FFD Variants for Virtual Machine Placement in Cloud Computing Data Centers

Aneeba Khalil Soomro, Mohammad Arshad Shaikh, Hameedullah Kazi
Department of Computer Science
ISRA University
Hyderabad, Pakistan

***Abstract*—Virtualization technology is used to efficiently utilize the resources of a Cloud datacenter by running multiple virtual machines (VMs) on a single physical machine (PM) as if each VM is a standalone PM. Efficient placement/consolidation of VMs into PMs can reduce number of active PMs which consequently reduces resource wastage and power consumption. Therefore, VM placement algorithms need to be optimized to reduce the number of PMs required for VM Placements. In this paper, two heuristic based Vector Bin Packing algorithms called FFDmean and FFDmedian are proposed for VM placement. These algorithms use First Fit Decreasing (FFD) technique. FFD preprocesses VMs by sorting all VMs in descending order of their sizes. Since a VM is multidimensional therefore, it is difficult to decide on its size. For this, FFDmean and FFDmedian use measures of central tendency, i.e. mean and median as heuristics, respectively, in order to estimate the size of a VM. The goal of these algorithms is to utilize the PM resources efficiently so that the number of required PMs for accommodation of all VMs can be reduced. CloudSim toolkit is used to carry out the cloud simulation and experiments. Algorithms are compared over three metrics, i.e. hosts used, power consumption and resource utilization efficiency. The results reveal that FFDmean and FFDmedian remarkably outperformed two existing algorithms called Dot-Product and L2 in all three metrics when PM resources were limited.**

Keywords*—Cloud computing; virtual machine placement; virtualization; first fit decreasing; first fit decreasing (FFD)

I. INTRODUCTION

Cloud computing is an internet based business model of computation for outsourcing computing resources such as processing power, networks, servers, storage, applications, and services [1]. NIST (National Institute of Standards and Technology) published their 16th and final definition of cloud computing, which is.

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [1].

Cloud computing provides a lot of opportunities for the IT industry. It is a rapidly enhancing and developing paradigm. The modern computational power has allowed it to become a utility that provides services to customers on a pay-as-you-go model i.e. the customers are required to pay only when they use the service. Hence, it is considered to become 5th utility [2] of our lives after other four utilities such as electricity, water, gas and telephony.

In cloud computing everything that is provided is a service. The services are available on-demand from anywhere in the world through internet. A cloud service provider provides services to its customers in three basic service models, which are:

1) Software-as-a-Service (SaaS): SaaS is software provided as a service. The software (application) is provided on demand which is built over an infrastructure and a platform. The software provided is a web application, accessible through a browser [3]. SalesForce, BaseCamp, GoToMeeting and NETSUITE are some examples of SaaS.

2) Platform-as-a-Service (PaaS): PaaS is a complete platform provided as a service to developers, which consists of all the required systems and the developing environment with underlying infrastructure. The end users (developers) of this service are allowed to develop their own software by testing, deploying and then hosting their custom web based applications [3]. Generally PaaS is a middleware (like OS) that allows communication between hardware and application [4]. Google APP Engine and IBM bluemix are some examples of PaaS.

3) Infrastructure-as-a-Service (IaaS): IaaS is a complete infrastructure provided as a Service. It provides Computing power with high level of adjustability as it allows the developers to create their own infrastructure like virtualization, etc. [3]. The customer is allowed to build their own platform and software over it. Amzaon EC2 and IBM Cloud are some examples of IaaS.

These services are highly scalable i.e. they may be increased or decreased according to the current demand of customers. The cloud customer does not need to worry about management and provisioning of resources when the demand rises or decreases. This is all done at the cloud providers' side. This relaxes customers from the hassle of managing, updating, over and under provisioning of resource (The customers and the service providers agree upon a contract called Service Level Agreements (SLAs) [27] to ensure Quality of Service (QoS).

The cloud Infrastructure is based on huge datacenter(s) comprised of millions of physical machines (PMs). These next-

generation datacenters are so powerful that millions of customers can be served. Every end user is assigned a dedicated virtual machine (VM) which eventually runs on a PM residing in a Cloud datacenter. This is accomplished by using virtualization technology which helps in sharing PM resources to multiple users by running multiple VMs on a single PM as if each VM is a standalone PM [5]. Since these PMs are hosts for VMs, PMs are also called as hosts.

A lot of electrical power is wasted due to inefficient utilization of datacenter's physical resources which results in high operational costs. About 70% of total data center's power is consumed by PMs [6]. The principal approach to power saving in cloud computing is to cut the operational costs of datacenters by efficiently utilizing the resources of a PM, in order to minimize the number of PMs in use and idle/inactive PMs are turned off or to low power mode [7]. Hence power efficient cloud computing [26] is an active research area and is often termed as Green Cloud Computing.

The VM placement (VMP) is a process of allocating VMs into PMs. A VM Placement algorithm is responsible to place/consolidate VMs into PMs. Efficient placement of VMs can reduce the number of PMs required for their accommodation, consequently increasing utilization of PM's resources and reducing power consumption. Therefore, there is a continuous need to design efficient algorithms for VMP so that VMs are packed in minimum possible PMs and resource utilization efficiency is increased by increasing utilization of each resource in a PM.

In this work two Vector Bin Packing (VBP) algorithms named as FFDmean and FFDmedian are proposed. These algorithms are essentially First Fit Decreasing (FFD) variants which are supposed to increase PM utilization to reduce number of PMs required and in turn reduce power consumption.

The remaining paper is structured as follows: Section II discusses the related work to this research. Section III formulates VMP problem as VBP problem. Section IV discusses the proposed algorithms. Section V describes the system model. Section VI explains the experimental setup. Section VII presents the results. Section VIII discusses the results. Section IX concludes this research and Section X presents the future recommendations.

II. Related Work

Cloud computing is a rapidly growing paradigm of computer science. A lot of academic and industrial research [24], [25] has been conducted in this field to enhance the quality of cloud. As our work is specific to VMP, in this section some of the work related to VMP in cloud computing is reviewed.

A. Power Aware VMP Approaches

Power conservation is a very important challenge in Cloud computing. In this section some of the VMP algorithms/policies that opt for power conservation are discussed.

1) Cutting back power usage and co_2 footprint by ECE algorithm: To become highly available, cloud data centers maintain different power sources. Mostly, these sources are not renewable and use carbon fuels to run, eventually leading to increased carbon footprint (cf) of environment which is injurious to our environment and health. The cf rate of these power sources is an important consideration, since data centers utilize electricity provided by these sources to run PMs and eventually VMs.

Khosravi et al. in [8] proposed ECE algorithm for VMP to minimize power consumption and cf. ECE integrates two parameters i.e. Power Usage Effectiveness (PUE) and cf. It aims to place a VM in to a PM so that the power consumption, cf and PUE of the PM, its datacenter and cluster is minimized. Initially, when a VM request is received, ECE sorts all the clusters by their PUExcf values in ascending order. After that for each PM in a cluster the change in power consumption (ΔP) after that VM's placement is calculated. These PMs are then sorted in ascending order of their ΔP. A suitable PM which has minimum ΔP after placement is selected. If a suitable PM is not found for the VM, algorithm tries the next cluster. The ECE algorithm decreased cf and saved power compared to many existing algorithms.

2) Power saving by demand forecasting: As cloud computing has an on-demand resource provisioning method; resource (de)allocation is dynamic. The service provider needs switched on PMs in spare to deal with dynamic resource demands. These idle PMs until they get some work waste a lot of power. On the other hand switching them on/off on demand also wastes power and takes 2-3 minutes to restart. One way to solve this issue is to standby the inactive PMs. It takes less time to restart and also consume lesser power compared to completely switching them off. If the future resource demands are estimated before time then according to the estimate some PMs could be set on standby while others shutdown to save power.

Cao et al., in [9] proposed a power efficient approach to solve VMP problem by forecasting the demand for VMs. The VMP is carried out in three steps. Initially demand forecasting is done by using Holt-Winter's exponential smoothing method assuming that in cloud computing a particular user's demands are identical and a particular demand succeeds a seasonal pattern. Next, a modification of multi-dimensional knapsack algorithm is used to allocate VMs to hosts; considering hosts as knapsacks and VMs as items with two types of costs i-e memory and cores. Finally a self-Optimizing module is used to update the forecasting parameter values in the demand forecasting model and mining the appropriate forecast periods by Hill Climbing method. The experiment result showed that proposed algorithm with forecast saves up to 60% power.

3) Power saving by decentralized VM migration technique for fault-tolerant load balancing: Mostly, centralized approaches are used for resource management in cloud computing. Management becomes easy with these approaches however they are not fault tolerant. If the centralized resource manager crashes, all of the system crashes and becomes unmanaged.

Wang et al. in [10] proposed a decentralized VM migration technique for resource management instead of using a central resource manager, for fault-tolerance and to efficiently balance load across the data center and save power. In this approach each PM sends its load information maintained in a load vector, to every other PM in the data center(s), thus all active PMs have each other's load information. The proposed DVM algorithm uses lower and upper thresholds to judge the over and underutilization of a PM's CPU respectively. VMs are migrated from PM's with over utilized CPU to avoid SLA violations. In an underutilized PM, all VMs are migrated to some other PM with minimum increment in utilization after addition of migrating VM and the PM is turned off/sleep for power conservation. DVM showed balanced resource utilization and saved up to 20% power compared to static and round robin algorithms.

4) Power saving by euclidean distance based algorithm: Srikantaiah et al. in [11] proposed a power aware multi dimensional bin-packing algorithm that uses a euclidean distance based heuristic for VMP. The algorithm considers two dimensions i.e. CPU and disk. It first finds out an optimal point where combination of CPU and disk utilization gives minimum power usage per transaction. Next, when it receives application (VM) requests it uses the euclidean distance heuristic which calculates euclidean distance of the utilization of each PM after allocation of the requested VM to the optimal point calculated in first step. The PM that has maximum euclidean distance of all is selected for VMP. Results showed that more power is saved as performance degradation tolerance is increased. At 20% tolerance the proposed algorithm used only 5.4% more power than optimal algorithm.

5) Power saving by live VM migration using vector based repacking: Consuegra et al. in [12] proposed a vector repacking algorithm called replicas to minimize operational costs of a datacenter, that include power and migration cost. This algorithm places copies of each item/VM on different PMs that are selected by replica allocation algorithm. Among these copies only one copy is considered to be active and the operations are executed on it. When a VM has an increase in demand and cannot fit anymore in the current PM where it resides, then a PM is selected by the help of active replica selection algorithm. The copy in the newly selected PM is activated, the newly activated copy is synchronized with the previous copy to update if there are any changes and the previous copy is sent to inactive mode. By using replicas algorithm migration cost is reduced or even eliminated.

6) Consolidating complementary VMs with spatial/temporal-awareness in cloud datacenters: Chen et al., in [13] proposed a spatial/temporal aware VBP algorithm for initial VMP. The algorithm forecasts resource usage patterns of VMs and packs together complementary VMs with spatial/temporal knowledge. This algorithm considers balanced utilization of CPU and memory of a PM. For e.g. a high CPU-intensive and low memory-intensive VM and a low CPU-intensive and high memory-intensive VM can be packed together to give a balanced utilization of PM resources. These types of VMs are complimentary VMs i.e. VMs for which overall requirement of every resource dimension (in the spatial space) almost arrives at their PM's capacity throughout VM lifetime phase (in the temporal space). The results show that the proposed techniques helped in utilization of PM's resources, reduction in VM migrations and active PMs in use alongwith maintaining SLAs.

B. Vector Bin Packing Algorithms for VMP

In cloud computing the VM resource demands are multi-dimensional. To solve the multidimensional VMP problem one of the solutions is to use a variant of bin packing called Vector Bin Packing (VBP) is used. Unlike the classic bin packing it considers the items and bins as multi-dimensional vectors.

1) FFD variants for VBP: VBP can be solved by classic FFD heuristic. In FFD, items are first sorted by decreasing order of their sizes and then placed in the first bin (container) they fit into. Since in VBP the items are considered as multi-dimensional, it is difficult to decide the size of an item. For this, many FFD variants have been put forward which use some heuristic to estimate the size of a multidimensional item.

Maruyama et al. in [14] proposed a generalized VBP algorithm for FFD and best fit decreasing (BFD) techniques where BFD chooses the best bin in which an item can fit into. The algorithm considers many heuristics to estimate the size of a multi-dimensional item. The heuristics include taking product of all dimensions, sum of average and standard deviation of all dimensions and different versions of sum of all dimensions. Among these they found taking sum of dimensions and sum of average and standard deviation, far better than taking product. Kou and Markowsky in [15] proposed multidimensional bin packing algorithms. Three heuristics were used to calculate the size of items for FFD and BFD. Heuristics used for deciding the size were lexicographical (Lex), maximum component (Max) and sum of components. In Lex an item a>b if a equals b or first component of a-b is positive. In Max a>b if maximum component of a is greater than that of b. In sum of components a>b iff sum of all components of a is greater than that of b.

Spieksma in [16] proposed FFD heuristic algorithm which sorts items by assigning priorities and a branch and bound algorithm for 2-dimensional (2d) VBP. Han et al., in [17] proposed heuristics and exact algorithms for 2d VBP for heterogeneous bins. Caprara and Toth in [18] also studied the 2d case of VBP in detail. They proposed enhanced lower bounding methods, heuristics, and exact algorithms for this problem. Stillwell et al. in [19] proposed some VBP algorithms for resource allocation in virtualized shared hosting platforms. To sort items for FFD and BFD they used heuristics such as Sum, Max and Lex from [15]. They also proposed choose pack and permutation pack algorithms, greedy algorithms, a genetic algorithm and relaxed linear programming solution. From all these, the algorithms which used Sum as a heuristic performed well. Panigrahy et al. in [20] proposed two FFD variants called Dot-Product (DP) and L2 which use dot product and L2 norm techniques, respectively as heuristics to estimate the size of a VM.

Measures of central tendency such as mean and median are used to calculate a particular value from a dataset which identifies the central point of that dataset. This technique can be used to estimate the size of a multi-dimensional VM. As far as related work is reviewed for this research, FFD variants that use these measures for VMP in cloud computing environment are not found. Therefore, in this work two new VBP algorithms called FFDmean and FFDmedian for VMP in cloud computing environment are proposed, which are essentially FFD variants. The algorithms use mean and median respectively as heuristics to estimate the size of a VM.

III. Problem Formulation

1) VBP problem: Vector Bin packing (VBP) problem is a variant of bin packing problem. It is NP-hard which means that it cannot be solved in polynomial time however approximation algorithms are proposed for this problem. The problem is all same as bin packing problem except the items here are not one dimensional but d-dimensional vectors where d≥2.

The resource demand of an item is represented by demand vector $\vec{D}$ and the bin capacity is represented by resource vector $\vec{R}$. Both vectors are d-dimensional where d≥2 and are represented in (1) and (2), respectively, here *i* is item's demand and *c* is the bin's capacity in a particular dimension.

$$\vec{D} = \{i_1, i_2 \ldots i_d\} \quad (1)$$

$$\vec{R} = \{c_1, c_2 \ldots c_d\} \quad (2)$$

In VBP, there is B number of bins each with a resource vector $\vec{R}$ and I number of items each with a demand vector $\vec{D}$. The problem is to place all items in minimum number of bins, such that in each bin *b*, the sum of demands in each dimension of all accommodated items does not exceed the bin capacity in each corresponding dimension.

2) VMP problem: VM Placement (VMP) problem is a NP-hard problem. The VMP problem receives a set *V* of VMs, each with d-dimensional resource demands and a Set *P* of PMs, each with d-dimensional resource capacity. The problem is to place all VMs in minimum number of PMs, such that in each PM the sum of demands in each dimension of all VMs does not exceed the PM capacity in each corresponding dimension.

3) VMP problem formulation as VBP problem: Since VMP problem is multi-dimensional, it is difficult to formulate it as general bin packing problem. VMP Problem can be transformed into a VBP problem by considering VMs as items with d-dimensional demand vector $\vec{D}$ and PMs as bins with d-dimensional resource vector $\vec{R}$. In this work $\vec{D}$ in (3) for a VM v has four-dimensions which are CPU mips, ram, bandwidth (bw) and Storage (strg). $\vec{R}$ in (4) for PM p which is used as a host for VMs also has the same dimensions.

$$\vec{D} = (v_{mips}, v_{ram}, v_{bw}, v_{strg}) \quad (3)$$

$$\vec{R} = (p_{mips}, p_{ram}, p_{bw}, p_{strg}) \quad (4)$$

IV. Proposed Algorithms

FFDmean and FFDmedian are two new VBP algorithms proposed for VMP in cloud computing environment. The algorithms use classic FFD heuristic.

FFD is a simple heuristic approach to solve bin packing problem, which receives set of items and bins. Since FFD is used here for VMP, items are considered as VMs and bins as PMs/hosts. In FFD, VMs are first sorted in descending order of their sizes. After that starting from the largest VM, each VM is placed in the first available PM they fit into. The process continues until all VMs are placed.

The first requirement of FFD heuristic is to sort all the VMs into descending order of their sizes; however in this case where a VM is four-dimensional vector, it is difficult to decide which dimension should be selected as size of a VM. Therefore, a particular value/number is required in order to sort VMs. For this, measures of central tendency such as mean and median can be used to calculate a particular value from demand vector $\vec{D}$ of a VM. This value identifies the central point of the vector $\vec{D}$ and is the estimated size of a VM.

A. FFDmean and FFDmedian

FFDmean and FFDmedian use mean and median respectively to estimate the size of a multi-dimensional VM. The algorithms work in three steps, i.e. Normalization, size estimation and sorting and placement. Step 1 and Step 3 are same for both algorithms, however Step 2 is different.

1) Step 1 (Normalization): In this step, all the VMs are normalized to bring them on same scale. VMs are normalized by dividing all dimensions of a VM by their corresponding dimension in the PM as shown in (5).

$$\text{normalized\_VM} = \left(\frac{v_{mips}}{p_{mips}}, \frac{v_{ram}}{p_{ram}}, \frac{v_{bw}}{p_{bw}}, \frac{v_{strg}}{p_{strg}}\right) \quad (5)$$

2) Step 2 (Size Estimation): In this step, the size of each VM is estimated by using a Size function S. This function is different for FFDmean and FFDmedian.

a) Size function for FFDmean: For FFDmean this function calculates mean of VM dimensions in demand vector $\vec{D}$. The larger the mean the larger is the VM size. The size function S for FFDmean is presented in (6) where v represents a VM, i is a dimension and d is the total number of dimensions ($1 \le i \le d$) from $\vec{D}$.

$$S(v) = \frac{\sum_{i=1}^{d} v_i}{d} \quad (6)$$

b) Size function for FFDmedian: For FFDmedian this function calculates median of VM dimensions in demand vector $\vec{D}$. The larger the median the larger is the VM size. For median there are two cases. The first case is for odd number of dimensions in $\vec{D}$. In this case the middle value is taken as the median. The second case is for even number of dimensions in $\vec{D}$, which is true for this research. In this case two middle values are taken and then average of these values is calculated. The size function S for the second case is presented in (7)

where v represents a VM and x and y represent two middle values in $\vec{D}$.

$$S(v) = \frac{x+y}{2} \quad (7)$$

3) Step 3 (Sorting and placement): In this step, all the VMs are sorted in decreasing order of their sizes and then placed in PMs by using FFD technique.

Bin-centric version of FFD [20] is used in which a PM is taken at a time *t*. The process continues from Step 1 to Step 3 for each PM until no VM is left for placement.

V. System Model

This work is based on the IaaS layer of cloud computing. CloudSim toolkit [21] is used to simulate the IaaS environment and the experimental setup. CloudSim is a framework for modeling and simulation of cloud computing infrastructures and services.

The system consists of one datacenter which contains 800 identical PMs. Since the cloud computing datacenters are multi-tenant i.e. VMs of different users might run on same PM, the VM instances are designed to bring the effect of different resource requirements by different users. For example, a user may demand a VM with high CPU capacity while another user may demand a VM with high memory etc. There are six types of VMs inspired by Amazon EC2 VM instances, used in our experiments. VM requirements are designed to be within PM capacities.

This work is limited to only initial VMP. This means that VMs complete their lifetime in only one PM in which they are initially placed and they are not migrated. Four algorithms are used separately for VMP. The algorithms are FFDmean, FFDmedian, Dot-Product (DP) and L2. FFDmean and FFDmedian are the proposed algorithms in this work while DP and L2 [20] are existing algorithms used for comparative analysis. DP uses dot product and L2 uses L2Norm as heuristic to estimate the size of a VM. In DP the larger the dot product between $\vec{D}$ and $\vec{R}$ the larger is a VM's size whereas in L2 the smaller the difference between $\vec{D}$ and $\vec{R}$ the larger is a VM's size.

The system receives user requests for VMs in the form of PlanetLab workload. It is a real workload of user requests (tasks/Jobs) selected by CloudSim from PlanetLab and provided by CoMon project [22].

A. PlanetLab Workload

PlanetLab is a research network spread worldwide which supports the advancement of new network services. From the start of 2003, above 1,000 researchers from industrial research labs and remarkable academic institutes have chosen PlanetLab to develop new techniques for distributed storage, peer-to-peer systems, network mapping, distributed hash tables, and query processing.

PlanetLab workload is the real workload traces from real systems. It is the data made available as a part of the CoMon project [22] which is a monitoring infrastructure for PlanetLab. This data consists of CPU utilizations of more than a 1,000 VMs from PMs (servers) situated at more than 500 places around the world. The utilizations of VMs are measured at an interval of five minutes.

To make our experiments more authentic it is important to test our algorithms with a real workload. For this we have chosen the PlanetLab Workload traces of 10 random days collected during March and April 2011, provided in CloudSim.

B. Power Model

Power consumption in an IaaS is decided by data center resources such as electric power supplies, cooling systems, CPU, memory, disk storage, network equipment, etc. According to recent research [23], among these resources CPU utilization has a linear relationship with PM's power consumption. This is because there are least amount of states that can be assigned to the voltage and frequency of a CPU and furthermore performance scaling such as DVFS (Dynamic Voltage and Frequency Scaling) is not functional to other resources that use power, for instance network devices, memory, storage, etc.

Due to the increased use of virtualization technology, state-of-the-art PMs are designed with multi-core CPUs and large quantities of other resources. Memory, network and other devices also add in the power consumption of a PM. For the reason that modeling power expenditure of current PMs is a difficult research problem, in this research genuine information on power expenditure made available by SPEC (Standard Performance Evaluation Corporation) power benchmark is used rather than making an analytical power model [23]. SPEC power model provides real power usage of PMs at different CPU usage/load levels.

VI. Experimental Setup

Three Experiments are designed to evaluate FFDmean and FFDmedian algorithms compared with DP and L2. In each experiment, there is one datacenter with 800 identical PMs of types 1, 2 and 3, respectively as shown in Table 1. The difference between the experiments is the PM type (configuration). Since the goal is to test the performance sustainability of algorithms as the resources increases under different PM types. VMs are created according to the Instances provided in Table 2.

Jobs/tasks for VMs are taken from PlanetLab Workload provided in CloudSim which is divided into 10 parts or workloads. All algorithms i.e. FFDmean, FFDmedian, DP and L2 are used separately for VMP for all PlanetLab Workloads. Therefore, we have four algorithms, 10 workloads and three experiments; we get total 120 simulations (4*10*3).

Power model made available by SPEC power benchmark for each host type is used. Power consumption of these PMs at different CPU usage/load levels is presented in Table 3.

TABLE I. PM CONFIGURATION

Physical Machine		PM Configurations				
		CPU	*Cores*	*RAM*	*Strg*	*BW*
1	HP ProLiant ML110 G5	2.66 GHZ (2660 MHZ)	2	4 GB (4096 MB)	160 GB (163840 MB)	1000 Mb/s
2	IBM System x3250 M3	3.07 GHz (3067 MHz)	4	8 GB (8192 MB)	160 GB (163840 MB)	1000 Mb/s
3	IBM System x3550 M3	2.93 GHz (2933 MHz)	6	12 GB (12288 MB)	160 GB (163840 MB)	1000 Mb/s

TABLE II. VM INSTANCE TYPES

VM Type [a]		VM Configurations				
		CPU	Cores	RAM	Strg	BW
1	High CPU	1700 MHz (1.7 GHz)	1	2048 MB (2 GB)	1024 MB (1 GB)	400 Mb/s
2	High Memory	1200 MHz (1.2 GHz)	1	3072 MB (3 GB)	2048 MB (2GB)	300 Mb/s
3	High BW	1300 MHz (1.3 GHz)	1	512 MB (0.5 GB)	2048 MB (2 GB)	500 Mb/s
4	High Strg	1400 MHz (1.4 GHz)	1	1024 MB (1 GB)	12288 MB (6 GB)	200 Mb/s
5	General	1500 MHz (1.5 GHZ)	1	1024 MB (1 GB)	3072 MB (3 GB)	200 Mb/s
6	General	1000 MHz (1.0 GHz)	1	512 MB (0.5 GB)	512 MB (0.5 GB)	250 Mb/s

a. *Inspired from https://aws.amazon.com/ec2/instance-types/*

TABLE III. POWER MODEL OF EACH PM

PM Workload (%)		Power Consumption of PMs (Watts) [b]		
		HP ProLiant ML110 G5	IBM System x3250 M3	IBM System x3550 M3
1	0%	93.7	42.3	66
2	10%	97	46.7	107
3	20%	101	49.7	120
4	30%	105	55.4	131
5	40%	110	61.8	143
6	50%	116	69.3	156
7	60%	121	76.1	173
8	70%	125	87	191
9	80%	129	96.1	211
10	90%	133	106	229
11	100%	135	113	247

b. Taken from *www.spec.org*

A. Performance Metrics

There are three performance metrics used to analyze the algorithms. These are described below:

1) Hosts Used: Hosts used is the number of PMs required to host (accommodate) all VMs in the workload. The goal is to minimize this number. The smaller the number the better is the performance.

2) Power Consumption: It is the total electrical power consumed by all PMs in a datacenter. The goal is to reduce the power consumption. Power consumption of a PM is calculated by using SPEC power model. Power consumption of each active PM at different load levels in a datacenter is calculated using the values provided in Table 3. At the end of the simulation, power consumed by all active PMs is then added together to get total power consumption of the datacenter.

3) Resource Utilization Efficiency: Resource utilization efficiency is how well the resources of a PM in each dimension are utilized. The goal is to increase utilization efficiency in all dimensions of a PM. For each PM, utilization percentage in each of its dimension i.e. CPU, RAM, BW, and Strg is calculated first. After that average utilization percentage of each PM in each dimensions is calculated.

VII. RESULTS

The experiments were conducted on a HP notebook PC running Windows 7 Home Premium with Core i5 CPU @ 1.60 GHz and 4 GB RAM. In each experiment all VMs were placed separately by FFDmean, FFDmedian, DP and L2. Average results of each algorithm in each performance metric are presented here. Fig. 1 shows a graphical representation of average hosts used by each algorithm in each experiment and Fig. 2 shows a graphical representation of average power consumption in kilo watts (KW) by each algorithm in each experiment. Fig. 3 shows a graphical representation of average resource utilization efficiency (%) by each algorithm in each experiment. To save space in figures FFDmean and FFDmedian are represented as Mean and Median. Table 4 shows a summary of results in each experiment.

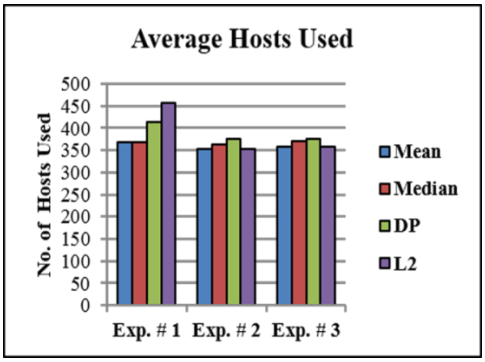

Fig. 1. Average hosts used in each experiment.

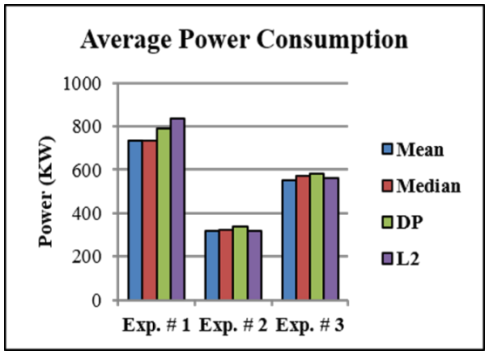

Fig. 2. Average power consumption in each experiment.

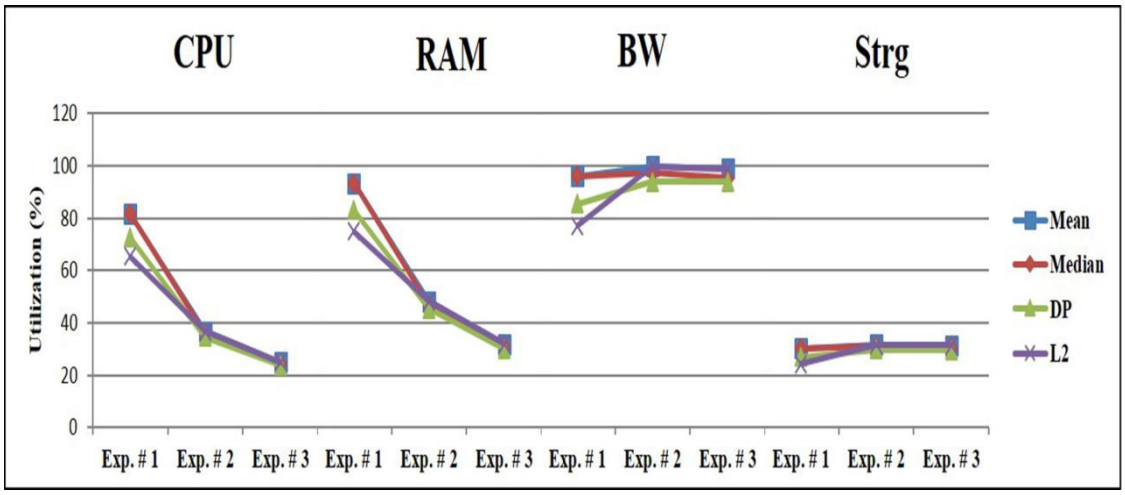

Fig. 3. Average resource utilizataion (%) in each experiment.

TABLE IV. SUMMARY OF AVERAGE RESULTS BY ALL ALGORITHMS IN ALL METRICS

Exp. #	Algorithms	Hosts Used	Power Consumption	Resource Utilization Efficiency (%)			
				CPU	RAM	BW	Strg
1	FFDmean FFDmedian DP L2	368 368 413 458	734.241 734.522 790.843 835.828	81.55 81.55 72.49 65.38	93.34 93.34 83 74.99	95.92 95.92 85.39 77.06	30.28 30.28 26.86 24.15
2	FFDmean FFDmedian DP L2	353 363 376 353	315.745 324.919 336.343 318.602	36.64 35.68 34.42 36.7	48.29 46.73 45.29 48.35	99.85 97.13 93.82 99.83	31.67 30.85 29.64 31.59
3	FFDmean FFDmedian DP L2	357 370 376 357	552.971 572.475 584 560.968	24.89 24.14 23.68 24.88	31.86 30.66 29.93 31.84	98.97 95.4 93.75 98.9	31.35 30.13 29.62 31.33

VIII. DISCUSSION

A. Hosts Used

In Exp. 1 FFDmean and FFDmedian saved 11% hosts compared to DP and saved 20% hosts compared to L2. In Exp. 2 FFDmean saved 6% hosts compared to DP and performed equally well as L2. FFDmedian saved 3% hosts compared to DP however it used 3% more hosts compared to L2. In Exp. 3 FFDmean saved 5% hosts compared to DP and performed equally well compared to L2. FFDmedian saved 2% hosts compared to DP however it used 4% more hosts compared to L2.

This shows that in Exp. 1 where PM resources were very limited compared to other experiments, FFDmean and FFDmedian remarkably saved hosts leaving behind DP and L2, however when resources were comparatively increased in Exp. 2 and Exp. 3 FFDmean outperformed DP and performed equally well as L2 however FFDmedian only performed better than DP.

B. Power Consumption

In Exp. 1 FFDmean saved 7.16% power compared to DP and 12.15% power compared to L2. FFDmedian saved 7.12% power compared to DP and saved 12.12% power compared to L2. In Exp. 2 FFDmean saved 6.12% power compared to DP and saved 0.9% compared to L2. FFDmedian saved 3.4% power compared to DP; however, it used 1.98% more power compared to L2. In Exp. 3 FFDmean saved 5.31% power compared to DP and saved 1.43% power compared to L2. FFDmedian saved 1.97% power compared to DP; however, it used 2.05% more power compared to L2.

This shows that like hosts used metric, in Exp. 1 where PM resources were limited compared to other experiments FFDmean and FFDmedian both outperformed DP and L2 in saving power. When PM resources were comparatively increased in Exp. 2 and Exp. 3 FFDmean outperformed both DP and L2 however FFDmedian only performed better than DP.

C. Resource Utilization Efficiency

Table 4 shows that in Exp. 1, FFDmean and FFDmedian performed equally in increasing utilization efficiency in all dimensions and outperformed both DP and L2. The effect of this utilization efficiency can be seen in other two metrics where FFDmean and FFDmedian outperformed very well than both DP and L2. In Exp. 2 FFDmean and L2 showed same utilization values and better than FFDmedian and DP, therefore their performance in other two metrics was almost same. FFDmedian increased up to 1% utilization in all dimensions compared to DP. Even with 1% increase in utilization

FFDmedian saved 3% hosts and 3.4% power compared to DP. This shows that even minor increase in utilization can bring a lot of effect. In Exp. 3 FFDmean and FFDmedian performed equally to L2. FFDmean and FFDmedian increased up to or more than 1% utilization compared to DP in all dimensions. With this minor change FFDmean saved 5% hosts and saved 5.31% power compared to DP, and FFDmedian saved 2% hosts and 1.97% power compared to DP.

This shows that with very limited resources in Exp. 1 FFDmean and FFDmedian remarkably outperformed DP and L2 in increasing resource utilization efficiency. By increasing PM resources in Exp. 2 and Exp. 3 utilization values of all algorithms became almost equal with minor difference. It is observed that even minor changes in utilization can bring a lot of effect in saving hosts and power.

From all experiment results, it is observed that our proposed algorithms FFDmean and FFDmedian remarkably outperformed DP and L2 when PM resources were extremely limited. Our algorithms also performed well when resources were increased, however the performance was not that remarkable compared to the prior situation. This shows that our algorithms can be useful in situations where resources are very limited.

IX. Conclusion

This research proposed two Vector Bin Packing algorithms for virtual machine placement called FFDmean and FFDmedian. These algorithms are essentially FFD variants which use mean and median respectively as heuristics to estimate size of a multi-dimensional VM. The performance of proposed algorithms was evaluated by comparing them to existing algorithms DP and L2 over three metrics i-e hosts used, power consumption and resource utilization efficiency. DP and L2 use dot product and L2-norm respectively to estimate the size of a VM.

Proposed algorithms were tested over three types of PMs (different in configuration) to evaluate their performance sustainability as the PM resources increase. It is observed that the proposed algorithms remarkably outperformed DP and L2 when PM resources were comparatively limited to other experiments. The algorithms also performed well when resources were increased, however the performance was not that remarkable compared to the prior situation. This shows that the algorithms can be useful in situations where resources are very limited.

X. Future Recommendations

Since this work was limited to only initial VMP, live VM migrations can be incorporated in this work. The algorithms were measured over only three metrics i.e. hosts used, power consumption and resource utilization efficiency this work can also be extended in future by using other metrics such as SLA performance degradation, etc. Moreover non-identical PMs can be used in the experiments.

Acknowledgement

Greatest thanks to Ms. Suhni Abbasi, for her help and encouragement.

References

[1] Mell, P., & Grance, T. The NIST definition of cloud computing, 2011.

[2] Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation computer systems, vol. 25, issue 6, pp. 599-616, June 2009.

[3] Rimal, B. P., Choi, E., & Lumb, I. A taxonomy and survey of cloud computing systems. In Proceedings of the 2009 Fifth International Joint Conference on INC, IMS and IDC, pp. 44-51, 2009.

[4] Ruest, N., & Ruest, D. Virtualization, A Beginner's Guide. McGraw-Hill, Inc, 2009, pp. 423.

[5] MALHOTRA, L., AGARWAL, D., & JAISWAL, A. Virtualization in Cloud Computing. J Inform Tech Softw Eng, vol. 4, issue 2, pp.136, 2014.

[6] Basmadjian, R., Niedermeier, F., & De Meer, H. (2012). Modelling and analysing the power consumption of idle servers. IEEE. In proceedings of Sustainable Internet and ICT for Sustainability (SustainIT), pp. 1-9 October 2012.

[7] Beloglazov, A., Abawajy, J., & Buyya, R. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future generation computer systems, vol. 28 issue 5, pp. 755-768, May 2012.

[8] Khosravi, A., Garg, S. K., & Buyya, R. Energy and carbon-efficient placement of virtual machines in distributed cloud data centers. In proceedings of Euro-Par 2013 Parallel Processing, Springer Berlin Heidelberg, vol. 8096, pp. 317-328, 2013.

[9] Cao, J., Wu, Y., & Li, M. Energy efficient allocation of virtual machines in cloud computing environments based on demand forecast. In Advances in Grid and Pervasive Computing, Springer Berlin Heidelberg, vol. 7296, pp. 137-151, 2012.

[10] Wang, X., Liu, X., Fan, L., & Jia, X. A decentralized virtual machine migration approach of data centers for cloud computing. Mathematical Problems in Engineering, vol. 2013, Article ID 878542, 10 pages, 2013. doi:10.1155/2013/878542

[11] Srikantaiah, S., Kansal, A., & Zhao, F. Energy aware consolidation for cloud computing. HotPower'08 Proceedings of the 2008 conference on Power aware computing and systems, pp. 10-10, 2008.

[12] Consuegra, M. E., Narasimhan, G., & Rangaswami, R. Vector repacking algorithms for power-aware computing. In Green Computing Conference (IGCC), 2013 International, pp. 1-8, June 2013.

[13] Chen, L., & Shen, H. Consolidating complementary VMs with spatial/temporal-awareness in cloud datacenters. In INFOCOM, 2014 Proceedings IEEE, pp. 1033-1041, May 2014.

[14] Maruyama, K., Chang, S. K., & Tang, D. T. A general packing algorithm for multidimensional resource requirements. International Journal of Computer & Information Sciences, vol. 6, issue 2, pp. 131-149, June 1977.

[15] Kou, L. T., & Markowsky, G. Multidimensional bin packing algorithms. IBM Journal of Research and development, vol. 21 issue 5, pp. 443-448, September 1977.

[16] Spieksma, F. C. A branch-and-bound algorithm for the two-dimensional vector packing problem. Computers & operations research, vol. 21 issue 1, pp. 19-25, January 1994.

[17] Han, B. T., Diehr, G., & Cook, J. S. Multiple-type, two-dimensional bin packing problems: Applications and algorithms. Annals of Operations Research, vol. 50, issue 1, pp. 239-261, December 1994.

[18] Caprara, A., & Toth, P. Lower bounds and algorithms for the 2-dimensional vector packing problem. Discrete Applied Mathematics, vol. 111, issue 3, pp. 231-262, August 2001.

[19] Stillwell, M., Schanzenbach, D., Vivien, F., & Casanova, H. Resource allocation algorithms for virtualized service hosting platforms. Journal of Parallel and distributed Computing, vol. 70, issue 9, pp. 962-974, September 2010.

[20] Panigrahy, R., Talwar, K., Uyeda, L., & Wieder, U. "Heuristics for vector bin packing. Research," unpublished.

[21] Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., & Buyya, R. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms.

Software: Practice and Experience, vol. 41, issue 1, pp. 23-50, January 2011.

[22] Park, K., & Pai, V. S. CoMon: a mostly-scalable monitoring system for PlanetLab. ACM SIGOPS Operating Systems Review, vol. 40, issue 1, pp. 65-74, January 2006.

[23] Beloglazov, A., & Buyya, R. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. Concurrency and Computation: Practice and Experience, vol. 24, issue 13, pp. 1397-1420, September 2012.

[24] Pires, F. L., & Barán, B. A virtual machine placement taxonomy. In 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), pp. 159-168, May 2015.

[25] Madni, S. H. H., Latiff, M. S. A., & Coulibaly, Y. Recent advancements in resource allocation techniques for cloud computing environment: a systematic review. In Cluster Computing, vol. 20, issue 3, pp. 2489-2533, September 2017.

[26] Hameed, A., et al. A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems. In Computing, vol 98, issue 7, pp. 751-774, July 2016.

[27] Radha, K., et al. Service Level Agreements in Cloud Computing and Big Data. International Journal of Electrical and Computer Engineering, vol 5, issue 1, pp. 158-165, February 2015.