Efficient HPC and Energy-Aware Proactive Dynamic VM Consolidation in Cloud Computing

Rukshanda Kamran
Faculty of Computer Science and
Information Technology,
University Malaysia Sarawak
94300 Kota Samarahan Sarawak, Malaysia

Ali A. El-Moursy
Computer Engineering Department,
University of Sharjah
Sharjah, UAE

Amany Abdelsamea
Computers and Systems Department,
Electronics Research Institute
Giza, Egypt

***Abstract*—The adoption of High-Performance Computing (HPC) applications has gained an extensive interest in the Cloud computing. Current cloud vendors utilize separate management tools for HPC and non-HPC applications, missing out on the consolidation benefits of virtualization. Non-HPC applications executed in the cloud may interfere with resource-hungry HPC applications, which is a key performance challenge. Furthermore, correlations between application major performance indicators, such as response time and throughput, with resource capacities reveal that conventional placement strategies are impacting virtual machine efficiency, resulting in poor resource optimization, increased operating expenses, and longer wait times. Since applications often underutilized the hardware, smart execution of HPC and Non-HPC applications on the same node can boost system and energy efficiency. This research incorporates proactive dynamic VM consolidation to enhance the resource usage and performance while maintaining energy efficiency. The proposed algorithm generates a workload-aware fine-grained classification by employing machine learning techniques to generate complimentary profiles that alleviate cross-application interference by intelligently co-locating non-HPC and HPC applications. The research used CloudSim to simulate real HPC workloads. The results verified that the proposed algorithm outperforms all heuristic methods with respect to the metrics in key areas.**

***Keywords*—Cloud computing; HPC (High-Performance Computing); virtual machine consolidation; placement; optimization**

I. Introduction

Cloud computing [1] provides organizations with an affordable, high-performance computing [2] infrastructure. HPC programs have a repetitive and predictable nature and based on the characteristics of the data that is utilized as input, their resource consumption patterns (CPU and memory, I/O, and network) are predictable. Strategies can be developed to increase queue throughput and resource utilization while reducing performance impacts on applications. HPC applications' performance can be adversely affected by virtualized layers, heterogeneous hardware, HPC-agnostic schedulers such as MOAB [3] or Load Leveler [4], and resource sharing policies. Currently, cloud providers for HPC either offer specialized clouds with dedicated nodes, lacking the consolidation advantages of virtualization, or cloud scheduling that is HPC-agnostic, resulting in inadequate performance. Although modern systems have enormous compute power per node, HPC applications seldom use all of the resources assigned to them. Other applications, such as non-HPC applications, can make use of this feature by utilizing underutilized resources.

Separate technologies have been used to manage the resources and applications on dedicated systems for HPC and Non-HPC applications. This isolation has become an increasing burden, and hence there is an increased demand for the adoption of a standardized shared platform. Execution of both HPC/Non-HPC applications on the same cluster boosts the system efficiency, allowing programs to take advantage of all the available hardware resources. However, even with these advantages, there is still a hindrance in using the full potential of sharing the resources. It is vital that the researcher must devise a method of bridging the gap between dedicated infrastructures and standardized shared platforms by balancing the trade-off between resource utilization, performance, and energy usage, by choosing a suitable VM to physical machine placement methods, with the intention of exploring the best physical machine (PM) that can be used to host the virtual machines. Workload heterogeneity has become norm in cloud computing [5]. Workload characterization is critical since it can group various resource-intensive workloads based on their defining qualities. Classifying workloads that have common consumption patterns can enhance resource management, which improves system performance while maintaining Quality of Service. Clustering techniques are frequently used to cluster workloads in the cloud data center to reduce energy usage and SLA violations for resource allocation.

In terms of VM placement strategies, there are two types: reactive and proactive/predictive strategies. Reactive strategies enhance the initial VM after the system reaches a certain undesired state. While proactive/predictive strategies attempt to enhance VM placement results by projecting future workloads or resource demands using prediction techniques. Identifying proper co-allocated combinations of applications that can be run on the common platform guarantees optimum utilization of resources [6]. The optimum utilization of active resources will allow the reduction of the number of operating servers, which will lead to saving energy spent on computation [7]. Hence, the total energy consumed by a data center will be reduced and optimum utilization of hardware will maintain the performance of applications. Most of the earlier well-established research refers to VM consolidation as a key approach for data centers to save energy and achieve a balance between utilization and SLA violations. The goal behind this strategy is to carefully consolidate VMs or workloads onto a smaller number of PMs and then convert the unused (idle) PMs into a power-saving state or shut them down when they are no longer needed.

Available literature indicates that there have been only a few attempts that proposed automatic workload clustering, virtual machine placement and VM consolidation techniques [8] to eliminate or at least reduce the impact of energy [9], performance [10], and interference [11] in co-located HPC and Non-HPC applications. Moreover, the HPC application characteristics of the strongly coupled processes that perform constant interprocess communication and synchronizations have also rarely been studied.

This study aims to combine and execute both HPC and non-HPC applications on cloud resources using a smart and innovative technique which can balance the trade-off between energy, performance, and resource utilization [12]. The researchers have proposed proactive dynamic VM consolidation for co-scheduling of HPC and non-HPC applications that is based on utilization predictions and an application's profile employing machine learning techniques. Our approach looks for applications that work well together and can be deployed on the same hardware, and the execution profiles for these applications do not compete with each other. This will allow for more efficient usage of the hardware. As part of the VM consolidation process, the proposed approach to VM consolidation also examines the application's resource usage requirements across multiple dimensions, such as the CPU, RAM, and the network. By co-locating suitable virtual machines on hosts during consolidation, resource contention and virtualization overhead can be reduced on application performance without losing the benefits of VM consolidation. Thus, the major contributions to this study can be stated as follows:

1) Introduces and implements HPC and energy-aware, efficient proactive dynamic virtual machine (VM) consolidation technique in cloud datacenter that makes cloud schedulers as well as VM placement HPC-aware.
2) Attempt to make VM consolidation application-centric while considering the requirements of the applications' resource utilization (i.e., CPU, memory, and bandwidth). The paper explores automatic clustering of workload and virtual machines using k-mean cluster technique. It also aims to explore smart VM placement strategies to intelligently schedule HPC and non-HPC applications on a single pool of resources to increase utilization of resources and overcome the performance issues caused by resource contention.
3) Explicitly examines the real-world HPC workload in order to demonstrate the reliability of our numerical analysis in terms of how well our proposed approach is suited to both HPC and non-HPC applications.

This paper is structured as follows: Section II presents the background and motivation for the study. Section III presents the related work to gain knowledge of contemporary research and discusses the importance of the proposed algorithm. while Section IV demonstrates the proposed energy-efficient HPC aware proactive dynamic VM consolidation (EAMDOBP). Section V introduces the evaluation methodology. Section VI discusses the simulation results and analysis. Lastly, the conclusion and future work is discussed in Section VII.

II. BACKGROUND AND MOTIVATION

One of the primary goals of modern datacenter architecture is to cut down on energy usage. Data center energy demands are expected to rise to 752 TWh in 2030. This means that data centers consume 2.13% of total global electricity demand [13]. The inefficient utilization of hardware resources is the root cause of high energy consumption. Idling servers might use 60% of their maximum power [14]. Clustering of workload [15] and Virtual machine consolidation are the most effective and crucial approach for optimizing resource consumption and improve energy efficiency in cloud datacenters. VM consolidation can be implemented in two ways, static and dynamic. Fig. 1(a) is static consolidation. When a job arrives, the size and location of virtual machines on physical machines (VMs) are explicitly pre-determined, and the placement does not change during execution. Static VM placement resembles the N-dimensional bin-packing issue where the bins symbolize physical machines (PM). The items for packing represent the VMs and the size of the bin depends on the volume, types and nature of resources. Static VM consolidation is better suited for small jobs spanning a few hours, where PMs resources for various kinds of VMs can be defined explicitly [16]. Simple heuristics or historical VM demand patterns are the basis of energy minimization. However, during low-demand resource periods, an increase in the cost of application providers is likely to occur. Similarly, the available resources may be insufficient in high utilization periods [17].

On the other hand Fig. 1(b) shows dynamic VM consolidation. To improve the effectiveness of a placement, dynamic VM consolidation allows relocation during execution. Mostly virtual machine workloads are bursty in nature, dynamic VM consolidation is highly beneficial in a cloud computing environment, assuming that monitoring is in place to prevent any violations of Service-Level Agreements (SLAs). Accordingly, dynamic VM consolidation conserves energy and enhances the consumption of resources by using the minimum resources necessary to meet the workload requirements. Consequently, if the workload requirement decreases, unused servers are shut down or kept in low-power mode. Similarly, as consumption grows, additional servers are brought online.

The VM migration and VM placement are considered as the backbone of the VM consolidation technique. The problems like scalability of resources, heterogeneity, migration cost, and unpredictable workloads cause that the VM consolidation process becomes very challenging. The virtual machines (VMs) placement to physical machines (known as VM-Placement) can significantly impact performance. It is very important to choose an appropriate host to enhance power efficiency, better use of resources and support for QoS to attain Resiliency in the Cloud [34].

Clustering is an unsupervised learning strategy for subdividing a big group into multiple smaller groups. As a result, it may be used to find patterns in massive datasets. In current research, the advantages of this group building technique of clustering have been applied to locate groupings in the jobs (incoming request). The ability to locate groupings within jobs based on the number of resources consumed is helpful since this methodology allows to find groupings within jobs.

Cross application interference can occur when different

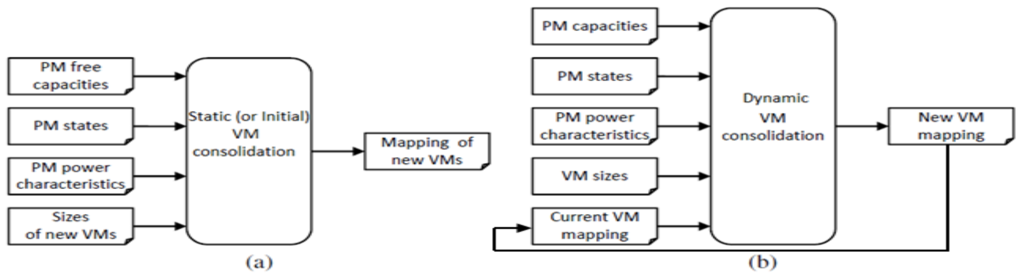


Fig. 1. Inputs and Outputs of (a) Static VM Consolidation and (b) Dynamic VM Consolidation.

cloud workload compete for shared resources, resulting in considerable performance degradation and, results in increase Service Level Agreement violations. Despite this, state-of-the-art VM scheduling still relies heavily on resource capacity, employing heuristics like bin-packing and ignoring the cross application interference overhead. To the best of our knowledge, only a few studies were made for VM scheduling algorithms that consider the tightly connected processes that perform frequent inter-process communication and synchronizations. Improved resource efficiency, cost savings, and, ultimately, broader acceptance of high-performance computing in clouds can be achieved by the strategic placement of virtual machines and the execution of HPC and non-HPC workloads in cloud environments.

Hence, this study focused on a multi-objective challenge that seeks to explore automatic online workload clustering using machine learning, smart VM consolidation and placement strategies to intelligently schedule or provision HPC and non-HPC applications on cloud resources to overcome the aforementioned gap and issues of rigorous consolidation.

III. Related Work

Primarily due to the enhanced efficiency, a significant number of heuristic algorithms have already been suggested for handling VM consolidation challenges in recent years. Among them, Mosa et al. [18] suggest the solution to virtual machine placement that dynamically reallocates virtual machines depending on their actual request for the individual VMs. The suggested approach evaluates various resource categories (particularly CPU and memory) to minimize under- and over-utilization in cloud-based data centers. The conducted experiments highlighted the significance of incorporating a variety of resource types. Finally, they concluded for dynamic VM placement, the genetic algorithm outperforms the Best-Fit algorithm. However, the paper neither takes into account the HPC workload nor discusses the issues of cross-application interference. Kraemer et al. [19] developed a job migration mechanism for transferring jobs from the cloud environment to the high-performance computing environment. The primary goal is to reduce the amount of response time violations associated with cloud jobs while not interfering with the execution of HPC jobs.

V. Antonenko [20] developed a strategy for migrating jobs from the cloud environment to the high-performance computing environment. The major objective is to reduce cloud job response time violations without conflicting HPC task execution. The author's study discusses the suggested job scheduling methods using the SimGrid simulator in various execution scenarios, and recorded findings revealed no reaction time violations. However, the authors did not incorporate support for parallel jobs and did not run experiments with higher rates of incoming cloud jobs.

Alves et al. [11] describes the Interference-aware Virtual Machine Placement Issue (IVMPP) in small-scale High-Performance Computing (HPC) applications executing in Clouds. When applications run on a common physical machine, cross-interference is likely to happen, which harms the application′s performance. This problem is very common in HPC that is executed in clouds. The iterated local search framework is proposed as a new solution to prevent the Interference-aware Virtual Machine Placement Problem (IVMP) from happening in HPC applications in clouds. In this study, they limited the interference that happens to HPC applications when sharing common physical machines. The results indicated that the proposed method limited interference by more than 40% in contrast to the most commonly applied heuristics to address the issue. However, the energy and the effect of consolidation are not taken into consideration.

A. Souza [12] proposed a hybrid resource management system for both DI (Data-Intensive) workload systems and HPC systems that will allow combining both of them on the same platform. The most significant feature between HPC systems and DI (Data-Intensive) systems is the fixed set of resources allocated completely to an application in HPC systems. Contrary to the DI (Data-Intensive) systems, in which allocation of resources and control are dependent on application needs. It also describes the design of a hybrid framework which helps for dual-level scheduling of DI jobs on the HPC infrastructure. The core benefit of this hybrid system is that it relies on real-time resource utilization monitoring that could successfully co-schedule high-performance computing (HPC) and data-intensive workloads. It can easily be adapted and extended to different types of workloads. For HPC and DI workloads, the architecture is based on the resource managers Slurm and Mesos. In a particular cluster, the hybrid architecture raises resource consumption by 20%, allowing it to meet all the constraints for HPC jobs, with a 12% reduction in queue makespan. Nevertheless, the paper does not explore ways to reduce the interference and co-location effects on energy as well as resource utilization.

Gupta et al. [21] presented scheduler for cloud platforms that is HPC-aware and incorporates topological needs for

HPC applications. Their scheduler uses benchmarking data to classify the application's network requirements and how resource sharing affects performance. Using three apps and NAS benchmarks, their scheduler outperformed the HPC-agnostic scheduler. Despite all the benefits, Gupta et al. [21] ignored SLA and energy breaches in their study. Furthermore, the trade-off among optimal HPC performance and ideal resource usage is primarily discussed in terms of throughput. Static VM consolidation techniques and Off-line applications profiling were used for classification. Hence, it is necessary to investigate the trade-off in terms of other dimensions as well. As part of earlier research, the researchers investigated CloudSim's default VM placement technique for energy consumption in contrast to the VM placement technique proposed by Gupta, which uses Multi-Dimensional Online Bin Packing (MDOBP). Fig. 2 illustrates that MDOBP uses more energy than CloudSim's default VM placement technique. The findings show an increase in energy consumption due to the possible restrictions of static VM placement. The findings of this experiment served as the foundation for the current research.

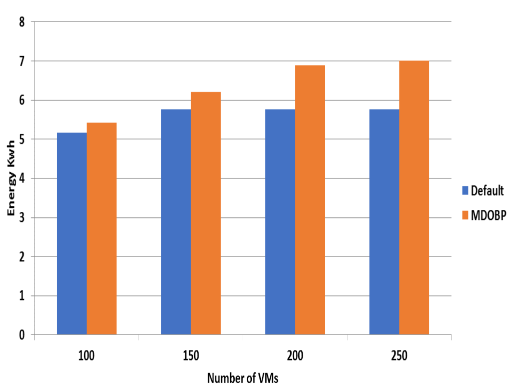


Fig. 2. Energy Comparison of Default vs MDOBP.

Cluster analysis is critical for detecting workloads with similar resource usage patterns. Several approaches for clustering workloads using K-Means have been proposed. Di, Sheng at al. [22] characterize a one-month Google cluster trace by CPU and memory utilization using the Forgy approach for centroid initialization. A merge ratio threshold is used to calculate the optimal number of clusters. Moreno et al. [23] employ K-Means clustering to group workloads based on user behavior and task characteristics. The number of clusters is obtained by comparing the variability of all items within a cluster to a threshold value.

In a previous project [24], the authors studied and implemented the Hybrid Local Regression Host Overload Detection algorithm (HLRHOD). An innovative energy efficient VM consolidation method that uses hybrid factors for host overload detection in cloud datacenters. We concluded that using hybrid factors (CPU, Memory, Bandwidth) can provide a more accurate indication of host utilization and outperform the techniques based on single factors. Thus, using the HLRHOD, the server in operation can be optimally minimized and, hence helps in the reduction of energy utilization. Nevertheless, the paper does not explore ways to reduce the interference and co-location effects of the HPC and non-HPC applications on resource utilization as well as energy.

In this study, Gupta et al. [21] heuristics for initial VM placement for high-performance computing (HPC) applications were followed and expanded. Additionally, this study expands VM consolidation work [24] by making VM consolidation application-centric while simultaneously taking into account the requirements of the application's resource consumption (i.e., CPU, memory, and bandwidth). We also explore smart VM consolidation and placement strategies to intelligently schedule or provision HPC and non-HPC applications on cloud resources to increase utilization of resources and overcome the aforementioned issues of consolidation. Furthermore, automatic and dynamic classification of the application workload using k-mean is implemented.

IV. Energy-Aware Multi-Dimensional Online Bin Packing (EAMDOBP)

HPC applications are often constructed to operate in a homogenous as well as dedicated environment to eliminate unwanted interference by apps that are located concurrently. This is due to the fact that the performance of high-performance computing applications are significantly dependent on the slowest node. In contrast, cloud infrastructure is changing from homogeneous to heterogeneous. Heterogeneity dramatically lowers performance in parallel applications, especially in repetitive and bulk concurrent workloads. VM scheduling for HPC is challenging because of the trade-off between better HPC performance and improved resource consumption. A technique for intelligently optimizing the placement and execution of virtual machines (VMs) for HPC and non-HPC applications can improve resource usage, improve energy efficiency, and hence promote HPC cloud acceptance.

Sharing resources typically causes critical interference. Contention of shared resources has poor impact on applications performance. Caches are small stores of temporary memory. Cache memory has an impact on the program execution because its access time is less than the access time of the other memories. It is the fastest component in the memory hierarchy and approaches the speed of CPU components. They can degrade system performance if they become too large. System performance is a decreasing function of the cache miss rate, the cache access time, and the number of processor cycles taken to service a miss. They also can consume memory that other applications might need, negatively impacting application performance. Cache, memory, I/O channels, and network access are all shared resources, but cache are one of the most significant applications performance degradation factors [25]. It is advantageous to have cache-intensive applications; endure more LLC misses per second; co-located with applications that use little or no cache on the same node. Cache-sensitivity awareness assists in the avoidance of interference.

Most of the studies reviewed recently, a static classification of the behavior of different workloads/applications is created to examine their studies in order to find solutions for interference scheduling issues in cloud environments. EAMDOBP algorithm presents an automatic workload clustering using machine learning techniques. The EAMDOBP classifier initially cluster the workload based on memory utilization and workload length then VMs are grouped based on their processing capacity. Afterwards the individual cloudlets in each cluster is scheduled to the appropriate VM in the VM groups. The flowchart in Fig. 3 depicts the execution flow of the proposed algorithm EAMDOBP, which is detailed in the following subsections.

A. Clustering of Workload and VM

A workload can be defined as a certain amount of work operated inside the data center while consuming specific limited resources. In the current context, cloud computing data centers are defined as computer resource pools that can bear variable workloads whether long scientific jobs (HPC) or transactional operations (non-HPC). Normally, workloads are different in heterogeneous environments because of the placement constraints they have and the number of resources they consume [26]. In fact, the amount of resources a job consume is defined as job resource requirements, while the type and characteristics of resources are defined as job placement constraints. Therefore, these main elements must be addressed for intelligent and efficient scheduling. The proposed EAMDOBP initially classifies incoming requests according to job characteristics. This classification further is used to make crucial scheduling decisions, such as scheduling jobs from various clusters based on the amount of resources they consume.

1) Clustering Jobs using K-Means and Silhouette Method: The EAMDOBP classifier divides workload into specified classes by using the k-means algorithm [35] and the silhouette approach [36]. K in k-means algorithm denotes the quantity of pre-defined clusters that must be produced during the process. It is a centroid-based technique, with each cluster having its own centroid. The algorithm's main goal is to minimize the sum of distances between data points and their respective clusters. As input, the EAMDOBP classifier takes memory utilization and its length from METACENTRUM-02.swf logs of the parallel workload archive, separates it into k = 4 clusters, and then continues the procedure until the centroids don't change. The technique for creating job clusters using the K-Means clustering algorithm is summarized in Algorithm-1. Once the clustering process is completed, the average memory utilization for each cluster is calculated.

Algorithm 1 Basic K-Means Algorithm

1: Select k points as initial centroid
2: **repeat**
3: Form k clusters by assigning each point to its closest centroid
4: Re-compute the centroid of each cluster
5: **until** the centroids don't change

Clusters presented in Table I are obtained applying k-means algorithm in cloudSim on the first 250 logs from METACENTRUM-02.swf logs of the parallel workload archive [27]. K-means algorithm clusters data into four clusters. From the cluster centroid analysis, four different types of workloads can be outlined. These have been labeled as "IntenseHPC", "ConcurrentHPC", "DiscreteHPC", and non-HPC.

Table II displays the input (memory use and length) classified into each cluster using k-means algorithm. The number of objects by cluster represents the total number of objects per cluster. Within-cluster variance is the sum of squared distance between the average point (centroid) and every point of the cluster. The smaller the within-cluster variance value the better is the clustering. The average distance between observations and the cluster centroid is a measure of observation diversity within each cluster. A cluster with a smaller average distance is generally more compact than one with a bigger average distance. Clusters with higher values show more variation in the observations within the cluster. A larger maximum value of maximum distance to centroid, especially when compared to the average distance, suggests a cluster observation that is located further away from the cluster centroid.

Furthermore, silhouette score is utilized to determine the quality of clusters formed by using K-means clustering algorithms. The silhouette approach analyzes the quality of clustering by determining how well each point fits into its cluster. Using Eq. 1, the silhouette score is calculated for each cluster data (i.e. memory utilization and its length). where (a) is mean intra-cluster distance and (b) is the mean nearest-cluster distance. Fig. 4 depicts the mean silhouette score of each cluster. When the silhouette score equals 1, it means that the clusters are very dense and well separated. When the silhouette score equals 0, it means that clusters are overlapping. If the score is less than 0, this indicates that the data could be incorrect.

$$SilhouetteScore = (b - a)/max(a, b). \tag{1}$$

Subsequent workload clustering, the processing capability of each virtual machine is determined based on its MIPS, size, bandwidth, and RAM. The virtual machines are then grouped into four clusters Extreme High (EHVMs), High (HVMs), Medium (MVMs) and Low (LVMs), using the modified K-means clustering algorithm. Afterwords, all cloudlet clusters are mapped to Virtual Machine (VM) clusters. The cloudlet with the high memory consumption is assigned to high processing capability. For example, IntenseHPC applications are assigned to Extreme High (EHVMS). Fig. 5 shows the simulation time with kmeans and without kmeans. As depicted in the Fig. 5 the simultaion time has shown improved result.

B. Virtual Machine Placement

EAMDOBP selects the PM that will host the VM based on the VM's class. The EAMDOBP begins the process with the target that IntenseHPC applications VMs can be assigned to the same host and rack as much as possible to reduce cross-interference for HPC/ Non-HPC applications. When applications share a physical machine, cross-interference can occur, negatively impacting their performance. EAMDOBP VM request contains the application's cluster name, as well as any existing parameters, unlike traditional VM requests. Based on the VM provisioning request, VM and cloudlets are generated depending on the Application type, capacity (number of CPU, RAM, and bandwidth), and instance type. VMs are allocated to a host and the cloudlet is allocated to VM. The algorithm then calculates the current host and rack free capacity, which is the number of extra VMs of the

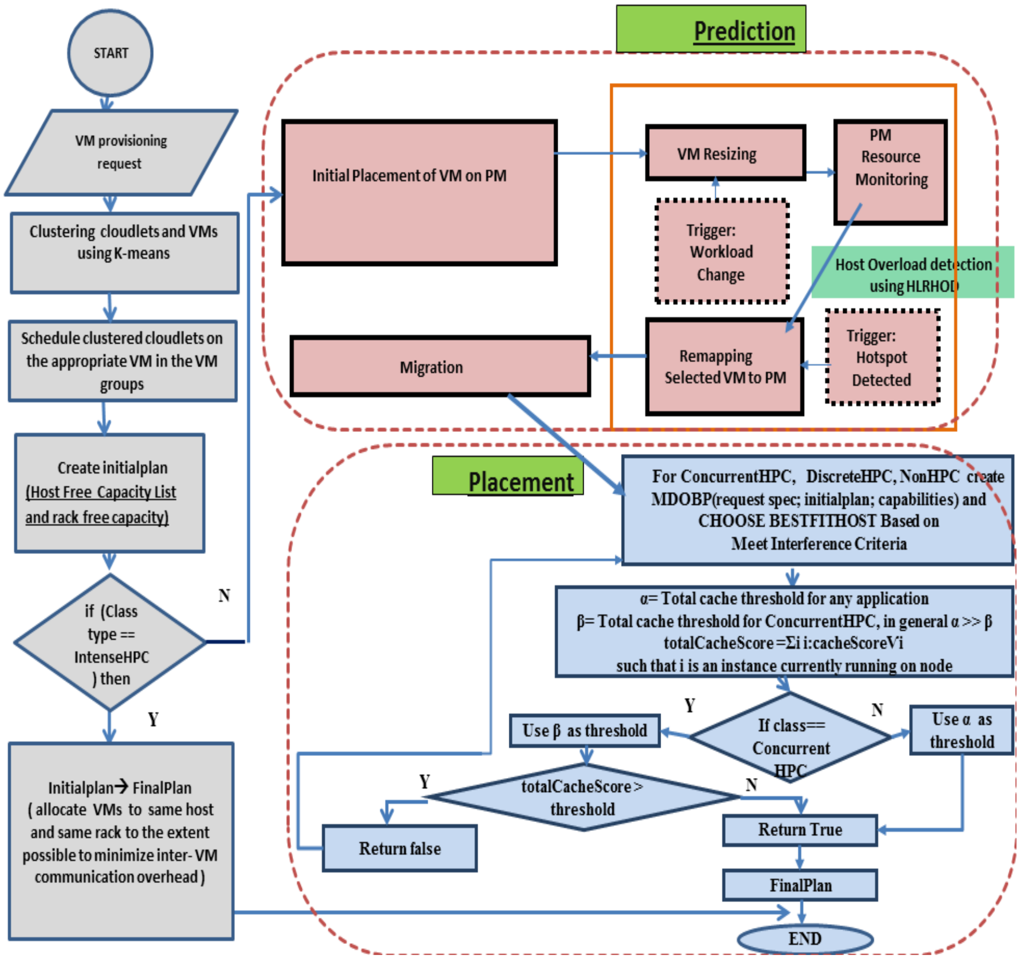


Fig. 3. Energy-Aware Multi-Dimensional Online Bin Packing (EAMDOBP).

TABLE I. POPULATION AND DESCRIPTION OF CLUSTERS RESULTING FROM CLUSTERING FIRST 250 DATA OF METACENTRUM-02.SWF LOGS OF THE PARALLEL WORKLOAD ARCHIVE

Cluster	Population	Memory usage	Clustering description
IntenseHPC (Cluster-4)	2	26749856.000	This cluster shows high memory consumption
DiscreteHPC (Cluster-3)	15	326042.000	This cluster shows moderate memory consumption
ConcurrentHPC (Cluster-1)	17	4732608.750	This cluster shows memory consumption is lower then DiscreteHPC.
non-HPC (Cluster-2)	215	86285.953	This cluster shows low memory consumption

TABLE II. K-MEAN CLUSTERS RESULT

Cluster	1	2	3	4
Number of objects by cluster	17	215	15	2
Sum of weights	17	215	15	2
Within-cluster variance	-0.223	-0.582	1.472	-0.667
Minimum distance to centroid	1.397	-0.540	-0.870	0.013
Average distance to centroid	0.276	-0.711	1.296	-0.860
Maximum distance to centroid	-0.384	0.713	0.906	-1.234

requested specification which can be deployed on a specific host and rack. To assure that IntenseHPC is only run on dedicated nodes, it sets all hosts with a active VM to zero, if the required VM category is IntenseHPC. The scheduler then prepares a initial plan, which is a list of hosts sorted by rackCapacity and hostCapacity for hosts in the same rack, if

Fig. 4. Mean Silhouette Score of Clusters.

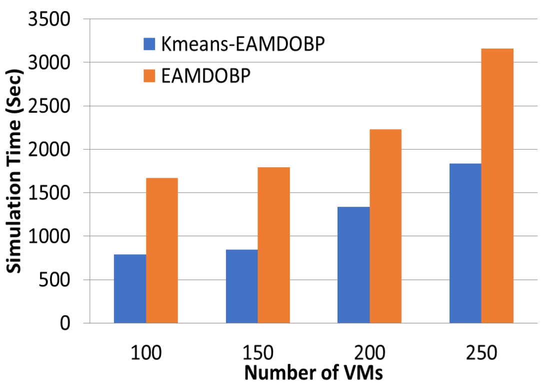


Fig. 5. Simulation Time with Kmeans and Without Kmeans

the required VM cluster is IntenseHPC or DiscreteHPC. The idea is to keep as many VMs on the same host and rack as possible to avoid inter or cross-VM communication overhead for these application types. This initialPlan is used for VM provisioning in IntenseHPC, whereas for the other classes (DiscreteHPC, ConcurrentHPC, and NonHPC), the method does multi-dimensional online bin packing to gruop VMs with variety of attributes on the same host.

C. Virtual Machine Consolidation

Since changing workloads modify VM resource utilization over time, initial VM placement should be complemented by a regular VM consolidation procedure. Status is examined for every scheduled interval by using the EAMDOBP proactive dynamic consolidation technique. Notably, the detection of host overload/underload is the initial stage in virtual machine consolidation whose main target is energy reduction. The default underload detection algorithm is applied every scheduled interval, and underutilized hosts are deactivated after moving all VM to other active PM. Afterward, Hybrid Local Regression Host Overload Detection algorithm (HLRHOD) [24] is applied to detect the overloaded hosts. VM is chosen from the overloaded hosts to migrate to available hosts. HLRHOD calculates host utilization based on hybrid factors by utilizing a metric which measures the combined CPU-network-memory load of physical and virtual servers. Following the calculation of host utilization, the program employs local regression. The basic concept behind the process of local regression is that it involves fitting simple models to confined subsets of data in order to construct a curve that approximates the original data. To identify whether the host is overloaded or underloaded, HLRHOD estimates the host utilization based on a hybrid factor. VM consolidation requires two more phases after the overloaded hosts are identified. The first phase is to identify the virtual machines (VMs) that will be migrated from the overloaded hosts to other hosts (known as VM migration), and the second phase is to replace the VMs that were selected for migration on new hosts (known as VM placement).

D. Virtual Machine Placement

Multi-dimensional online bin packing (MDOBP) algorithm is used for VM placement and it allows VMs with diverse characteristics to be placed on the same host to reduce energy and enhances the consumption of resources. MDOBP treats hosts as bins, and virtual machines as objects that must be packed into the bins. A host is represented as a d-dimensional vector, which is referred to as the host's vector of capacities. Each dimension indicates the host's capacity corresponding to a specific resource, such as CPU utilization, memory utilization, or disk bandwidth. Similarly, each virtual machine is represented by a vector of requirements. The aim is to place all of the VMs on as few hosts as appropriate, while ensuring that, across all dimensions, the total demand of VMs placed on a host does not exceed the capacity of the host.

Additionally, each job is assigned a cache score from (0-30) The cache score represents the amount of pressure being placed on the both memory controller subsystem as well as shared cache. The chosen host is furthermore verified for compliance with the interference requirements i.e estimated demand for contested resources from the physical hosts. If the sum of cache scores for the requested VM and all other VMs running on the host exceeds a certain threshold alpha α (Total cache threshold for any application), the request will be denied. which needs to be determined through experimental analysis. A different threshold β (Total cache threshold for DiscreteHPC) is used when a requested virtual machine (VM) or one or more virtual machines operating on that host are of the class DiscreteHPC, because applications of this type can tolerate less interference compared to IntenseHPC. When the cache threshold β is too high, the efficiency is decreased as cache-intensive applications on the same node are aggressively packaged. Furthermore, very low thresholds lead to an excessive waste of certain CPU cores if very small cache scores do not exist. A record of interference indices is saved to examine interference between the applications that face a big performance penalty when being executed on the same host. Having this information is beneficial in avoiding co-locations, which are detrimental to the performance of high-performance computing (HPC) applications. Following that, a FinalPlan is established, which contains a list of hosts on which the virtual machines should be provisioned. After each repetition, a log is also created, which can be used to track energy consumption and quality of service.

V. Evaluation Methodology

A. Experimental Setup

As it is very challenging to conduct repeatable large-scale experiments on a real infrastructure [28], simulations are recommended to show the improvement of our suggested algorithms. The CloudSim toolkit facilitates the modeling of

cloud system resources from both a system and a behavior perspectives specifically virtual machines, data centers, and resource management policies. It incorporates common application delivery techniques that can be extended quickly and with minimal effort. Therefore, the experiments were conducted with the CloudSim 3.0 simulation toolkit [29]using four VM types and two PM types. Cloudsim is extended with energy-aware simulations, originally not present in the core framework [29].
Furthermore, while implementing our scheduling and migration techniques the researchers identified multiple limitations in the CloudSim 3.0, that need to be addressed. They include that CloudSim is designed and implemented for the cloud, and primarily operates for tasks involving just one processor while HPC machines require quite a huge number of processors. In this research work, the researchers extended CloudSim 3.0 to provision the simulation of HPC in the cloud. Hence, the key change performed to enhance the execution of multi-core jobs to simulate HPC in a cloud thus the PowerDatacenter class that empowers simulation of power-aware data centers in the cloud environment.

In order to efficiently map jobs to virtual machines (VM), a predetermined number of virtual machines (of various types) will be constructed at the beginning of the simulation, and jobs (cloudlets) will be submitted to the DatacenterBrokerEAMDOBP broker. The cloudlets class is also modified, in addition to existing parameters, it includes the application class and name for VM provisioning. Furthermore, for cache-awareness, used a uniform distribution random number generator by giving a cache value from 0 to 30 to each job.

Further, DatacenterBroker class was also extended to DatacenterBrokerEAMDOBP , with two additional features: i) it allocates a cloudlet to VM after determining the characteristics of both VM and Cloudlets, to assign lengthy cloudlets (jobs) to the more efficient VMs so that the VM is not idle in a data center and the cloudlet execution time will be reduced. This does not only results in the efficient and improved utilization of the system but also helps to overcomes the draw backs of the defaults cloudSim 3.0 DatacenterBroker policy. ii) Furthermore, cache-awareness is also added to DatacenterBrokerEAMDOBP policy to efficiently address the issue of cross-interference. The cross-interference problem arises at a high scale when high-performance applications are executed in clouds. The proportion of additional time spent by one program when it runs simultaneously with another is known as the slowdown of that application.Thus the accurate forecasting of the slowdown due to interference in each application has many advantages: for example, it can help to enhance efficient shared resource utilization to avoid unreasonable application slowdowns from consolidation. DatacenterBrokerEAMDOBP policy will allocate cloudlets to VM that do not surpass the criteria of interference. Similar approached as Gupta [21], the interference is calculated based on following criteria: The sum of cache value of the VM requested does not exceed a threshold for any VM running on a host. The value of the threshold value is set to 60 after careful and thorough experimentation. If the threshold is set to a large value, it will decrease efficiency as a result of aggressive cache-intense applications packaged on the same node. On the other hand, too small value of threshold will result in an unreasonable waste of certain CPU cores if few applications are having insignificant cache values. Interference indices maintained to record and save interference between applications that experience high-performance penalties when hosts are shared. While DatacenterBrokerEAMDOBP policy assigns suitable cloudlet(job) to VM, the knowledge of recorded interference indices is used to prevent co-locations of HPC/non-HPC applications that consumed more resources.

Furthermore, the default CloudSim VmAllocationPolicySimple class extended to PowerVmAllocationPolicyMigrationEAMDOBP which manages a user request encompasses multiple types of VM. Two methods findHostForVm (for initial VM allocation) and optimizeAllocation (for VM consolidation) in the PowerVmAllocationPolicyMigrationEAMDOBP class carries out proposed Energy-aware multi-dimensional online bin packing scheduling. When HLRHOD [24] detects a host overload, certain VMs will be chosen to be migrated from the overloaded host to other hosts.

Furthermore, the researchers compare the proposed Energy-Aware Multi-Dimensional Online Bin Packing (EAMDOBP) algorithm against the following algorithms from the literature:

- The Power-Aware-Best-Fit Decreasing algorithm (PABFD) algorithm [28].
- The Modified- Worst-Fit Decreasing algorithm (MWFD) algorithm [30].

B. Power Model

The CPU, disk storage, memory, and cooling systems utilize the majority of the power in cloud data centers [28]. Establishing exact analytical models for modern multicore CPUs is a difficult research problem due to the complex power model of modern multicore CPUs. Hence, we employ real data on power rate obtained from the results of the SPECpower benchmark [28] as an alternative to the use of an analytical model of a host's power consumption. The host overload is evaluated on a regular basis according to the scheduling interval, which is set at 300 seconds. The host types are: HP ProLiant ML110 G4 (Intel Xeon 3040, 2 cores 1860 MHz, 4 GB), and HP ProLiant ML110G5 (Intel Xeon 3075, 2 cores 2660 MHz, 4 GB). The power consumption features of the chosen hosts are presented in Table III.

C. Performance metrics

To conduct in-depth analysis of the suggested algorithm and to evaluate and contrast the algorithm's performance, several number of experiments were carried out and examined in the current research using the following metrics:

Simulation Time (ST): It is defined as the amount of time spent conducting an experiment in seconds, commonly known as the makespan. Based on the number of applications that were processed, the total time required to generate the simulation was calculated. The simulation time (MakeSpan) is used to determine the efficiency of the algorithm.

Throughput: In computing, it is the quantity of work that a computer or a system of computers is able to perform in a given period of time. Increasing throughput is an ongoing challenge that IT managers, researchers, and scientists must

TABLE III. POWER CONSUMED BY THE CHOSEN HOSTS AT VARIOUS LOAD LEVELS IN WATTS [28]

SERVER	**0%**	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
HP PROLIANT G4	86	89.4	92.6	96	99.5	102	106	108	112	114	117
HP PROLIANT G5	93.7	97	101	105	110	116	121	125	129	133	135

meet and exceed. The throughput of a system is used to evaluate its overall performance.

Efficient Resource Utilization (ERU): It is used determine how much CPU, memory and network bandwidth are needed for HPC and non-HPC workloads deployed in cloud-based data centers

Power Usage Effectiveness (PUE): The Power Usage Effectiveness (PUE) is used as indices for measurement of datacenter performance. PUE is considered the most used datacenter metric. The PUE is calculated using Eq. 2, where P_{CS} is the cooling power, and P_C is the computing power of the host. This calculation is iterated over each host.

$$PUE = \frac{P_{CS} + P_C}{P_C} \tag{2}$$

Table IV shows the PUE values based on several experiments which were performed in a small datacenter [31].Low PUE indicates higher efficiency as a significant amount of the power has been consumed by computing power [32].

TABLE IV. PUE EFFICIENCY VALUES [31]

PUE	**Level of efficiency**
3.0	Very inefficient
2.5	Inefficiency
2.0	Average
1.5	Efficient
1.2	very Efficient
1.1	Standard

The number of VM migrations: For dynamic VM consolidation, when the overloaded or under-loaded hosts are detected, the VMs are then chosen to move. Reducing the VM migration time is the most significant obstacle in the migration step and the default method to achieve that is by reducing the total number of VM migrations.

Energy and SLA Violations (ESV): There is an adverse relation between the energy consumed by physical hosts and SLAV because energy can be frequently reduced by allowing more SLA violations. The objective of the host management framework is to decrease both energy consumption and service level agreement violations. Thus, a mixed metric denoted by Energy and SLA Violations (ESV) is proposed in [28] and is shown in Eq. 3. For the ESV metric lower is better.

$$ESV = E \times SLAV \tag{3}$$

D. Workloads

The workload contains several months of accounting records from the national grid of the Czech Republic, called Metacentrum. This grid is composed of 14 clusters (called nodes), each with several multiprocessor machines, for a total of 806 processors [27]. The workload data contains CPU and memory usage [33]. Standard workload format (SWF) includes the log. METACENTRUM-2013-1.swf. is used while relying on accounting data collected by the scheduler. three utilization models rebuilt to examine the CPU and RAM, and calculate the BW, respectively, from the workload and send it to the cloudlets.

VI. SIMULATION RESULTS AND ANALYSIS

Memory utilization, workload length and number of VMs are crucial parameters that are used to configure the algorithms implemented besides the performance metrics presented in Section V

A. Sensitivity Analysis

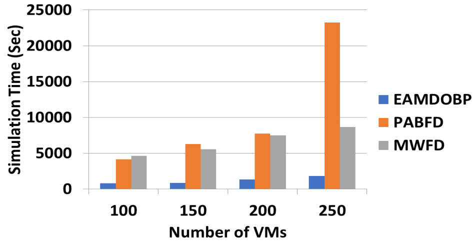
The effect of changing the number of VMs on the performance of the proposed algorithms in terms of metrics is provided in Section 5.3.

1) Number of VMs:

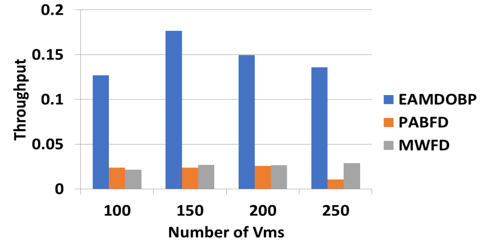
Our sensitivity analysis is based on changing the number of virtual machines while fixing the number of hosts to 800 and scheduling interval to 300 sec. The effect of varying the number of VMs using Metacentrum workload traces on the energy consumption and other system and performance metrics for different algorithms are examined.

Algorithms comparison are performed on (CPU, RAM, and Network) traces using Metacentrum HPC workload. Measurement of simulation time is one of the most critical indicators for measuring performance in a dynamic system. Observe in Fig. 6(a), EAMDOBP algorithm reduces the execution times better than the other algorithms. This proofs that application awareness results in fewer contentions for resources as only the most compatible VMs are consolidated. The new EAMDOBP algorithm results in the lowest simulation time compared to PABFD and MWFD regardless of the number of VM. The simulation time is the highest in the case of PABFD when the number of VM is high. Besides, the simulation time is always less in the case of MWFD when compared to PABFD.

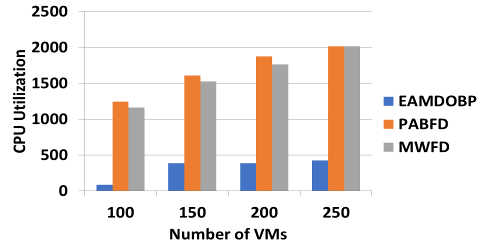
The HPC framework involves the co-scheduling of tasks with the same nature to make some synchronous development. In the present virtualized environment, all VMs progress independently of each other, so that the need that all VMs of the same HPC function must be arranged together using EAMDOBP algorithm. Results are shown in Fig. 6(b) indicate



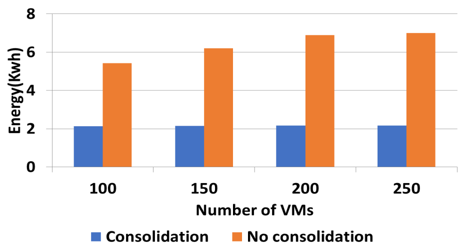
(a) Simulation Time



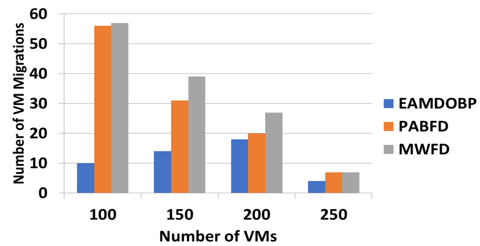
(b) Throughput Metric



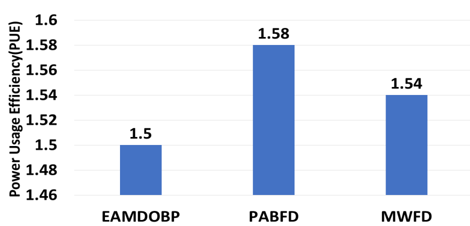
(c) CPU Utilization Metric



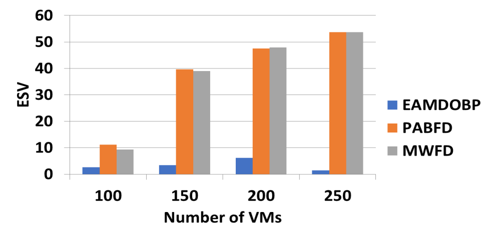
(d) Energy Consumption



(e) Number of VM Migration



(f) PUE Metric



(g) ESV Metric

Fig. 6. Algorithms Comparison for Metacentrum HPC workload, where the various metrics are depicted in log-scale

that the throughput levels are the highest when the EAMDOBP algorithm is used. While throughput levels are almost equal when using PABFD and MWFD algorithms.

It is important to address the issue of machine utilization or "busy" time that is commonly used to evaluate throughput effectiveness. It is often only moderately correlated with a hardware measure such as utilization. The prediction method of the resource utilization based on HLRHOD is used, Fig. 6(c) indicates better CPU utilization. The use of HLRHOD prediction technique improves CPU utilization as well as increase the effectiveness of HPC-application execution

B. Algorithms Comparative Analysis

The EAMDOBP forehand has the expertise to forecast the best node for VM allocation without venturing with the energy consumption. While the energy and interference-aware co-location of the VMs is used for all the workload allocations to the VMs running on the system nodes. The need for rising VM migrations is therefore low and it helps to reduce energy depletion. The same is shown through Fig. 6(d) and (e). The graph in Fig. 6(d) indicates how energy usage increases with the increase in the number of VM when no consolidation is allowed. Significantly, energy consumption is fixed regardless of the increase in the number of VMs when consolidation is allowed, meaning that the VMs are allowed to migrate to run on fewer physical servers. Thus, VM consolidation helps reduce energy consumption. As observed in Fig. 6(e) VM migrations are the lowest when using the EAMDOBP algorithm compared to PABFD and MWFD. VM migration is the highest when using MWFD. Each host's cooling and processing power, as well as its power utilization efficiency (PUE), are measured after each migration, and the impacts are averaged. Results depicted in Fig. 6(f) indicate that PUE is the lowest in the case of EAMDOBP compared to PABFD and MWFD. It is worth mentioning that low PUE indicates better efficiency. This is because a considerable part of the power is consumed by computing power. However, PUE is the highest in the case of PABFD. Results revealed in Fig. 6(g) indicate that ESV is the lowest when using the EAMDOBP algorithms compared to PABFD and MWFD. ESV levels increase significantly with the increase in the number of VM when using PABFD and MWFD algorithms.

VII. Conclusion

Cloud computing has become a popular solution for the exponential growth in the demand for high-performance computing. Huge data centers use a considerable amount of energy, which leads to an increase in operating costs. Therefore, virtual machine consolidation is a perfect solution as it allows VM live migration to run on fewer physical servers to save energy consumption. Many studies have tried to investigate the currently available cloud service architectures for running HPC applications in the most effective approach. The major obstacles are cross-application interference, energy concerns, and guaranteeing SLAs in different metrics, e.g. response time (web application) vs. execution time (HPC application). However, for a successful VM consolidation host overload detection is necessary to predicts if a physical server will be overloaded with VMs. This paper proposes new algorithm named Energy-Aware Multi-Dimensional Online Bin Packing (EAMDOBP) algorithm that will provide better results. Specifically, In comparison to PABFD and MWFD, experiments reveal that the Energy Aware Multi-Dimensional Online Bin Packing (EAMDOBP) has improved CPU, RAM, and bandwidth consumption by a relative improvement of 77%, 84%, and 70%, respectively. The Energy Aware Multi-Dimensional Online Bin Packing (EAMDOBP), according to experiments, improves CPU usage, lowers PUE, and rationalizes energy consumption. Additionally, EAMDOBP achieves faster throughput, less VM migration, and lower ESV when compared to PABFD and MWFD. According to the results analysis, the EAMDOBP algorithm outperforms PABFD and MWFD in terms of all the parameters employed. In current paper the researcher has focused on bridging the HPC-cloud gap by enhancing application performance and resource and energy efficiency however, there are two important limitations in this study that could be addressed in future investigations. First, the only information provided by current performance analysis tools is the application's performance. They don't go into depth about the additional tasks that could have used up other parameters such the network and I/O. Also, the work should be extended to be run on a real cloud system not simulation. In future, we plan to run our algorithm on a real cloud system; also we will consider other factors such as I/O which can affect performance of VMs. We can apply other machine learning techniques such as Naive Bayesian classifiers for classification of VMs.

Acknowledgment

This research was supported in part by the the cloud computing center of excellence grant number 5220, Science and Technology Development Fund (STDF), Egypt, and in part by the Distributed and Networked Systems Research Group Operating Grant number 150410, University of Sharjah, UAE.

References

[1] P. Mell, T. Grance *et al.*, The nist definition of cloud computing (2011).

[2] V. Mauch, M. Kunze and M. Hillenbrand, High performance cloud computing, *Future Generation Computer Systems* **29**(6) (2013) 1408–1416.

[3] H. Moab, Suite, *URL: adaptivecomputing. com/products/hpc-products/moab-hpc-basic-edition/basic-edition-solution-architecture* (2019).

[4] S. Kannan, M. Roberts, P. Mayes, D. Brelsford and J. F. Skovira, Workload management with loadleveler, *IBM Redbooks* **2**(2) (2001) p. 58.

[5] D. A. Menasce and V. A. Almeida, *Capacity Planning for Web Performance: metrics, models, and methods* (Prentice-Hall, Inc, 1998)

[6] M. Gienger, High performance computing in the cloud: A survey on performance and usability, in *Sustained Simulation Performance 2015*, (Springer, 2015) pp. 29–40.

[7] A. Shehabi, S. Smith, D. Sartor, R. Brown, M. Herrlin, J. Koomey, E. Masanet, N. Horner, Azevedo and W. Lintner, United states data center energy usage report, tech. rep., Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States) (2016).

[8] A. A. El-Moursy, A. Abdelsamea, R. Kamran and M. Saad, Multi-dimensional regression host utilization algorithm (mdrhu) for host overload detection in cloud computing, *Journal of Cloud Computing* **8**(1) (2019) 1–17.

[9] Z. Li, X. Yu, L. Yu, S. Guo and V. Chang, Energy-efficient and quality-aware vm consolidation method, *Future Generation Computer Systems* **102** (2020) 789–809.

[10] H. Jin, H. Qin, S. Wu and X. Guo, Ccap: a cache contention-aware virtual machine placement approach for hpc cloud, *International Journal of Parallel Programming* **43**(3) (2015) 403–420.

[11] M. M. Alves, L. Teylo, Y. Frota and Drummond, An interference-aware virtual machine placement strategy for high performance computing applications in clouds (2018) 94–100.

[12] A. Souza, M. Rezaei, E. Laure and J. Tordsson, Hybrid resource management for hpc and data intensive workloads, *19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2019, May 14-17, 2019, Larnaca, Cyprus* (2019) 399–409.

[13] M. Koot and F. Wijnhoven, Usage impact on data center electricity needs: A system dynamic forecasting model, *Applied Energy* **291** (2021) p. 116798.

[14] L. A. Barroso and U. Hölzle, The case for energy-proportional computing, *Computer* **40**(12) (2007) 33–37.

[15] A. Shahidinejad, M. Ghobaei-Arani and M. Masdari, Resource provisioning using workload clustering in cloud computing environment: a hybrid approach, *Cluster Computing* **24**(1) (2021) 319–342.

[16] Y.-C. Shim, Performance evaluation of static vm consolidation algorithms for cloud-based data centers considering inter-vm performance interference, *International Journal of Applied Engineering Research* **11**(24) (2016) 11794–11802.

[17] Q. Zheng, R. Li, X. Li, N. Shah, J. Zhang, F. Tian, K.-M. Chao and J. Li, Virtual machine consolidated placement based on multi-objective biogeography-based optimization, *Future Generation Computer Systems* **54** (2016) 95–122.

[18] A. Mosa and R. Sakellariou, Dynamic virtual machine placement considering cpu and memory resource requirements, *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)* (2019) 196–198.

[19] A. Kraemer, C. Maziero, O. Richard and D. Trystram, Reducing the number of response time service level objective violations by a cloud-hpc convergence scheduler, *Concurrency and Computation: Practice and Experience* **30**(12) (2018) p. e4352.

[20] V. Antonenko, A. Chupakhin, I. Petrov and R. Smeliansky, Improving resource usage in hpc clouds (2019).

[21] A. Gupta, L. V. Kale, D. Milojicic, P. Faraboschi and S. M. Balle, Hpc-aware vm placement in infrastructure clouds, *2013 IEEE International Conference on Cloud Engineering (IC2E)* (2013) 11–20.

[22] S. Di, D. Kondo and F. Cappello, Characterizing cloud applications on a google data center, in *2013 42nd International Conference on Parallel Processing*, IEEE2013, pp. 468–473.

[23] I. S. Moreno, P. Garraghan, P. Townend and J. Xu, Analysis, modeling and simulation of workload patterns in a large-scale utility cloud, *IEEE Transactions on Cloud Computing* **2**(2) (2014) 208–221.

[24] A. Abdelsamea, A. A. El-Moursy, E. E. Hemayed and H. Eldeeb, Virtual machine consolidation enhancement using hybrid regression algorithms, *Egyptian Informatics Journal* **18** (2017) p. 161–170.

[25] S. Zhuravlev, J. C. Saez, S. Blagodurov, A. Fedorova and M. Prieto, Survey of scheduling techniques for addressing shared resources in multicore processors, *ACM Computing Surveys (CSUR)* **45**(1) (2012) 1–28.

[26] S. Li, L. Wang, W. Wang, Y. Yu and B. Li, George: Learning to place long-lived containers in large clusters with operation constraints, in *Proceedings of the ACM Symposium on Cloud Computing*, 2021, pp. 258–272.

[27] "The metacentrum hpc workload", Cs.huji.ac.il, 2022. [Online]. Available: http://www.cs.huji.ac.il/labs/parallel/metacentrum. [Accessed: 22-Sep- 2022]

[28] A. Beloglazov and R. Buyya, 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* **24**(13) (2012) 1397–1420.

[29] R. N. Calheiros, R. Ranjan, A. Beloglazov, B. De Rose and Rajkumar, 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.

[30] M. R. Chowdhury, M. R. Mahmud and R. M. Rahman, Implementation and performance analysis of various vm placement strategies in cloudsim, *Journal of Cloud Computing* **4**(1) (2015) p. 20.

[31] M. Uddin, A. Shah, R. Alsaqour and J. Memon, Measuring efficiency of tier level data centers to implement green energy efficient datacenters, *Middle-East Journal of Scientific Research* **15**(2) (2013) 200–207.

[32] M. J. Usman, A. S. Ismail, G. Abdul-Salaam, H. Chizari, O. Kaiwartya, A. Y. Gital, M. Abdullahi, A. Aliyu and S. I. Dishing, Energy-efficient nature-inspired techniques in cloud computing datacenters, *Telecommunication Systems* **71**(2) (2019) 275–302.

[33] "Parallel Workloads Archive", Cs.huji.ac.il, 2022. [Online]. Available: http://www.cs.huji.ac.il/labs/parallel/workload. [Accessed: 22-Sep- 2022]

[34] Chinnasamy, P and Vinothini, B and Praveena, V and Subaira, AS and Sujitha, B Ben: Providing Resilience on Cloud Computing, in *2021 International Conference on Computer Communication and Informatics (ICCCI)/*, IEEE 2021, pp. 1–4.

[35] Hartigan, J. A., and M. A. Wong: Algorithm AS 136: A K-Means Clustering Algorithm.*Journal of the Royal Statistical Society. Series C (Applied Statistics*, 1979, pp"100–08" .

[36] P. J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, *Journal of computational and applied mathematics*, Elsevier, 1987 .