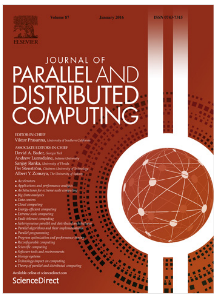
Accepted Manuscript



Workload aware VM consolidation method in edge/cloud computing for IoT applications

Irfan Mohiuddin, Ahmad Almogren

PII: S0743-7315(18)30676-2
DOI: https://doi.org/10.1016/j.jpdc.2018.09.011
Reference: YJPDC 3949

To appear in: *J. Parallel Distrib. Comput.*

Received date : 9 May 2018
Revised date : 5 September 2018
Accepted date : 17 September 2018

Please cite this article as: I. Mohiuddin, A. Almogren, Workload aware VM consolidation method in edge/cloud computing for IoT applications, *J. Parallel Distrib. Comput.* (2018), https://doi.org/10.1016/j.jpdc.2018.09.011

***Highlights (for review)**

Highlights

- A workload aware virtual machine consolidation model is proposed in Edge Cloud Environment
- A classification based heuristic is used for the placement of virtual machines in edge cloud
- A promising virtual machine migration strategy is proposed for better resource management

Workload Aware VM Consolidation Method in Edge/Cloud Computing for IoT Applications

Irfan Mohiuddin and Ahmad Almogren
College of Computer & Information Sciences
King Saud University
Riyadh 11543, Kingdom of Saudi Arabia
irfanm@.ksu.edu.sa, ahalmogren@ksu.edu.sa

Abstract- **Wide-ranging edge cloud data centers are a vital part of the solution for the problems caused by enormous growth in the IT industry for high computational power by advanced service applications. Majority of IoT applications switched to the Cloud and this stimulated the emergence of Edge technology to better manage the computing applications, data, resource and services. Consequently, with the massive client size and enormous applications trying to benefit from the cloud service, it makes it a challenging task for the edge cloud data centers to work in a power saving mode. In this paper, we propose a virtual machine consolidation method to switch the idle physical servers into hibernation mode, resulting in reduced power usage. We know that edge cloud data centers offer storage as a service, in this study we address the issues pertaining to storage units in the data centers. A unique classification approach is adopted to ensure load is balanced accordingly during allocation and our main contribution is on the VM migration technique. The VM migration is aimed in consolidating the VMs based on the workload to reduced number of physical machines to mitigate the energy consumption and promoting green computing. Therefore, we name the approach as Workload Aware Virtual Machine Consolidation Method (WAVMCM). We validate the proposed method with a competitive analysis of experimental results gathered from comparing it with Artificial Intelligence based probabilistic algorithm like Simulated Annealing, Genetic Algorithm and a case of no migration. Experimental results demonstrate that the proposed WAVMCM reduces 9% active servers saving 15% of power consumption when compared to genetic algorithm based method.**

***Keywords*-Edge Computing, Cloud Computing, Energy Efficient Allocation, Virtual Machines, Workload aware, Green Computing, Resource Management, Dynamic Consolidation.**

I. Introduction

Resource management in the context of virtual machines refers to assigning resources like- computing processes, networks, nodes and storage on-demand to a set of applications in cloud computing environment. The cloud service providers are supposed to provide the resources to clients within the limits of the service level agreements (SLAs). This goal is accomplished by the cloud service provider (CSP) by means of multiplexing the resources through virtualization techniques. Applications with unpredictable load or startups rent the resources from the cloud provider on pay-per-use method. Today, with the increasing growth in the cloud computing, the requirement is to improvise the services being offered to match the surge in the cloud computing data centers with hundreds of thousands requests being received. The Internet of Things (IoT) technology is expanding at high pace and consequently the supporting technologies like Cloud and Edge computing are fine tuning themselves to meet the demands. Due to huge number of organizations and companies moving to the cloud services, it is getting tougher for the cloud data centers to manage the resources in a way that it does not encourage unnecessary operations, for example leading to inefficient power consumption by the data centers.

In previous methods, the VM resource management problem is dealt as a bin packing problem and several methods have been proposed ranging from Best Fit, First Fit, Best Fit Decreasing, Genetic Algorithm based and Simulated Annealing based [1]. But all the previous methods present an optimal solution to the problem and could not deliver a global optimum. These methods not only change the VM layout constantly because of its aim to find an optimal solution but increase the number of active servers in an attempt to migrate the overloaded servers. There are several challenges confronted in aiming to enhance the virtual machine consolidation process to reduce the wastage of resources. One of the main challenge is the dynamic nature of the workload received by the data centers. Secondly, the realization of the need to consolidate the virtual machine to reduce the energy consumption. The other challenge is the heterogeneity of workload types and moreover the consolidation process should not be an added burden in terms of security and data retrieval [2, 3]. Environments with deterministic workloads are rare and therefore a method to deal with dynamic non-deterministic workload aware method is needed.

A recent study shows that the energy consumption of the data centers alone will be 2% of the global energy consumption by the year 2020 [4]. Meng et al. [5] suggest the use of traffic patterns among the VMs and optimize the placement of VMs to enhance the network scalability. The VMs with the large bandwidth usage are placed in closer proximity to the host machine when compared to VMs with lesser bandwidth usage. However, the migration strategy is not addressed. A particle swarm optimization method and multi-resource allocation model based virtual machine allocation algorithm is proposed by Xiong et al. [6]. Though the proposed algorithm works on the lines of energy saving and utilizing the system resources in a reasonable way, it fails to address the issue of system instability and resource wastage, after any VM is withdrawn from the host machine. Wang et al. [7] addressed the issue of energy conservation through the VM allocation. The authors attempt to resolve the issue of centralized VM allocation and migration of VMs with cost consideration through a multi-agent based method. For each host machine, an agent is deployed and unquestionably it requires constant communication between these agents and this in turn increases the cost of the operation and computation time. Wolke et al. [8] discuss about workload aware migration but in a deterministic environment where the workload is known a priori.

In this paper, we overcome the demerits of the other methods by designing a technique, which takes into consideration the load of the server and cost of migration. Therefore, keeping in mind the above mentioned challenges, a workload aware virtual machine consolidation method is designed. In this method, after the first allocation is processed, the workload is analyzed in each physical server before consolidating the virtual machines to less number of active physical servers. The aim of this method is to utilize the resource efficiently by reducing the number of active servers and conserve power.

2. Related Work

There is an increasing interest being shown by the global research community on VM allocation and migration techniques to reduce the power consumption in cloud computing and edge computing environments. The current researches and reviews are drawing the attention of researchers and practitioners towards the power consumption issue pertaining to the excess underutilized active servers in the cloud data centers.

Kansal et al. [9] highlights that resource allocation is a fundamental part of resource scheduling. If the resources are allocated in a systematic method then it is much likely to effect the overall need for the consolidation of the VMs to minimal physical servers through migration. An efficient resource allocation procedure is one which satisfy the QoS aware standards, is less expensive and energy efficient. Wang et al [7] dispatches an agent to each physical machine to assist the physical machine in managing the resources. An auction based VM allocation decide the allocation of the VMs to the PMs. Most importantly, a negotiation based mechanism to consolidate the VMs for energy cost saving is put forward. However, there is an increase in computation time because of the intensive negotiation and cost calculations. Vasu et al. [4] dicusses the dissemination of the load in a balanced fashion based on the previous records of downtime. However, the authors proposed the use of a reliability record at the server end and they have not discussed the implication of maintaining such a record. Wang et al. [10] addresses an important aspect of undeterministic jobs. The cloud service provider is not familiar with the future jobs they receive and therefore a prediction based model is proposed to avoid the affect of unpredicted jobs to the performance of the resource allocation. An energy conserving resource allocation scheme with prediction (ECRASP) for VM allocation to PM in cloud computing predicts the trends of arriving job and related features for the future demand, which helps the system to take sufficient decisions.

In [11], Ali et al. highlights the inefficiency of conventional works which do not consider more than one energy host. More specifically the authors deal with the large scale cloud datacenters. However, the solution is not applicable for all size of data centers. A gossip protocol is designed by Yanggratoke et al. [12], in which the changing of load pattern is taken as a criteria for the consolidation of the VMs. However, it is a heuristic solution and the author hints that it works efficiently in certain conditions and may not be applicable to all types of environments. Jha et al. [13] introduced a load balancing algorithm for Infrastructure as a Service Cloud to mitigate the power consumption in the cloud data centers. However, the computational time is not discussed after implementing the load balancing strategy. Genetic algorithms are evolution based algorithms and are said to be providing better results after a certain number of generations. Ranjbari et al. [14] proposed a learning automata based algorithm to enhance the resource utilization and mitigate the energy consumption. However, the proposed method is a prediction approach depending on the changes in the user's resource demands.

A time-series forecasting approach is presented by Zhou et al. [15] for the non-deterministic workload environment when there is no a priori information about the incoming jobs. Nevertheless, a threshold specific to the simulation environment is used by the proposed system to adjust to the dynamic workload. Wang et al. [16] proposed Ada-Things to monitor and enable live migration method for the IoT application in the edge cloud architecture. The workload information is the main criteria adapted for migration to achieve a fair balance in the performance. Genetic programming based and swarm intelligence based optimization models pose drawbacks like high error rate and low feasibility in addition to the prevailing scalability challenge [17].

Wu et al. [18] proposed a power estimating model to predict the CPU-intensive workloads and estimate the power consumption of the single VM and physical server hosting the VM. Huda et al. [19] proposed Combinatorial Ordering First-Fit Genetic Algorithm (COFFGA) and Combinatorial Ordering Next Fit Genetic Algorithm (CONFGA) to reduce the running physical servers by consolidating the VMs. However, the proposed approach is a genetic algorithm based probabilistic based solution and not a global optimal. An energy-efficient solution with prediction of upcoming jobs based on Gaussian process regression method is proposed by Bui et al. [20]. However, an exhaustive approach is adapted with an assumption that confined number of physical servers can manage to serve the incoming VM requests.

A multipbjective with fuzzy rule-based system is proposed by Alberto et al. where the consolidation decision is taken by an operator in the data center in combination with a prediction technique of the load. Nevertheless, the prediction and heuristic models are not completely reliable with the magnitude of applications request to the edge cloud architecture [21].

Table 1 Literature Review for Energy Efficient Virtual Machine Allocation Methods

Reference	Algorithm	Problem addressed	Improvement	Weakness
An-ping and Chun-xiang **[22]**	Particle swarm optimization algorithm	VM allocation	Minimize energy	No comparison with recent methods
Wang et al. **[7]**	VM allocation based on auction strategy	VM allocation	Minimize energy	In dynamic environment the results are same to the compared technique
Wolke et al. **[8]**	Dynamic Server allocation problem (DSAP) linear program	Dynamic VM allocation	Reduce server demand and increase energy efficiency	Simulation does not show correct estimation of migration overhead
Fetahi et al. **[23]**	A gossip protocol	Network resource allocation and computing resource allocation	Energy efficiency and service differentiation	Minimum requirement for resource allocation system is 100,000 machines
Dabbagh et al. **[24]**	Energy aware resource provisioning framework	VM allocation	Save energy for the data center	Cloud provider is the only consideration in the study
Vasu et al. **[4]**	Fast up slow down (FUSD) algorithm	Predicting load and consumption of energy	Maximize utilization	Only focus on server and do not consider CPU, Storage and VMs
Wang et al. **[10]**	Energy conserving resource allocation scheme with prediction (ECRASP)	Power consumption	Improve the performance	No practical implementation
Ali et al. **[11]**	Energy efficient (EE) algorithm	VM allocation	Improve the performance	No comparison with standard methods
Jha and Gupta **[13]**	Power and load aware VM allocation policy	VM allocation	Improve the performance	No attention to load balancing and no comparison with standard methods
Yanggratoke et al. **[12]**	GRMP, a generic gossip protocol for resource management	Reducing power consumption	Reduce the energy consumption in data center	No comparison with other protocols
Kansal and Chana **[9]**	Artificial Bee colony	Energy consumption	Reduce execution time and energy consumption	No consideration for the workload of the nodes

Energy is a strength or vitality required for execution of cloudlets or tasks for certain resources of the cloud users demand in cloud computing. Simply, it is a form of an electricity to run the Physical Machines (PM's) in data centers. The energy consumption of given resource i at a time T with placement F is given as follows. [13]

$$Energy_{\ Total} = \sum_{resource\ i} \int_{Str\ Time}^{Fnh\ Time} E_i(F,T)$$

In Table 1 we present the existing energy efficient resource allocation methods with their advantages and limitations. In the literature, several researchers proposed methods, which either operate a single VM allocation and migration or a cluster of VM migration to decrease the frequency and cost of migration. But the migrations are based on approximation algorithms, which do not generate a final solution. The generated solutions are a result of unplanned consolidation of VMs, which might make the system unstable. Further, some researchers work on static and dynamic power consumption in homogenous environment. There are many researchers, who utilize the scheduling algorithms for example the Pegasus large scale workload distributed system [25] but without considering energy efficiency issues in the data center. Hence, in this paper, we present a workload aware VM consolidation for energy efficient cloud environment. The proposed approach thereby improves the resources utilization and reduce the overall energy consumption.

3. Proposed Method

We propose a new method for live VM migration, Workload Aware Virtual Machine Consolidation Method (WAVMCM), that reduces number of active servers by consolidating VMs to less number of servers in order to reduce the energy consumption.

The physical servers in the data centers are of higher resource capacity and allocating these higher resource capacities to lower VM resource requests, leads to improper utilization of resources and moreover higher energy consumption by running more number of servers, as and when a new request arrives.

Thin provisioning technique and virtualization help in overcoming this improper utilization of resources and help in managing the resources efficiently. But, after allocating, to optimize the system, migration of the virtual machines is performed and we observe in the experimental study that a calculative migratiion is better when compared to unaccounted migrations.

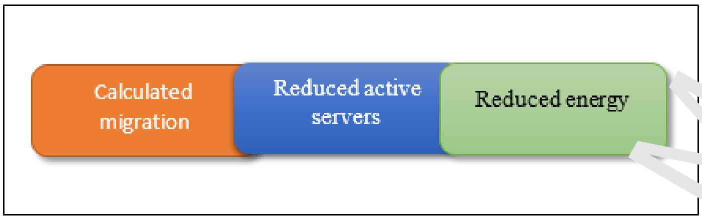


Figure 1 General Approach

We classify the physical resources into four classes with varying resource capacities to match the diverse VM requests. The number of classes is not limited to four and could be scaled up or down as per the size of the data center. Classes vary from one another in terms of their computing capabilities like CPU, Network Bandwidth and Memory.

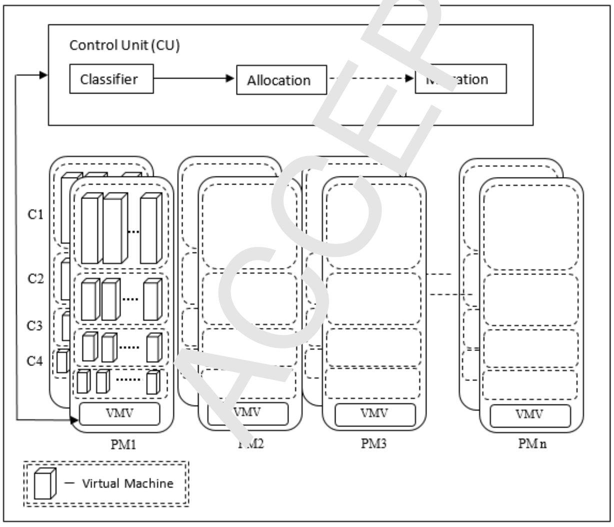


Figure 2 Virtual Machine Resource Allocation Architecture

The components and their functionalities in Figure 2 are as follows.

Table 2 Components and functionalities of proposed architecture

Component Name	Functionalities
PM-Physical Machines	Typical Physical servers in the cloud infrastructure
VMO-Virtual Machine Observer	To monitor the working of the local virtual machines and send the information to the control unit.
VM- Virtual Machines	Clients own VMs and each VM may be a different application of various clients.
(C1,C2,C3,C4)- Class Layers	Each PM is classified into classed layers hosting varying capacity resources
CU-Control Unit	Receives information from the virtual machine observer and works accordingly
Classifier	Classifies the incoming request into classes
Allocation	Receives information from the classifier and allocates the virtual machines accordingly
Migration	From the status information of the PM, checks if there is possibility to turn a PM to IDLE state to save power

Classifier-Each time a new VM request arrives to the Edge/Cloud from distnguished IoT applications, the request is received by the classifier. The classifier, which is an integral part of the Control Unit, checks the available classes in the physical machines and maps the request for the virtual machine. This mapped details are then forwarded to the Allocation component, which takes charge of allocating a virtual machine to the requesting IoT application.

The reason to classify the physical servers into classes are as follows:

1. Resource requirements are heterogeneous and therefore a classification of resources help in fitting the requirements to a best fit resource.
2. To avoid congestion of VM requests by possible over provisioning of the resources to one client and starving the other. Avoiding the large resource holding users to unnecessarily block the lower resource holding users.

3. To avoid aggressive consolidation of physical servers at the time of migration of VMs. In a rush for reducing the active servers, if there is no classification there is a high chance of disorientation of the resource management.
4. Migration of whole class is also possible if necessary and it does not require extra effort because the VMs inside the migrating class are of identical properties.

Divide the physical machines into four classes (class 1, class 2, class 3 and class 4) with class 1 being the class with higher capabilities of resources (CPU, Bandwidth and Memory) and class 4 being the minimal basic level class with reasonable amount of resource capacity.

When the VM request arrives to the cloud control unit (CU), we have a classifier which reads the VM request and classify the VM before allocating it to the matching class.

A Virtual Machine Viewer (VMV) maintains a status matrix which works as a monitoring unit which monitors the status of all the physical servers in the pool of servers to see all the possibilities for migration.

The designed approach is suitable for allocating a single VM in a class or multiple VMs in a class. As shown in the Figure 12 Classes are flagged as 1 if they have any virtual machines class whose utilization is less than 1/3rd the class resource capacity. From this we understand that the lower bound is set to 1/3rd of its capacity.

Allocation- The allocation component takes the mapped details of the available resources in the form of empty slots in the various classes of the physical machine and the specification of the requests from the IoT applications. Thereafter the allocation of requested resources like network, bandwidth, processing time and storage are accomplished. The allocation algorithm implemented in this approach is through the well established First Fit algorithm. In the Figure 3, we demonstrate the allocation approach through a flowchart.

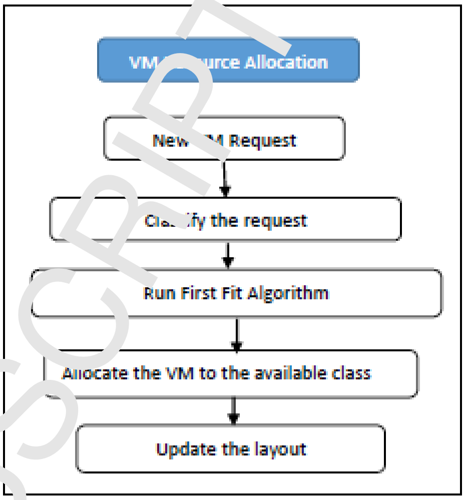


Figure 3 Flowchart of VM Allocation

Migration – Contrary to VM allocation where the resources are assigned to the IoT applications in the form of a VM specifically designed for them, consolidation of VM refers to the oncept of moving the created VM from one PM to another in order to deduce the power consumption caused by operation of more PMs. The consolidation of the virtual machines may start at a set period or when the load of a particular server has reached to a threshold value 'X' after which there is possibility of performance degradation. In previous works, we observe that the consolidation procedure is initiated when the total load of the servers decreases less than 50%. The number of physical machines should be reduced while maintaining satisfactory performance of the VMs.

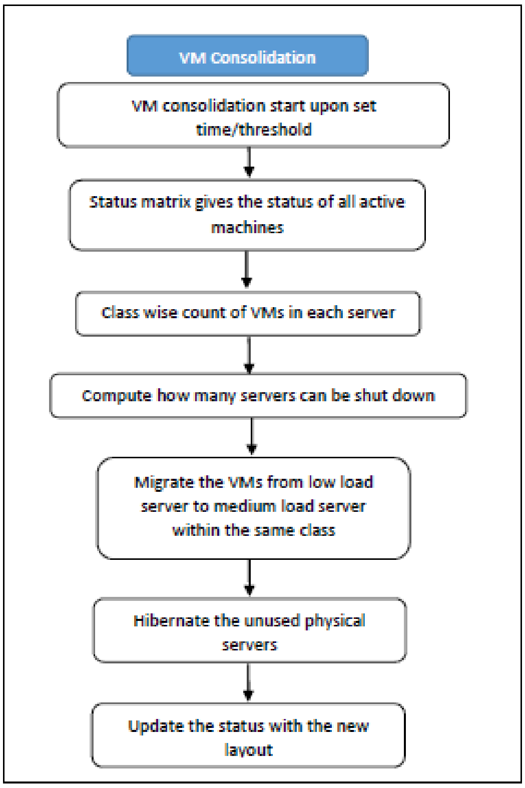

VM Consolidation
VM consolidation start upon set time/threshold
Status matrix gives the status of all active machines
Class wise count of VMs in each server
Compute how many servers can be shut down
Migrate the VMs from low load server to medium load server within the same class
Hibernate the unused physical servers
Update the status with the new layout

Figure 4 VM consolidation

Migration algorithm takes the feed from the VMV and migrates the VMs from Potential Idle Server (PIS) to hibernate the server in order to reduce the power consumption..

The migration procedure is triggered by a set time duration or through a threshold trigger. In practical environment mostly the threshold is taken as parameter to initiate migration. Aim of migration algorithm is to reduce the number of active servers by performing possible migrations of the VMs to accommodate the VMs to a location such that migration cost is as minimal as possible and the outcome of the migration result in reduced active server leading to lesser energy consumption.

The strength of the proposed method is to perform workload aware live migration of the VMs located at low load physical servers. Workload aware refers to the overall load of the physical server which is calculated as sum of workload of all the classes in order to qualify for the migration. There are several cases which account a virtual machine to be migrated such as the physical server is overloaded, the physical server is under-loaded or the physical server could be turned to hibernation in order to reduce the number of active servers. In this study, we test the performance of the proposed approach through the number of reduced servers it achieves.

Following is the procedure to turn an active low load physical server into hibernation to reduce number of active servers.

For instance, if the number of active servers are N= 13 and the 1's represent the classes with VMs and 0's represent a vacant position.

$$\begin{array}{c} C1 \\ C2 \\ C3 \\ C4 \end{array} \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Figure 5 After Initial Allocation

We check the maximum number of classes which have underutilized resources in the total number of active servers. In the below figure we can observe that the maximum is Max =8 classes. This is also the least possible active servers to be in Power ON state out of the total active server before migration.

$$\begin{array}{c} C1 \\ C2 \\ C3 \\ C4 \end{array} \left(\begin{array}{ccccccccccccc} 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{array} \right) \begin{array}{l} \to 8 \\ \to 8 \\ \to 7 \\ \to 5 \end{array}$$

Figure 6 Counting VMs

To calculate the possible serves that can be turned to hibernation we perform Potential Idle Server (PIS) identification, where PIS=N-Max. We get PIS = 13-8=5.

To calculate the energy saved through migration of the underutilized servers we have,

ConsumedEnergy= Energy(N) – Energy(PIS)

Saved Energy = Energy(PIS)

Our objective is to migrate 5 server's classes to fit into other servers in order to reduce the total number of active servers by 5.

To decide which 5 servers we want, we calculate the sum each physical server is calculated. For example, Workload(P0) =6 (i.e. the workload of the first physical server is 6 because a fully occupied physical server's workload is denoted by 8+4+2+1 = 15). The workload here is 6 because there are two VMs only with 4 and 2 as their workload. Our proposed approach reduces the number of active servers by migrating the VMs classes which cost minimum migration if required. The workload calculated in the step 6 indicates the cost of migration in migrating the VM classes. From the total 13, we sort the minimum 5 physical servers (PS) which are **PS (1), PS (2), PS (5), PS (6), PS (9)**.

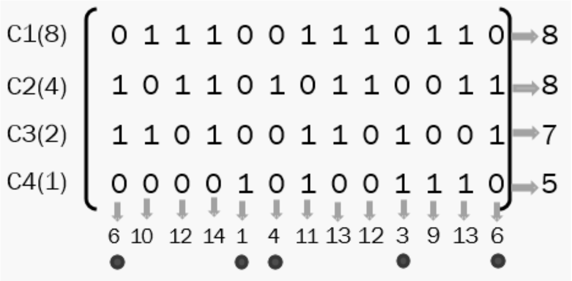

C1(8) 0 1 1 1 0 0 1 1 1 0 1 1 0 8
C2(4) 1 0 1 1 0 1 0 1 1 0 0 1 1 8
C3(2) 1 1 0 1 0 0 1 1 0 1 0 0 1 7
C4(1) 0 0 0 0 1 0 1 0 0 1 1 1 0 5
6 10 12 14 1 4 11 13 12 3 9 13 6

Figure 7 Weighing the load on physical machines

In a case where a class cannot be migrated to a same class to reduce the number of active machines then it is liable to move the smaller class for instance there are two VMs with load designation 4 for VM1 and 1 for VM2. The choice to turn of the physical server will be made for the one which is holding VM2 because the migration of VM1 will cost more. The final solution with reduced number of active servers is achieved through a global optimum method instead of approximation or local optimal solution.

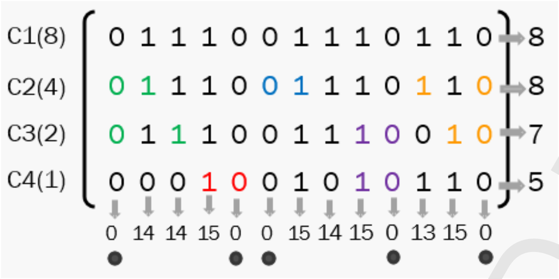

C1(8) 0 1 1 1 0 0 1 1 1 0 1 1 0 8
C2(4) 0 1 1 1 0 0 1 1 1 0 1 1 0 8
C3(2) 0 1 1 1 0 0 1 1 1 0 0 1 0 7
C4(1) 0 0 0 1 0 0 1 0 1 0 1 1 0 5
0 14 14 15 0 0 15 14 15 0 13 15 0

Figure 8 Post Migration of VMs

In the above figure, the color code is used to represent the migration from source to destination. After successful migration, the physical servers with all 0 value classes represent they have no reason to stay active so they are turned into hibernation. Therefore, we have reduced the number of active servers in the above example from 13 active servers to 8 active servers. From a Cloud Service Provider's (CSP) perspective it is a remarkable achievement and globally also it contributes in lesser carbon footprint when physical servers are turned to hibernation. Contrary to our global optimum solution giving a final solution, we compare our approach with the basic migration less allocation procedure, Genetic Algorithm based technique and Simulated Annealing approach.

4. Implementation and Experiments

We simulate the designed algorithm using MATLAB and compare the approach with two popular intelligent algorithms, they are Genetic Algorithm and Simulated Annealing.

Genetic Algorithm (GA) is probabilistic searching method which results in an adaptive optimization strategy to solve a problem. It finds an exact or approximate solutions to optimization and search problems. GA are characterized as global search heuristics. GA is based on the biological evolution and they have operations like inheritance, mutation, selection and crossover. Any typical GA requires the genetic representation of the solution domain and a fitness function to evaluate the solution domain. The fitness function is subjective to the problem to which the GA is applied [5]. We represent the solution domain as the physical server with resources. The fitness function can be defined over the number of VMs in a physical server. The aim would be to deliver a solution that is nearly optimal to the number of physical servers utilized at a time. The initial layout of the VMs in physical servers is considered as the initial solution and this initial solution is important in GA because the genetic operators are applied to this solution. If we would like to get a solution in a fixed time, then the number of generation an initial solution can generate could be fixed. Therefore, we end up having a solution better than the previous solution from where we began but not an optimal solution. "Depending on the set population length it selects a population of VMs and calculates the load of each VM. The load of physical server is computed. Fitness for each population is calculated and one population is chosen randomly. Crossover and mutation is performed before calculating the fitness value again and this process repeats for a desired iteration, for instance iteration is set to 100 times. After 100th iteration the results are compared and chosen best population."

Chromosome Encoding

A first fit decreasing algorithm may result different output for different combinations of allocation. If there are n service request, then there maybe n! type of combinations and as n goes bigger the value for n! also gets huge. One can get an acceptable result using first fit decreasing but we cannot test all the combinations. To solve this problem, we use genetic algorithm. And, the first step to genetic algorithm is chromosome coding. Since our aim is to find the best combination of VMs to PMs we use the chromosome coding as a representation of the combinations for the VMs. For example, if there are virtual machines with different loads for instance X1 = {2,1,3,4,6,5} then the result of encoding will be X1 = {2,1,3,4,6,5} and other method for instance FFD will be X2 = {1, 2, 3, 4, 5, 6}.

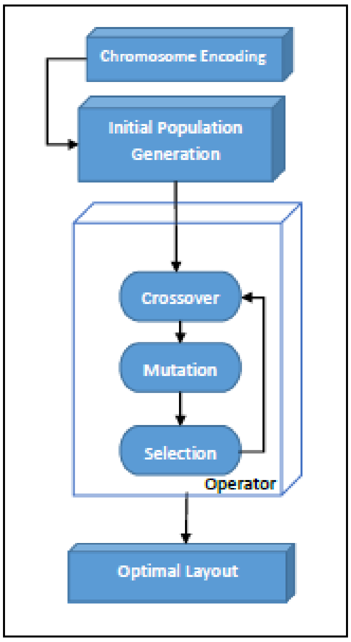


Figure 9 Flow chat of Genetic Algorithm

Fitness Function

Once, we get the chromosome, we can define the fitness value. The aim of the fitness function is to deduce the number of active servers by finding an optimal combination of servers to accommodate other VMs from the other underutilized servers and switch the servers which are not in use to hibernation state. Because each chromosome represents an allocation order, we need to calculate the fitness value for the GA.

Fitness function= $F = \sum_{i=1}^{length} a_i T_d / (1 + N_{idle})$

$length$:$Length\ of\ Chromosome$

ai:$weight\ of\ cost\ in\ each\ column$

Td:$is\ the\ basic\ unit\ of\ execution\ time\ for\ VM\ migration$

We assume execution time for VM migration in Class 1 is ‘4Td’, Class 2 is ‘3Td’, Class 3 is ‘2Td’ and Class 4 is ‘Td’.

Numerator: Target for a smaller value

Denominator: The number of closed or idle servers compared with previous matrix. Higher value is the target value for denominator expression.

Therefore, the fitness function F smaller is the target, which result in lower cost and close more number of active servers.

Selection operation

Selection is an important step in the Genetic Algorithm. We can take some action to select from the population after the fitness function value is calculated for each chromosome. The main objective of the selection operation is to generate a new generation of the population from the previous generation to ensure the diversity of the population. In our fitness function the cost of the migration Td should be less and the number of idle servers turned to hibernation should be more, which interprets to the minimum the fitness function F, the better. The common selection methods are Roulette Wheel Selection, tournament selection and so on. However, in order to avoid the population to fall into degradation we avoid tournament function for selection of chromosomes. Instead we choose by sorting the fitness values from less to high because our aim is to have a low fitness value as per our derived function. In this process we must make sure that no duplicates values are selected.

Crossover

The main aim in the crossover step of the GA is to change the gene's order of the chromosome. We use a homologous crossover operator. We make use of the sequential crossover operator from the several other operators like single-point, double-point and partial crossovers. A sequential crossover is initiated in the population to generate new populations of all the set of VMs selected initially. These new populations are repeat for all individuals to calculate the fitness for all the individuals.

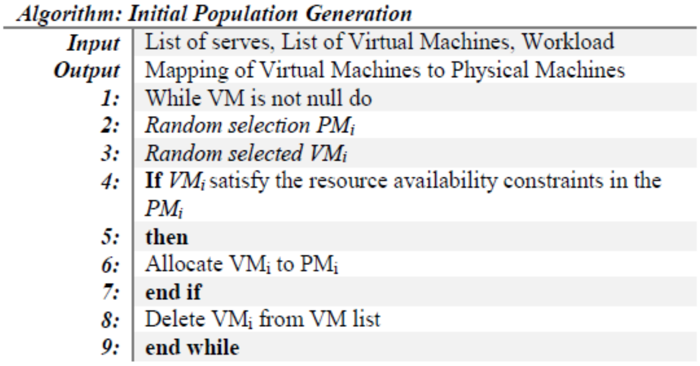
Mutation operation

In the mutation step of the GA, the chromosome which represent virtual machines in a physical server are modified by exchanging the genes in the same chromosome.

Termination condition

The process of computation is terminated if the number of iteration in finding the optimal solution is reached or when there is an improvement in successive iterations.

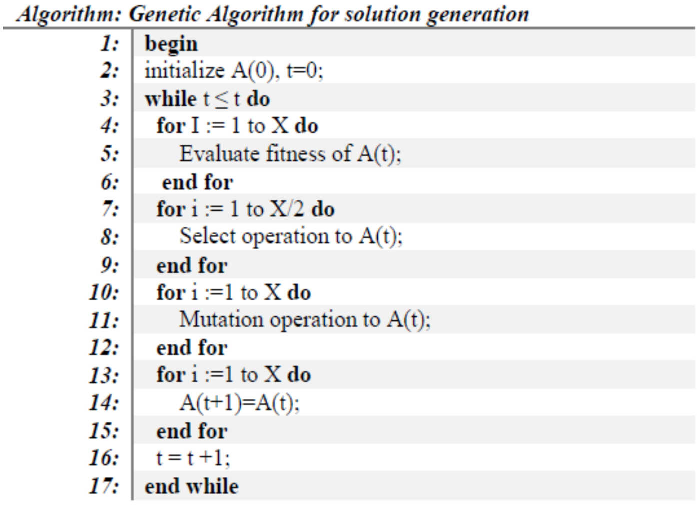
So, basically when the requirements are received from the clients for new VMs, chromosome encoding operation allocate the VMs to the PMs according to the best population after the last iteration. The fitness function is evaluated to this newly set layout of VMs in the PMs in each chromosome. Below we briefly present the genetic algorithm utilized for initialization.



Algorithm: Initial Population Generation

```
Input   List of serves, List of Virtual Machines, Workload
Output  Mapping of Virtual Machines to Physical Machines
While VM is not null do
    Random selection $PM_i$
    Random selected $VM_i$
    If $VM_i$ satisfy the resource availability constraints in the $PM_i$
    then
        Allocate $VM_i$ to $PM_i$
    end if
    Delete $VM_i$ from VM list
end while
```

Algorithm 1 Algorithm for Initial Population Generation



```
Algorithm: Genetic Algorithm for solution generation
begin
initialize A(0), t=0;
while t ≤ t do
  for I := 1 to X do
    Evaluate fitness of A(t);
  end for
  for i := 1 to X/2 do
    Select operation to A(t);
  end for
  for i :=1 to X do
    Mutation operation to A(t);
  end for
  for i :=1 to X do
    A(t+1)=A(t);
  end for
  t = t +1;
end while
```

Algorithm 2 Genetic Algorithm for solution generation

Simulated Annealing based technique:

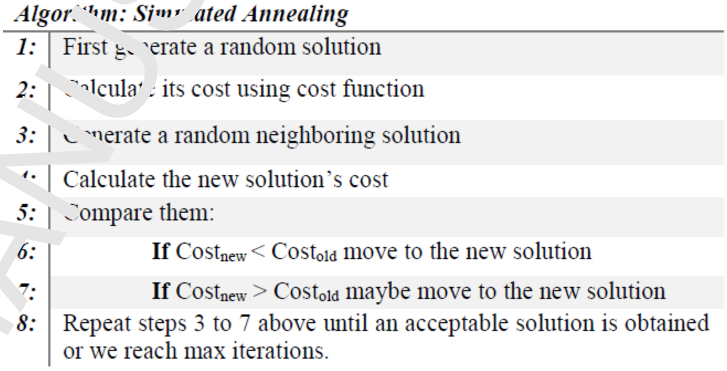
Simulated annealing (SA) was proposed by Kirkpatrick et al. [26] to be a general approach of solving some NP complete optimization problems in heuristic way. The effectiveness of the approach comes from two heuristics of divide and conquer and iterative improvement approach. In [27] Wu et al. proposed a simulated annealing based algorithm called Simulated Annealing Virtual Machine Placement algorithm (SAVMP) that intend to improve the VM placement. The objective of the improvement of the VM placement is to optimize the power consumption. The results obtained by the author are claimed to be saving 25% more energy than first fit decreasing in a particular time frame. The objective is to make better use of the hardware and save energy. The theory is built on the relation between the statistical mechanics and multivariate or combinatorial optimization. In this technique, the algorithm divides the large changes to a set of high temperature and low changes to a low temperature. It tries not to take any big steps in a risk of ending up at high temperatures, so slow and steady changes are opted.

This phenomenon of Simulated Annealing is applied to the migration of VMs in Edge/Cloud because as the number of virtual machines increases in the data center it becomes computationally intensive to check the number of physical machines to be turned off and which virtual machines to be selected and moved to other physical machine. Simulated annealing is an effective approach in this scenario which can find an approximate solution if not an optimal solution. However, the reshuffling of the VMs can be expensive due to reasons like the migration cost when being used in the VM consolidation. Similar to the GA, the initial VM layout is considered as the initial solution and the following steps are applied [28, 29, 30-33].

1. A random solution with mapping of VMs to PMs is generated
2. After defining a cost of migration function, calculate the cost of migration of VMs in the current solution
3. Random neighbor solution is generated and its cost of migration is also calculated
4. Reshuffling of VMs to PMs is done according to the minimum cost.

Algorithm Description

1. A simulated annealing algorithm consist of four components:
2. Problem configuration
3. New configuration generator
4. Optimization problem and its fitness function
5. A schedule of the temperatures and length of times to volve



```
Algorithm: Simulated Annealing
1: First generate a random solution
2: Calculate its cost using cost function
3: Generate a random neighboring solution
4: Calculate the new solution's cost
5: Compare them:
6:         If Cost_new < Cost_old move to the new solution
7:         If Cost_new > Cost_old maybe move to the new solution
8: Repeat steps 3 to 7 above until an acceptable solution is obtained
   or we reach max iterations.
```

Algorithm 3 Simulated Annealing

For the resource allocation and migration of VMs from highly loaded physical servers, we formulate the SA into four components as follows.

A. Problem configuration

In the cloud computing scenario, the assignment of virtual machines to the physical machines is the problem configuration. To avoid depleted space, a simple array is used instead of matrix to represent the assignment of the virtual machines. The constraint set is that one physical machine hosts one virtual machine. An array of represent the host number in which the virtual machines are assigned. For instance VM={2,3,1,2,3,2,1,3,2,2} . This represent an array of 10 virtual machines and virtual machines 1, 4,6,9 and 10 are in the host machine 2. Virtual machines 2, 5 and 8 are in host machine 3. Host machine 1 host the virtual machines 3 and 7.

B. New configuration generator

With an objective of achieving a computed fitness function, in the second component of the SA the algorithm tries to find the better configuration to meet the fitness function. To avoid workload overhead because of large search space, SA search for a better configuration in the neighboring states with better results close to the fitness function. With each step, configuration moves towards better state. A new configuration is formed by randomly swapping the pairing of the VMs and PMs. The locations of the virtual machines are changed to the other physical machines and the results are monitored. This operation forms a new neighboring state. It is similar to the bin packing operation to strive to achieve to a better result.

C. Optimization problem and its fitness function

The acceptance criteria for any new configuration to be the next new state is to satisfy all the above mentioned equations according to the capacity availability. The other constraint applied is to obtain configuration with low energy consumption. If a configuration is not suitable to the estimated result, then the energy consumption parameter is increase so that it will rejected in the screening. Therefore, we identify that energy parameter is a special parameter in the acceptance decision. Each new configuration is assigned an anomaly energy which represent a new state with an allowed value to go up than the previous state. This permission of higher anomaly energy for state is allowed to eliminate getting stuck at the local optimization.

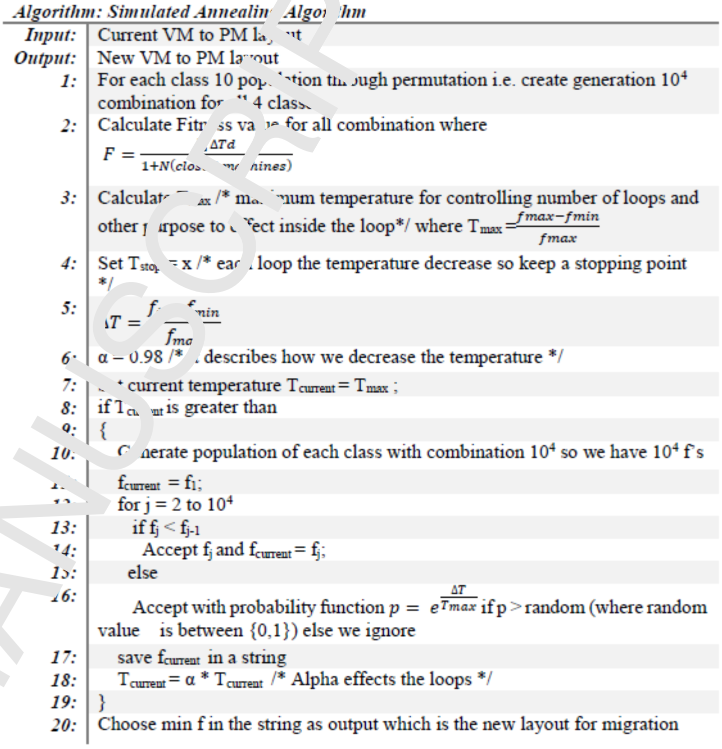
$$AnamolyEnergy = \max(E_{max}^{y}) * \frac{t_c}{t_0}$$

Where tc is the current temperature and $t0$ is the initial temperature. The energy increase at the high temperatures and the next following states might have energy with much larger values. Energy of the accepted states at low temperatures is stated stable.

D. Scheduling Temperatures

The most important operation in the simulated annealing method is the temperature scheduling procedure. In the absence of the temperature scheduling the quick quenching might lead to getting the state very far from the optimal state. The count of iterations at each temperature is a critical issue in the temperature scheduling operation. The range of the temperature to be used is also a concern. Therefore, the designing of the scheduling temperature is explained in an empirical way. Best of the several applied settings is picked out of the range. Initially temperatures start from 1000 degree and gradually it is reducing to 0 with an interval of 5 degrees each time, at which, new temperatures are 10,000 * N times of evolutions.

The initial resource allocation for the VM requests are performed by First Fit Decreasing algorithm. The strategy utilized for the enhancement of the VM placement is basically a procedure to find a set of random combinations of the VM to physical machines. Periodically, the incremental energy is monitored for each pair gained during the random process. Suppose PM1 is accommodating certain VMs and the simulated annealing algorithm intends to improve the energy utilization and resource utilization of the current layout, then energy consumption Ey is calculated for the concerned physical machine. For physical machine PM1 is $E1$and a random physical machine selected is PM2 and the energy consumption is $E2$ then SA finds that $E1$< $E2$ then the layout of energy consumption $E1$ is selected. This iteration is performed until a satisfactory approximate value for the fitness function is achieved. However, it is to be noted that SA is after all an approximation algorithm and cannot produce a global optimum solution.



```
Algorithm: Simulated Annealing Algorithm
Input:  Current VM to PM layout
Output: New VM to PM layout
For each class 10 population through permutation i.e. create generation $10^4$ combination for all 4 classes
Calculate Fitness value for all combination where
$F = \frac{\Delta Td}{1+N(closed\ machines)}$
Calculate $T_{max}$ /* maximum temperature for controlling number of loops and other purpose to effect inside the loop*/ where $T_{max} = \frac{fmax-fmin}{fmax}$
Set $T_{stop}$ = x /* each loop the temperature decrease so keep a stopping point */
$\Delta T = \frac{f_i - f_{min}}{f_{max}}$
α = 0.98 /* α describes how we decrease the temperature */
Set current temperature $T_{current} = T_{max}$ ;
if $T_{current}$ is greater than
{
  Generate population of each class with combination $10^4$ so we have $10^4$ f's
  $f_{current} = f_1$;
  for j = 2 to $10^4$
    if $f_j < f_{j-1}$
      Accept $f_j$ and $f_{current} = f_j$;
    else
      Accept with probability function $p = e^{\frac{\Delta T}{Tmax}}$ if p > random (where random value is between {0,1}) else we ignore
  save $f_{current}$ in a string
  $T_{current} = α * T_{current}$ /* Alpha effects the loops */
}
Choose min f in the string as output which is the new layout for migration
```

Algorithm 4 Simulated Annealing for VM Migration

The single point failure of SA based technique is that for larger trends, the method fails to give positive results because it is difficult for improvements, when the searching space grows bigger.

Experiment

We implemented the algorithms described above in MATLAB simulation software. The computation experiments are run on our PC with 4G memory and 2.3 Ghz CPU. For simulation purpose, we use randomized test data generated programatically in MATLAB.

The setup for the WAVMCM.

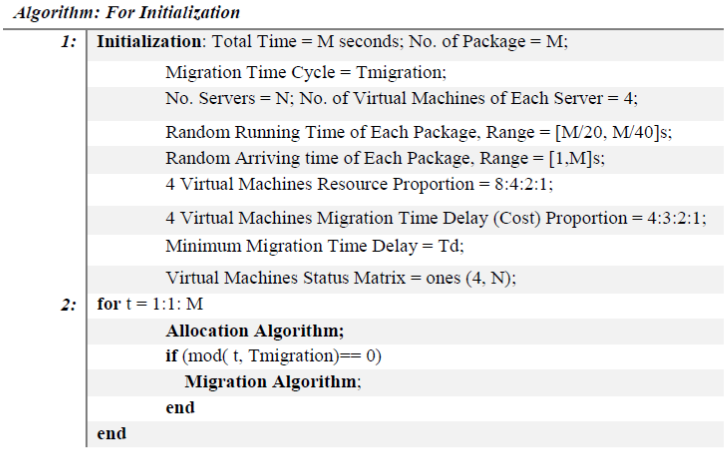
1. We configure N servers. In our example N=500 servers.

2. T denotes time where T=2000 seconds.

3. M denotes number of jobs where M=2000 jobs.

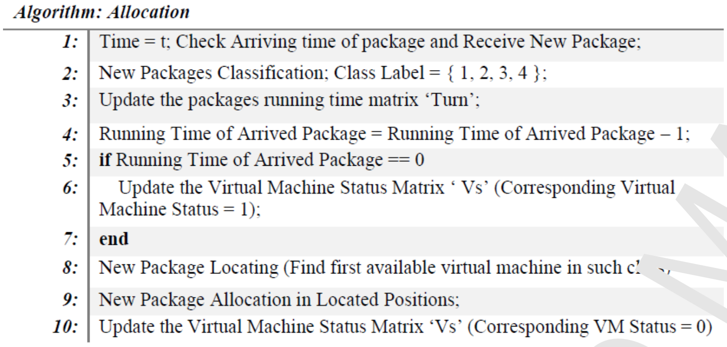
4. Execution time of jobs is arranged between the range (100,500) seconds.

5. Arriving time of job is in the range (1, 2000) seconds.



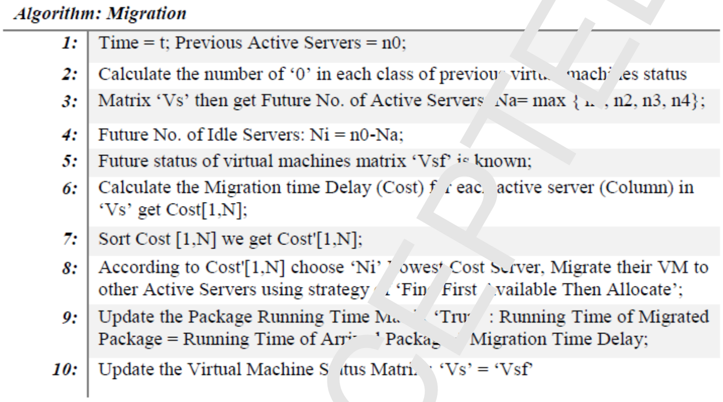
```
Algorithm: For Initialization
Initialization: Total Time = M seconds; No. of Package = M;
        Migration Time Cycle = Tmigration;
        No. Servers = N; No. of Virtual Machines of Each Server = 4;
        Random Running Time of Each Package, Range = [M/20, M/40]s;
        Random Arriving time of Each Package, Range = [1,M]s;
        4 Virtual Machines Resource Proportion = 8:4:2:1;
        4 Virtual Machines Migration Time Delay (Cost) Proportion = 4:3:2:1;
        Minimum Migration Time Delay = Td;
        Virtual Machines Status Matrix = ones (4, N);
for t = 1:1: M
        Allocation Algorithm;
        if (mod( t, Tmigration)== 0)
            Migration Algorithm;
        end
end
```

Algorithm 5 Main Algorithm for initialization of setup



```
Algorithm: Allocation
Time = t; Check Arriving time of package and Receive New Package;
New Packages Classification; Class Label = { 1, 2, 3, 4 };
Update the packages running time matrix 'Turn';
Running Time of Arrived Package = Running Time of Arrived Package – 1;
if Running Time of Arrived Package == 0
    Update the Virtual Machine Status Matrix ' Vs' (Corresponding Virtual Machine Status = 1);
end
New Package Locating (Find first available virtual machine in such class);
New Package Allocation in Located Positions;
Update the Virtual Machine Status Matrix 'Vs' (Corresponding VM Status = 0)
```

Algorithm 6 Allocation Algorithm



```
Algorithm: Migration
Time = t; Previous Active Servers = n0;
Calculate the number of '0' in each class of previous virtual machines status
Matrix 'Vs' then get Future No. of Active Servers: Na= max { n1, n2, n3, n4};
Future No. of Idle Servers: Ni = n0-Na;
Future status of virtual machines matrix 'Vsf' is known;
Calculate the Migration time Delay (Cost) for each active server (Column) in 'Vs' get Cost[1,N];
Sort Cost [1,N] we get Cost'[1,N];
According to Cost'[1,N] choose 'Ni' lowest Cost Server, Migrate their VM to other Active Servers using strategy of 'Find First Available Then Allocate';
Update the Package Running Time Matrix 'Turn': Running Time of Migrated Package = Running Time of Arrived Package + Migration Time Delay;
Update the Virtual Machine Status Matrix: 'Vs' = 'Vsf'
```

Algorithm 7 WAVMCM Migration

In our experimental setup, we simulate 500 physical servers. At one point, the curve gets straight, which resembles that the systems get balanced and virtual machines have reached a specific point, where there is a tradeoff between incoming jobs and leaving jobs. This is stable status. That is the reason the curve is flat which is called convergence, meaning the system is stable.

Each algorithm has its performance and have their own convergence line. We can easily say that our proposed algorithm provides better migration because it is a precise global solution. Without any migration, we can observe the lowest performing red line. The other two are simulated annealing and genetic algorithms are represented in the plot

At the beginning, the whole system is empty and gradually we accommodate the incoming virtual machine requests. Each virtual machine has tasks with execution time. The incoming requests are accommodated as best-fit policy to the empty spaces and therefore, there is overlap between the curves we observe in the graph. The migration technique implied at this moment yields the same result and leading to the overlap.

The other two algorithms are approximation algorithms and the proposed algorithm is a global solution. Because the design gives the final status of the virtual machine matrix, whereas the approximation does not know the final status matrix and look for approximation.

Definition for the percentage of saved energy:

Number of active servers after allocation= A

Number of active server before applying the migration algorithm = B

Percentage of saved energy = (A-B)/A*100

In Figure 12, we observe the following conditions:

(a) Zero/Overlap: Saved energy percentage is 0% because A=B (In this experiment we can observe at approx. 300 seconds according to the configured setting it monitors some works/packages leave the system and migration sets in at this point). Therefore, until this point there is an overlap in the curves and therefore no power could be saved.

(b) Convergence: There are three stages of the curve, first is incoming requests and no output. In this period, whether we apply migration or not, the application does not affect the system. That is why same result. Even we apply all the three algorithms; migration does not affect giving 0% saved energy.

The second stage is some application requests have started to leave the system. In this case some VMs are empty and therefore the algorithm has opportunity to effect the system and this trend of effecting the performance is observed and gradually it deviates from the overlap and this trend increases to reach the convergence point at approx. 700 seconds in our experiment. At this stage, the incoming and outgoing requests has certain balance, that is why the curve is linear and system is stable.

5. EVALUATION

Server consolidation plays an important role in the operational costs of data center and cloud servers. In this paper, we presented a workload aware virtual machine consolidation algorithm which takes the workload into consideration of each VM before migrating it to another physical server.

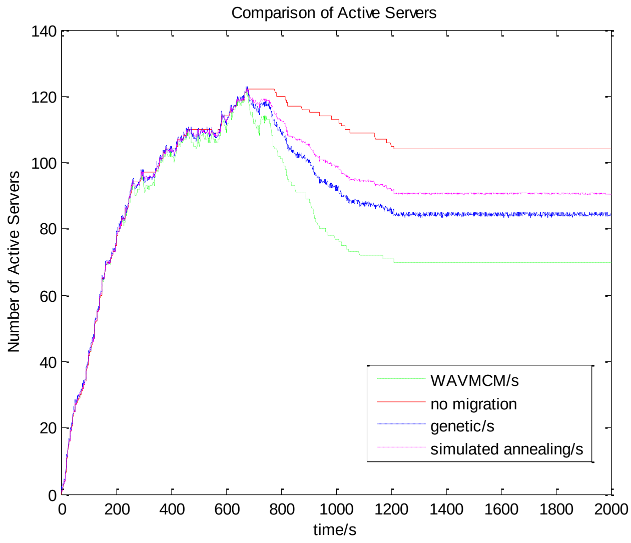


Figure 10 Comparison of Active Servers for every 200 seconds

The above figure, shows comparison of active servers in four scenarios. From top if we observe, the red line in the graph represent the allocation and no migration technique. Here there are no VMs being migrated so it will keep accommodating VMs as and when they are coming to the data center. It has highest number of active servers. We can observe that it has more than 105 servers active and consequently spend more amount of energy.

Next we have the pink curve, which represent the simulated annealing technique. This technique is an approximation method and gives a local optimal solution. In this experiment we can observe that it has more than 90 servers in active status and spends reasonably high amount of energy.

Thirdly, we have Genetic Algorithm based method, which is a probabilistic approac, resulting in an approximate solution which shows nearly 82 active servers which is comparatively better than the simulated annealing technique and migration free technique.

Lastly, we have our proposed migration algorithm, which reduces the highest number of active servers, as we can observe that there are 70 active serves saving energy of nearly 35 servers by putting them to IDLE state as shown in Table 3 i.e Comparison of number of active servers in (a) No migration (b) SA (c)GA (d) WAVMCM. The following values represent the results achieved shown in Figure 2.

Table 3 Comparison of Number of Active Servers

Method	*No. of Active Servers*
Without Migration	105
Simulated Annealing Technique	90
Genetic Algorithm	82
WAVMCM	70

In the figure below, we have compared the active server's frequency when the WAVMCM is performed for 50 seconds and then 100 seconds and the normal mode, which is migration free.

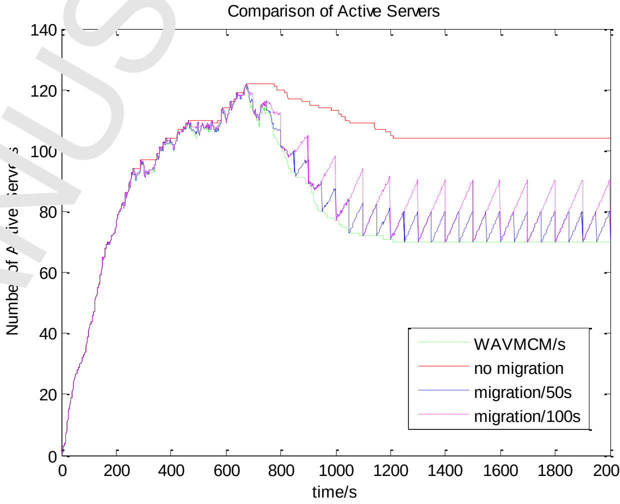


Figure 11 Comparison of Active Servers for every 50 seconds, 100 seconds and 200 seconds time interval

In the figure below, we have compared the percentage of saved power by performing the WAVMCM and the other two algorithms for every 200 seconds interval.

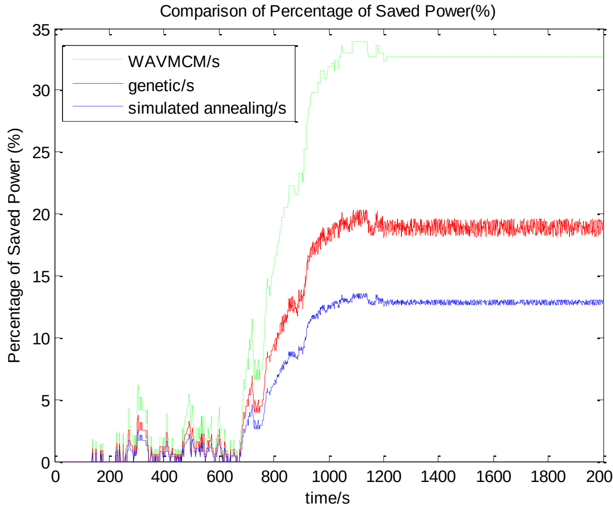


Figure 12 Comparison of Percentage of Saved Power

In the above figure, we demonstrate the percentage of saved power by our method while comparing it with Simulated Annealing technique and Genetic Algorithm based technique. We can observe that our proposed method saved highest percentage of power.

Secondly, GA is close to 18% of saved power and SA method is close to 14% of saved power. Therefore, we can safely say that, our proposed method outperforms the GA and SA approaches.

Table 4 Percentage of power saved

Method	*Percentage of Saved Power*
Simulated Annealing Technique	14%
Genetic Algorithm	18%
WAVMCM	33%

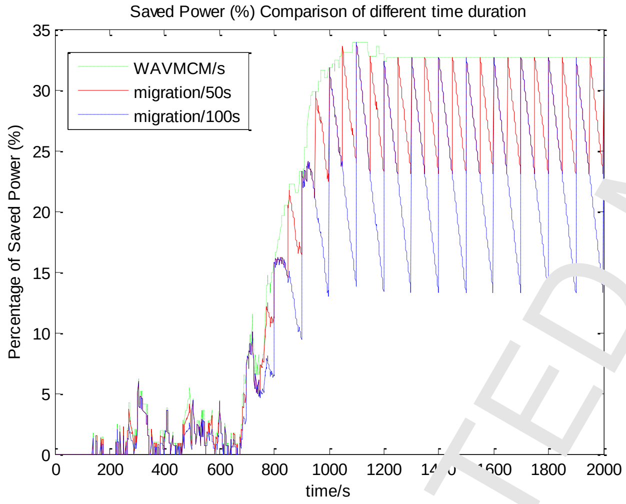


Figure 13 Comparison of saved power, / migration performed at different time period

In the figure above, we perform the workload aware virtual machine consolidation method along with two other methods which are based on genetic algorithms and simulated annealing. The time interval is set to three variations and therefore the figure shows varying levels of saved power with migrations performed at 50 seconds, 100 seconds and 200 seconds.

Therefore, from the results obtained, we can evaluate that the proposed method is performing better than two popular migration methods based on GA and SA.

6. Conclusions and Future Work

Green computing is going to be limitless with the rapid growth of business in the future. It is a procedure to use computing resources environmentally and user friendly while maintain overall computing performance. To reduce the use of hazardous materials, minimize the energy consumption, less heat generation and resource wastage are problematic issues in the cloud computing.

To prove the credibility of our designed algorithm, we compare the two other models based on Genetic Algorithm and Simulated Annealing method. The results obtained clearly proves that our designed algorithm efficiently manages the resources in the data center by reducing the number of active servers and eventually reducing the overall energy consumption, promoting green computing. Secondly, the classification of physical servers into varying capacities not only helps the cloud service provider to reduce the energy cost, but also benefits the clients by processing the VMs request swiftly.

7. Acknowledgement

The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through research group no (RGP-1437-35).

8. References

[1] S. Jangiti and S. S. VS, "Scalable and direct vector bin-packing heuristic based on residual resource ratios for virtual machine placement in cloud data centers," Computers & Electrical Engineering, vol. 68, pp. 44-61, 2018.

[2] S. F. Manoel C., C. C.Monteiro, P. R.M.Inácio and M. M.Freire, "Approaches for optimizing virtual machine placement and migration in cloud environments: A survey," Journal of Parallel and Distributed Computing, vol. 111, pp. 222-250, 2018.

[3] D. Kesavaraja and A. Shenbagavalli, "QoE enhancement in cloud virtual machine allocation using Eagle strategy of hybrid krill herd optimization," Journal of Parallel and Distributed Computing, vol. 118, pp. 267-279, 2018.

[4] R. Vasu, E. I. Nehru and G. Ramakrishnan, "Load Forecasting for Optimal Resource Allocation in Cloud Computing Using Neural Method," Middle-East Journal of Scientific Research, vol. 24, no. 6, pp. 1995-2002, 2016.

[5] X. Meng, V. Pappas and L. Zhang, "Improving the scalability of data center networks with traffic-aware virtual machine placement," in Proceedings IEEE INFOCOM, 2010.

[6] A.-p. Xiong and C.-x. Xu, "Energy efficient multiresource allocation of virtual machine based on PSO in cloud data center," Mathematical Problems in Engineering, 2014.

[7] W. Wang, Y. Jiang and W. Wu, "Multiagent-Based Resource Allocation for Energy Minimization in Cloud Computing Systems," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 47, no. 2, pp. 205-220, 2016.

[8] A. Wolke and L. Ziegler, "Evaluating Dynamic Resource Allocation Strategies in Virtualized Data Centers," in 2014 IEEE 7th International Conference on Cloud Computing (CLOUD), 2014.

[9] N. J. Kansal and I. Chana, "Artificial bee colony based energy-aware resource utilization technique for cloud computing," Journal of Concurrency and Computation: Practice & Experience , vol. 27, no. 5, pp. 1207-1225 , 2015.

[10] C.-F. Wang, W.-Y. Hung and C.-S. Yang, "A Prediction Based Energy Conserving Resources Allocation Scheme for Cloud Computing," in IEEE International Conference on Granular Computing (GrC) , 2014.

[11] A. Ali, L. Lu, Y. Zhu and J. Yu, "An Energy Efficient Algorithm for Virtual Machine Allocation in Cloud Datacenters," in ACA 2016: Advanced Computer Architecture. Communications in Computer and Information Science, Singapore, 2016.

[12] R. Yanggratoke, F. Wuhib and R. Stadler, "Gossip-based resource allocation for green computing in large clouds," in Proceedings of the 7th International Conference on Network and Services Management, Paris, France, 2011.

[13] R. S. Jha and P. Gupta, "Power aware resource allocation policy for hybrid cloud," 2015 Third International Conference on Image Information Processing (ICIIP), pp. 1-45, 2015.

[14] M. Ranjbari and J. A. Torkestani, "A learning automata-based algorithm for energy and SLA efficient consolidation of virtual machines in cloud data centers," Journal of Parallel and Distributed Computing, vol. 113, pp. 55-62, 2018.

[15] H. Zhou, Q. Li, K.-K. R. Choo and H. Zhu, "DADTA: A novel adaptive strategy for energy and performance efficient virtual machine consolidation," Journal of Parallel and Distributed Computing, vol. 121, pp. 15-26, 2018.

[16] Z. Wang, D. Sun, G. Xue, S. Qian, G. Li and M. Li, "Ada-Things: An adaptive virtual machine monitoring and migration strategy for internet of things applications," Journal of Parallel and Distributed Computing, pp. 1-45, 2018.

[17] P. Arroba, J. M. Moya, J. L. Ayala and R. Buyya, "Dynamic Voltage and Frequency Scaling-aware dynamic consolidation of virtual machines for energy efficient cloud data centers," Concurrency and Computation: Practice and Experience, vol. 29, no. 10, p. e4067, 2017.

[18] W. Wu, W. Lin and Z. Peng, "An intelligent power consumption model for virtual machines under CPU-intensive workload in cloud environment," Soft Computing, vol. 21, no. 19, pp. 5755-5764, 2017.

[19] H. Hallawi, J. Mehnen and H. He, "Multi-Capacity Combinatorial Ordering GA in Application to Cloud resources allocation and efficient virtual machines consolidation," Future Generation Computer Systems, vol. 69, pp. 1-10, 2017.

[20] D.-M. Bui, Y. Yoon, E.-N. Huh, S. Jun and S. Lee, "Energy efficiency for cloud computing system based on predictive optimization," Journal of Parallel and Distributed Computing, vol. 102, pp. 103-114, 2017.

[21] A. Cocaña-Fernández, J. Rodríguez-Soares, L. Sánchez and J. Ranilla, "Improving the energy efficiency of virtual data centers in an IT service provider through proactive fuzzy rules-based multicriteria decision making," The Journal of Supercomputing, pp. 1-16, 2017.

[22] A.-p. Xiong and C.-x. Xu, "Energy Efficient Multiresource Allocation of Virtual Machine Based on PSO in Cloud Data Center," Mathemtatical Problems in Engineering, vol. 2014, pp. 1-8, 2014.

[23] F. Wuhib, R. Yanggratoke and R. Stadler, "Allocating Compute and Network Resources Under Management Objectives in Large-Scale Clouds," Journal of Network and Systems Management, vol. 23, no. 1, pp. 111-136, 2015.

[24] M. Dabbagh, B. Hamdaoui and M. Guizani, "Energy-Efficient Resource Allocation and Provisioning Framework for Cloud Data Centers," IEEE Transactions on Network and Service Management, vol. 12, no. 3, pp. 377-391, 2015.

[25] D. Lo, L. Cheng, R. Govindaraju, L. A. Barroso and C. Kozyrakis, "Towards energy proportionality for large-scale latency-critical workloads," in ACM/IEEE 41st International Symposium on Computer Architecture, 2014.

[26] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by simulated annealing," Science, vol. 220, no. 4598, pp. 671-680, 1983.

[27] W. Yongqiang, M. Tang and W. Fraser, "A simulated annealing algorithm for energy efficient virtual machine placement," in IEEE International Conference on Systems, Man, and Cybernetics, 2012.

[28] A. Marotta and S. Avallone, "A Simulated Annealing Based Approach for Power Efficient Virtual Machines Consolidation," in IEEE 8th International Conference on Cloud Computing, 2015.

[29] M. A. Khan, A. Paplinski, A. M. Khan, M. Murshed and R. Buyya, "Dynamic Virtual Machine Consolidation Algorithms for Energy-Efficient Cloud Resource Management: A Review," Sustainable Cloud and Energy Services, pp. 135-165, 2018.

[30] I. Mohiuddin, A. Almogren, M. Al Qurishi, M.M. Hassan, I. Al Rassan, and G. Fortino, “Secure distributed adaptive bin

packing algorithm for cloud storage”. Future Generation Computer Systems, Vol. 90, pp. 307-316, 2019

[31] M.G.R. Alam, M.M. Hassan, M.Z. Uddin, A. Almogren, A. and G. Fortino, “Autonomic computation offloading in mobile edge for IoT applications”. Future Generation Computer Systems, Vol. 90, pp. 149-157, 2019

[32] S. Basu, M. Karuppiah, K. Selvakumar, K.C. Li, S.H. Islam, M.M. Hassan, and M.Z.A. Bhuiyan. “An intelligent/cognitive model of task scheduling for IoT applications in cloud computing environment”. Future Generation Computer Systems. Vol. 88, pp. 254-261, 2018

[33] A. Enayet, M.A. Razzaque, M.M. Hassan, A. Alamri, and G. Fortino. “A mobility-aware optimal resource allocation architecture for big data task execution on mobile cloud in smart cities”. IEEE Communications Magazine, 56(2), pp.110-117, 2018

*Author Biography & Photograph

Biography of Authors

Irfan Mohiuddin received his M.Sc. in Computer Science from King Saud University, Riyadh-Saudi Arabia, where he is currently working as a Researcher while pursuing his Ph.D. degree in Computer Science. His research interests include Cloud Computing, Networking, Resource Allocation, Internet of Things and Security.

Ahmad Almogren has received PhD degree in computer sciences from Southern Methodist University, Dallas, Texas, USA in 2002. Previously, he worked as an assistant professor of computer science and a member of the scientific council at Riyadh College of Technology. He also served as the dean of the college of computer and information sciences and the head of the council of academic accreditation at Al Yamamah University. Presently, he works as an associate professor and the vice dean for the development and quality at the college of computer and information sciences at King Saud University in Saudi Arabia. He has served as a guest editor for several computer journals. His research areas of interest include mobile and pervasive computing, computer security, sensor and cognitive network, and data consistency.