centres require a lot of energy. Energy expenses account for about 42% of total operational costs, according to Amazon's data centre research [2]. Another reason to save energy is the ongoing discussion about climate change. Running servers is projected to produce 0.5% of world CO_2 emissions [3]. As a result, lowering data centre energy usage without sacrificing the QoS provided is an incipient research domain.

Large numbers of physical servers are commonly seen in data centres. The IT infrastructure, which is subjugated by PMs, accounts for about 60% of the overall energy usage in a data centre. Virtualisation is a technique that allows customers to access cloud computing resources in the form of several VMs. Since numerous VMs may be deployed to a similar physical server, Virtualisation is critical for attaining both energy efficiency and high server utilisation. Hence, employing an effective Virtual Machine Placement (VMP) method can have a significant impact on the power usage of a data centre. VMP is an NP-hard optimisation problem [4].

Virtual machines (VMs) share resources through Virtualisation on hosts to process user requests over physical machines (PMs). Virtualisation may be used to conduct three different operations: VM migration, VM isolation and VM consolidation. The virtual machine migration technology moves virtual machines from one PM to another. Virtual machines operating on separate hosts will leave that host and congregate on fewer ones during the VM consolidation process to save energy by turning off or transferring the initial running host to hibernate mode [5].

The issue of energy consumption has improved because of recent developments in hardware technology. It is still a major issue for sustainable computing though, because how computing and auxiliary hardware resources are used has a significant impact on how much energy is used by those resources. In contrast to resources that are employed effectively, under-utilisation or over-utilisation of the resources (CPU and RAM) results in increased energy consumption. This necessitates the creation of several software energy-saving strategies, such as scheduling and Virtualisation. With lower resource utilisation, the energy efficiency of the system will be lower. Additionally, there will be more active hosts.

A PM supplies all essential VM resources such as storage, network bandwidth, memory, and CPU as each PM can hold several VMs. Consolidation of virtual machines is a method for making intelligent and efficient use of the resources of the cloud. One of the most difficult components of VM consolidation is VM allocation. It is described as locating the best PM for a VM to decrease the number of operating physical machines in data centres. As a result, many goals have been proposed for improving load balancing, lowering costs and network usage, mitigating SLA (Service Level Agreement) violations, increasing energy efficiency, and maximising resource utilisation.

This work presents a Harris Hawk Optimisation (HHO) model for multi-objective virtual machine placement in the cloud data centre. The proposed HHO model aims to optimally place VMs on appropriate physical hosts. HHO is a meta-heuristic approach for determining the global ideal solution. The system model is depicted in Fig 1. A data centre is made up of multiple physical machines. Many virtual machines can run on a single physical machine.

should be explained in a header comment. For example, it is possible to use partition numbers to identify which machine in a cluster was used.; Preceding Job Number – this is the number of a previous job in the workload, such that the current job can only start after the termination of this preceding job. Together with the next field, this allows the workload to include feedback as described below.; Think Time from Preceding Job – this is the number of seconds that should elapse between the termination of the preceding job and the submittal of this one. 1 0 224904 37349 128 37349 -1 -1 -1 -1 1 7 -1 -1 29 2 -1 -1 2 4751 257510 43349 128 42924 -1 -1 -1 -1 1 7 -1 -1 29 2 -1 -1 3 91769 213864 22 128 -1 -1 -1 -1 -1 1 7 -1 -1 29 2 -1 -1 4 138658 86135 4138 8 4138 -1 -1 -1 -1 1 3 -1 -1 28 2 -1 -1 5 138682 86354 58 4 -1 -1 -1 -1 -1 1 1 -1 -1 2 2 -1 -1 6 140276 84762 681 1 -1 -1 -1 -1 -1 1 2 -1 -1 2 2 -1 -1 7 141888 265755 94 256 60.92 -1 -1 -1 -1 1 18 -1 -1 17 2 -1 -1 8 141902 265840 60 256 52.10 -1 -1 -1 -1 1 18 -1 -1 17 2 -1 -1 9 141918 265888 74 256 67.95 -1 -1 -1 -1 1 18 -1 -1 17 2 -1 -1 10 141934 265952 90 256 75.78 -1 -1 -1 -1 1 18 -1 -1 17 2 -1 -1 11 143227 264753 86 256 79.24 -1 -1 -1 -1 1 18 -1 -1 17 2 -1 -1 12 146537 261534 255 256 246.04 -1 -1 -1 -1 1 18 -1 -1 17 2 -1 -1 13 148219 76566 12717 64 12717 -1 -1 -1 -1 1 3 -1 -1 28 2 -1 -1 14 164187 64753 25646 8 25626 -1 -1 -1 -1 1 3 -1 -1 28 2 -1 -1 15 176763 60750 29196 16 29174 -1 -1 -1 -1 1 3 -1 -1 28 2 -1 -1

Funding: The author(s) received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

Virtual Machine Manager (VMM) also identified as Hypervisor, is software that makes it easier to create, manage, and monitor virtual machines. On top of physical hosts, it also controls a Virtualised environment. When the data center manager receives a request for VM execution it first gathers status information from all accessible physical machines and sends it to the VM scheduler. The HHO model was used to create the VM scheduler. The VM scheduler then evaluates the status information and assigns VMs to appropriate PMs.

Harris Hawk is a meta-heuristic approach for determining the global ideal solution. The significant contributions of the proposed work are listed below:

- A resource and energy-efficient VM deployment model for diverse cloud environments is proposed. This contribution intends to increase the resource utilisation and then the energy consumption can be minimised while satisfying the QoS expressed by the cloud providers.
- Load balancing is addressed by migrating VMs from overloaded to underloaded physical machines and vice versa.
- Reducing the running time of the VM Placement algorithm: The reduction in execution time required to process all the requests from users plays a vital role from the cloud provider's perspective since it directly affects the performance of the cloud provider.

The remainder of this article is structured as follows, a review of the existing techniques is presented in the literature review section. The problem formulation section discusses the problem and the proposed HHO model, while the experimental setup and comparisons section evaluates the proposed technique. Finally, the conclusion section summarises the paper with a discussion on future work.

Literature review

One of the challenges with cloud computing is VMP which has an impact on many aspects of cloud computing. As a result, several research efforts have been carried out to map the best position for VMs among accessible PMs. This section summarises pertinent studies on VMP and Table 1 depicts the studies and their parameters. Also, virtual machine placement technquies are categorized into Nature/Bio Inspired methods, Metaheuristic approaches and Machine Learning technquies.

Nature/bio inspired methods

The Salp Swarm Algorithm and the sine-cosine algorithm were combined to create a hybrid multiobjective VM placement technique [6]. The proposed technique aimed to reduce the mean time before host shutdown (MTBHS), the number of SLA violations and power consumption. The proposed method was compared to several meta-heuristics, and the findings showed that it was superior. When discussing VMs and PMs, the bandwidth has not been fully considered. Furthermore, the balanced use of multidimensional resources in physical hosts remains uncertain.

A bandwidth-aware VMP algorithm, developed on the enhanced Whale Optimisation Algorithm (WOA) and a novel bandwidth allocation methodology, was proposed in [7]. The outcomes reveal that the suggested method outperforms several meta-heuristics and heuristics. I suggested approach focuses solely on bandwidth optimisation, neglecting to consider other critical resources like memory and CPU use. Also, the study did not focus on the problem of optimising power consumption.

The authors in [8] proposed an energy-aware VM placement technique using Binary Particle Swarm Optimisation (BPSO) algorithm. The work is based on the modification of local

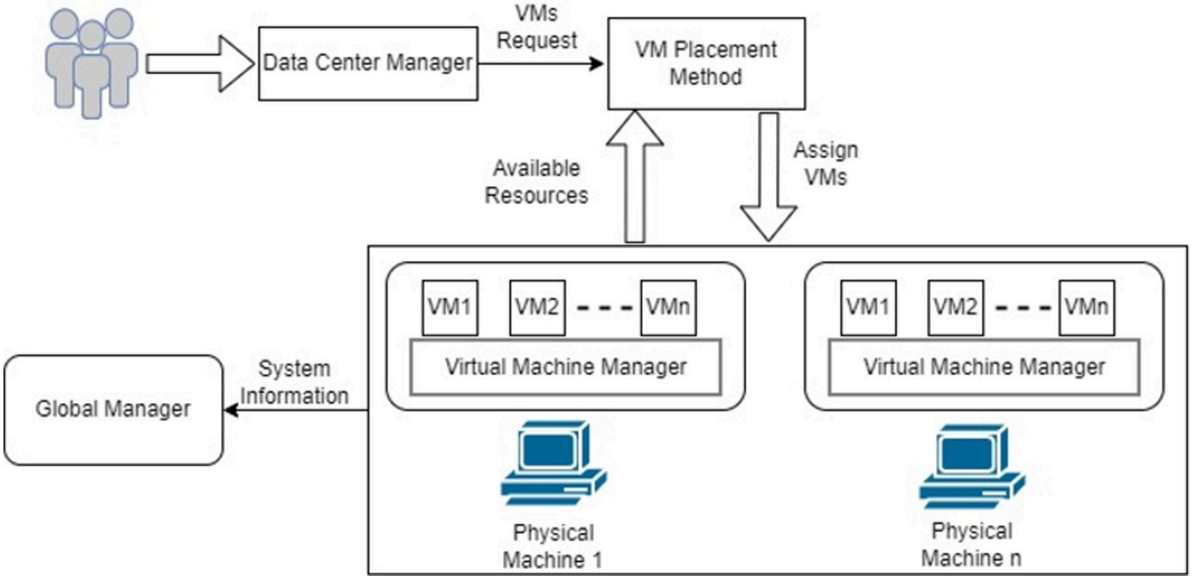

Fig 1. System design for VMP in the cloud.

https://doi.org/10.1371/journal.pone.0289156.g001

optimum placement and global optimal placement, to get optimal VM placement with the lowest energy usage.

To minimise energy consumption and fulfil uers' experience, an enhanced ant colony algorithm is used to propose an energy-saving VMP approach which attains a balance between user experience and energy consumption in data centres [9]. The original ant colony algorithm's pheromone and heuristic parameters were modified, ensuring that the improved algorithm may transition from a local to a globally optimal solution, avoiding the algorithm's early maturity. Dolphin Partner Optimisation along with optimised security for resource allocation, has been presented in [10]. Memory-aware Optimisation and Energy-based Prioritisation are utilised to pick memory and energy-established VMs for safety, and this work has also incorporated hypervisor safety into the two groups of VMs acquired. The Dolphin Partner Optimisation then enhances the two sets of virtual machines to provide the best capable VM for each set. Finally, streamlining security is applied to boost security, and the chosen virtual machine is essentially the most secure.

Authors in [11] proposed a research model that uses VM consolidation to minimise data centre power consumption while maintaining stable operation. An adaptive harmony search approach is created to achieve the best solution for the suggested VM consolidation model, which requires less effort to establish the model's parameters than current harmony search methods.

The authors in [12] presented an energy-efficient container placement using the WOA technique. Two stages of placement, that is placing containers on suitable VMs and mapping VMs to suitable PMs are solved as one optimisation problem. The proposed method is evaluated in a heterogeneous environment and results show, it minimises the power consumption, reduces the number of PMs used and maximises the resource utilisation but increases the number of migration increasing the cost.

Authors [13] developed a hybrid approach using PSO and Flower Pollination Optimisation techniques to reduce power consumption, placement time, and resource wastage and increase

Table 1. Classification of different approaches in the existing literature.

Reference	Algorithm Used	Parameters				Limitations
		Execution Time	Resource Utilisation	Energy Consumption	SLA Violation	
6	Salp Optimisation			✓	✓	Doesnot consider Resource Utilisation
7	Whale optimisation algorithm					Focused only on bandwidth efficiency, does not consider resource utilisation and energy consumption
8	PSO			✓		Doesnot consider resource utilisation, execution time and SLA violations.
9	ACO	✓		✓		No real-time dataset is used, and resource utilisation is not considered
10	Proportionate resource utilisation (PRU) based policy		✓	✓		Does not consider execution time and SLA violations
11	Dolphin partner optimisation		✓	✓	✓	Does not consider VM migrations
13	Original harmony search			✓	✓	Does not consider resource utilisation and execution time
14	Whale optimisation		✓	✓		Randomly generated data were used and does not consider SLA violation
15	HPSOLF-FPO		✓	✓		Randomly generated data were used and does not consider load balancing
18	ACLR			✓		Focused only on energy consumption
20	Firefly algorithm	✓				Resource and energy optimization was not considered
22	Q Learning			✓		Focused only on energy consumption
23	Flower pollination	✓	✓	✓		SLA violation is not considered
29	Artificial ant colony	✓	✓			The work aims to improve only Makespan and resource utilisation
30	Self-adaptive PSO	✓				The work aims to improve only the cost and execution time of the task.
31	Multi-agent system	✓				Focused on execution time and parallel resource utilisation. Energy consumotion is not considered
32	Double deep Q-network			✓		Only to improve power and network load
33	Flower Pollination Algorithm		✓	✓		Improve power efficiency of host and number of migration. Does not consider resource utilization
34	krill herd			✓	✓	Improves only power consumption and minimize SLA violation. Resource utilization is not considered.
35	Deep reinforcement learning algorithm	✓		✓		Does not cosider resource utilization and SLA violation
36	Ant lion optimizer and sine cosine algorithm		✓	✓	✓	Does not consider execution time
38	Symbiotic Organisms Search Algorithm		✓	✓		Doesnot consider execution time and SLA violation
39	Enahnced Cuckoo search algorithm			✓	✓	Does not consider resource utilization
40	Squirrel search algorithm	✓	✓	✓	✓	Migration cost is not taken into consideration
41	Clonal optimization	✓	✓			SLA and resource utilization is not taken into consideration
42	Hybrid BAT and ABC	✓				Doesnot consider Resource Utilisation and energy optimization
43	Jelly Fish	✓	✓			Energy and SLA is not considered in this work
46	Elephand herd	✓		✓		SLA and migration cost is not taken into consideration
47	Whale	✓	✓	✓		SLA and migration cost is not taken into consideration

(*Continued*)

Table 1. (Continued)

Reference	Algorithm Used	Parameters				Limitations
		Execution Time	Resource Utilisation	Energy Consumption	SLA Violation	
48	Ant Lion	✓		✓		Resource utilization and SLA is not considered
49	Buterfly optimization	✓	✓			Energy and SLA is not considered in this work
50	Gray Wolf	✓	✓	✓		SLA and migration cost is not taken into consideration

https://doi.org/10.1371/journal.pone.0289156.t001

server utilisation. Placements of the virtual machines onto physical machines are accomplished based on the fitness values derived from the above objectives.

Adlin Sheeba et al. proposed a VM placement technique using the Firefly Optimisation Technique [14]. In this work, the authors used the K-Means clustering technique to minimise the migration time of VMs. An enhanced Firefly Optimisation Algorithm was used to design the VMP model. To decide the optimal cluster for VMP, a combination of PSO and coyote algorithm was used.

In [15], the authors proposed a Water Wave Optimisation technique to handle virtual machine consolidation problems in the cloud. The proposed method is employed to find the proper migration plan to minimise the load on the overloaded hosts and maximise resource utilisation. In [16], a Flower Pollination-Based Nondominated Sorting Optimisation (FP-NSO) algorithm is presented to handle VM placement to minimise energy consumption and maximise resource utilisation. The method that aids in identifying the most suitable PMs for deploying VMs in a cloud environment is linked to many resource-constraint parameters.

In a recent work [17], the author has proposed a modified ant colony-based load balancing algorithm for cloud resource optimisation to improve makespan and resource utilisation in the cloud. Similar work using self-adaptive PSO (Particle Swarm Optimisation) [18] is proposed to improve cost using a combination of machine learning to predict the cost model and PSO for finding the best resource over the cloud. The work aims to improve the cost of the resources and execution time. In [19], a resource allocation algorithm is proposed using The Flower Pollination Algorithm to improve power efficiency as compared to a genetic algorithm in the cloud. This work also tried to reduce the number of migrations to improve resource utilisation (CPU and RAM utilisation). From the nature-inspired algorithms, the krill herd model has been proposed to improve SLA violation and energy efficiency in the cloud [20]. The work shows an improvement in SLA and power efficiency as compared to the genetic algorithm and the MBFD (Modified Best Fit Decreasing) algorithm. Authors in [21] proposed a hybrid approach for multi objective virtual machine placement in cloud. Ant lion optimization and sine cosine algorithm were used to optimally place VMs over suitable physical machines. Performance metrics like power consumption, resource wastage, reosuce utilization, number of active PMs, VM migrations and SLA were considered.

In [22], authors presented a Variable Neighborhood Search-Based Symbiotic Organisms Search Algorithm to enhance energy efficiency in cloud. Authors aimed to minimize energy consumption and maximize resource utilization. A minimum of active hosts and the energy-saving turnoff of inactive servers allowed for the best VM allocation. Esha Barlaskar et al., [23] proposed an enhanced cuckko search algorithm to obtain optimal solution for virtual machine placement in cloud. This work aims to reduce energy consumption, VM migrations and SLA violation. Hetergeonous hosts were used for experimentation work. In [24] authors has proposed an nature inspired squirel search optimization algorithm for resource scheduling is cloud. the work is compared with genetic algorithm, PSO and ACO using energy, cost and

SLA as performance parametrs. In [25] a clonal optimization model is proposed for resource allocation for cloud infrastructure to improve power efficiency in cloud. In recent years many other work are been proposed using nature inspired algorithms like work inspired from BAT algorithm [26], jelly fish [27], wild horse [28], FOX inspired model [29], Elephant herd [30], Whale Optimization Algorithm (WOA) [31], Ant Lion [32], Butterfly Optimization Algorithm (BOA) [33] and Gray Wolf Optimization (GWO) [34].

Metaheuristic approaches

A VM allocation policy has been proposed that assigns VMs to hosts proportionally based on their RAM and CPU use. It employs the idea of skewness to assess the unevenness in host resource utilisation and assigns VMs to the host machine with the lowest skew value [35].

In [36], a combination of a mixed integer linear program and a heuristic approach was proposed for virtual machine placements in edge-cloud computing. The objective is to meet the varied latency requirements of applications while minimising the consumption of IT infrastructures for the placement of VMs in cooperative edge-cloud computing. To defend against co-location assaults in IaaS (Infrastructure As A Service) cloud providers, the authors in [37] presented a VM allocation strategy which considers 3 different types of incoming virtual machines. The proposed algorithm focuses on the secure placement of VMs over physical machines. This work aims to minimise energy consumption.

In [38], the authors presented an open-source development model algorithm to address dynamic virtual machines' placement in the cloud. ODMA(Open Source Development Model Algorithm) is one of the meta-heuristic approaches that is used in this work to consolidate several VMs into a reduced number of hosts. The objectives of this work are to minimise the number of active hosts, achieve load balancing and improve performance. In [39], the author has proposed a multi-agent-based resource optimisation algorithm is proposed which aims to solve the optimisation problem using parallel scheduling and a multi-agent system. The work proposes a mathematical model to find an optimal solution in parallel resources. Canosa-Reyes et al., [40] proposed energy tradeoff consolidation with contention-aware resource provisioning, here authors used containers to optimally place the jobs. Cloudsim was used for experimention purpose. The proposed method reduces resource contention and makes job placement more efficient with the energy-utilization tradeoff.

Machine learning technquies

Ashawa et al. proposed the LSTM technique for load balancing to enhance cloud efficiency via resource allocation [41]. LSTM provided a dynamic resource allocation mechanism that evaluates the resource usage of an application using heuristics to determine the optimal additional resources to make available for that application. Based on the result, the proposed model shows the accuracy rate is enhanced by approximately 10–15%.

Ali Aghasi et al. employed the Q Learning technique to address virtual machine placement [42]. Reinforcement learning along with state action representation is utilised. The objectives of this work were to minimise energy consumption and reduce CPU temperature. In this generation of machine learning, various hybrid approaches have been proposed using machine learning and deep learning, like the Double Deep Q-network to improve network performance in cloud radio access networks [43]. This work [43] aims to improve performance by managing and optimising the load on network paths using Q-Network approaches. The result showcases an improvement in power consumption (Kwh) and network delay. Another work using deep learning was proposed in [44] to optimise energy efficiency and resource optimisation. This model tries to predict the best resource of a VM based on the prior performance in terms of CPU utilisation and power consumption. The work uses a deep reinforcement learning model for training and model prediction. The proposed model is compared with a greedy algorithm using power consumption and average waiting time as performance parameters.

Work has also been carried out in job scheduling. For example, Ibrahim Attiya et al. presented a hybrid job scheduling approach in cloud computing using a modified Harris Hawk Optimisation and simulated annealing algorithm [45]. This work aims to minimise the makespan and improve the convergence speed. Both standard and synthetic workloads were employed to analyze the performance of the this work.

Authors in [46] proposed a multi-objective task scheduling technique, based on Gaussian Cloud Whale Optimisation Algorithm (GCWOAS2) in cloud computing. A three-layer scheduling model was presented in this work. The goal is to reduce the operating cost of the system by minimising task completion time by effectively utilising virtual machine resources and maintaining the load balancing of each virtual machine. To develop the best scheduling scheme in the GCWOAS2 approach, an opposition-based learning mechanism is initially employed to establish the scheduling strategy. Then, to dynamically widen the search range, an adaptive mobility factor is provided. To improve the unpredictability of the search, a Whale Optimisation technique based on the Gaussian cloud model is presented.

To summarise, prior research shows that the meta-heuristic approaches listed above can identify an appropriate solution for VM scheduling in cloud computing. However, experiments were carried out using randomly generated data in some works and most of the work focused on two to three objectives without taking into account load balancing, SLA violation and execution time concurrently. The proposed work in this article focuses on multi-objective VM placement along with load balancing which was not addressed in the existing approaches.

Motivation

The motivation of this work is to develop a new meta-heuristic algorithm to achieve better performance in the field of cloud computing. Where existing work as shown in the literature work uses traditional algorithms, this work proposes the Harris Hawk Optimisation (HHO) model to improve the performance of the cloud environment in terms of power consumption and utilisation of the system. The existing models are being compared with our proposed model to study the performance.

Problem formulation

The cloud data centre in this work consists of N VM and K PMs. The resource requirements of VMs are CPU and RAM. The requirements of CPU and memory of $\mathrm{VM_i}$ are represented as VM_{cpu_i}. and VM_{ram_i} respectively. The CPU and memory capacity of PMj are represented as PM_{cpu_j} and PM_{ram_j} respectively. Table 2 depicts the terminologies used in this work.

Each PM has enough capacity in this cloud data centre to allocate a set of VMs. Let $r = (r_{pm}, pm \in PM)$ denote the VM placement approach satisfying the resource allocation policy is feasible i.e. resources allocated to every VM are fewer than the overall capacity of the PM as represented in Eq 1.

$$\sum_{pm:pm \in PM} W_{pm} \cdot r_{pm} \leq 0, \tag{1}$$

Where W_{pm} represents the se'ver's willingness to offer resources or performance weight. Considering the proposed VMP method, let γ_{pm} be the fairness among the association of PMs. Once $\gamma_{pm} = 1$, the utility function of the pm is represented as $UT_{pm}(r_{pm}(t)) = W_{pm} log\, r_{pm}(t)$.

Table 2. Key terminologies.

Terminologies	Description
$VM = \{vm1, vm2, ...vmn\}$	Set of VMs
$PM = \{pm1, pm2, ...pmk\}$	Set of PMs
$VM = (pm)$	Set of VMs hosted by a PM $j \in PM$
$r_{pm}(t) = L_{pm}(t)$	VM resource needs to be aggregated at a PM
$r_{vp}(t) = L_{vp}(t)$	The VM resource demands placed on PM
$Capacity_{pm}$	PM capacity (e.g., CPU power, memory)

https://doi.org/10.1371/journal.pone.0289156.t002

Next, maximising the cumulative utilities of all PMs in the data centre is expressed as:

$$\max \sum_{pm \in PM} W_{pm} \, log \, r_{pm}(t) \tag{2}$$

Virtual machine placement problem statement

Let $L_{vp}(t)$ symbolise the load of $VM\ i$ which is hosted on physical machine pm and $L_{pm}(t)$ denote the aggregate load of PM, the following condition (3) must be satisfied:

$$L_{pm}(t) = \sum\nolimits_{i:i \in VM(PM)} L_{vp}(t) \tag{3}$$

Here $L_{vp}(t)$ is the VM's load requirements as the d dimensional vector, where d = 2 when CPU and memory are considered, $L(t)$ is given by

$$L(t) = \left(VM_{cpu_i}, VM_{ram_i}\right) \tag{4}$$

Further, $Capacity_{pm}$ is defined as the available server capacity on PM $j \in PM$ regarding its CPU and RAM. The following formula must hold true to confirm that the overall load on any PM is not more than its capacity.

$$\sum\nolimits_{vm:vm \in VM(PM)} L_{vp}(t) \leq \ Capacity_{\ pm} \tag{5}$$

Typically, optimal placement of virtual machines on servers and turning off other servers leads to maximisation of utilisation and minimising server power consumption. To reflect this, in our analysis, the following equation is utilized:

$$(Y_1) : \max \sum_{pm:pm \in PM} UT_{pm}\left(L_{pm}(t)\right) \tag{6}$$

Subject to

$$\sum_{vm:vm \in VM(pm)} L_{vp}(t) = L_{pm}(t), \forall \ pm \in PM, \tag{7}$$

$$\sum_{vm:vm \in VM(pm)} L_{vp}(t) \leq \ Capacity_{\ pm}, \forall \ pm \in PM \tag{8}$$

$$Over \ L_{vp}(t) \geq 0, vm \in VM, pm \in PM$$

Based on the constraints below, a single PM can host a set of VMs:

$$\sum_{i}^{n} VM_{cpu_i} \le PM_{cpu_j}, \forall\ vm_i \in VM,\ and\ pm_j \in PM \tag{9}$$

$$\sum_{i}^{n} VM_{ram_i} \le PM_{ram_j}, \forall\ vm_i \in VM,\ and\ pm_j \in PM \tag{10}$$

The above equation ensures that the total resources used by a group of VMs should not surpass the CPU and memory capacities of PM.

When only CPU and RAM are considered, the PM resource utilisation problem (Y_1) will be equivalent to:

$$\left(Y_1'\right) : \max \sum_{pm:pm\in PM} UT_{pm}\left(PM_{cpu_j}(t) \times PM_{ram\ j}(t)\right) \tag{11}$$

Subject to

$$\sum_{vm:vm\in VM(PM)} PM_{cpu_j}(t) \le\ Capacity_{pm}^{cpu}, \forall\ pm \in PM \tag{12}$$

$$\sum_{vm:vm\in VM(PM)} PM_{ram\ j}(t) \le\ Capacity_{pm}^{ram}, \forall\ pm \in PM \tag{13}$$

$$Over\ L_{vp}(t) \ge 0, vm \in VM, pm \in PM$$

To facilitate the subsequent derivation of the formula, let $r_{pm}(t) = L_{pm}(t)$ To maximise the data ce'tre's' overall aggregate utilities and find the best solution, a Lagrange function is defined as:

$$\begin{aligned} &LR\left(r_{vp}, r_{pm}, \gamma, \beta\right) \\ &= \sum_{pm:pm\in PM} \left\{ UT_{pm}\left(r_{pm}(t)\right) + \gamma_{pm}\left(\sum_{vm:vm\in VM} r_{vp}(t) - r_{pm}(t)\right)\right\} \\ &\quad + \sum_{pm:pm\in PM} \beta_{pm}\left(Capacity_{\ pm} - \sum_{vm:vm\in VM} r_{vp}(t) - \boldsymbol{\varepsilon}_{pm}^2\right) \end{aligned} \tag{14}$$

Where $\gamma = (\gamma_{pm}, pm \in PM)$ and $\beta = (\beta_{pm}, pm \in PM)$ are Lagrange multiplier vectors, $\boldsymbol{\varepsilon}^2 = \left(\boldsymbol{\varepsilon}_{pm}^2, pm \in PM\right)$ is the relaxation factor vector. Let γ_{vm} denote the load requirement of the virtual machine vm. Let β_{pm} be the available capacity of the physical machine pm. Let the resource occupied by all VMs on physical machine pm be expressed as $\sum_{vm:vm\in VM(pm)} r_{vp}(t)$ and $\sum_{vm:vm\in VM(pm)} r_{vp}(t) \cdot \boldsymbol{\varepsilon}_{pm}^2 \ge 0$ represents the enduring resources present on the physical machine pm.