IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 07, Issue 03 (March. 2017), ||V1|| PP 49-62

Power Consumption and Temperature Minimization in Cloud Computing

Raed A. I. Alsini[1], Medhat A. Tawfeek[2]
[1](*Information Systems Department, Faculty of Computers and Information Technology, Kau, Saudia Arbia*)
[2](*Department of Computer Science, Faculty of Computers and Information, Menoufia University, Egypt*)
(medhattaw@yahoo.com)

Abstract: - Cloud computing resources are utilized in forms of different virtual machines that provide large-scale computation for complicated tasks. The allocation process of virtual machines on physical machines is a critical portion of cloud computing trends. Efficient allocation of the virtual machines on available physical servers is necessary to exploit the high-performance resource utilization and improving power efficiency. In this paper, virtual machine allocation problem is handled by particle swarm optimization with fuzzy multi-objective method. The main goal of the proposed method is to efficiently obtain a near-optimal solution that minimizes power consumption, total processing resource wastage and the peak temperature among the servers. The proposed method has been compared with well-known algorithms for virtual machine allocation problem existed in the literature. The comparison results prove that the proposed algorithm significantly outperforms the compared methods on the basis of power consumption, processing resource wastage and temperature metrics.

Keywords: - *Cloud computing, Fuzzy multi-objective, Particle swarm optimization, Power consumption, Peak temperature*

1. Introduction

Cloud computing is considered a natural evolution of grid computing in its approach for providing computing resources to remote users [1]. The providers of cloud computing have resources of multiple data centers at different geographical locations in order to optimally serve the requirements of costumers around the world [2]. These cloud resources appear to be infinite to the users who can rent computing power as they need [3]. The providers and users have conflicts goals. Providers want to gain more revenue, while users want to decrease expenses with meeting their requirements [4]. In this regard, the resources power in the cloud is becoming a harder challenge especially it control the operational costs [5]. Moreover, consuming more power also raises another serious issue that is a carbon dioxide [4]. The minimization of total power consumption with increasing the resource utilization leads to reducing the cost [6]. Virtualization is a one from the most important related technologies that make cloud computing possible [7]. Virtualization provides an efficient approach which hardware resources on one machine can be divided through partial or complete machines. It allows multiple applications to run on different virtual machines (VMs) by hiding the technical complexity from users [8]. The problem of virtual machine (VM) allocation has become a challenging problem and a favored research topic for improving power and thermal efficiency in cloud infrastructures [9]. Most of the research studies concentrate on only one specific objective of management, such as minimizing wastage of resources, minimizing power consumption or balancing thermal distribution. When considered these objectives all together, it may lead to an obvious conflicting. For one hand, allocating VMs on a small number of servers and turning off other servers is an effective way to reduce power consumption and energy costs. For another hand, accumulation workloads on a subset of server's resources can cause heat imbalances that create hot spots. Hot spots may degrade server performance and cooling costs so an effective strategy should consider tradeoffs among all these objectives [10]. The importance of needing efficient methods for VMs allocation in physical machines is addressed in several research works. Many meta-heuristic algorithms have been proposed for VM placement problem [4] [11]. The Particle Swarm Optimization (PSO) is a metaheuristic inspired by the social foraging behavior of some birds and the schooling behavior of fish. PSO solves successfully many optimization problems likes graph coloring problem, vehicle routing problem, traveling salesman problem and scheduling problem [11] [12]. The basic idea that can be concluded from PSO is to simulate the behavior of birds when trying to search for the food sources. During a tour, a group of particles adjusts their values closer to the value of member who is very close to the target at any given moment [13] [14]. In this paper, the problem of VM allocation is formulated as a multi-objective optimization problem aiming to minimize power consumption, total processing resource wastage and the peak temperature among the servers. PSO algorithm is proposed and designed to deal effectively with the formulated VMs allocation problem. The performance of the proposed PSO with fuzzy multi-objective is compared to first-fit decreasing (FFD) [15], best-fit decreasing (BFD) [16], Max-Min Ant System (MMAS) [17], The Multi-objective Grouping Genetic Algorithm (MGGA) [10] and Virtual machine placement based on ant colony system (VMPACS) [4]. The experimental results state that the proposed method compete efficiently the compared algorithms to the VM allocation problem. The rest of this paper is organized as follows. The background and the related work are presented in section 2. Section 3 formulates the VM allocation problem. The details about proposed PSO with fuzzy multi-objective are presented in section 4. The implementations of the proposed PSO algorithm and simulation results are covered in section 5. Finally, section 6 concludes this paper.

2. Background and Related Work

2.1. Green Cloud Computing

Cloud computing allows consuming large amounts of power by using pool of resources and offering a single system view for cloud consumer [18]. It is defined as a large pool of usable virtualized resources such as software, hardware and development platforms. This pool of resources is dynamically managed for best resource utilization based on service level agreements established through negotiation between the service provider and consumers [19]. It is the responsibility of the provider to manage his resources in an efficient way to make the needed resources available on demand to the consumers. The growing direction to using cloud, increases the energy consumption that has become a critical challenge for society and industry. Increasing energy consumption also increases carbon emission. High energy cost decreases cloud providers’ profit and high carbon emission is very bad for the environment. Hence, energy efficient solutions are demanded. This scenario is simulated in Fig. 1. [20].

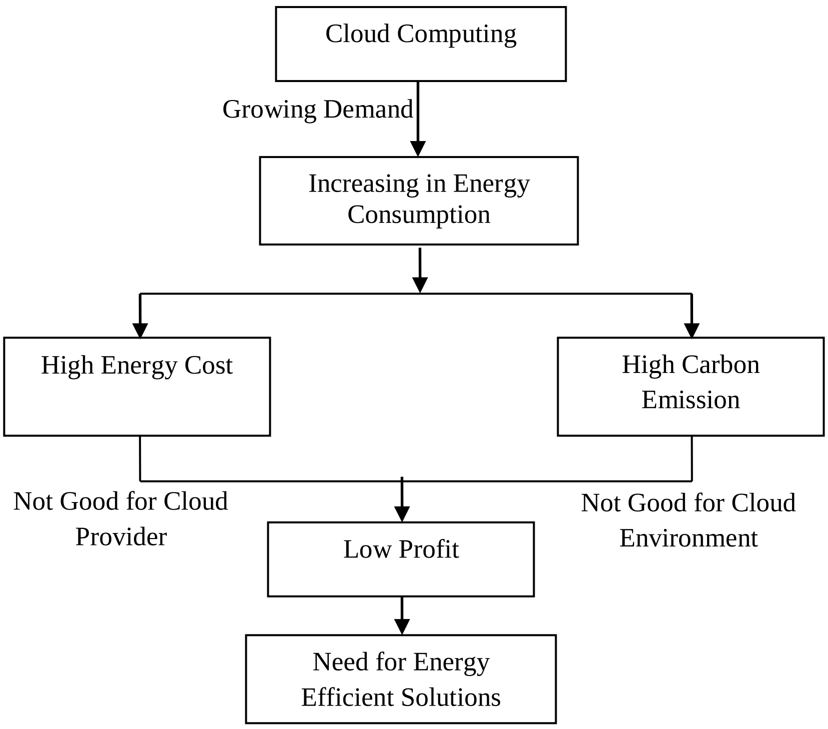


Fig. 1: Cloud computing energy issues

Taking the power as an objective during VMs allocation improves performance and utilization that will decrease the energy consumption and reduce carbon emission [4] [6]. So this goal is required to achieve green computing as follow.

1. Reducing energy consumption avoids high costs by allocating the VMs across the servers of a cloud in a way that helps in decreasing the amount of the consumed power.
2. As the consumed power is decreased, the carbon emission will be decreased that helps in achieving green computing [20].

2.2. Particle Swarm Optimization (PSO) and Metaheuristic

Metaheuristics have a popularity in solving complex problems. Metaheuristic can be defined as an iterative function which combines different concept intelligently to obtain near-optimal solutions efficiently. It has the ability to explore a search space efficiently and effectively by using two contradictory criteria: exploration and exploitation. This class of algorithms includes ant colony system, PSO, genetic algorithms, simulated annealing and tabu search [21]. PSO was introduced in 1995 [14]. The particles of PSO fly over an environment by biasing their direction toward good areas. The goal of the PSO algorithm is exchanging information to share experiences of searching. The PSO algorithm gives to all particles initially random positions and assigns small random velocities for each one [22]. The PSO algorithm is working as a simulation, changing the position of each particle using its velocity that is computed by best position and best global position. Over iterations, the particles reach together around good solution [23]. Fig. 2 shows the pseudo code of the standard PSO algorithm [13].

```
Initialize all particles randomly
Repeat
   Evaluate f (xi)
   For each particles i
       Update velocities by Eq. (1)
       Move to the new position by Eq. (2)
      If f (xi) < f (pBesti) Then pBesti = xi
      If f (xi) < f (gBest) Then gBest = xi
   EndFor
Until reaching stopping criteria
Output: gBest solution.
```

Fig. 2: Pseudo code of standard PSO.

A particle has its own position and velocity, which means the flying direction of the particle. At each iteration, a particle moves from one position to another in the decision space. Each particle will evaluate its position in the search space according to the objective function *f*. The velocity value of a particle is calculated according to how far an individual is from the target. The velocity value is computed by Eq. (1).

$$V_i(t+1) = V_i(t) + U_1 C_1 \times (pBest_i - X_i(t)) + U_2 C_2 \times (gBest - X_i(t)) \tag{1}$$

Where, $V_i(t+1)$ represents the new velocity of a particle and $V_i(t)$ represents its current velocity. U_1 and U_2 are two random variables in the range [0, 1]. The constants $C_1$and C_2 represent the learning factors. The parameter C_1 is the self-learning factor and the parameter C_2 is the social learning factor. The *x-vector* records the current position of the particle in the search space. Each particle keeps track of its achieved best fitness value called personal best ($pBest_i$). Another best value that is tracked by the PSO is the best value obtained by any particle in the neighborhood of that particle called global best (*gBest*) [13]. After updating the velocity of each particle, each particle will moves to the new position in the decision space by Eq. (2).

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{2}$$

2.3. Fuzzy Logic

Fuzzy Logic (FL) was initiated in 1965. FL uses intermediate values between conventional evaluations like tall/short, 1/0 etc. There are many alternatives can be found between the boundaries 0 and 1, namely [0~1] interval [24]. The number one means that the element is belongs to the set S and the number zero means that the element is not belongs to the set S. Other values refer to a gradual membership belongs to the set S. Fig. 3 shows an example of the membership functions that represent three fuzzy sets for the variable "height".

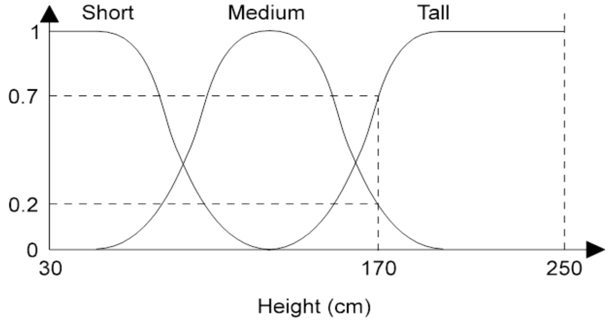


Fig 3. Membership functions of three fuzzy sets for the variable "height"

The membership function is a representation of the affection of participation of each input. It assign a weight with each input, determines the overlap between inputs, and generates the output response. The fuzzy rule-based system is the most important applications of FL. These systems use “IF-THEN” rules whose antecedents blocks and consequents blocks use FL statements to extract the knowledge. The rules takes the inputs as a membership values that are weight factors to aggregate their influence on the final conclusion of fuzzy output. The fuzzy outputs from all rules are finally accumulated to one fuzzy set to get a crisp decision from this fuzzy output. FL is considered as a well tool for complex and controlling industrial processes, as well as it is a preferable method for expert knowledge and handling conflicting goals [25].

2.4. Related work

Because of the fundamental significance of VM allocation optimization, extensive study has been made in that field and many algorithms exist in the literature. The benefit of packing VMs efficiently in server consolidation is handled in [26]. A simple process for VM allocation has been introduced in [27]. This simple method begins by choosing a target host with compatible requests. After that, the first VM will be placed on the first server. The second VM will be on the same server only if it can satisfy the requirements. If the current server cannot satisfy the requirements, a new server is appended and the VM will be mapped to it. These steps will be continued until all the VMs have been placed. The best-fit decreasing (BFD) and the first-fit decreasing (FFD) algorithms were used to pack list of VMs into a minimal number of hosts and deal with the VM allocation problem as a bin packing problem [15] [16]. Traditional analytical approaches based on linear and quadratic programming are proposed in [28] to minimize the number of used hosts. The linear programming formulations of host consolidation problems were covered in [29]. This approach restricts the number of VMs in a single physical host ensuring that some VMs are assigned to different physical hosts and the total number of migrations will be limited. A genetic algorithm was proposed to adaptively self-reconfigure the VMs in cloud data centers that hold heterogeneous servers [30]. The Multi-objective Grouping Genetic Algorithm (MGGA) was proposed in [10] for combining possibly conflicting objectives when searching the solution space. Max-Min Ant System (MMAS) metaheuristic based single-objective to minimize the required number of physical hosts was proposed in [17]. VMPACS algorithm for the VMs allocation is proposed in [4].The VMPACS algorithm aims to collect a set of non-dominated solutions by using ant colony system that simultaneously decreases power consumption and resource wastage. The performance of the VMPACS algorithm outperformed a single-objective ACO algorithm and a multi-objective grouping genetic algorithm (MGGA) as in [4]. The virtual machine placement based on Ant Colony Optimization (ACO) for minimizing resource Wastage is proposed in [31]. In MGGA approach, the problem of VM allocation is formulated as a multi-objective optimization problem that enhances simultaneously total memory resource wastage and total processing resource wastage. After that ACO algorithm is proposed for solving the formulated problem. In this paper, VMs allocation based on PSO with Fuzzy Multi-Objective (PSOFM) is designed and proposed to optimize the total power consumption, processing resource wastage and the peak temperature of the servers.

3. Problem Formulation

3.1. The Processing Wastage Model

The resources wastage of processing from used server may be differed greatly with different VMs allocation solutions. To fully utilize the available resources, the Eq. (3) is used to calculate the potential wasted CPU processing. The processing resource wastage measures the total CPU resource wastage with respect to total

CPU resources usage. The effective target from the VMs allocation is that W_{pj} must be minimized as much as it can.

$$W_{pj} = \frac{T_{pj} - U_{pj}}{U_{pj}} \tag{3}$$

Where, W_{pj} denotes the processing resource wastage. T_{pj} is the threshold of CPU processing utilization associated with $server_j$. U_{pj} is the total used CPU processing. The main idea of using the above threshold for the processing utilization is that 100% utilization leads to server performance degradation. Another reason the VM live migration needs some amount of processing [4].

3.2. Power Consumption Model

The consumed power of the server can be estimated by a linear relationship between its CPU utilization and its power consumption. Moreover, this linear relationship is also posted by and their idle power does not be considered as part of the total energy consumption. The power consumption of the $server_j$ is defined as a function of the CPU processing utilization as in Eq. (4).studying conducted on a Dell server in [10]. In order to save energy, idle servers are turned off.

$$P_j = (P_j^{busy} - P_j^{idle}) \times U_{pj} + P_j^{idle} \tag{4}$$

Where, P_j^{busy} is the power value when $server_j$ is fully utilized and P_j^{idle} is the power when $server_j$ is idle.

3.3. Thermal model

Thermal performance is an important factor in cloud resources management. The generated hotspots from fully server utilization may lead to disruptive downtime. Eq. (5) is used to handle the temperature of a server as in [31].

$$T_j = P_j \times R + T_{am} \tag{5}$$

Where, P_j denotes the power consumption of $server_j$, R denotes the thermal resistance, and T_{am} is the ambient temperature. The goal of thermal management is keeping the temperature of used servers within a safe operating range.

3.4. Formulation of VMs Allocation Problem

Suppose that n VMs need to be allocated on m servers. The variables i and j are used to index the VM and the server respectively considering that none of the VMs needs more resource than can be supplie by a single server. D_{pi} is the processing demand of each VM while D_{mi} is the memory demand of each VM. Two binary variables x_{ij} and y_j are used such that the binary variable x_{ij} indicates if VM_i is assigned to $server_j$ or not and the binary variable y_j indicates whether $server_j$ is in use or not. The proposed algorithm objective is to power consumption. The VM allocation optimization problem can be formalized as follows.

$$Min \sum_{j=1}^{m} W_{pj} = \sum_{j=1}^{m} (y_j \times \frac{(T_{pj} - \sum_{i=1}^{n}(x_{ij} \times D_{pi}))}{\sum_{i=1}^{n}(x_{ij} \times D_{pi})}) \tag{6}$$

$$Min \sum_{j=1}^{m} P_j = \sum_{j=1}^{m} \left(y_j \times \left((P_j^{busy} - P_j^{idle}) \times \sum_{i=1}^{n} (x_{ij} \times D_{pi}) + P_j^{idle} \right) \right) \tag{7}$$

$$Min\ \ Max\ \ T_j = P_j \times R + T_{am} \tag{8}$$

Subject to:

$$\sum_{j=1}^{m} x_{ij} = 1\ \forall i \in I \ and\ \forall\ j \in J \tag{9}$$

$$\sum_{i=1}^{n} (x_{ij} \times D_{pi}) \leq T_{pj} \times y_j \tag{10}$$

$$\sum_{i=1}^{n} (x_{ij} \times D_{mi}) \leq T_{mj} \times y_j \tag{11}$$

$$x_{ij}, y_j \in \{0,1\}\ \forall i \in I \ and\ \forall\ j \in J \tag{12}$$

The first objective function is to minimize the total processing resource wastage from all the servers, the second objective function is to minimize the power consumption by all the servers and the third objective function is to minimize the peak temperature among the servers. Allocating a VM to only one server is treated by constraint (9). Constraint (10) model the processing capacity constraint of the server while the memory capacity of the server is handled by constraint (11). Constraint (12) specifies the variables domain of the problem. There are a total of m^n possible VM allocation solutions if we have n VMs and m physical servers. For example, the number of possible solutions to place 25 VMs on 10 physical servers just computes 10^{25} =10,000,000,000,000,000,000,000,000 possible solution. Even if 1 quadrillion solutions can be compared per second by an efficient mainframe computer, examining all 10^{25} possible solutions would demand more than three hundred years. Therefore, it is not logic to make a complete listing and enumeration of all possible solutions to find the optimal solutions.

4. The Proposed PSO with Fuzzy Multi-Objective (PSOFM)

This section shows the details of how to apply the proposed algorithm based on PSO to efficiently search for an acceptable solution in large solution spaces. The pseudo code of the proposed PSO with the fuzzy multi-objective procedure is shown in Fig. 4.

```
Input: VMs set and servers with utilization thresholds set
Output: The acceptable solution (near-optimal) for VMs allocation
1. Initialize parameters:
Set value for Number_of_particles, tmax, V_Max.
Start t from 1.
Assign gBest=null.
Generate for each particle solution randomly
2.  For each particle
   {
     Calculate solution fitness value using FL by Eq. (13)
     If the fitness value is more efficient than gBest
             Update gBest
    End If
    }
3.  For each particle
    {
    Compute particle Velocity
    Use velocity to upgrade particle location
    Calculate solution fitness value using FL by Eq. (13)
     If the fitness value is more efficient than pBest
             Update pBest
     End If
    If pBest is more efficient than gBest
             Update gBest
    End If
    }
4. Increment t by one.
5. If (t <t_max)
      Goto step 3
   Else
     Print out the gBest
  End If
6. Return
```

Fig 4. Pseudo code of the proposed PSO with fuzzy multi-objective (PSOFM)

PSOFM proceeds with some particles, each maintaining one potential solution to the entire VMs allocation problem. The position of a particle is being placed in a search space randomly. The PSO algorithm tracks the overall best solution found by any particle in the PSO and the fitness value of the solution here is evaluated using the proposed fuzzy multi-objective optimization.

4.1. Fuzzy Logic Based Multi-Objective Optimization

The VMs placement problem considers three conflicting objectives (processing resource wastage, power consumption and temperature). A fuzzy set is defined for each objective. The membership functions of these three fuzzy set are decreasing function which means the smaller value is a higher degree of satisfaction. The following fuzzy rule is used to represent the evaluation of the solutions.

If solution s_1 has Lower Processing Resource Wastage (*LPRW*), Lower Power Consumption (*LPC*) and Lower Temperature (*LT*) than solution s_2 then s_1 is a better solution than s_2.

The solution with the highest membership in the fuzzy sets *LPRW*, *LPC* and *LT* is the most efficient solution. The Eq. (13) that evaluates the above fuzzy rule, uses the weighted-averaging fuzzy operator as in [32].

$$\mu(sol) = \beta \min(\mu_W(sol)\mu_P(sol)\mu_T(sol)) + (1-\beta) avg(\mu_W(sol)\mu_P(sol)\mu_T(sol)) \qquad (13)$$

Where, $\mu(sol)$ is the membership value for solution *sol*. $\mu_W(sol)$ represents the membership degree of solution *sol* in the fuzzy set defined by *LPRW*. $\mu_P(sol)$ represents the membership degree of solution *sol* in the fuzzy set defined by *LPC*. $\mu_T(sol)$represents the membership degree of solution *sol* in the fuzzy set defined by *LT*. Finally, β is set to 0.5. The membership functions for *LPRW*, *LPC* and *LT* are linear decreasing function. The lower bound and the upper bound of the membership functions are as follows. The maximum number of servers needed to serve all the VMs is m_{max} = min(m,n) and the minimum number of servers needed to serve all the VMs m_{min}=max((CPU requirements of all VMs/server CPU capacity), (memory requirements of all VMs/server memory capacity)). The lower bound of power consumption P_{low}= $(m_{\min} P_j^{idle} + ((P_j^{busy} - P_j^{idle}) \times \sum_{i=1}^{n} (D_{pi}))$. The upper bound of power consumption P_{up}= $(m_{max} P_j^{idle} + ((P_j^{busy} - P_j^{idle}) \times \sum_{i=1}^{n} (D_{pi}))$ as in [10]. The lower bound of CPU temperature is T_{low}= $P_j^{idle} R + T_{am}$ and the upper bound of CPU temperature is T_{up}= $P_j^{busy} R + T_{am}$

4.2. Generation Random Solutions for Particles

In the initialization phase of the proposed PSOFM algorithm, the total number of used particles is assigned and the other parameters are initialized. The *V_Max* variable determines the allowed maximum velocity. The *gBest* variable is set to null and *t* will be 1 to refer that the PSO starts with the first iteration (*t* used to index the iteration). The t_{max} variable refers to the maximum number of allowed iterations allowed. The PSO algorithm includes two parts, randomly generating a solution for each particle and regenerating a new solution from an existing solution. The initial solution for a particle is generated as follows. Sort the list of physical servers randomly. After that, the list of VMs is allocated on the randomly sorted physical servers using the best-fit algorithm. By using this approach, different initial solutions for particles are produced.

4.3. Iteration Part of the Proposed PSOFM

The PSO iterative phase simulates the real behavior of particles. During an iteration, the *pBest* value of each particle is upgraded when this particle found fitness value of the reached solution is more efficient than its *pBest*. The value of *gBest* will be altered only when any particle's *pBest* value is more suitable than it. The *gBest* variable gradually moves closer to the optimal solution during reaching the stopping criteria. The velocity value of each particle is computed by Eq. (1) using the proposed fuzzy multi-objective optimization. After evaluating velocity for a particle, the velocity is compared with *V_Max*. If it is over than the determined value, it will be reset to equal *V_Max*. Once the velocity of the particle has been calculated, the new position is reached by swapping VMs within current particle with VMs of its nearest particle.

5. Implementation and Experimental Results

A lot of researchers depend on the CloudSim platform for simulating cloud environment because it can imitate host, service brokers, data centers, scheduling and allocation policies of cloud platform [33]. In the experiments that are implemented here using CloudSim platform, problem instances are randomly generated. The instances were a demanding set of CPU processing and memory capacity for different numbers of VMs. The total number of servers was set to equal the number of VMs in order to handle the worst VM allocation scenario, in which only one VM is assigned per a server. After PSOFM with fuzzy multi-objective was finished, the VM placement is applied by the reached *gBest* solution. The VM allocation algorithms to be compared in the experiments include: first-fit decreasing (FFD) [15], best-fit decreasing (BFD) [16], Max-Min Ant System (MMAS) [17], Multi-objective Grouping Genetic Algorithm (MGGA) [10], Virtual machine placement based on ant colony system (VMPACS) [4] and the proposed (PSOFM).

First-fit-decreasing (FFD) places VMs in a decreasing order of size after that the next VM is allocated to the first available server. FFD-CPU is the FFD solution sorted by VM CPU processing requirements and FFD-MEM represents the FFD solution sorted by memory requirements. Best-fit-decreasing (BFD) likes FFD but it places a VM in the fullest server that still has enough capacity. BFD-MEM and BFD-CPU are the BFD solutions sorted by memory requests and CPU processing requests respectively. Pseudo codes for the compared algorithms were coded by using java language under CloudSim platform that ran on an Intel(R) Core(TM) 2 Duo CPU with 2.20 GHz 2.20 GHz and 2 GB RAM.

The parameter settings of MGGA algorithm are as follows. The population size is 12. The initial population was generated randomly. The crossover rate in MGGA is 0.7 and the mutation rate equals 0.05. The maximum number of generations for each search process is 10 as in [10]. In the case of VMPACS algorithm, NA (number of ants) = 10, t_{max} (number of iterations) = 100, $\alpha = 0.45$, $\rho_l = \rho_g = 0.35$, and $q_0 = 0.8$ as in [4].

The total CPU resources wastage and the total memory resource wastage are shown in Fig. 5 - Fig. 8. It is shown from these figures that the proposed PSOFM algorithm can find the solutions with high resource utilization compared to FFD-CPU, FFD-MEM, BFD-CPU, BFD-MEM, MMAS, MGGA and VMPACS algorithms and produces the lowest resource wastage in cases of a different number of VMs allocation. The proposed PSOFM algorithm outperforms other algorithms because it is able to search the solution space more efficiently based on models for minimizing total CPU processing resources wastage, the total power consumption and peak temperature. The MMAS, MGGA, VMPACS and PSOFM algorithms take into account the resources wastage when searching for the near-optimal placement. This is the reason that those algorithms outperform the FFD-CPU, FFD-MEM, BFD-CPU and BFD-MEM algorithms.

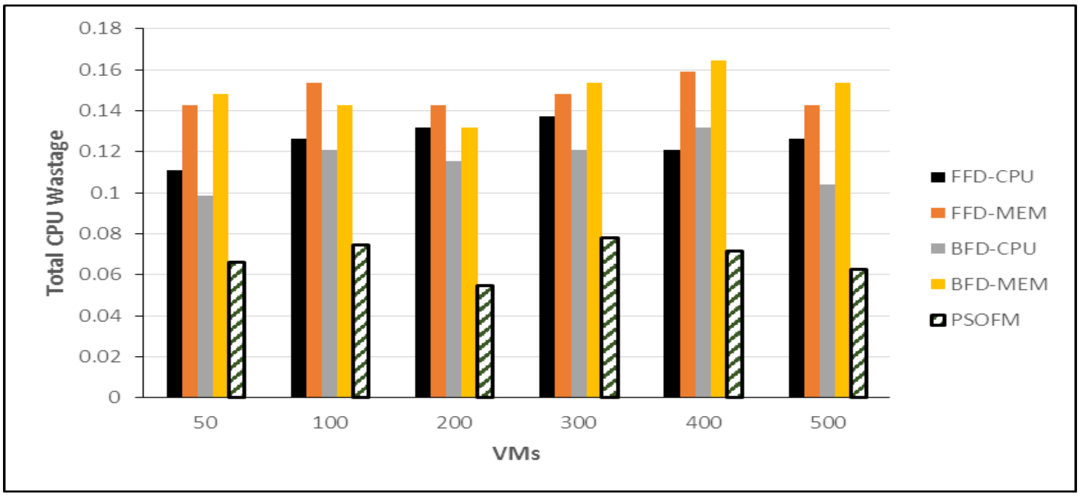


Fig 5. Total CPU processing resources wastage for FFD-CPU, FFD-MEM, BFD-CPU, BFD-MEM and PSOFM

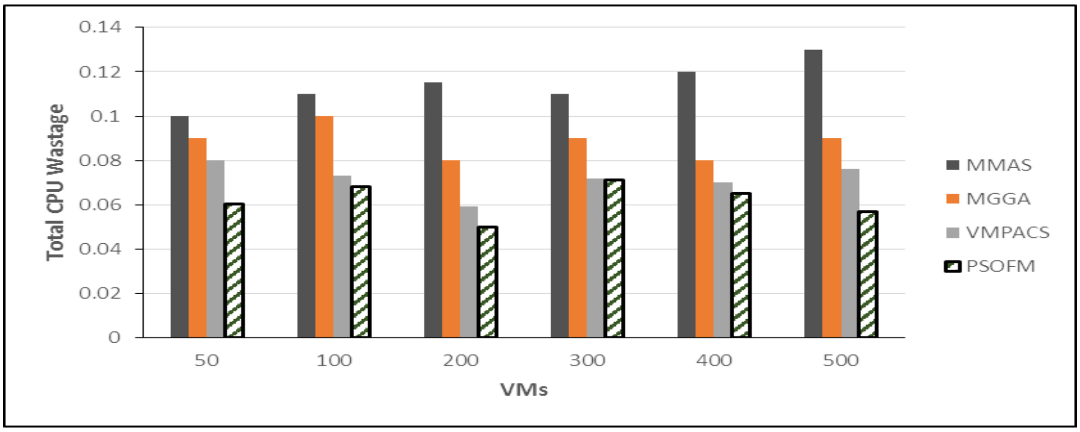


Fig 6. Total CPU processing resources wastage for MMAS, MGGA, VMPACS and PSOFM

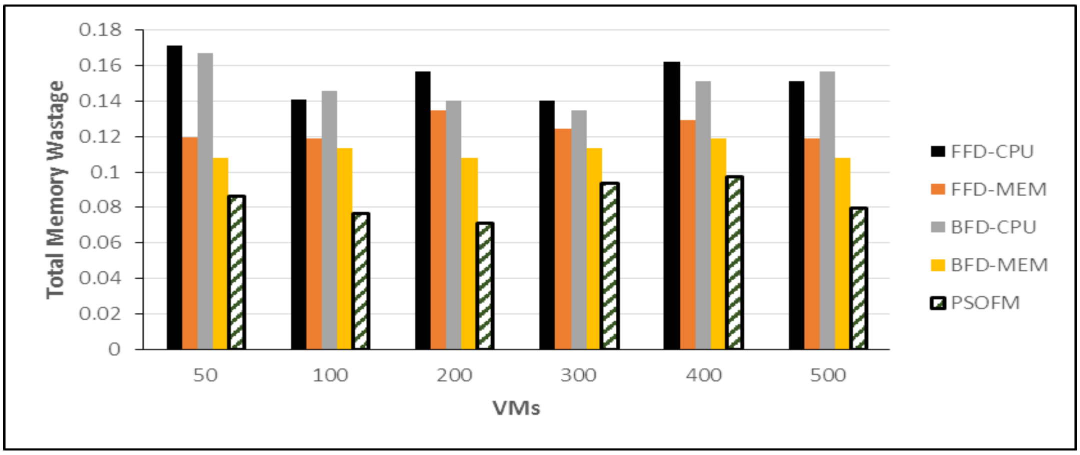


Fig 7. Total memory processing resources wastage for FFD-CPU, FFD-MEM, BFD-CPU, BFD-MEM and PSOFM

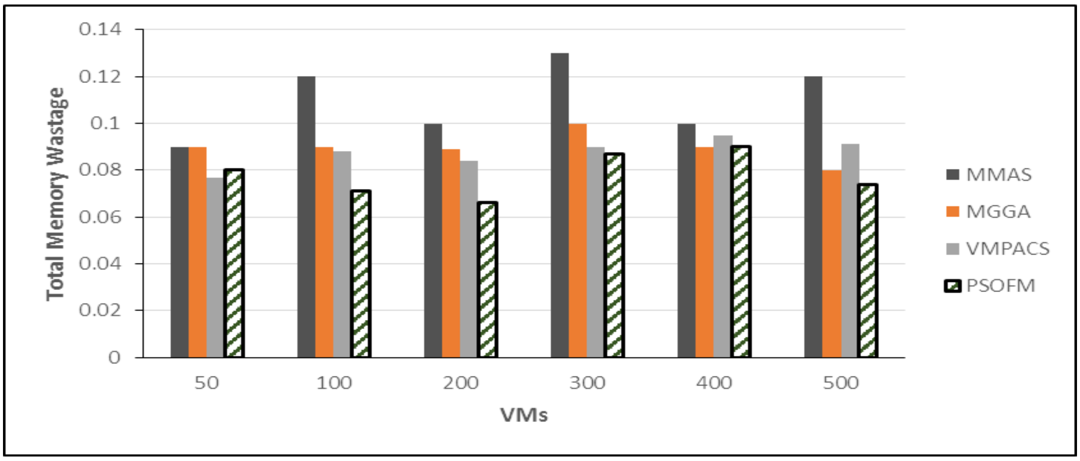


Fig 8. Total memory processing resources wastage for MMAS, MGGA, VMPACS and PSOFM

The proposed PSOFM algorithm outperforms other algorithms because PSOFM deals with the CPU resources wastage as a separated objective function. It has its own measure that should be optimized. The PSOFM algorithm tracks the overall best solution found by any of the particles in the PSO and the fitness value of the solution is evaluated using the proposed fuzzy multi-objective optimization. So, the PSOFM algorithm can find the solutions with a smaller number of used servers and high resource utilization compared to other algorithms

Fig. 9 and Fig 10 compare the total power consumption for each algorithm. The values of P_j^{busy} *and* P_j^{idle} have been fixed to 215 and 162 Watt according to the measurements performed on a Dell server

[4] [10]. The proposed PSOFM produces relatively low values for power consumption because it takes the objective function of power consumption into consideration besides other objective functions and achieves a good balance among different goals. It tries to find solutions that optimize power consumption, temperature performance and resources utilization. FFD-CPU, FFD-MEM, BFD-CPU and BFD-MEM algorithms yield the highest power consumption, CPU resource wastage and memory resource wastage because they tend to use a larger number of servers compared with reminder algorithms.

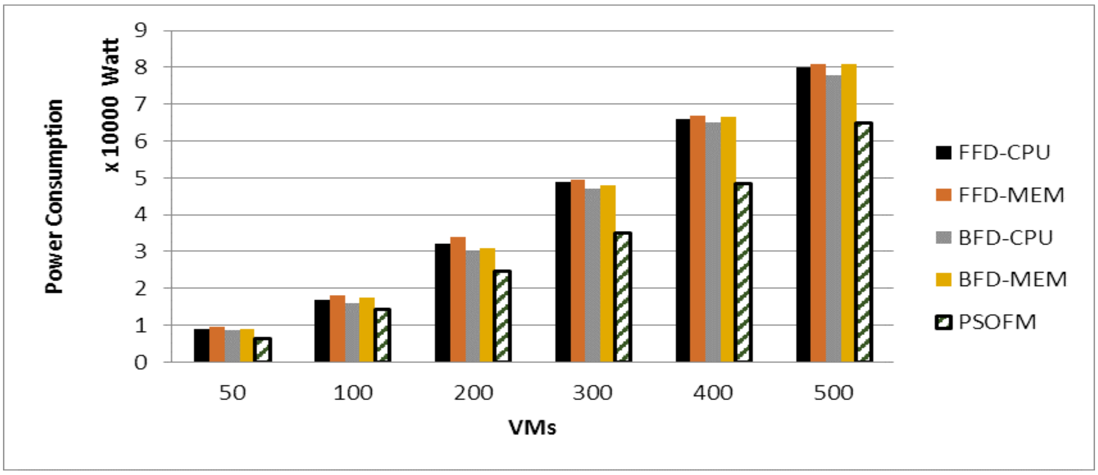


Fig 9. Total power consumption for FFD-CPU, FFD-MEM, BFD-CPU, BFD-MEM and PSOFM

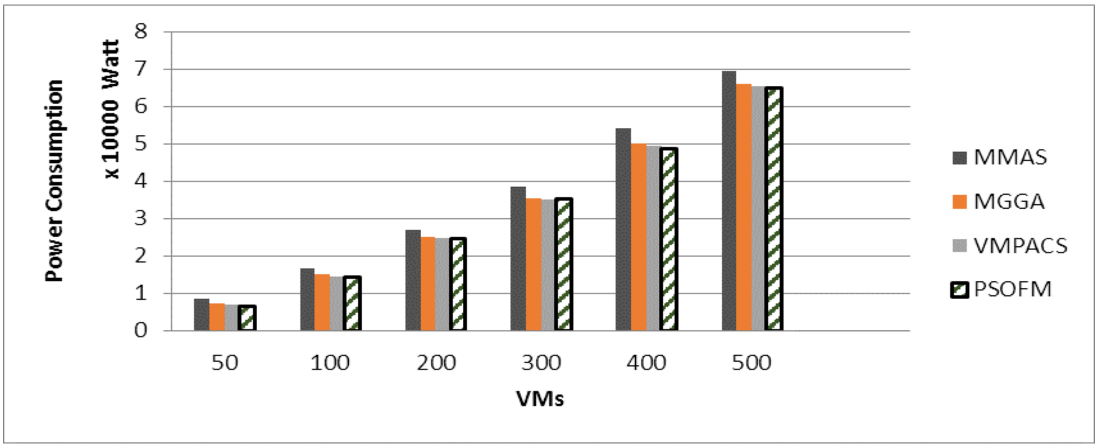


Fig 10. Total power consumption for MMAS, MGGA, VMPACS and PSOFM

Fig. 11 and Fig. 12 compares the peak temperature among the servers for each algorithm. Thermal performance that is one of the critical issues in data-center management, can be computed using Eq. (14).

$$Eff\,(T) = 1 - \left(\frac{T_j - T_{low}}{T_{high} - T_{low}} \right)^d \qquad (14)$$

Where *Eff(t)* is temperature efficiency, T_j is the temperature of $server_j$, T_{low} and T_{high} represent the temperature of idle server and overloaded server respectively and *d* represents the degree [10]. Temperature efficiency value decreases rapidly when the CPU temperature goes over the safe range. Fig. 13 and Fig. 14 compare the temperature efficiency of each algorithm using T_j equal the peak temperature among the servers. T_{low} and T_{high} are assumed to be 15^{o}c and 55^{o}c respectively as in [10] and *d* assumed to be 1.

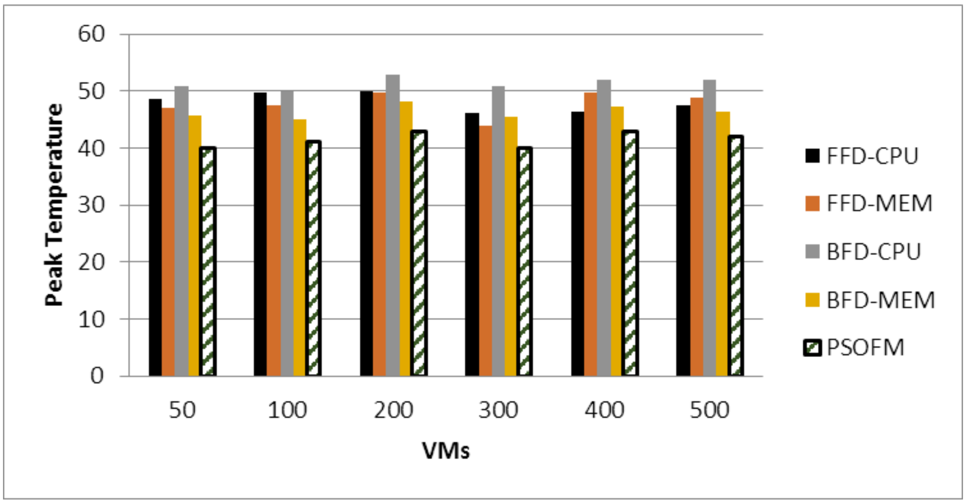


Fig 11. Peak temperature among the servers for FFD-CPU, FFD-MEM, BFD-CPU, BFD-MEM and PSOFM

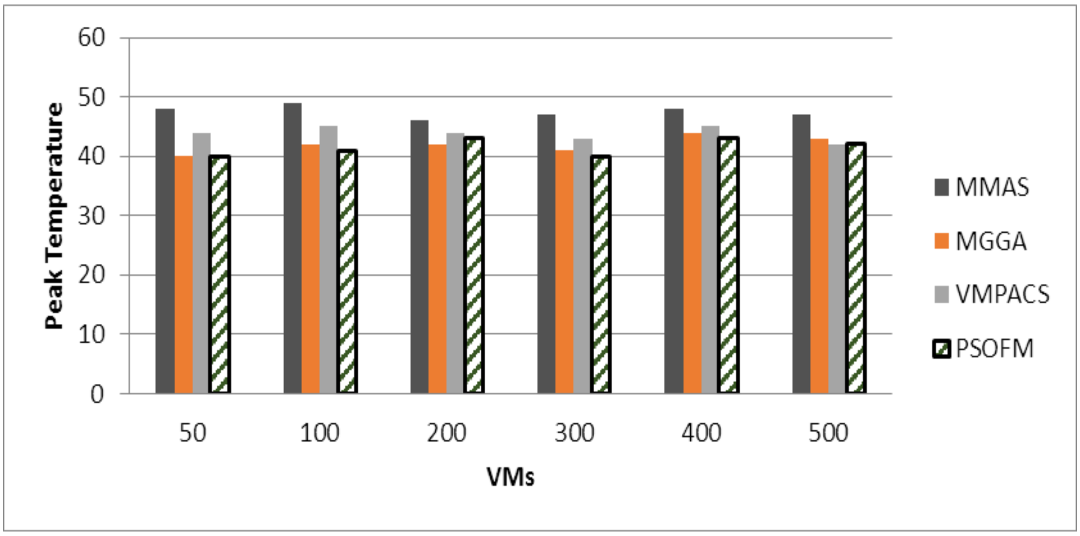


Fig 12. Peak temperature among the servers for MMAS, MGGA, VMPACS and PSOFM

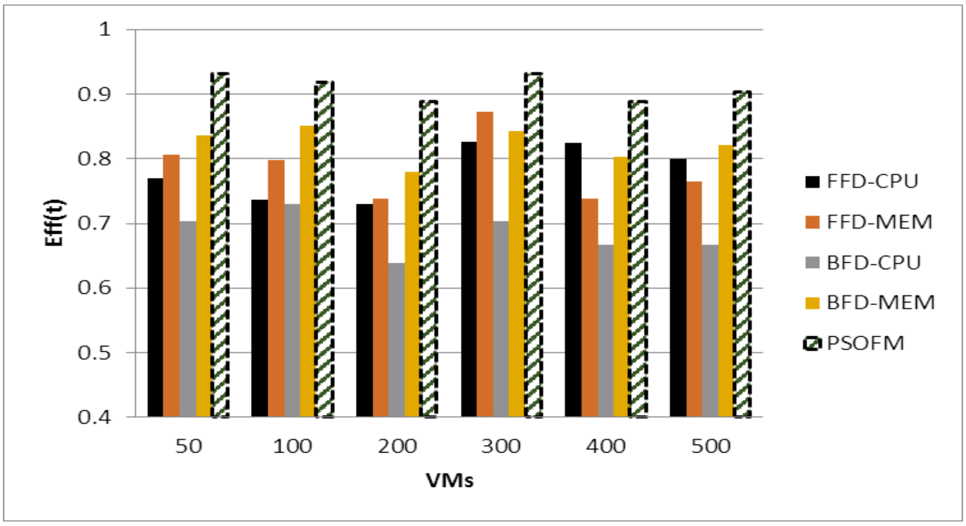


Fig 13. Temperature efficiency (Eff(t)) for FFD-CPU, FFD-MEM, BFD-CPU, BFD-MEM and PSOFM

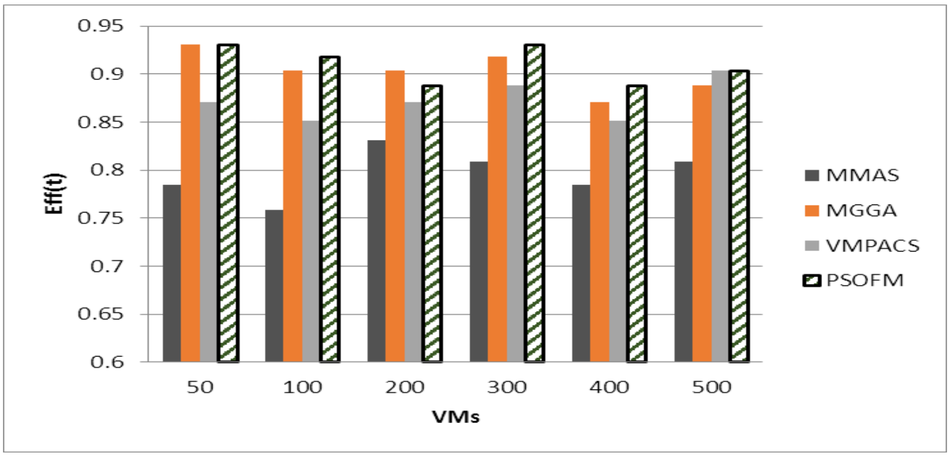
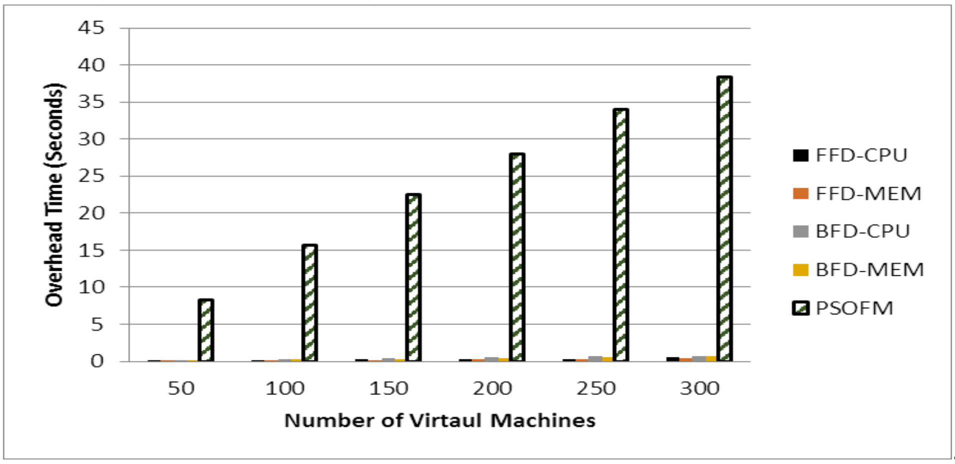


Fig 14. Temperature efficiency (Eff(t)) for MMAS, MGGA, VMPACS and PSOFM

The key observations concerning Fig. 11 - Fig. 14 are as follows. The temperature efficiency and peak temperature do not depend on the total power consumption for all used servers but they depend only on the highest power consuming server. The temperature efficiency of the proposed PSOFM algorithm is high and they have low value for peak temperature because it tends to consolidate VMs into an acceptable number of servers, resulting in efficient resource utilization and low CPU temperature. The VMs placement by the proposed PSOFM algorithm considers three conflicting objectives (processing resource wastage, power consumption and temperature). A fuzzy set is defined for each objective. The PSOFM algorithm searches and keeps track the solution with the highest membership in the fuzzy sets.

Fig. 14 and Fig. 15 draw the time required to generate a solution with different problem sizes. The proposed PSOFM algorithm takes less than one minute to obtain a solution for allocating 500 VMs. It is clear that the execution time is approximately linear with respect to the values of VMs. Therefore, it can be said that the proposed PSOFM algorithm is suitable for large numbers data centers.



.

Fig 15. Overhead(running time relative to number of VM requests) for FFD-CPU, FFD-MEM, BFD-CPU, BFD-MEM and PSOFM

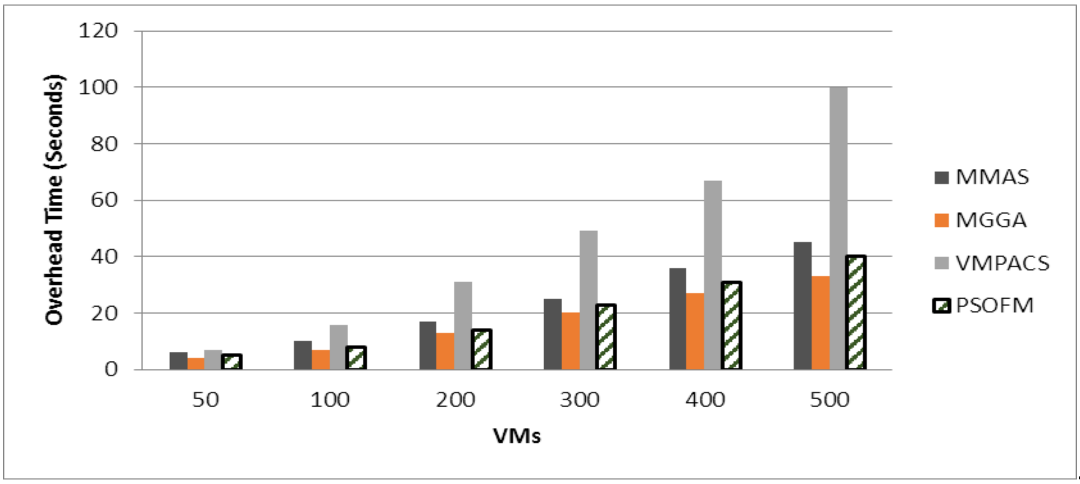


Fig 16. Overhead(running time relative to number of VM requests) for MMAS, MGGA, VMPACS and PSOFM

6. CONCLUSIONS AND FUTURE WORK

This research paper experimentally demonstrates that the proposed particle swarm optimization with a fuzzy multi-objective algorithm for the virtual machine allocation problem is more convenient than first-fit decreasing, best-fit decreasing, max-min ant system, multi-objective grouping genetic and virtual machine placement based on ant colony system algorithms. The virtual machines allocation problem is formulated in this paper as a multi-objective optimization problem that works simultaneously toward total processing resource wastage, power consumption and peak temperature enhancement. The proposed particle swarm optimization with the fuzzy multi-objective algorithm is developed to effectively handle the potential large solution space. It follows the overall best solution found by any of the particles using the proposed fuzzy multi-objective rule. The proposed particle swarm optimization with the fuzzy multi-objective approach was studied with respect to its performance by simulation based experiments. The results demonstrate that proposed algorithm is the corroborative and outperforms well-known algorithms for virtual machines allocation problem. In future work, the carbon emission will be handled.

REFERENCES

[1] A.O. Akande, N.A. April, and J.V. Belle, Management Issues with Cloud Computing, *in Proceedings of the Second International Conference on Innovative Computing and Cloud Computing, New York, NY, USA*, 2013, pp. 119-124.

[2] R, Buyya, J. Broberg, and A.M. Goscinski, Cloud Computing Principles and Paradigms, *Wiley Publishing*, 2011.

[3] P. Balaji and R. Buyya, Special issue on Cluster, Grid, and Cloud Computing, *Future Generation Computer Systems, 29(8)*, 2013.

[4] Y. Gao, H. Guan, Z. Qi, Y. Hou, and L. Liu, A multi-objective ant colony system algorithm for virtual machine placement in cloud computing, *Journal of Computer and System Sciences,79(*8),2013,1230-1242.

[5] K. Chandrasekaran, Tutorial: Resource Management in Cloud Computing, *in Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing, Washington, DC, USA*, 2013.

[6] M. Cardosa, M.R. Korupolu, and A. Singh, Shares and utilities based power consolidation in virtualized server environments, , *IFIP/IEEE International Symposium on in Integrated Network Management*, June 2009, 327-334.

[7] F. Durao, J.F. Carvalho, A. Fonseka, and V.C. Garcia, A Systematic Review on Cloud Computing, *J. Supercomput., 68(3),* 2014,1321-1346.

[8] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility, *Future Generation Computer Systems, 25(6)*, 2009,599-616.

[9] L. Grit, D. Irwin, A. Yumerefendi, and J. Chase, Virtual Machine Hosting for Networked Clusters: Building the Foundations for "Autonomic" Orchestration, *in Proceedings of the 2Nd International*

Workshop on Virtualization Technology in Distributed Computing, Washington, DC, USA, 2006.

[10] J. Xu and J.A.B. Fortes, Multi-Objective Virtual Machine Placement in Virtualized Data Center Environments, *in Int'l Conference on Cyber, Physical and Social Computing (CPSCom)*, 2010, 179-188.

[11] E. Pacini, C. Mateos, and C.G. Garino, Distributed job scheduling based on Swarm Intelligence: A survey, *Computers & Electrical Engineering, 40(1)*, 2014, 252-269.

[12] Q. Kang and H. He, A novel discrete particle swarm optimization algorithm for meta-task assignment in heterogeneous computing systems, *Microprocess Microsyst, 35(1)*, 2011,10-17.

[13] E. Talbi, Metaheuristics from Design to Implementation. *Hoboken, New Jersey: John Wiley & Sons, Inc.*, 2009.

[14] S. Pandey, L. Wu, S. Guru and R. Buyya , A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments, *In International conference on advanced information networking and applications. IEEE Computer Society*, 2010, 400-407.

[15] G. Dosa, R. Li, X. Han, and Z. Tuza, Tight absolute bound for First Fit Decreasing bin-packing:, *Theoretical Computer Science,510(0)*, 2013,13-61.

[16] J. Wang et al., Best fit decreasing based defragmentation algorithm in semi-dynamic elastic optical path networks, *in Communications and Photonics Conference (ACP), Asia, Nov*, 2012, 1-3.

[17] E. Feller, L. Rilling, and C. Morin, Energy-Aware Ant Colony Based Workload Placement in Clouds, *in12th IEEE/ACM International Conference on Grid Computing (GRID)*, 2011, 26-33.

[18] L.M. Vaquero, L.R. Merino, J. Caceres, and M. Lindner, A Break in the Clouds: Towards a Cloud Definition, *SIGCOMM Comput. Commun. Rev., 39(1)*, 2008, 50-55.

[19] F. Owusu and C. Pattinson, The Current State of Understanding of the Energy Efficiency of Cloud Computing, *in Proceedings of the 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, Washington, DC, USA*, 2012, 1948-1953.

[20] N.J. Kansal and I. Chana, Cloud Load Balancing Techniques: A Step Towards Green, *International Journal of Computer Science Issues, 9(1)*, 2012, 238-246.

[21] X. Yang, Metaheuristic Optimization: Nature-Inspired Algorithms and Applications, *in Artificial Intelligence, Evolutionary Computing and Metaheuristics, Springer Berlin Heidelberg,427()*, 2013, 405-420.

[22] C.D. Sudholt, Runtime Analysis of a Binary Particle Swarm Optimizer, *Theoretical Computer Science, 411()*, 2010, 2084-2100.

[23] P. Mathiyalagan, U.R. Dhepthie, and S.N. Sivanandam, Grid Scheduling using Enhanced PSO Algorithm, *international Journal on Computer Science and Engineering, 2(2)*, 2010, 145-150.

[24] R. Kruse, J. Gebhardt, F. Klawon, Foundations of Fuzzy Systems, *Wiley, Chichester* 1994

[25] H.J. Zimmermann, Fuzzy Sets, Decision Making and Expert Systems, *Boston, Kluwer* 1987

[26] W. Vogels, Beyond Server Consolidation, *Queue, 6(1)*, 2008, 20-26.

[27] G. Khanna, K. Beaty, G. Kar, and A. Kochut, Application Performance Management in Virtualized Server Environments, *in Network Operations and Management Symposium*, 2006, 373-381.

[28] S. Chaisiri, Bu-Sung Lee, and D. Niyato, Optima Virtual Machine Placement across Multiple Cloud Providers, *in IEEE Asia-Pacific Services Computing Conference (APSCC), Dec* 2009, 103-110.

[29] B. Speitkamp and M. Bichler, A Mathematical Programming Approach for Server Consolidation Problems in Virtualized Data Centers, *IEEE Transactions on Services Computing, 3(4)*, 2010, 266-278.

[30] H. Mi et al., Online Self-Reconfiguration with Performance Guarantee for Energy-Efficient Large-Scale Cloud Computing Data Centers, *IEEE International Conference on in Services Computing (SCC)*, 2010, 514-521.

[31] AKrum, The CRC Handbook of Thermal Engineering, *Frank Kreith, Ed. USA: CRC Press*, 2000.

[32] R. Yager, On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decision Making, *in IEEE Trans. on Systems, Man and Cybernetics*, 1998.

[33] R.N. Calheiros, R. Ranjan, A. Beloglazov, C.D. Rose, and R. Buyya, CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, *Software: Practice and Experience, 41(1)* , 2011,23-50, 2011.