International Journal of Innovation and Applied Studies
ISSN 2028-9324 Vol. 8 No. 4 Oct. 2014, pp. 1504-1516

http://www.ijias.issr-journals.org/

Virtual Machine Consolidation Challenges: A Review

Amany Abdelsamea[1], Elsayed E. Hemayed[2], Hesham Eldeeb[1], and Hanan Elazhary[1]

[1]Computers & Systems Department,
Electronics Research Institute,
Giza, Egypt

[2]Computer Engineering Department,
Faculty of Engineering, Cairo University,
Giza, Egypt

ABSTRACT: Virtualization is a powerful technology that facilitates better use of the available data center resources using a technique called Virtual Machine (VM) consolidation which involves gathering of several virtual machines into a single physical server. To address the problem of high energy usage, it is necessary to eliminate inefficiencies and waste in the way electricity is delivered to computing resources, and in the way these resources are utilized to serve application workloads. This can be done by improving the physical infrastructure of data centers as well as resource allocation and management algorithms. VM consolidation involves live migration, which is the capability of transferring a VM between physical servers with a close to zero down time is an effective way to improve the utilization of resources and energy efficiency in cloud data centers. VM placement and VM migration act as a backbone to the VM consolidation process. Issues such as heterogeneity and scalability of physical resources, volatile workloads and migration cost make the VM consolidation process difficult. This paper presents a comprehensive survey of different VM consolidation challenges such as host underload detection, host overload detection, VM selection, VM live migration and VM placement algorithms. The paper discusses these VM consolidation challenges and presents a comparison between different state-of-the-art VM consolidations algorithms.

KEYWORDS: Cloud computing, Hypervisor, Virtualization, VM selection, VM live migration, VM placement.

1 INTRODUCTION

Cloud computing is defined by NIST [1] as a model for enabling convenient, on demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimum management effort or service provider interaction. Several other definitions have been proposed for cloud computing [2], but they all imply the existence of a shared pool of computing resources. Virtualization plays a vital role in managing and coordinating access to the resource pool via a software layer called Virtual Machine Monitor (VMM) or hypervisor. It hides the details of the physical resources and provides virtualized resources for high level applications. Moreover, it virtualizes all of the resources of a given physical machine (PM) allowing several virtual machines (VM) to share its resources [3]. It should be noted that an essential characteristic of a virtual machine is that the software running on it is limited to the resources and abstractions provided by the virtual machine. Virtualization also allows gathering several virtual machines into a single physical server using a technique called VM consolidation.

VM consolidation can provide significant benefits to cloud computing by facilitating better use of the available data center resources. It can be performed either statically or dynamically. In static VM consolidation, the VMM allocates the physical resources to the VMs based on peak load demand (overprovision). This leads to resource wastage because the workloads are not always at peak. On the other hand, in case of dynamic VM consolidation, the VMM changes the VM capacities according to the current workload demands (resizing). This helps in utilizing the data centers resources efficiently.

In dynamic VM consolidation, the VMs can be dynamically reallocated using live migration according to the current PM resource demand to minimize the number of active physical servers, referred to as hosts, required to handle the workload. The idle hosts are switched to low power modes with fast transition times to eliminate the static power (power consumed while there is no circuit activity so it is considered as wasted energy) and reduce the overall energy consumption. The hosts are reactivated with the increase of the resource demand to avoid violating QoS requirements. This approach has basically two objectives, namely minimization of energy consumption and maximization of the quality of service (QoS) delivered by the system, which forms an energy-performance tradeoff [4]. VM consolidation is faced by a large number of challenges that are discussed in this paper.

The paper is organized as follows: Section 2 explains virtual machine consolidation steps and challenges. Section 3 explains host overload detection algorithms. VM selection algorithms, VM live migration algorithms, and VM placement algorithms are discussed in sections 4, 5, and 6 respectively. Finally, the conclusions are provided in section 7.

2 VIRTUAL MACHINE CONSOLIDATION STEPS AND CHALLENGES

As mentioned in Section1, dynamic consolidation of virtual machines can improve the utilization of resources and reduce energy consumption by live migration, the capability of transferring a VM between physical servers (referred to as hosts, or nodes) with a close to zero downtime. VM consolidation involves the following steps [3] as shown in Fig. 1:

- Gathering the requirements of each VM, sizing the VM and placing the VM on a suitable PM (initial placement).
- Resizing each VM according to workload demands.
- Monitoring and profiling the resource utilization for hotspot detection (overloaded and underloaded PMs).
- Remapping the VM to another PM and thus migration according to a placement plan.

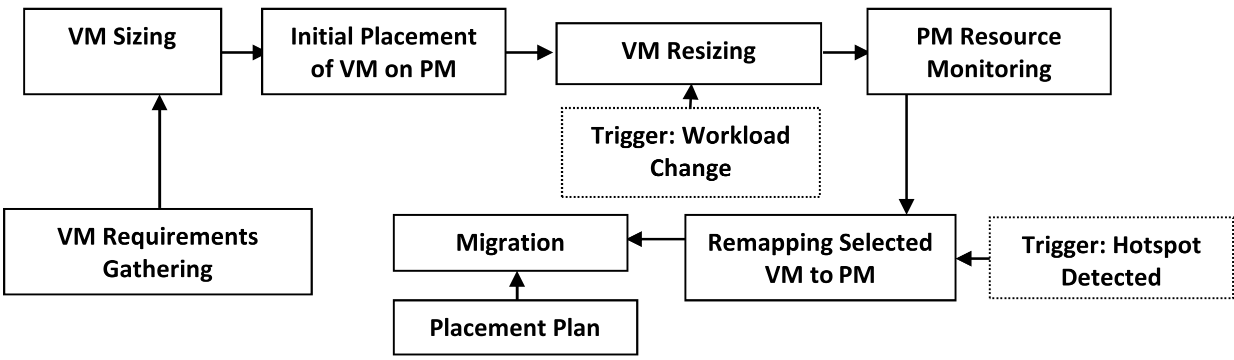

Fig. 1. Virtual Machine Consolidation Steps

VM placement and VM migration both act as a backbone to the VM consolidation process. The issues such as heterogeneity and scalability of physical resources, volatile workloads and migration cost make the VM consolidation process difficult. The basic challenges for efficient VM consolidation can be summarized as follows:

- **Host underload detection:** Deciding if a host is considered to be underloaded so that all VMs should be migrated from it and the host should be switched to a low-power mode (to minimize the number of active physical servers).
- **Host overload detection:** Deciding if a host is considered to be overloaded so that some VMs should be migrated from it to other active or reactivated hosts (to avoid violating the QoS requirements).
- **VM selection:** Selecting VMs to migrate from overloaded host.
- **VM live migration**: Performing VM migration process with minimal service downtime and resource consumption during migration process.
- **VM placement:** Virtual machine placement by placing VMs selected for migration on other active or reactivated hosts.

For host underload detection the VMs can be migrated from the least utilized host or set a static threshold to decide if a host is underloaded as presented in [4].The other challenges and the corresponding algorithms proposed in the literature are discussed in more details in the following sections.

3 HOST OVERLOAD DETECTION ALGORITHMS

Detecting when a host becomes overloaded directly influences the QoS, because if the resource capacity is completely utilized, it is highly likely that the applications are experiencing resource shortage and performance degradation calling for the migration of some VMs from it. What makes the problem of host overload detection complex is the necessity to optimize the time-averaged behavior of the system, while handling a variety of heterogeneous workloads placed on a single host. To address this problem, most of the current approaches to dynamic VM consolidation apply either heuristic-based techniques, such as static utilization thresholds, adaptive utilization threshold based on statistical analysis of historical data; or regression based algorithms as shown in Table 1. The limitations of these approaches are that they lead to suboptimal results and do not allow the administrator to explicitly set a QoS goal [5].

Table 1. Types of Host Overload Detection Algorithms

Type	**Static utilization threshold based algorithms**	**Adaptive utilization threshold based algorithms**	**Regression based algorithms**
Explanation	Based on fixed CPU utilization threshold	Based on statistical analysis of historical data of VM	Based on estimation of future CPU utilization
Pros	Simple	Robust in case of dynamic environments	Better predictions of host overloading
Cons	Unsuitable for dynamic environment	Poor prediction of host overloading	Not simple
Examples	THR (averaging threshold-based algorithm)	MAD (Median Absolute Deviation), IQR (Inter Quartile Range)	LR (Local Regression), LRR (Local Robust Regression)

3.1 STATIC CPU UTILIZATION THRESHOLD ALGORITHMS

One of the simplest overload detection algorithms is based on an idea of setting a CPU utilization threshold distinguishing the non-overload and overload states of the host. When the algorithm is invoked, it compares the current CPU utilization of the host with the defined threshold. If the threshold is exceeded, the algorithm detects a host overload [6]. An example of static CPU utilization threshold based algorithms is the averaging threshold-based algorithm (THR) [4]. The algorithm calculates the mean of the n latest CPU utilization measurements and compares it to the specified threshold. It detects overload situations if the mean of the n last CPU utilization measurements is higher than the specified threshold. Unfortunately, fixed values of utilization thresholds are unsuitable for an environment with dynamic and unpredictable workloads, in which different types of applications can share a physical PM. The system should be able to automatically adjust the utilization threshold depending on the workload patterns exhibited by the applications.

3.2 ADAPTIVE UTILIZATION THRESHOLD BASED ALGORITHMS

These algorithms provide auto-adjustment of the utilization thresholds based on a statistical analysis of historical data collected during the lifetime of the VMs. Its main idea is to adjust the value of the upper utilization threshold depending on the strength of deviation of the CPU utilization. The higher the deviation, the lower the value of the upper utilization threshold, as the higher the deviation, the more likely that the CPU utilization will reach 100% and cause an SLA violation. Example algorithms are Median Absolute Deviation (MAD) and Interquartile Range (IQR) [7]:

- Median Absolute Deviation (MAD): It is a measure of statistical dispersion that behaves better with distributions without a mean or variance and is a more robust estimator of scale in comparison to sample variance or standard deviation, as it. In case of the standard deviation, the distances from the mean are squared, so on average, large deviations are weighted more heavily, and thus outliers can heavily influence it. On the other hand, in case of MAD, the magnitude of the distances of a small number of outliers is irrelevant.
- Interquartile Range (IQR): It is another measure of statistical dispersion. It is also called the midspread or middle fifty as it equals the difference between the third and first quartiles in descriptive statistics. The interquartile range is a robust in comparison to the (total) range, having a breakdown point of 25%, and is thus preferred to the total range.

Adaptive utilization threshold algorithms are more robust than static CPU utilization threshold algorithms in case of dynamic environments. But, unfortunately, they provide poor prediction of host overloading.

3.3 REGRESSION BASED ALGORITHMS

They are based on the estimation of the future CPU utilization. They provide better predictions of host overloading, but are more complex. Example algorithms include Local Regression algorithms (LR) [8] such as Loess method and Local Regression Robust (LRR) [9], which is a modification of LR robust to outliers.

Table 2. Host Overload Detection State-of-Art Algorithm Comparison

Technique	Type	Parameters considered				Performance metrics accounted					
		Utilization threshold	Median absolute deviation (MAD)	Interquartile range (IQR)	Future expectations of CPU utilization	CPU utilization	Energy (E)	SLA violation (SLAV)	E.SLAV (ESV)	SLA violation Time per Active Host (SLATAH)	Performance degradation due Migrations (PDM)
THR [4]	Static CPU utilization threshold	✓				✓	✓	✓	✓	✓	✓
MAD [7]	Adaptive utilization threshold		✓				✓	✓	✓	✓	✓
IQR [7]	Adaptive utilization threshold			✓			✓	✓	✓	✓	✓
LR [8]	Regression based				✓		✓	✓	✓	✓	✓
LRR [9]	Regression based				✓		✓	✓	✓	✓	✓

Meeting QOS requirements is extremely important for cloud computing environments. Qos requirements are commonly formalized in the form of SLAs which can be determined in terms of such characteristics as minimum throughput or maximum response time delivered by the deployed system. As these characteristics can vary for different applications, it is necessary to use a workload independent metric that can be used to evaluate SLA delivered to any VM deployed in an IaaS.

The main metrics used are energy consumption by physical nodes and SLAV (SLA violation), however these metrics are typically negatively correlated as energy can usually be decreased by the cost of the increased level of SLA violations. Since the objective of resource management system is to minimize both energy and SLA violations a combined metric is denoted by Energy and SLA violation (ESV) is used. Also two other metrics are used which are the percentage of time during which active hosts have experienced the CPU utilization of 100%, SLA violation Time per Active Host (SLATAH) and the overall performance degradation by VMs due to migrations, Performance Degradation due to Migrations (PDM). Both SLATAH and PDM metrics independently and with equal importance characterize the level of SLA violations. Since SLAV = SLATAH.PDM.

A comparison of the state of art of host overload detection algorithms is summarized in Table 2. The following conclusions have been assessed [7]:

- Heuristic-based dynamic VM consolidation algorithms substantially outperform the static CPU utilization threshold algorithm (THR) due to a vastly reduced level of SLA violations.
- Dynamic VM consolidation algorithms based on local regression outperform the threshold-based and adaptive-threshold based algorithms due to better predictions of host overloading, and therefore decreased SLA violations due to host overloading (SLATAH) and the number of VM migrations.
- The algorithm based on local regression produces better results than its robust modification, which can be explained by the fact that for the simulated workload it is more important to react to load spikes instead of smoothing out such outlying observations.

4 VM SELECTION ALGORITHMS

Once a host overload has been detected, it is necessary to determine what VMs are the best to be migrated from the host. This problem is solved by VM selection algorithms. VM selection algorithms can be classified into:

- Techniques that use fixed criteria for selecting VMs.
- Techniques that apply multiple criteria for selecting VMs.

4.1 TECHNIQUES THAT USE FIXED CRITERIA

They employ a fixed criterion for decision-making so they are not suitable in dynamic environments. Some examples of those techniques are presented

- Dynamic management algorithm (DMA): It defines a VM selection criterion based on CPU utilization of VMs. In fact, the VMs with the lowest CPU utilization are selected. It was proved that selecting VMs based on this criterion can result in minimized migration cost [10].
- The Minimum Migration Time Policy (MMT): This policy selects a VM that requires the minimum time to complete a migration relative to the other VMs allocated to the host. The migration time is estimated as the amount of RAM utilized by the VM divided by the spare network bandwidth available for the host [7], [11], [12].
- The Random Choice Policy (RC): This policy selects a VM to be migrated according to a uniformly distributed discrete random variable whose values index a set of VMs allocated to a host [7], [11].
- The Maximum Correlation Policy (MC): This policy is based on the idea that the higher the correlation between the resource usage by applications running on an oversubscribed server, the higher the probability of the server overloading. According to this idea, the VMs to be migrated are those that have the highest correlation of the CPU utilization [13]. To estimate the correlation between CPU utilization with VMs, the multiple correlation coefficient is applied. The multiple correlation coefficient is used in multiple regression analysis to assess the quality of the prediction of the dependant variable. It corresponds to the squared correlation between the predicted and the actual values of the dependant variable. It can also be interpreted as the proportion of the variance of dependant variable explained by the independent variables [6] [7].

4.2 TECHNIQUES THAT APPLY MULTIPLE CRITERIA FOR SELECTING VMS

The VM selection task is considered as a dynamic decision making task. An example of these techniques is VM selection using Fuzzy Q-Learning (FQL).

- VM Selection using Fuzzy Q-Learning (FQL): VM selection, on the other hand, has been somehow neglected in the past, often using fixed criteria instead of dynamic ones. The VM selection task is considered as a Dynamic Decision-Making (DDM) task and thus can be modeled using Fuzzy Q-Learning (FQL). It is shown that FQL can be used to dynamically choose VM selection strategies from a set of possible strategies in order to achieve better results than each of the individual strategies achieves when used alone [14].

A comparison between state-of-art existing VM selection algorithms is presented in Table 3.

Table 3. VM Selection State-of-Art Algorithm Comparison

Technique	Parameters considered					Performance metrics accounted					
	Upper Utilization threshold	Lower Utilization threshold	Memory Size	Network bandwidth	No. VMs	CPU utilization	Energy (E)	SLA violation (SLAV)	ESV	No. VM migration	Migration cost
DMA [10]			✓	✓	✓	✓					✓
MMT [15]	✓	✓	✓	✓			✓	✓	✓	✓	
RC [15]	✓	✓	✓	✓			✓	✓	✓	✓	
MC [7]			✓	✓			✓	✓	✓	✓	
FQL [14]					✓	✓	✓	✓	✓	✓	

It has been concluded that:

- The MMT policy [15] produces better results compared to the MC policy [7], meaning that the minimization of the VM migration time is more important than the minimization of the correlation between VMs allocated to a host and the resources.
- DMA [9], MMT [15], and MC [7] employ a fixed criterion for decision-making so they are not suitable for decision-making in dynamic environments.
- A Fuzzy Q-Learning (FQL) approach as an online decision making strategy enhances the VM selection task in a dynamic VM consolidation procedure. It is able to integrate multiple VM selection criteria to benefit from all advantages and possible synergetic contributions of them in long-term learning. In fact, this approach learns how to find an optimal

strategy to applying multiple criteria for selecting VMs during a dynamic VM consolidation procedure towards improving the energy performance trade-off.

5 VM LIVE MIGRATION ALGORITHMS

Live migration makes possible for VMs to be migrated without considerable downtime. The transfer of a VM actually refers to the transfer of its state. This includes its memory, internal state of the devices and that of the virtual CPU. Among these, the most time-consuming one is the memory transfer. Two parameters to be considered while performing the live VM-migration are downtime and migration time. Migration time refers to the amount of time required to transfer a virtual machine at source node to destination node without affecting its availability. Down time refers to the time during which the service of the VM is not available. Total migration time and downtime are two key performances metric that the clients of a VM service care about the most; because they are concerned about service degradation and the duration that the service is completely unavailable [16].

Virtual machine migration techniques are classified according to their goals into [17]:

- Fault Tolerant Migration Techniques: Fault tolerance allows the virtual machines to continue its job even any part of system fails. This technique migrates the virtual machine from one physical server to another physical server based upon the prediction of the failure occurred ,fault tolerant migration technique is to improve the availability of physical server and avoids performance degradation of applications.
- Load Balancing Migration Techniques: The Load balancing migration technique aims to distribute load across the physical servers to improve the scalability of physical servers in cloud environment. The Load balancing aids in minimizing the resource consumption, implementation of fail-over, enhancing scalability, avoiding bottlenecks and over provisioning of resources etc.
- Energy Efficient Migration Techniques: The power consumption of Data center is mainly based on the utilization of the servers and their cooling systems. The servers typically need up to 70 percentage of their maximum power consumption even at their low utilization level. Therefore, there is a need for migration techniques that conserves the energy of servers by optimum resource utilization.

Also, the migration mechanisms have been divided into three main branches [18]:

- Process migration: Process migration which was born along with distributed systems is the beginning of VM's migration. The mechanisms based on process migration have some characteristics such as complexity, performance, transparency, fault resilience, scalability and heterogeneity that have a major impact on the effectiveness and deployment of process migration.
- Memory migration: The memory state migration has been widely studied and basically there are two methods post copy and pre-copy. Post copy migration transfers a VM's memory contents after its processor state has been sent to the target host. It differs from the pre-copy approach which first copies the memory state to the destination through a repetitive process after which its processor state is transferred to the target.
- Suspend /Resume Migration: It refers to the motion of a VM from one host to another where the VM is locally inactive during the translation. Usually this is a migration across a WAN. The network connections are typically dropped and need to be re-established. For a WAN migration, it is crucial to not just transfer the VM images but also transfer the local persistent state, its ongoing network connections, and the support for the disconnected operation.

Some of live VM migration techniques are discussed below.

- Pure stop-and-copy approach: This involves halting the original VM, copying all pages to the destination, and then starting the new VM. This has advantages in terms of simplicity but the service downtime is proportional to the amount of physical memory allocated to the VM. This can lead to an unacceptable outage if the VM is running a live service [19].
- Post-copy approach: A short stop-and-copy phase transfers essential kernel data structures to the destination. The destination VM is then started, and other pages are transferred across the network on demand. This results in a much shorter downtime, but produces a much longer total migration time [20], [21].
- Pre-copy migration: It iteratively copies the memory pages from the source machine to the destination host, without ever stopping the execution of the VM being migrated. The iterative nature of the algorithm is due to the dirty pages, i.e. memory pages that have been modified in the source host since last page transfer must be sent again to the destination host. If the rate of updating of pages is very high, migration time will rise to a very high value. But the advantage of this approach is that all updating are available at the destination host. It can be activated any time. Every VM will have some

(hopefully small) set of pages that it updates very frequently and which are therefore poor candidates for pre-copy migration [22].

- Modification of pre-copy approach: It is a modification of pre-copy approach; a framework is constructed that includes pre-processing phase in traditional pre-copy based live migration so that the amount of transferred data is reduced [23].
- Hybrid pre and post copy: It is a technique that performs single round of pre-copying which precedes the virtual CPU state transfer. This is followed by post-copying of the remaining dirty pages from the source PM. Therefore, it gets the benefits of both approaches and improves live migration process [24].
- Memory compression: Virtual machine migration performance is greatly improved by cutting down the amount of transferred data. In the source node, data being transferred in each round are first compressed by the algorithm. When arriving on the target, compressed data are then decompressed. Compression time is a performance bottleneck of additional overhead introduce by compression operations. If the compression overhead outweighs the advantage of memory compression, live migration would not get any benefit from it [25], [16], [3].
- Delta page transfer: It is a technique that reduces the network bandwidth consumption by maintaining a cache of previously transferred memory pages. This optimizes the transmission of dirtied pages by sending the difference between cached page content and page content that is going to be transferred. This improves the live migration process with reduced risk of service interruption [26], [27].
- Dynamic self Ballooning (DSB): Ballooning is an existing technique that allows a guest OS to reduce its memory footprint by releasing its free memory pages back to the hypervisor. DSB automates the ballooning mechanism so it can trigger frequently (every 5 seconds) without degrading application performance. Our DSB implementation responds directly to OS memory allocation requests without the need for kernel patching. It neither requires external introspection by a co-located VM nor extra communication through the hypervisor. DSB thus significantly reduces total migration time by eliminating the transfer of free memory pages [28].
- Data deduplication: It is a technique that uses hash based finger print technology that finds the similar and identical data inside the memory and disk of a single VM and optimizes the live migration process. This technique requires transferring memory data from source to target PM only once [3], [29].
- Adaptive prepaging: Adaptive prepaging is another technique where we intelligently identify the pages which may be accessed by the VM in near future and transfer them beforehand. This helps in quick transfer of the VMs working set [28].
- Content bashed hashing: To be able to fetch memory pages from the memory of other VMs in the site, the destination node needs to discover which VMs in the network have a copy of a specific page. Since these other VMs are running and their memory content change over time, this information has to be discovered dynamically. To solve this problem hash table allowing to locate nodes having a copy of a given page using its hash value is used [29], [30], [31].
- Multiple VM Migrations: Whenever there is a need to shift entire virtual clusters to different locations, it requires large amount of data to be transferred over the network which results in network and CPU overhead. This leads to performance degradation of applications running in the PM. So identical contents of co-located VMs in the physical machine can be pro-actively tracked and transferred only once to the target PM. This improves the performance of multiple VM migration. This is known as “Gang Scheduling” of virtual machines. It optimizes both memory and network overhead of migration [28], [32].

Table 4. Comparison of State-of-art Existing Live Migration Approaches as Presented in [3]

Technique	Parameters considered				Performance metrics accounted			Single VM/ Multiple VM	WAN		
	VM size	Page dirty rate	N/W traffic	CPU cycles	Total migration down time	N/W BW	CPU util.		Y/N	N/W	Storage
Pre-copy [18]		✓	✓		✓			Single VM	No		
Pre-copy + memory compression [23]		✓			✓			Single VM	No		
Pre-copy + delta page transfer [24]		✓	✓		✓	✓		Multiple VM	Yes		✓
Post copy + adaptive pre-paging + dynamic self ballooning [28]		✓			✓			Single VM	No		
Pre-copy + Post-copy [22]		✓			✓			Single VM	No		
Pre-copy + multiple migrations [28]		✓	✓		✓	✓		Multiple VM	No		
Pre-copy + Content based hashing [29]		✓			✓			Multiple VM	yes		✓
Delta page transfer + data deduplication + VPN configuration [25]		✓	✓		✓	✓		Multiple VM	yes	✓	✓
Pre-copy + data deduplication [27]		✓			✓			Single VM	No		

The live migration approach consumes resources such as CPU, memory and network which can seriously impact the performance of both the VM being migrated as well as other VMs in the PM. The performance of the migration approach depends on the parameters such as VM size, memory, page dirty rate, network traffic and amount of available CPU resource.

It has been is concluded that:

- The performance of a migration approach depends on the parameters such as: VM size, memory page dirty rate, network traffic and amount of available CPU resource.
- Also, it is observed that the two important parameters VM size and available CPU cycles have not been taken into consideration during migration. For efficient migration, all the mentioned parameters should be taken into account and also the performance of the migration algorithm needs to be measured in terms of performance metrics.
- Current research efforts on live virtual machine migration take considerable amount of migration time and down time. A new model for live migration of VMs should be designed with reduced overheads of downtime and migration time

6 VM PLACEMENT ALGORITHMS

The virtual machine placement involves two main steps which are provisioning of resources for the virtual machines according to the capacity requirements of corresponding applications (VM sizing) and actual placement of VMs onto PMs. The placement approach should consider multiple resources such as CPU, memory, disk storages and network bandwidth to reduce the energy consumption at data centers and also maintain the energy performance tradeoff. The goal of virtual machine placement is to do VM consolidation either for power savings or load balancing to deliver best possible QoS to the applications running in VMs. Automating the process of VM placement has become important with the increase in size of the data centers [33]. Some of the factors that are typically used to quantify the cost functions are:

- Processor usage
- Storage usage
- Memory usage
- Network usage
- Power usage

The choice of which performance factors and how many performance factors are to be taken into consideration varies from cloud provider to cloud provider.

6.1 Classification of VM Placement Algorithms

The VM placement algorithms can be broadly classified into two categories on the basis of their placement goal [34]:

- **Power Based approach:** The main aim of these approaches is to map virtual machines to physical machines in such a way, so that the servers can be utilized to their maximum efficiency, and the other servers can be either hibernated or shut down depending on load conditions.
- **Application QoS based approach:** These algorithms manage the mapping of virtual machines onto physical hosts with the aim of maximizing the quality of service (QoS) delivered. By continuously monitoring virtual machine activity and employing advanced policies for dynamic workload placement, such algorithms can lead to better utilization of resources and less frequent overload situations eventually leading to savings in cost.

VM Placement algorithms can be further classified as shown in Fig. 2.

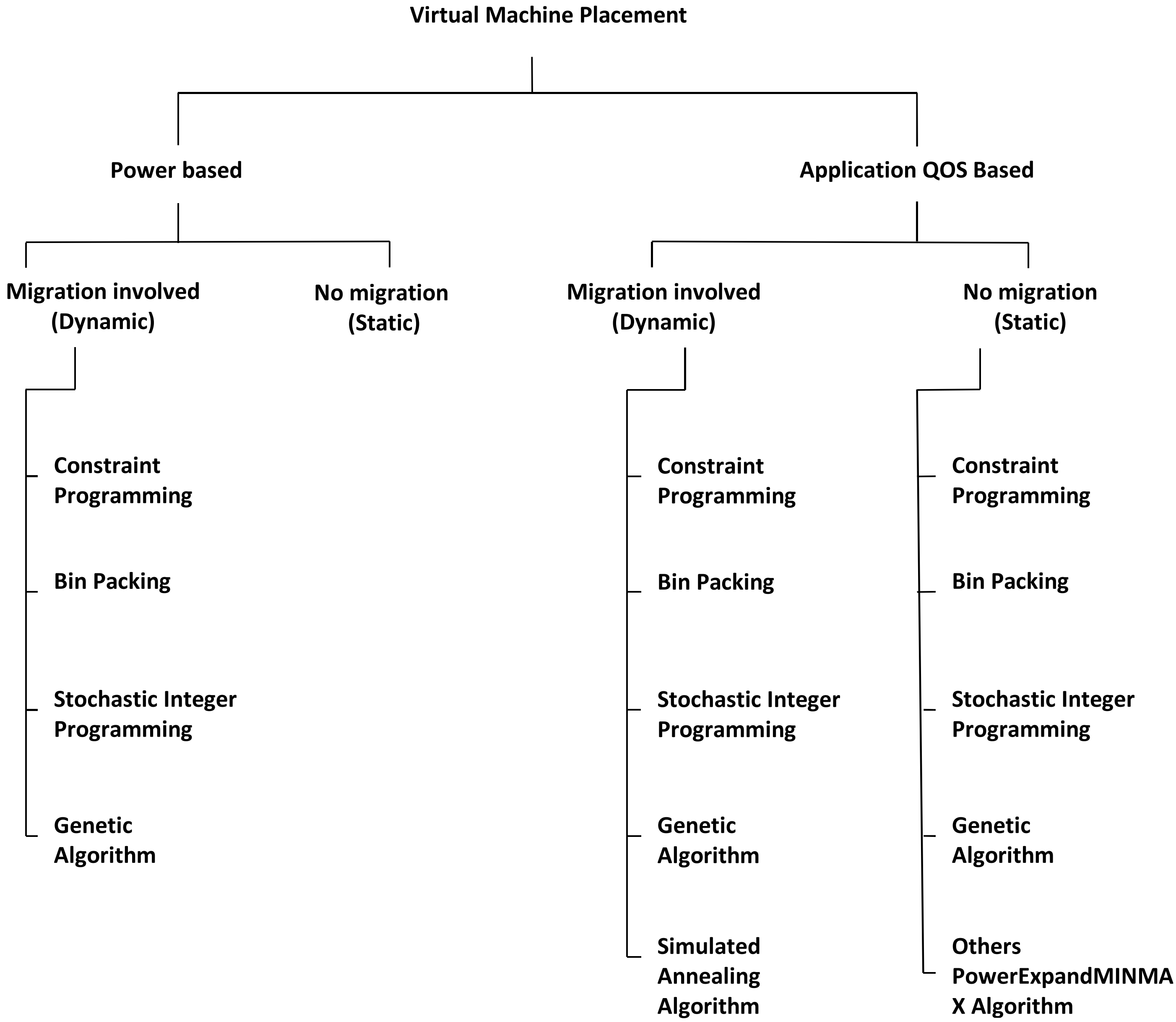

Fig. 2. Classification of VM Placement Algorithms

6.2 Discussion and Comparison of Different Techniques

This sub-section provides a discussion and comparison among different VM placement algorithms [34]:

- **Constraint programming:** It is useful in cases where we have the input data with us, that is, before we compute the cost functions, we know the demands of the Virtual Machines. The techniques using constraint programming are easily extendable so as to take additional constraints into account. The time taken to generate an optimal solution is high as the number of constraints increases [35].
- **A Bin packing approach:** It can be considered as a subset of Constraint Programming approach. It is useful for dynamic VM Placement, especially where the demand is highly variable. It is a heuristic based approach. Thus it may not give us an optimal solution. However, it will always generate a good solution in considerable amount of time. A Bin-packing approach is really useful when all physical machines have the same amount of memory and processing capabilities. However, we can even model Bin packing algorithms with constraints. There is always a tradeoff between packing maximum number of virtual machines on a single physical machine versus distributing the load across all the physical machines. Multi-dimensional bin-packing algorithm can also be applied to VM placement problem. In this the dimensions correspond to amount of memory and number of processing units [34], [4].
- **Stochastic integer programming:** It is useful where the future demands and prices of resources are not known, but their probability distributions are either known or can be computed. This is the best technique to be used in the case where we have two or more uncertain parameters on which the cost depends [36].
- **Genetic algorithm:** It is a way to solve the bin packing problem with certain constraints. The Grouping Genetic Algorithm requires more computing time and higher computing resources as compared to bin packing. It is particularly useful for static placements, that is, in scenarios where the demands do not vary over a considerable period of time. GA is also useful for specifying VM-VM and VM-PM interference constraints [37].

Each of the virtual machine placement algorithm works well under certain specific conditions. Thus, it is important to choose a technique that suits the needs of the cloud user and cloud provider. Also, the parameters to these algorithms should be properly specified. The performance metrics are measured at both system level and application level. The system level metrics are measured in terms of CPU load and the application level metrics are measured in terms of response time of applications.

Physical and virtual machines are characterized by their CPU (MIPS), RAM (MB) and bandwidth (Mbits/s). The goal of virtual machine placement problem is to determine the minimum number of physical machines required by the set of virtual machines.

In [36] vm sizes are generated as random variables following some distribution and actual Violation Probability is the failure probability of ensuring the target overflow probability for all the hosts.

It is concluding from table 5 that:

- Constraint programming algorithm can compute approximately global optimization solution within timeout. It can efficiently reduce the number of physical servers but it suffers from relatively long search times.
- Bin packing approach is a heuristic based approach .Thus it may not give us an optimal solution. However,it always generate a good solution in considerable amount of time.
- Stochastic integer programming is the best technique to be use in the case where we have two or more uncertain parameters on which the cost depends
- The computation time of the GA increased with the number of VMs and the number of PMs.

Table 5. Comparison of State-of-Art Existing Bin Packing VM Placement Approaches

Technique	Type	Input parameters							Output		
		Host CPU (MIPS)	Host RAM (MB)	Host BW (Mbits/s)	VM CPU (MIPS)	VM RAM (MB)	VM BW (Mbits/s)	VM size	Actual violation probability	Allocations of VMs	Average running time
CP-Based VM Placement Algorithm [35]	Constraint Programming	✓	✓	✓	✓	✓	✓			✓	✓
Modification of Best Fit Decreasing[4]	Bin Packing	✓	✓		✓	✓				✓	
Stochastic VM Multiplexing [36]	Stochastic Integer Programming	✓	✓					✓	✓	✓	
GA energy efficient VM placement Algorithm [37]	Genetic Algorithm	✓	✓		✓	✓				✓	✓

7 CONCLUSIONS

This paper presented a survey of different virtual machine consolidation challenges in addition to a comparison among corresponding algorithms. From the survey, it can be concluded that:

- Dynamic VM consolidation algorithms based on local regression outperform the threshold-based and adaptive-threshold based algorithms due to better predictions of host overloading, and therefore decreased SLA violations due to host overloading (SLATAH) and the number of VM migrations.
- A Fuzzy Q-Learning (FQL) approach as an online decision making strategy enhance the VM selection task in a dynamic VM consolidation procedure.
- New models for live migration of VMs should be designed with reduced overheads of downtime and migration time.
- Automating the process of virtual machine placement has become important with the increase in size of the data centers. The choice of which performance factors and how many performance factors are to be taken into consideration varies from cloud provider to cloud provider.

REFERENCES

[1] S. Thakur, A. Kalia and J. Thakur, "Server Consolidation Algorithms for Cloud Computing Environment: A Review," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, no. 9, pp. 379–384, 2013.

[2] H. Elazhary, "Cloud Computing for Big Data," MAGNT Research Report, vol. 2, no. 4, pp. 135-144, 2014.

[3] T. Veni and S. Bhanu, "A Survey on Dynamic Energy Management at Virtualization Level in Cloud Data Centers," Computer Science & Information Technology , pp. 107-117, 2013.

[4] A. Beloglazov and R. Buyya, "OpenStack Neat: A Framework for Dynamic and Energy-Efficient Consolidation of Virtual Machines in OpenStack Clouds," Concurrency and Computation: Practice and Experience (CCPE), pp. 32–36, 2014.

[5] A. Beloglazov and R. Buyya, "Managing Overloaded Hosts for Dynamic Consolidation of Virtual Machines in Cloud Data Centers under Quality of Service Constraints," IEEE Transactions on Parallel and Distributed Systems (TPDS), vol. 24, no. 7, pp. 1366–1379, 2013.

[6] A. Beloglazov, "Energy-Efficient Management of Virtual Machines in Data Centers for Cloud Computing," Ph.D. thesis, The University of Melbourne, 2013.

[7] 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 (CCPE), vol. 24, no. 13, pp. 1397–1420, 2012.

[8] W. Cleveland, C. Loader, "Smoothing by Local Regression: Principles and Methods," Statistical Theory and Computational Aspects of Smoothing, 1996.

[9] W. Cleveland, "Robust Locally Weighted Regression and Smoothing Scatterplots," Journal of the American Statistical Association, 1979.

[10] G. Khanna, K. Beaty, G. Kar, and A. Kochut, "Application Performance Management in Virtualized Server Environments," Proc. Network Operations and Management Symposium, IEEE, pp. 373–381, 2006.

[11] A. Singh, S. Kinger, "Virtual Machine Migration Policies in Clouds," IJSR, vol. 2, no. 5, pp. 364–367, 2013.

[12] A. Beloglazov and R. Buyya, "Adaptive Threshold-Based Approach for Energy-Efficient Consolidation of Virtual Machines in Cloud Data Centers," Proceedings of the 8th International Workshop on Middleware for Grid, Clouds and E-science, Bangalore, India, 2010.

[13] A.verma, G. Dasgupta, T. Nayak, and R.Kothari, "Server Workload Analysis for Power Minimization Using Consolidation," Proc. the 2009 USENIX Annual Technical Conferene, San Diego, USA, 2009.

[14] S. Masoumzadeh and H. Hlavacs, "Integrating VM Selection Criteria in Distributed Dynamic VM Consolidation Using Fuzzy Q-Learning," Proc. the 9th International Conference on Network and Service Management (CNSM 2013), pp. 332–338, Oct. 2013.

[15] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing," Future Generation Computer Systems, vol. 28, no. 5, pp. 755–768, 2012.

[16] A. Mohan and S. Shine, "Survey on Live VM Migration Techniques," International Journal of Advanced Research in Computer Engineering and Technology, vol. 2, no. 1, pp. 155–157, 2013.

[17] P. Leelipushpam and J. Sharmila, "Live VM migration techniques in cloud environment—A survey," Information & Communication Technologies, pp. 408–413, 2013.

[18] V. Medina and J. García, "A survey of migration mechanisms of virtual machines," ACM Computing Surveys (CSUR), vol. 46, no. 3, 2014.

[19] C. Sapuntzakis, R. Chandra, B. Pfaff, J. Chow, M. Lam, and M. Rosenblum, "Optimizing the Migration of Virtual Computers," Proc. of the 5th Symposium on Operating Systems Design and Implementation (OSDI-02), December, 2002.

[20] C. Clark, K. Fraser, S. Hand, J. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live Migration of Virtual Machines," Proc. the 2nd conference on Symposium on Networked Systems Design & Implementation, pp. 273-286, 2005.

[21] S. Sharma and M. Chawla, "A Technical Review for Efficient Virtual Machine Migration," 2013 International Conference on Cloud & Ubiquitous Computing & Emerging *Technologies*, pp. 20–25, Nov. 2013.

[22] B. Hu, Z. Lei, Y. Lei, and D. Xu, "A Time-Series Based Precopy Approach for Live Migration of Virtual Machines," Proc. the IEEE 17th International Conference on Parallel and Distributed Systems (ICPADS), pp. 947-952, Dec. 2011.

[23] E. Zaw and N. Thein, "Improved Live VM Migration using LRU and Splay Tree Algorithm," International Journal of Computer Science and Telecommunications, vol. 3, no. 3, March 2012.

[24] S. Sahni and V. Varma, "A Hybrid Approach to Live Migration of Virtual Machines," International Conference on Cloud Computing in Emerging Markets, India, 2012.

[25] H. Jin, L. Deng, S. Wu, X. Shi, and X. Pan, "Live Virtual Machine Migration with Adaptive Memory Compression," IEEE Cluster, New Orleans, 2009.

[26] S. Hacking and B. Hudzia, "Improving the Live Migration Process of Large Enterprise Applications," Proc. 3rd International Workshop on Virtualization Technologies in Distributed Computing, pp. 51-58, USA, 2009.
[27] T. Wood, P. Shenoy, K. Ramakrishnan, and J. Merve, "Cloudnet: Dynamic Pooling of Cloud Resources by Live WAN Migrations of Virtual Machines," Proc. 7th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, pp. 121-132, USA, 2011.
[28] M. Hines and K. Gopalan, "Post-Copy Based Live Virtual Machine Migration Using Adaptive Pre-Paging and Dynamic Self Ballooning," Proc. International Conference on Virtual Execution Environments, pp. 51-60, USA, 2009.
[29] X. Zhang, Z. Huo, J.Ma., and D. Meng, "Exploiting Data Duplication to Accelerate Live Virtual Machine Migration," Proc. the IEEE International Conference on Cluster Computing, pp. 88-96, 2010.
[30] U. Deshpande, X. Wang, and K. Gopalan, "Live Gang Migration of Virtual Machines," Proc. International Conference on High Performance Distributed Computing, pp. 135-146, USA , 2011.
[31] P. Riteau, P. Morin, and T. Priol, "Shrinker: Efficient Wide Area Live Virtual Machine Migration Using Distributed Content Based Addressing," Proc. 17th International Conference on Parallel Processing, pp. 431-442, Heidelberg, 2011.
[32] R. Bradford, E. Kotsovinos, A. Feldmann, and H. Schioberg, "Live Wide Area Migration of Virtual Machine Including Local Persistent State," Proc. 3rd International Conference on Virtual Execution Environments, pp. 169-179, USA, 2007.
[33] D. Jiang, P. Huang, P. Lin, and J. Jiang, "Energy Efficient VM Placement Heuristic Algorithms Comparison for Cloud with Multidimensional Resources," Information Computing and Applications, pp. 413–420, 2012.
[34] P. Sayeedkhan, "Virtual Machine Placement Based on Disk I/O Load in Cloud," vol. 5, no. 4, pp. 5477–5479, 2014.
[35] Y. Yu and Y. Gao, “Constraint Programming-Based Virtual Machines Placement Algorithm in Datacenter,” *Intelligent Information Processing*, pp. 295–304, 2012.
[36] B. B. Nandi, A. Banerjee, S. C. Ghosh, and N. Banerjee, “Stochastic VM Multiplexing for Datacenter Consolidation,” *2012 IEEE Ninth International Conference on Services Computing*, pp. 114–121, Jun. 2012.
[37] G. Wu, M. Tang, Y. Tian, and W. Li, “Energy-efficient virtual machine placement in data centers by genetic algorithm,” Neural Information Processing, pp. 315–323, 2012.