Distributed and Parallel Databases (2022) 40:441–474
https://doi.org/10.1007/s10619-021-07387-2

Virtual machines pre-copy live migration cost modeling and prediction: a survey

Mohamed Esam Elsaid[1] · Hazem M. Abbas[2] · Christoph Meinel[1]

Accepted: 24 November 2021 / Published online: 6 December 2021

Abstract

Live migration is an essential feature in virtual infrastructure and cloud computing datacenters. Using live migration, virtual machines can be online migrated from a physical machine to another with negligible service interruption. Load balance, power saving, dynamic resource allocation, and high availability algorithms in virtual data-centers and cloud computing environments are dependent on live migration. Live migration process has six phases that result in live migration cost. Several papers analyze and model live migration costs for different hypervisors, different kinds of workloads and different models of analysis. In addition, there are also many other papers that provide prediction techniques for live migration costs. It is a challenge for the reader to organize, classify, and compare live migration overhead research papers due to the broad focus of the papers in this domain. In this survey paper, we classify, analyze, and compare different papers that cover pre-copy live migration cost analysis and prediction from different angels to show the contributions and the drawbacks of each study. Papers classification helps the readers to get different studies details about a specific live migration cost parameter. The classification of the paper considers the papers' research focus, methodology, the hypervisors, and the cost parameters. Papers analysis helps the readers to know which model can be used for which hypervisor and to know the techniques used for live migration cost analysis and prediction. Papers comparison shows the contributions, drawbacks, and the modeling differences by each paper in a table format that simplifies the comparison. Virtualized Data-center and cloud computing clusters admins can also make use of this paper to know which live migration cost prediction model can fit for their environments.

Keywords Virtual datacenter · Cloud computing · Live migration · Overhead · Cost · Modelling · Prediction

✉ Mohamed Esam Elsaid
Mohamed.Elsaid@HPI.de

[1] Hasso-Plattner Institut, Prof.-Dr.-Helmert-Straße 2-3, 14482 Potsdam, Germany

[2] Faculty of Engineering, Ain Shams University, Cairo, Egypt

1 Introduction

Virtualization concept relies on resources sharing such that the datacenters resources of compute, network and storage can be shared between different operating systems and applications with a physical layer based isolation [1]. Virtual Machines (VMs) is the virtualization of the compute layer in the datacenter resources; such that the physical server CPU, cache, memory and all other compute hardware can be shared across several VMs. Each VM has its dedicated Operating System (OS), platforms and applications. Virtual machines isolation, configuration and resource allocation is basically done by the system hypervisor which manages the relationship between the VMs and the physical machines. The benefits of datacenter virtualization are basically higher physical resources utilization, less power consumption, lower infrastructure Total Cost of Ownership (TCO), higher system availability, dynamic resource management, less vendor lock-in and adoption to cloud computing. These vital benefits to the IT industry lead to having more than eighty percent of the global datacenters utilizing virtualization technologies.

Live Migration is a key technology and essential feature in datacenter virtualization. With live migration, the VMs can be moved from a physical host to another with almost no impact on the running applications availability. This makes the running applications do not get impacted by the entire physical server issues; which enhances the service availability dramatically. Live migration traffic is sent over the TCP/IP protocol that utilizes the Ethernet network which interconnects the cluster servers. The content that should be migrated is basically the CPU cache, memory and buffers content; however the big bulk to be migrated is the memory content. The CPU cache and buffers content is almost negligible compared to the memory content and that what most of the papers assume in live migration modelling.

Live migration is supported by almost all hypervisors in the market; VMware ESXi, Microsoft Hyper-V, Xen and KVM. Clusters load balance, power saving, resource allocation flexibility and fault tolerance are all dependent on live migration.

- For load balance, live migration is used to update the allocation mapping between the VMs and the physical machines from time to time. This update is based on the physical machines utilization to keep all the cluster physical servers utilization balanced by avoiding system bottlenecks.
- In power saving, live migration is used to concatenate the VMs within less number of physical machines during the low utilization hours and so to minimize the number of active physical servers and switch the other idle servers into sleep mode.
- Fault tolerance also relies on live migration between two physical servers at least with keeping two copies of the VM; one at the source host and another copy at the target host. So in case of failure in the primary VM at the primary host, the secondary VM on the secondary host will takeover and act as a new primary VM.

VMs live migration is an essential feature in cloud computing environments, however live migration cost can not be ignored. The challenges that the readers face in live migration cost topic is finding tens of papers that discuss different cost parameters,different hypervisors and different analysis methodologies. So in this survey paper, we summarize the related work in pre-copy live migration cost topic. The proposed summary is based on related work cost parameters classification that shows the modeling comparison and research methodologies differences between the related papers.

Live migration of VMs has three different types; pre-copy, post-copy an hybrid-copy. In this paper, we focus on Pre-copy live migration as the most robust migration type. In pre-copy migration, the source host memory content copy starts and keeps transferring until meeting a stopping condition. Then the VM stops at the source host and be initiated at the target host. This is the most commonly used migration type by commercial and open source hypervisors as the most reliable migration type.

Compared to other survey papers, to the best of our knowledge, this paper is the first paper that provides the below contributions:

- We summarize in this paper tens of research articles that discussed the live migration cost modeling and prediction topic for different hypervisors.
- Cost parameter based classification that shows the modeling comparison for the same cost parameter proposed by different papers.
- The provided classification is presented in a table format that highlights the missing points that are still open for research.
- To introduce live migration with persistent memory technology as a new trend in servers memory technology.

The rest of this paper is organized as follows, in Sect. 2, we introduce the different configurations and types of live migration in general to explain the definition of pre-copy migration and its difference with other migration types. Because networking is an essential topic in VMs migration performance, we discuss networking set up of virtualized cluster in Sect. 3. The networking set up section discussed the networking details for VMware vSphere, Microsoft Hyper-V, KVM and Xen hypervisors clusters. Migration cost parameters are presented in Sect. 4 and the cost parameters modeling is discussed in Sect. 5. Then the prediction techniques of migration cost is presented in Sect. 6. The proposed classification of live migration cost modeling and prediction papers is proposed in Sect. 7. In Sect. 8, we discuss PMem and live migration support for servers with PMem DIMMs as a new trend in servers memory technology that is recently supported by many servers vendors. Finally, we list the open research areas in Sect. 9 that can extend the work in live migration cost modeling and prediction and conclude the paper in Sect. 10.

2 Live migration configurations and types

Live migration of VMs is supported in different configuration scenarios depending on which resource should be migrated. The migration process sequence has also different mechanisms that define the migration types. In this section, we discuss in details Live migration configurations and types.

2.1 Live migration configurations

Live migration can be achieved with different configuration scenarios; as shown in Fig. 1.

- The first scenario is to migrate the VM compute resources to another physical host without the VM storage virtual disk migration. This can be applied only under the condition of having a shared storage environment between the source and the target servers. In this case, mainly the memory content is migrated. For example the VM in Fig. 1 can be migrated with this scenario only between S1 and S2 hosts through the management IP network of the cluster.
- The second scenario is to migrate the compute and storage resources of the VM from a source to a target host through the management IP network of the cluster. In this case the memory and the virtual disk storage content should be migrated. So the VM in Fig. 1 can be migrated from S1 or S2 to S3 host or vice versa.
- The third scenario is to migrate the compute and storage resources of the VM from the source to the target host through the WAN or the Internet network. This scenario is mainly useful for datacenter migrations or disaster recovery solutions between datacenters in different locations. So to migrate the VM in Fig. 1 from S1, S2 or S3 to S4
- The fourth scenario of live migration is to have multiple VMs migration simultaneously. The number of simultaneous VMs to be migrated has a maximum limit. This limit is defined by the source host of the migration that is responsible

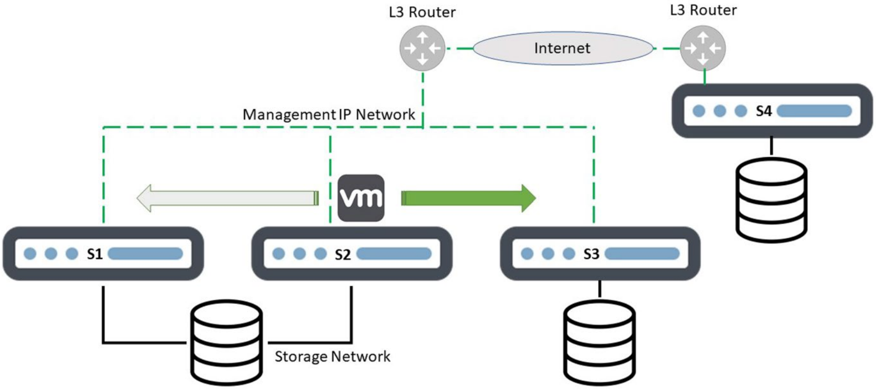

Fig. 1 Live migration configurations

for resources allocation and migration success verification process. Referring to Fig. 1, in multiple simultaneous VMs migration, there can be many VMs in different hosts that can be migrated from any of the hosts to another.

2.2 Live migration types

From migration processes point of view, live migration has three different types; as shown in Fig. 2. The three types are Pre-copy, Post-copy and Hybrid-copy.

- In Pre-copy, live migration starts with transferring the whole content of the source host memory to the target host, however due to the fact that the application is still writing data on the source host memory, this new data is called dirty pages that should be transferred also to the target host in other iterations. This iterative copy runs until a stopping condition is met. There are different stopping conditions, as we will discuss. After the stopping condition is met, the last copy of the memory and the CPU state is transferred to the target host and the this is the time when the VM is handed-over to the target host. During this handover, there is a down-time that should be very short to avoid the running application interruption. This means that in Pre-copy, the handover of the VM only occurs when there is little amount of data to be transferred to minimize the down-time and to have robust migration. That is the the reason for considering Pre-copy live migration as the most reliable live migration type. VMware, KVM, Hyper-V and Xen are all using Pre-copy live migration. The dis-advantage of Pre-copy live migration is the migration time which is not predictable because the number of copy iterations can not be predicted. It depends basically on the dirty pages rate and the network transmission rate. In some cases the migration might take too long time or even fails due to high dirty pages rate with lower network transmission rate. But when this case happens, the VM continues running on the source host without disruption, which make Pre-copy as the most reliable technique.

  The stopping condition in Pre-copy differs from a hypervisor to another. The number of pre-copy iterations, the residual amount of data to be migrated in the source host memory, or ratio between the transferred data and the memory content to be migrated are the main stopping conditions for pre-copy. The stopping conditions in the Xen platform are [2]:

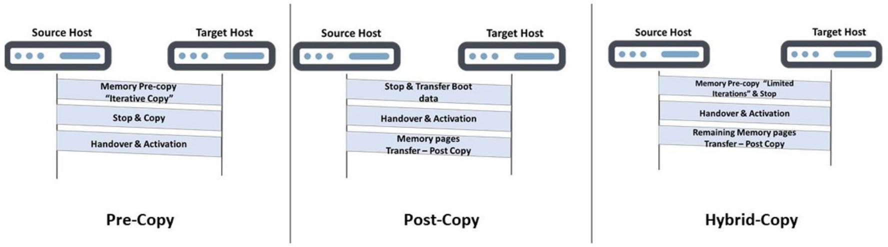
Source Host
Target Host
Memory Pre-copy "Iterative Copy"
Stop & Copy
Handover & Activation
Pre-Copy
Source Host
Target Host
Stop & Transfer Boot data
Handover & Activation
Memory pages Transfer – Post Copy
Post-Copy
Source Host
Target Host
Memory Pre-copy "Limited Iterations" & Stop
Handover & Activation
Remaining Memory pages Transfer – Post Copy
Hybrid-Copy

Fig. 2 Live migration types

(a) Less than 50 pages are dirtied during the last pre-copy iteration.
(b) To have 29 pre-copy iterations have been carried out.
(c) More than three times the total amount of RAM allocated to the VM has been copied to the destination.

While the stopping conditions for VMware are [3]:

(a) Less than 16 megabytes of modified pages are left.
(b) There is a reduction in changed pages of less than one megabyte.

- In Post-copy migration, the source host transfers only the data required for the VM boot to the target host and then stops the VM at the source host to hand it over. After the VM activation at the target host, the source host starts sending the memory data in one iteration to the target host. This means that the memory copy is done in a one shot after the VM handover, and so post-copy migration time is predictable. However, this means that if the memory content transfer is fails for any reason, the VM will be destroyed and data loss might occur [4]. So, it is not a reliable migration technique as Pre-copy. And so, post-copy is not used by any commercial hypervisor. In [5] a detailed comparison between pre-copy and post-cop migration is presented. The comparison in [5] shows that pre-copy technique is a more safe and reliable and so it is the commonly used technique by the hypervisors Xen, VMware, Microsoft Hyper-V, Oracle VM server, KVM, Virtuozzo, OVirt and Google Compute Engine. Post-copy migration is used also in KVM, Virtuozzo, OVirt and Google Compute Engine.
- Hybrid-copy technique has several algorithms that try to mix steps of pre-copy and post-copy to get higher robust migration with migration time prediction. One of these algorithms firstly migrates [6] and [7] the memory content of the VM is transferred to the target host and during this migration, new dirty pages are written to the source host memory, so several pre-copy iterations are run but with limited number to keep the migration time predictable. Then the VM state is transferred and handover occurs to activate the VM at the target host. The residual memory pages are transferred to the target host in a post-copy manner. Hybrid-copy depends on having low amount of residual memory pages in the post-copy phase to enhance the migration robustness compared to post-copy, however it does not show the same reliability and robustness level of pre-copy. So in case of transfer failure in the post-copy phase, data loss might occur.

3 Live migration networking

Virtual networking is an essential requirement for virtualized datacenters and cloud computing platforms [8]. Each VM has a virtual network adapter and at least one virtual port. The VMs are inter-connected to virtual switches (vSwitches) that use physical Ethernet switches in the back-end. In this section we discuss in more details the concept of network virtualization and how live migration is implemented in the four hypervisors; VMware vSphere, Microsoft Hyper-V, Xen and KVM.

In virtual networking, each VM has virtual Network Interface Cards (vNICs). Each vNIC has one or more virtual ports (vPorts). Each vPort is connected to a vSwitch. This virtual switch can be a local switch inside the physical host only to connect the VMs within this host, or can be a cluster virtual switch to connect between the VMs in a cluster. Each vSwitch has at least one uplink port which is mapped to a physical switch port. Each group of ports in the vSwitch can create a separate vLAN or port group that can be labeled. For one or more physical hosts connection, a cluster vSwitch is used as a centralized vSwitch that connects all the VMs of the cluster physical nodes. This vSwitch concept applies to all hypervisors [9]. However, the hypervisors are different to each other when it comes to live migration networking set up. We discuss in this section live migration networking configuration details for VMware vSphere, Microsoft Hyper-V, Xen and KVM.

3.1 VMware vSphere live migration networking

ive migration feature in VMware is called vMotion. Figure 2 shows an example of the best practice for vMotion networking using a cluster of two physical machines. These two servers are connected to a shared storage using FC-SAN switch and connected to the IP network using an Ethernet switch. The solid lines represent physical connections and the dotted lines represent the virtual connections for the virtual distributed switch. Figure 2 represents a commonly used scenario in enterprise datacenters where a storage array is shared between the cluster servers using FC-SAN network. Live migration uses TCP/IP protocol and so it utilizes the IP network. From best practice point of view, each physical host should have at least 2 physical NICs and each VM should have at least 2 NICs [10]. The VMs in the cluster are connected to a virtual distributed switch. Using port groups, the IO traffic of the VMs can be isolated. There are two types of port groups in VMware; VMkernel distributed port group and VM network distributed port group. VM network port group is responsible for the production traffic like applications traffic. VMkernel port group is responsible for the special classes of traffic like vMotion, management, iSCSI, NFS, Fault tolerance, replication traffic and VMware vSAN as a Software Defined Storage (SDS) traffic [11]. The physical machines NICs ports should be mapped to the distributed switch as uplink ports. The uplink port is responsible for the in-going and the out-going traffic into and from the distributed switch. Each port group should have at least one uplink port from each physical host. Each uplink port can be shared between many port groups. For vMotion traffic, it is a best practice to create a dedicated VMkernel port group between the VMs in the cluster. This vMotion distributed port group should include at least one uplink port from each physical host [10]. This uplink port assignment is actually not only for vMotion port group, but also for any other VMkernel based port group. From physical port isolation, vMotion traffic is physically isolated on the host port level from the applications traffic. However, depending on the back-end network topology, vMotion and workload traffic might compete on the back-end network bandwidth.

3.2 Microsoft hyper-V live migration networking

Virtual layer two switches in Hyperv-V have the same concept like VMware. It is basically a software based switch that connects the VMs vNICs with the physical ports uplinks [12]. Also, live migration in Microsoft Hyper-V has the same concept like VMware vMotion. The best practice for Hyper-V is to configure a separate virtual network or VLAN for live migration in order to isolate the migration network traffic from the applications traffic [13].

3.3 Xen hypervisor live migration networking

In Citrix Xen, virtual Switch concept is also used as in vSphere and Hyper-V, such that each VM has at least one vNIC and vport that are connected to a distributed vSwitch. The distributed vSwitch connects the VMs across the cluster and includes the hosts physical NICs as the vSwitch uplinks. The difference in Xen compared to vSphere and Hyper-V is having a separate OpenFlow controller. This OpenFlow controller is a centralized server that controls the Xen servers virtual network and is responsible for the vSwitches configuration, traffic management and performance monitoring [14]. Live migration feature in Xen is called XenMotion. XenMotion networking best practice is to create a cross server private network that isolates XenMotion traffic from other other management or workload traffic. This private network provisions dedicated virtual management interface of the VMs for live migration traffic [15].

3.4 KVM live migration networking

Libvirt is used for KVM Hypervisor virtual networking [16]. Libvirt uses APIs that talks to Quick EMUlator (QEMU) for network and storage configurations. Each VM has its own QEMU instance. The vSwitch that is created by libvirt can connect the VMs vNICs across the KVM cluster with the physical hosts uplink ports. For KVM live migration networking, Redhat best practice is to create separate the storage network from the migration network. So live migration isolation from other management traffic or workload traffic is not mentioned [17]. This means that live migration network traffic might be in contention with the workload traffic or with other management traffic.

4 Live migration cost parameters

Live migration is a key technology for data-centers and cloud computing environments, however there is a cost for the live migration process. The cost includes down time, migration time, network overhead, power consumption overhead, memory overhead and CPU utilization overhead.

Cost modeling and prediction of live migration is a very important topic. Based on the definition of live migration cost modeling and prediction techniques, IT admins can:

- Correlate the relationship between the VM size and the migration cost.
- Get an estimation about VM migration cost before proceeding with the migration event [18].
- Receive a recommendation with the optimal timing for a specific VM live migration; depending on the VM size and the DC network utilization [19].

There are many research articles that analyze and model live migration cost [20–22] and [23]. In addition, there are also many other papers that predict the migration cost for pre-copy migration in order to resolve the disadvantage of not having a predictable migration cost [24, 25]. However, it is a challenge for readers of these papers to track the differences and map the suitable use-case for each model of these papers.

In this paper, we review, compare, classify and summarize different up to date research articles in the area of live migration overhead modelling and prediction. Also, we cover in this paper live migration overhead for servers with persistent memory inside; as an emerging memory technology recently used in modern datacenters. Finally, we discuss in this paper the outstanding research topics in the area of live migration cost analysis and prediction.

This paper contribution can be summarized in the following points:

(1) Analyze the other related survey papers that cover pre-copy live migration topic and show their differentiation compared to this paper.
(2) Categorize and compare related live migration cost papers based on the papers focus (analysis or prediction), the cost parameters that are discussed and the hypervisors used.
(3) Discuss live migration in modern datacenter servers with persistent memory technology.
(4) Share the outstanding research problems in the area of live migration cost.

The rest of this paper is organized as following; the next section provides more details about live migration overhead background. Sections 3 and 4 classify and compare research articles that cover live migration cost analysis, modeling and prediction. In Sect. 5, we discuss live migration cost for modern datacenters with persistent memory servers. We share the open research areas in the area of live migration cost in Sect. 6 and then conclude in Sect. 7.

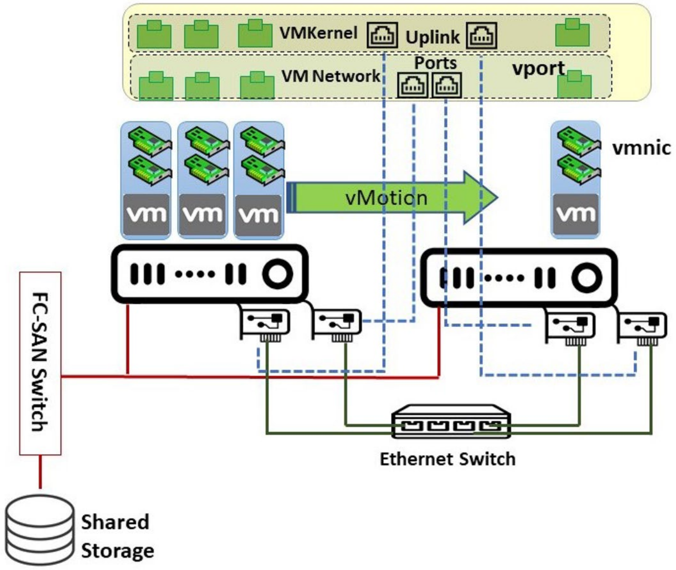

Fig. 3 Network topology for VMware vMotion

In this paper, we focus on the pre-copy live migration cost as the mainly used technique by all virtualization hypervisors due to higher robustness; as mentioned in the previous section. In this section, we discuss in more details pre-copy live migration phases, the migration cost parameters considering their root causes, and the different methods for migration cost analysis and prediction.

Pre-copy live migration has mainly six phases; as shown in Fig. 3 [26]. These phases are:

(a) *Initialization* Initiating the migration by selecting the VM to be migrated and selecting the target machine.
(b) *Reservation* The source machine sends a request to the target machine for resources reservation and the target machine answers with an acknowledgment after reserving the required resources for the migration.
(c) *Iterative pre-copy* The entire RAM is sent in the first iteration, then pages modified during the previous iteration are transferred to the destination. Using shadow page table for memory dirty pages mapping.
(d) *Stop-and-copy* When the stop conditions are met, the VM is halted on the source for a final transfer round. At the same round of stop- and-copy while transferring the final dirty pages, the migrated VM’s CPU state is transferred to the destination.
(e) *Commitment* the destination host checks if it has received successfully a consistent copy of the migrated VM. Then the target machine sends a message telling the source that it has successfully synchronized the migrated VM states.

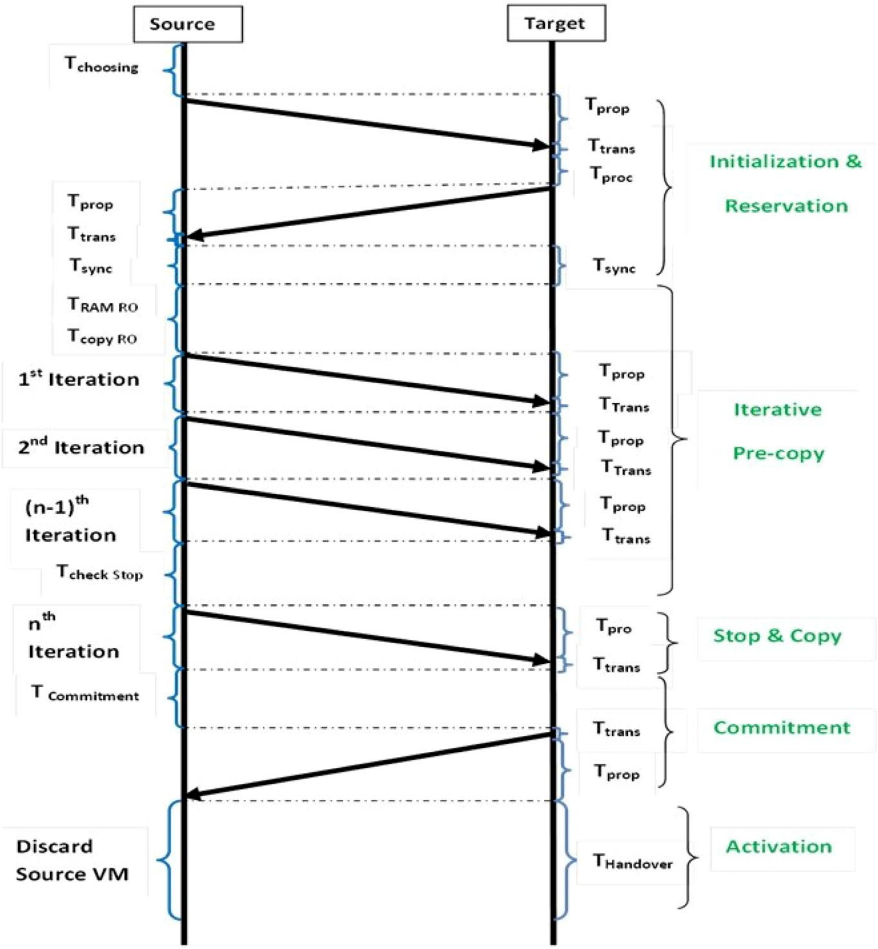

Fig. 4 Pre-copy live migration phases

(f) *Activation* After target host informs source host that it has synchronized their states, source VM can be discarded. The migrated VM running on target host is the primary host now and takes over the services offered by source VM (Fig. 4).

5 Live migration cost modeling

Pre-copy live migration cost is basically a result of the six phases of live migration that are mentioned in the previous section. In this section, we discuss in details the definition of each cost parameter, the root cause of it and its verified modeling for different hypervisors. The cost parameters are:

(1) *Migration time* Migration time is the period between the VM migration request initialization and having the VM activated at the destination server. This time can take from seconds to minutes depending on the VM memory content size, the network transmission rate during the migration and the dirty pages rate.
(2) *Down time* This is the time consumed in the stop and copy phase, when the VM stopping condition applies and the last iteration of the migration copy should start and then the VM networking being attached to the target server and until being activated. Down time should typically be in the range of milli-seconds and so the applications and the users do not feel interruption, however in some cases it takes several seconds [27].
(3) *Network throughput overhead* Network average rate is the average throughput at which data was transmitted from the physical host NIC card during the migration time interval. This represents the consumed bandwidth of the network in Bps for live migration process. Live migration process is managed by the cluster manager server which uses the Transmission Control Protocol/Internet Protocol (TCP/IP) in the networks layers 3 and 4 for the live migration management and the iterative copies of memory pages.
(4) *Power consumption overhead* Live migration process consumes CPU cycles from the source and the target servers [28]. This overhead parameter should not be ignored especially when live migration is used for data-centers power saving algorithms. Live migration transmission rate is the dominant parameter that controls the power consumption during the migration process [29].
(5) *CPU overhead* VMs live migration consumes also from the source and target servers CPU resources due to handling the iterative copy phase; as a CPU intensive phase of the migration [30]. Meanwhile, the more available CPU resources, the less migration time in case of having available network bandwidth.

Optimization of live migration is proposed by many research articles with an objective to minimize one or many of the migration cost parameters. This survey paper [31] summarizes and classifies the different approaches in live migration optimization algorithms. As presented in [31], the optimization techniques can be based on compression, de-duplication, check-pointing and other optimization techniques. There is a major difference between the survey proposed in [31] and this survey paper. This paper focuses on classification and comparison between the papers discuss live migration cost modelling and prediction techniques for the built-in live migration algorithms in different hypervisors. However in [31], the focus is to classify and summarize the papers that discuss the live migration optimization algorithms to minimize the migration cost.

Live migration cost is covered by different researchers, we list many of them in Table 1 and classify the articles based on research focus if it is cost prediction or just analysis, the validated hypervisors and the cost parameters that are considered.

We proposed empirical modeling techniques in [20] for VMs live migration in VMware environments to characterize live migration time, network rate and power consumption overhead. The proposed modeling is based on applying the regression techniques on the obtained test results to present a linear or non-linear

Table 1 Summary of related work

Paper	Research scope	Mig. time	Down time	CPU	Network	Power	Testing env.
[20]	Regression modeling	X	–	–	X	X	VMware
[21]	Perf. comparison	X	X	–	–	–	All
[22]	Analysis and comparison	X	X	–	–	–	Xen VMware
[23]	Perf. evaluation	X	X	–	–	–	Xen
[32]	Analysis on apps perf.	X	X	–	–	–	Xen
[33]	Multi-VMs scheduling	X	–	–	–	–	VMware
[34]	Analytical & regression based modeling	X	X	–	X	X	Xen
[35]	Analysis and model checker	X	–	–	–	–	Xen
[36]	Analytical modeling	X	X	–	–	–	All
[37]	Cost analysis	X	–	–	–	X	KVM
[24]	Cost prediction	X	X	–	–	X	Xen
[38]	Cost prediction	X	–	–	–	X	KVM
[25]	Cost prediction	–	–	X	X	X	KVM
[39]	Prediction	X	–	–	–	–	VMware but not vMotion
[40]	Prediction	–	–	X	X	X	Oracle Virtual Box
[41]	Prediction	X	X	X	X	–	All
[42]	Prediction	X	X	–	–	–	Xen
[43]	Prediction	X	X	–	–	–	VMware/Xen/KVM
[44]	Prediction	X	–	X	–	–	Xen
[45]	Markov model prediction	–	–	X	–	–	CloudSim
[46]	Prediction	X	–	–	–	–	Xen
[18]	ML based prediction	X	–	–	X	X	VMware
[47]	Analytical modeling	X	X	–	X	–	All
[48]	Cost modeling	–	–	–	–	X	Xen

regression based models for these migration cost parameters. In Reference [21], an analysis of live migration time and downtime is provided and then a comparison between Xen, KVM, Hyper-V and VMware vSphere hypervisors is presented in terms of storage migration and live migration time and downtime. A comparison between Xen and VMware live migration time and downtime is also presented in [22] with more investigation on the parameters that control the live migration time and downtime duration. The authors [49] show the impact of a VM live migration on the running applications performance from client side. The performance degradation of the application from client side was measured in operations per second. The impact of live migration on Internet Web 2.0 applications performance is discussed in [23]. This is important for environments with SLA requirements. For this purpose, a test-bed is built in [23] where the running Web 2.0 workload is Olio application, combined with Faban load generator that access the Apache 2.2.8 Web server with MySQL database. In [33] the authors propose a scheduling weighted based approach for Multi-VMs live migration requests in VMware. The objective of the proposed technique is to minimize the total migration time for Multi-VMs. The weight assigned to each request is based on the memory usage and the network bandwidth and the article shows the impact of scheduling the migration requests using this weight on the total migration time of the VMs. Article [34] studies the impact of virtualization technology and live migration on multi-tier workloads as well as the migration performance. Experimental tests show that virtualization technology does not have significant overhead on Multi-Tier applications, however live migration causes performance decrease due to the migration cost and down time. This performance degradation is more significant with memory intensive multi-tier workloads.

The authors in [50] use Probabilistic Model Checking (PMC) and Markov Decision Process (MDP) to study the impact of VM size, page size, dirty pages rate, network rate and pre-copy iterations threshold on the live migration time and down time. The proposed approach uses numerical analysis and the results should be valid for any pre-copy based live migration. In [35], the authors build a performance model for live migration using several migration tests in a Xen hypervisor based test bed and then use Probabilistic Symbolic Model Checker (PRISM) for modelling verification. The proposed approach is used to model live migration time for single and multiple concurrent VMs migration. In [36], analytical modeling is also used to formalize live migration time and down time for single and multiple VMs. Then a Markov model is build for inter-DC network to study the impact of network bandwidth, number of migration requests rate and the number of interconnected DCs on the blocking probability for migration requests.

In [37], the author studies the relationship between live migration cost parameters; namely the migration time, the network bandwidth, the power consumption and their correlation with the size of the VM memory. Testing results show that the migration time exponentially decreases as the network rate increases. The average power usage of the source as well as the destination server linearly increases as the network rate increases. The migration time and the energy consumption linearly increase as the size (memory content) of the virtual machine increases. The models presented in this paper are experimental models that are obtained using KVM Hypervisor based test-bed.

6 Live migration cost prediction

The other category of papers focus on live migration cost prediction. Classification of Live migration cost is provided in [24] with an explanation of the parameters that control migration time, downtime and energy consumption. Also, Mathematical models are proposed to estimate live migration time, downtime and energy consumption. Machine learning is used in [40] for VM placement elements predictive modeling like (CPU, memory, network and energy). The authors in [42] analyze the parameters that control the migration time and the downtime and show the impact of the workload on the migration performance. Markov chains are used in [51] for hosts utilization prediction after live migration. The proposed Markov based prediction model is used for power saving algorithm that can achieve lower SLA violations, lower VM migrations as well as less power consumption [51]. Time series is used in [44] for time varying resources load prediction. The proposed model is used for power saving by minimizing the number of active physical machines with less live migration times and with satisfying the SLA requirements. The proposed technique is tested in a Xen cluster. A mathematical based prediction framework is also proposed in [25]. In [45], a Linear Regression based CPU Utilization Prediction (LiRCUP) method is proposed to determine the future CPU utilization. The objective is to minimize the power consumption and SLA violation level during VMs live migration. This article compares four benchmark algorithms using CloudSim simulation tool. The first algorithm migrates a VM when the current CPU utilization exceeds a certain threshold. The second and third algorithms adjust the utilization threshold dynamically based on the median absolute deviation (MAD) and the inter Quartile range (IQR). The fourth method utilizes a local regression (LR) technique to predict the CPU utilization.

Authors in [46] evaluate the pre-copy migration in Xen hypervisor. The study objective is to minimize live migration duration and down time in Xen hypervisor by optimizing the total number of memory pages that should be transferred. The proposed approach achieves this target by combining two techniques. Firstly by avoiding repeated dirty pages based on using LRU (Least Recent Used) Stack Distance or using Probability Prediction that can predict the repeated pages. Secondly by using memory pages compression technique.

In [52] the authors propose a host CPU load detection algorithm called Median Absolute Deviation Markov Chain Host Detection algorithm (MADMCHD). The objective of the proposed algorithm is to minimize the SLA violation and to reduce the number of VMs migrations. This objective is achieved by using the past readings for hosts CPU utilization as input for Markov Chains that can therefore predict the future CPU states. Based on the servers CPU prediction, the proposed algorithm decides when to migrate VMs to consider SLAs and with minimizing the VMs migrations numbers. The authors of [53] propose an idea to enhance the compression for VMs memory pages before live migration. The idea is to characterize the memory pages and to identify the pseudo-stable pages. So these pages are predicted to change slowly during the iterations of live migration; which enhances the compression ratio for the pages before being migrated. The proposed algorithm is tested in a KVM lab that shows enhancement in the compression from 10 to 16%.

We have proposed in [18] our machine learning based approach to predict live migration cost in VMware environments. The proposed prediction approach uses regression techniques to predict the migration transmission rate, migration time and peak power increases given the VM active memory size.

In [54], the authors make use of the Intel EPT A/D hardware feature to track the VM disk and memory access sequence during live migration and based on that prediction methods of the disk read working-set information and memory write working-set are proposed. Then the data transfer sequence is adjusted to based on the working-set information. Also de-duplication is used to minimize the data that should be migrated; which decreases the migration cost [54].

7 Live migration cost classification

In this section, we classify live migration cost related articles listed in Table 1 based on the the live migration cost. The objectives of this classification are to show:

- A simple presentation the different models for a cost parameter; which is useful for readers who search about specific live migration cost attribute modelling and the different testing environments used for modeling validation.
- How the same live migration cost parameter is modelled differently or similarly by different researchers with different tests and distinguished hypervisors. This should help the reader to identify for the same cost parameter how much modelling might change by different test-beds, or if it is a generic model that is independent on the testing environment.
- A comparison for prediction techniques of the same cost parameter in terms of prediction dependencies, accuracy and approach complexity.
- How the proposed classification list should help identifying the research area that are still open or need more research contributions for specific cost parameters modeling or prediction.

In the next subsections, we classify Table 1 papers in modeling and prediction summary for the different cost parameters that are listed in Table 1.

7.1 Migration time modeling and prediction

As shown in Table 1, live migration time is studied by many articles from modeling and prediction points of view. In Table 2 more details about live migration time modeling and prediction are discussed.

Table 2 Migration time modeling and prediction

Paper	Equation	Hyper.	Methodology
[20] & [18]	$\mathrm{T}_{mig} = a.(V_{mem}/R_s) + b$ T_{mig}: is the migration time duration in seconds a and b are the equation constants R_s: is the source host network throughput V_{mem}: is the source host active memory size in kB at the time when the live migration should start	VMware	Regression based approach is proposed to obtain the constants and predict the migration time
[34]	$\mathrm{T}_{mig} = \frac{V_{mem}}{R} . \frac{1-\lambda^{n+1}}{1-\lambda}$ T_{mig}: Total migration time V_{mem}: Current size of VM memory during migration R: Memory transmission rate during migration D: Memory dirtying rate during migration $\lambda = D/R$	Xen	Provides an analytical model for live migration time based on the analysis of pre-copy migration phases and iterative copy iterations
[35]	$y = \alpha + \beta * (VMsize(GB)) * \mathrm{x}$ y: Total migration time (sec) α & β: Constants x: Concurrent migrating VMs	Xen	Uses empirical modeling and then probabilistic Symbolic Model Checker (PRISM) for model verification
[36]	$T^{(z)}_{mig,j} = \frac{V_z}{R_j} . \frac{1-\gamma_j^{n_j^{(z)}+1}}{1-\gamma_j}$ $\mathrm{n}_j^{(z)} = min\{log_{\gamma_j} \frac{V_{th}}{V_z}, \mathrm{n}_{max}\}$ $T_{mig,j}$: Total migration time of the j-th VM. V_z: Total memory content to be migrated for request z R_j: transfer rate for the j-th VM in the requested set γ_j: ratio between the dirtying rate and the network bit rate	All	Single and multiple VMs migration time analytical modeling. Then a Markov model is built for inter-DC network to study the impact of network bandwidth, number of migration requests rate and the number of interconnected DCs on the blocking probability for migration request
[37]	$t = as + b$ t: Migration latency s: size of the VM memory	KVM	Experimental modeling using KVM platform

Table 2 (continued)

Paper	Equation	Hyper.	Methodology
[24]	$(\mathrm{V}_{mem}/b) \leq \mathrm{t}_{mig} \leq ((\mathrm{m}_{th} + 1)V_{mem}/b)$ V_{mem}: Total Memory Size b: Network transfer rate m_{th}: Number of pre-copy iteration	Xen	Analytical modeling for migration cost based on related work survey
[38]	$t = A + (B * s)/b$ t: Migration time s: VM memory size b: Network rate A and B are constants	Xen	Experimental modeling to study VM size and network bandwidth on the migration cost
[42]	$Overheads + (VMSize/LinkSpeed) \leq TotalMigrationTime \leq Overheads + (5 * (VMSize - 1) * \mathrm{page/LinkSpeed})$ *Overhead*: Pre- and post- migration overheads *VMSize*: VM active memory size	Xen	Studying link speed and page dirty rate impact on migration time
[43]	$\mathrm{Mig.\ Time} = \min(t_{c1}, t_{c2}) + f(\mathrm{t}_2)/\mathrm{r}_u - \mathrm{t}_0$ t_{cx}: The time when stop condition x is satisfied $f(t)$: Estimated number of memory pages that remain to be copied r_u: Memory dirty pages migration rate t_0: Start time of the live migration procedure	Xen, KVM and VMware	Migration cost prediction modelling given the live migration characteristic parameters
[44]	$y = 0.0904\,x + 2.455$ y: Migration time x: VM memory size	Xen	Proposes a Prediction based Dynamic Resource Scheduling (PDRS) technique to minimize the number of active physical machines with satisfying the SLA

Migration time is a critical cost parameter because this is the time consumed during the whole migration process. So in Table 2, there are many models with common parameters that control having long or short migration duration.

From Table 2 we get that the commonly used modeling and prediction formulas for migration time is directly proportional with the VM active memory size and inversely proportional with the network transmission rate, and this relationship is valid for all hypervisors. The next cost parameter to discuss is the migration down time; when the VM is handed over to the target host and not actually responding to the application requests.

7.2 Down time modeling and prediction

Live migration down time is studied also by many articles as shown in Table 1. In Table 3, we show more details about live migration down time modeling and prediction.

Different analysis and prediction models of live migration down time as presented in Table 3. As shown, there is no common formula that is shared by the related papers. However, the proposed formulas show that the down time is directly proportional with the dirty pages rate and the page size; while the down time is inversely proportional with the transmission rate. In the next section, we discuss live migration CPU cost modeling and prediction techniques.

7.3 CPU modeling and prediction

In this subsection, we present live migration CPU overhead modeling and prediction formulation.

Table 4 indicates that the contribution in live migration CPU cost is obviously lower than other cost parameters. This might be due to the modeling and prediction complexity for CPU overhead during the live migration process. So, we consider modeling and prediction of CPU overhead as one of the open research areas in pre-copy live migration cost. From Table 4, we see that the CPU cost of live migration has different formulations. In [25], the CPU overhead of the physical server is directly proportional with the number of vCPU and the estimated CPU utilization of the VM. In [44] and [45], the future physical host CPU overhead is obtained from the last CPU utilization. In the next subsection, we discuss the network cost of live migration.

7.4 Network modeling and prediction

In Fig. 3, we explained the network setup of the VMware cluster as an example of virtualized platforms to show how live migration network is configured. In this subsection, we present the network overhead in kBps as a result of live migration of VMs.

Network cost requires also more contribution as obvious in Table 5; same as the CPU cost. So it is added also as one of the open research areas that requires more

Table 3 Down time modeling and prediction

Paper	Equation	Hyper.	Methodology
[34]	$\mathrm{T}_{down} = T_n + T_{resume}$ $n = \log_\lambda \frac{V_{thd}}{V_{mem}}$ T_n: Remaining dirty pages transferred during the stop-and-copy phase T_{resume}: time spent on VM resuming at the destination host	Xen	Analytical model for live migration down time based on the analysis of pre-copy migration phases
[36]	$T^{(z)}_{s-down} = \frac{V_z}{b}\gamma^{n^z_s} + (M-1).\frac{V_z}{b}.\frac{1-\gamma^{n^{(z)}_s+1}}{1-\gamma} + \mathrm{T}_{res}$ $T^{(z)}_{p-down} = M\frac{V_z}{b}(M\gamma)^{n^z_p} + T_{res}$ T_{s-down}: Down time for request z of M VMs migration in series $T^{(z)}_{p-down}$: Down time for request z of M VMs migration in parallel V_z: Total memory content to be migrated for request z b : Network pipe bandwidth γ: ratio between the dirtying rate and the network bit rate	All	Single and multiple VMs down time analytical modeling Then a Markov model is built for inter-DC network to study the impact of network bandwidth, number of migration requests rate and the number of connected DCs on the blocking probability of migration request
[24]	$t_{down} = (d * l * t_n)/\mathrm{b}$ t_{down}: Migration down time d: Dirty pages rate l: Page size b: Link speed t_n: Last pre-copy round n duration	Xen	Analytical modeling for migration down time
[42]	$Post-migrationOverhead \leq TotalDownTime \leq Post-migrationOverhead + ((VMSize)/LinkSpeed)$ $Post-migrationOverhead$: Time consumed in commitment and activation phases of a VM migration	Xen	Studying link speed and page dirty rate impact on migration downtime
[43]	Down time $= f(t_2)/r_u$ $f(t_2)$: The number of remaining pages in the stop-and-copy phase. r_u: Transmission rate of non-empty pages	Xen, KVM and VMware	$f(t)$ depends on the hypervisor, and in this paper, different hypervisors are tested to measure $f(t)$ and so to predict the down time

Table 4 CPU overhead modeling and prediction

Paper	Equation	Hyper.	Methodology
[25]	$PMi_{Pred_U} = (\alpha * (\sum_{y=1}^{VM_{count}} (\mathrm{VMx}_{ReqvCPUs} * \frac{VMx_{Pred_U}}{100})) + \beta) + (PMi_{Curr_U} - PMi_{Idle_U})$ PMi_{Pred_U}:The predicted PMi CPU utilization $\mathrm{VMx}_{ReqvCPUs}$: Requested vCPU for each VM VMx_{Pred_U}:Predicted utilisation for each VM PMi_{Curr_U} : Current PMi utilisation PMi_{Idle_U} : Idle PMi utilisation	KVM	A linear regression model is applied to predict the PMs CPU utilisation based on the used ratio of the requested number of vCPU for the VMx with consideration of its current workload as the PMs
[44]	$R_{k+1} = \phi_0 R_k + .. + \phi_p R_{k-p+1} + \epsilon_{k+1} + \theta_0 \epsilon_k + ..\theta_{q-1} \epsilon_{k+1-q}$ R_{k+1}:Is the first step prediction for CPU resource usage R_k ϕ_i and θ_i :Constants estimated from the available date ϵ_k :Independent error terms	Xen	Auto-Regressive Integrated Moving Average (ARIMA) model is used as the basic prediction model to predict the time series of the CPU and memory usage at time k
[45]	$PredictUtil = \beta_0 + \beta_1 * CurrTotalUtil(h)$ $PredictUtil$: Host CPU utilization prediction β_0, β_1 :Regression coeff. parameters that estimate based on the last k CPU utilization of the host $CurrTotalUtil(h)$: Current CPU utilization of the host (h)	All	This technique used the last utilization over one hour ago and approximated a function. This function can forecast the short-term future utilization based on the current requested utilization in each host

Table 5 Migration network modeling and prediction

Paper	Equation	Hyper.	Methodology
[20] & [18]	$R_s = \alpha e^{V_{Mem}} + \beta$ R_s: is the source host network throughput α and β are the equation constants V_{mem}: is the source host active memory size in kB at the time when the live migration should start	VMware	Non-linear Regression approach is applied to model and predict the migration network rate given the V_{mem}
[34]	$R = D + \delta$ R: Memory transmission rate during migration D: Memory dirtying rate during migration δ :is a constant variable and its default value is empirically set as 100 Mb/s	Xen	Migration data transmission rate for each round is determined by adding a constant increment to the previous round's memory dirtying rate
[47]	$\mathrm{C}_s^z = \alpha C_{s-net}^z + (1-\alpha) C_{s-app}^z$ $\mathrm{C}_p^z = \alpha C_{p-net}^z + (1-\alpha) C_{p-app}^z$ $0 \leq \alpha \leq 1$ $\mathrm{C}_{s-net}^z = \frac{1-\gamma^{n_s^z+1}}{(n_{max}+1)(1-\gamma)}$ $\mathrm{C}_{p-net}^z = \frac{1-(M_{z\gamma})^{n_p^z+1}}{(n_{max}+1)(1-M_{z\gamma})}$ $\mathrm{C}_{s-app}^z = \frac{V_z\left(\gamma^{n_s^z}+(M_z-1)\frac{1-\gamma^{n_s^z+1}}{1-\gamma}\right)+T_{res}b}{V_z(1+(M_z-1)(n_{max}+1))+T_{res}b}$ $\mathrm{C}_{p-app}^z = \frac{M_z Vz(M_{z\gamma})^{n_p^z}+T_{res}b}{M_z V_z+T_{res}b}$ M_z: Number of migrated VMs for request z V_z: Allocated memory for each VM γ: Ratio of the dirtying rate to the maximum transfer rate T_{res}: Time required to resume a VM at the destination host b: Network pipe max. transfer rate n_{max}: Maximum number of iterations that trigger the stop and-copy phase n: Number of remote DCs in the cloud federation	All	Presents an analytical model for assessing inter-DC network performance in cloud federations, assuming that network load is caused by live migration of multiple VMs

study in modeling and prediction. The formulas proposed in Table 5 show that the network transmission rate has an exponential relationship with the VM active memory size as in [20] and [18]. In [34], the network transmission rate has directly proportional relationship with the dirty pages rate during the migration process. In [47] the transmission rate is function of the number of VMs to be migrated, the dirty pages rate, the maximum bandwidth of the network pipe and the time required to resume the VM at the destination host. The last cost parameter table to discuss is for the power and energy modeling and prediction in Table 6.

7.5 Power and energy modeling and prediction

In this subsection, we present the energy and power overhead modeling and prediction due to live migration

Table 6 lists different live migration energy cost models. The energy overhead in Joule is directly proportional with the memory content to be migrated as mentioned in [24, 34] and [38]. Another representation is shown in [20] and [18] for the power overhead in Watt as directly proportional with the transmission rate. Other models proposed in [25] and [48] say that the power consumption prediction in Watt is directly proportional with the CPU utilization.

Because live migration is basically a migration for the memory volume, so in the next section we discuss a new memory technology which is persistent memory that is recently provided by different servers manufactures and supported by many software vendors [55].

8 Live migration with persistent memory

Persistent memory is a recently developed technology that keeps the data stored at the memory tier even with system reboot or power off [56]. This technology is now supported by many hardware, operating systems, middle-ware and applications vendors [55]. Memory intensive and in-memory applications are one of the main drives for persistent memory utilization. These applications use memory capacity intensively to gain high performance and low latency. However, the admins for these applications face challenges in large memory cost with DRAM and system reboot duration due to running a volatile memory. Persistent memory is a new tier of storage that is added between DRAM and standard flash SSDs; as shown in Fig. 5. This new byte-addressable tier of storage minimizes the bottlenecks in read and write operations between the DRAM and NAND SSDs due to the big gap in performance. So adding this new storage tier provides a more balanced data management for computer systems [56].

Persistent memory comes in two main silicon technologies; Non-Volatile DIMMs (NVDIMMs) or Non-Volatile RAM (NVRAM) and 3D xPoint Persistent Memory (PMem) which is 3D-xPoint based silicon technology released only by Intel [57] and Micron [58] in 2019. From the CPU prospective, Intel Xeon Cascadelake CPU

Table 6 Power and energy modeling and prediction

Paper	Equation	Hyper.	Methodology
[20]& [18]	$P_{mig} = \text{c R}_s$ P_{mig}: peak power overhead in Watt c: is constant R_s: is the source host network throughput	VMware	Regression based approach is proposed to obtain the constants and predict the peak power overhead due to migration
[34]	$E_{mig} = (\alpha_s + \alpha_d)V_{mig} + (\beta_s + \beta_d)$ E_{mig}: Migration energy consumption overhead in Joules V_{mig}: Current size of memory content to be migrated in MB $\alpha_s, \alpha_d, \beta_s$ and β_d : model parameters to be trained	Xen	Analytical model for live migration energy overhead at source and target stations There is very little difference of energy consumption between source and target stations energy overhead
[24]	$E_{mig} = \alpha * V_{mig} + \beta$ E_{mig}: Energy overhead of VM migration V_{mig}: Total Memory Size α and β: trained parameters in offline experiments	Xen	Analytical modeling for migration cost. The authors validated their approach based on five different benchmarks running on the migrated VM
[38]	$E_{ov} = A + B * s + C * b$ E_{ov}: VM migration energy overhead . s: VM memory size b: Network rate A, B and C: are constants to be trained	Xen	Experimental modeling to propose a lightweight, linear model to estimate the energy cost of live migration of virtual machines
[25]	$PMi_{Pred_P} = (\alpha * \text{PMi}_{Pred_U} + \beta)$ PMi_{Pred_P} : The predicted PMi Power consumption for the source station in Watt PMi_{Pred_U}:The predicted PMi CPU utilization α and β : constant parameters	KVM	Propose steps to predict the PMs/VMs workload and power consumption, then estimate the total cost of the migrated VMs
[48]	$P = \text{a} * Util. + \text{b}$ P: The predicted Power consumption of the source server in Watt $Util.$: The predicted CPU utilization a and b: constant parameters	Xen	Proposes a practical experimental approach to evaluate the power consumption of VM migration. And then we quantify the power cost of VM migration both source and target servers

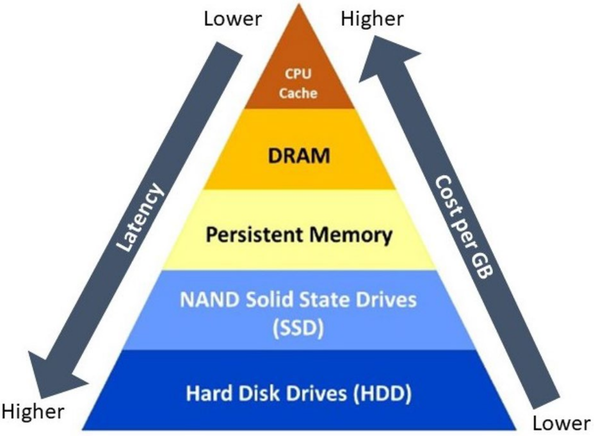

Fig. 5 Storage tiers hierarchy

[59]; released in 2019 is the first and so far the only CPU that supports PMem; this is up to the date of writing this article.

8.1 Non volatile DIMM (NVDIMMs)

NVRAM or NVDIMMs is based on the CMOS transistor that is used in the standard DDR4 RAM, however the memory architecture and power management ICs are modified to run with battery-backed power source that flushes the data from the DRAM to an internal NAND drive in order to have have data persistence during power failures or system restart [60–62] and [63]. So NVRAM has the same performance as the volatile DRAM but has the the endurance of NAND technology [64, 65] and [62]. NVDIMMs are supported by VMware vSphere 6.7 and above [66].

8.2 3D *x*Point PMem

3D xPoint is a new silicon technology developed by the memory vendors Intel and Micron [67]. Compared to NVRAM, 3D xPoint PMem has higher capacity density, lower cost per GB, higher durability and native persistent; without batteries [67]. From performance point of view, 3D xPoint PMem has comparable bandwidth, however PMem has 10x more latency compared to DDR4 DRAM memory. That is why 3D xPoint PMem is not proposed as a replacement for DDR4 DRAM, but to be added as a new tier of memory between DRAM and the standard flash NAND SSDs as a higher capacity and a more cost effective memory solution that provides also higher system availability due the storage persistence [68].

Table 7 shows a comparison between DRAM, NVDIMMs and 3D xPoint PMem specifications. As shown; DRAM has advantages in latency and endurance, however the dis-advantages in DRAM are the low capacity per DIMM, volatile storage

Table 7 Power and energy modeling and prediction

Mem. tech.	Max. cap. per DIMM (GB)	Latency (ns)	Endurance	Silicon tech	Persistent	Cost/GB
DRAM	128	10’s	CMOS	CMOS	No	Medium
NV-DIMM	128	10’s	NAND	CMOS and NAND	Yes	Highest
3D xPoint PMem	512	100’s	3D xPoint	3D xPoint	Yes	Lowest

medium and the cost per GB. NVDIMMs has advantages in latency, and in data persistent, however the disadvantages in NVDIMMs are the low capacity per DIMM which is exactly as DRAM, the medium endurance and the cost per GB. The 3D xPoint PMem has advantage in the capacity per DIMM, the medium endurance, the data persistent and the lower cost per GB. However the disadvantages in 3D xPoint PMem is the higher latency compared to DRAM and NVDIMMs.

3D xPoint PMem has two modes of operations; Memory Mode (MM) and app direct mode (AppDir); as shown in Fig. 6. In MM, the PMem acts as a volatile memory and the only gains of using it are the capacity per DIMM and the endurance. In MM, there is no need to do any development in the hypervisor, the OS and the application to discover the PMem as the system discovers it as normal DRAM. In MM also, if there are PMem DIMMs as well as DDR4 DIMMs in the same server, the DDR4 DIMMs act as memory cache for the PMem DIMMs [62].

In AppDir mode, the PMem acts as persistent memory and in this case DDR4 and PMem are presented as two different types of memory to the hypervisors, the OS and the applications. So all these three software stacks should be developed to decide which part of the data should be stored in the PMem to utilize the larger capacity and persistent storage and which part should reside on the DDR4 to get the lower latency. PMem in both modes is supported by many applications (like SAP HANA, Apache Cassandra, Apache Spark SQL and Microsoft SQL server 2019), OSs vendors(like Windows Server 2019, RHEL 7.6, Ubuntu 18.10, CentOS* 7.6 and SLES* 12 SP4), hypervisors vendors (VMware* ESXi 6.7 EP 10 or

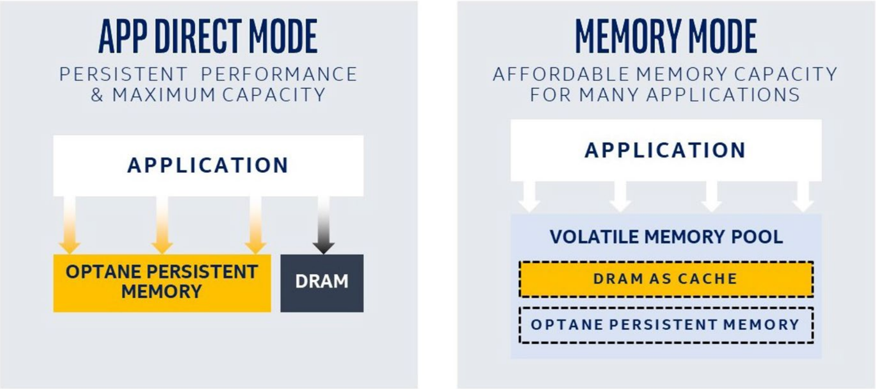

Fig. 6 AppDir mode vs memory mode [67]

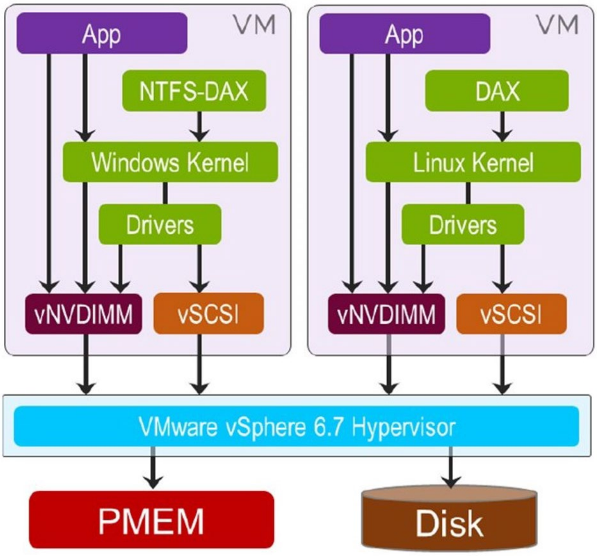

Fig. 7 vSphere support for PMem [62]

later, Xen project and Microsoft Hyper-V 2019) and servers vendors (like Dell, Supermicro, Lenovo and HPE) [69] and [57].

For the AppDir mode presented in Fig. 6, we show how VMware vSphere discovers PMem in Fig. 7 as an example of a commonly used hypervisor that supports PMem starting with vSphere 6.7 release [70]. Figure 7 is presented in [62] . As shown; for vSphere 6.7 and above, the hypervisor can actually recognize the PMem as a persistent memory device or as a very fast block storage disk; which is a memory class of storage. The decision of the PMem module representation is done using the server BIOS. For utilizing the PMem as a persistent memory, the logical persistent volume vNVDIMM should be used by the hypervisor to represent the PMem to the VM OS. For utilizing the PMem as a fast block storage, the logical volume vSCSI should be used to represent the PMem to the VM OS [69].

Live migration of VMware vSphere VMs with PMem inside is supported, however the high availability features is not supported yet. So for future research, it might be interesting to study if live migration cost modeling and cost prediction techniques there were studied on the DDR4 DRAM will be valid also for PMem as a new memory technology.

9 Open research areas

Based on this survey study, we can list the open research areas for pre-copy live migration cost in the below points:

(a) To propose lightweight and practical prediction techniques for live migration cost that can be integrated with the hypervisors interfaces to be used by clusters' admins. Most of the proposed prediction techniques consume time and CPU intensively to train the models and predict live migration cost prediction [71]; which results in a challenge to have a technique that can be practically implemented.

(b) Live migration cost aware load balance techniques for cloud environments. This is because load balance utilizes live migration in virtual and cloud computing environments [72]. So since live migration cost can not be ignored, it should be considered as an overhead for load balance techniques.

(c) Live migration cost aware power saving techniques for cloud environments. Same as load balance, power saving also utilizes live migration [73, 74] and [75]. So, it is important to consider live migration cost and especially the power overhead in power saving techniques for virtual and cloud computing environments.

(d) Live migration cost modeling and prediction for WAN scale migration [76] and [77]. Most of the proposed modeling and prediction techniques were tested and proposed for LAN scale live migration; where the source and target hosts are within the same datacenter. So, it is an open research area to provide models and tests for WAN scale live migration.

(e) Live migration cost analysis and prediction for VMware and Hyper-V. This is taking into account the fact that most of the proposed modeling and prediction techniques tests focus mainly on open source hypervisors like Xen and KVM. So more research work is needed toward commercially used hypervisors.

(f) Live migration cost modeling and prediction with different in memory data analytics applications like Apache Spark [78] and in memory data base applications like Redis [79]. These in memory applications are memory intensive application and show example for modern applications trend that run on private and public cloud environments.

(g) Referring to Table 4, more research work is still needed in live migration CPU cost modeling and prediction. Since few papers could be found that consider this cost parameter modeling and prediction.

(h) Referring to Table 5, more research work is still needed in live migration network cost modeling and prediction. Since also few papers could be found that consider this migration network modeling and prediction.

(i) From Tables 2, 3, 4, 5 and 6, to compare between different prediction techniques for the same cost parameter. The comparison can be in terms of the prediction accuracy and the prediction CPU consumption overhead.

(j) Live migration cost analysis and prediction for VMs with Persistent Memory (PMem) and to compare the cost versus the standard DRAM. This includes the different memory configurations of PMem as following:

- VMs with 3D xPoint PMem in memory mode.
- VMs with 3D xPoint PMem in App Direct mode.
- VMs with NVDIMMs in memory mode.
- VMs with NVDIMMs in App direct mode.

(k) Consider variant memory capacities in the PMem test, in order to add contribution about very large memory VMs migrations; especially with 3D xPoint PMem as a high capacity/DIMM memory technology.

10 Conclusion

Live migration is an essential feature for private, public and hybrid cloud environments. This feature is a basic requirement for dynamic management of the VMs as well as system load balance, power saving and fault tolerance. Pre-copy is a commonly used type of live migration by almost all the commercial hypervisors, and this is due to this type of copy robustness and reliability against failures during the migration process.

In this paper, we studied, compared and analyzed different articles that discuss pre-copy live migration cost modeling and prediction techniques. The paper starts with introducing live migration concept, benefits, types, configuration scenarios and networking set up. This is to give a detailed background to the readers about pre-copy live migration feature. Then we discuss live migration drawbacks in Sect. 4 in the form of cost parameters that are presented and analyzed as a result for the six phases of pre-copy live migration. In Sects. 5 and 6, many research articles are discussed to show the main related work in pre-copy live migration cost modeling and prediction.

The main contribution of this paper is presented in Sect. 7 which is a classification proposal of these papers based on the cost parameter. Such that, all papers that have proposed the same cost parameter modeling or prediction were added in the same table including the proposed models equations. This shows to the readers of this paper, how modeling or prediction of the same cost parameter might change from a research article to another depending on the hypervisor, the modeling or the prediction methodology. This classification shows also how different research articles might have similar cost modeling and prediction relations for a specific cost parameter; which gives a more credibility to this model or prediction relationship. Because live migration is basically data transfer for the memory pages of the VMs, we discussed the persistent memory technology topic in Sect. 8 as a new trend in servers industry that is recently supported by many hardware and software vendors. In this section, we showed the difference between DRAM and persistent memory technologies specifications to introduce that live migration cost migth change with using persistent memory. Finally, we proposed many research directions in Sect. 9 that are still open in the area of pre-copy migration cost modeling and prediction.

Funding Open Access funding enabled and organized by Projekt DEAL. This study is not funded by any party or funding institution.

Declarations

Conflict of interest There is no conflict of interest by any of the authors of this article.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Somani, G., Chaudhary, S.: Application performance isolation in virtualization. IEEE Int. Conf. Cloud Comput. **2009**, 41–48 (2009). https://doi.org/10.1109/CLOUD.2009.78
2. Akoush, S., Sohan, R., Rice, A., Moore, A.W., Hopper, A.: Predicting the performance of virtual machine migratio., In: Proceedings of the 2010 IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, MASCOTS '10, IEEE Computer Society, Washington, DC, USA, pp. 37–46 (2010)
3. Hu, B., Lei, Z., Lei, Y., Xu, D., Li, J.: A time-series based precopy approach for live migration of virtual machines. In: Proceedings of the 2011 IEEE 17th International Conference on Parallel and Distributed Systems, ICPADS '11, IEEE Computer Society, Washington, DC, USA, pp. 947–952 (2011)
4. Fernando, D., Terner, J., Gopalan, K., Yang, P.: Live migration ate my vm: Recovering a virtual machine after failure of post-copy live migration. In: IEEE INFOCOM 2019—IEEE Conference on Computer Communications, pp. 343–351 (2019). https://doi.org/10.1109/INFOCOM.2019.8737452
5. Le, T.: A survey of live virtual machine migration techniques. Comput. Sci. Rev. **38**, 100304 (2020). https://doi.org/10.1016/j.cosrev.2020.100304
6. Hu, L., Zhao, J., Xu, G., Ding, Y., Chu, J.: Hmdc: live virtual machine migration based on hybrid memory copy and delta compression (2013)
7. Sahni, S., Varma, V.: A hybrid approach to live migration of virtual machines. In: IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), pp. 1–5 (2012)
8. Gupta, T., Ganatra, J., Samdani, K.: A survey of emerging network virtualization frameworks and cloud computing. In: 2018 8th International Conference on Cloud Computing, Data Science Engineering (Confluence), pp. 14–15 (2018). https://doi.org/10.1109/CONFLUENCE.2018.8442995
9. Tseng, H., Lee, H., Hu, J., Liu, T., Chang, J., Huang, W.: Network virtualization with cloud virtual switch. In: 2011 IEEE 17th International Conference on Parallel and Distributed Systems, pp. 998–1003 (2011). https://doi.org/10.1109/ICPADS.2011.159
10. VMware virtual Distributed Switch: https://docs.vmware.com/en/VMware-vSphere/6.0/com.vmware.vsphere.networking.doc/GUID-3147E090-D9BF-42B4-B042-16F8D4C92DE4.html
11. VMware VMkernel: https://docs.vmware.com/en/VMware-vSphere/6.7/com.vmware.vsphere.networking.doc/GUID-D4191320-209E-4CB5-A709-C8741E713348.html
12. Hyper-V Switch: https://docs.microsoft.com/en-us/windows-server/virtualization/hyper-v-virtual-switch/hyper-v-virtual-switch
13. Hyper-V Live Migration: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/
14. Xen Virtual Switch: https://docs.citrix.com/en-us/citrix-hypervisor/technical-overview.html

15. Xen Live Migration: https://docs.citrix.com/en-us/xenserver/7-0/downloads/administrators-guide.pdf
16. KVM Hypervisor: https://libvirt.org/drvqemu.html
17. Redhat Migration: https://developers.redhat.com/blog/2015/03/24/live-migrating-qemu-kvm-virtual-machines/
18. Elsaid, M.E., Abbas, H.M., Meinel, C.: Machine learning approach for live migration cost prediction in vmware environments. In: Muñoz, V.M., Ferguson, D., Helfert, M., Pahl, C. (eds.) Proceedings of the 9th International Conference on Cloud Computing and Services Science, CLOSER 2019, Heraklion, Crete, Greece, May 2–4, 2019, SciTePress, pp. 456–463 (2019. https://doi.org/10.5220/0007749204560463
19. Elsaid, M.E., Sameh, M., Abbas, H.M., Meinel, C.: Live migration timing optimization integration with vmware environments. In: 10th International Conference, Cloud Computing and Services Science CLOSER 2020, Prague, Czech Republic, May 7–9. Revised Selected Papers 1399, (2021), pp. 133–152 (2020)
20. Elsaid, M.E., Meinel, C.: Live migration impact on virtual datacenter performance: Vmware vmotion based study. In: 2014 International Conference on Future Internet of Things and Cloud, pp. 216–221 (2014). https://doi.org/10.1109/FiCloud.2014.42
21. Hu, W., Hicks, A., Zhang, L., Dow, E.M., Soni, V., Jiang, H., Bull, R., Matthews, J.N.: A quantitative study of virtual machine live migration. In: Proceedings of the 2013 ACM Cloud and Autonomic Computing Conference, CAC '13, ACM, New York, NY, USA, , pp. 11:1–11:10 (2013) https://doi.org/10.1145/2494621.2494622
22. Salfner, F., Tr oger, P., Polze, A.: Downtime analysis of virtual machine live migration. In: The Fourth International Conference on Dependability, IARIA, IARIA, pp. 100–105 (2011)
23. Voorsluys, W., Broberg, J., Venugopal, S., Buyya, R.: Cost of virtual machine live migration in clouds: a performance evaluation. In: Proceedings of the 1st International Conference on Cloud Computing, CloudCom '09, Springer-Verlag, Berlin, Heidelberg, pp. 254–265 (2009). https://doi.org/10.1007/978-3-642-10665-1_23
24. Strunk, A.: Costs of virtual machine live migration: a survey. In: Proceedings of the 2012 IEEE Eighth World Congress on Services, SERVICES '12, IEEE Computer Society, Washington, DC, USA, pp. 323–329 (2012). https://doi.org/10.1109/SERVICES.2012.23
25. Aldossary, M., Djemame, K.: Performance and energy-based cost prediction of virtual machines live migration in clouds. In: Proceedings of the 8th International Conference on Cloud Computing and Services Science, CLOSER 2018, Funchal, Madeira, Portugal, March 19–21, pp. 384–391 (2018). https://doi.org/10.5220/0006682803840391
26. Clark, C., Fraser, K., Hand, S., Hansen, J.G., Jul, E., Limpach, C., Pratt, I., Warfield, A.: Live migration of virtual machines. In: Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation—Volume 2, NSDI'05, USENIX Association, Berkeley, CA, USA, pp. 273–286 (2005). http://dl.acm.org/citation.cfm?id=1251203.1251223
27. Salfner, P.T.F., Polze, A.: Downtime analysis of virtual machine live migration. In: The Fourth International Conference on Dependability (DEPEND 2011), ISBN: 978-1-61208-149-6, pp. 100–105, French Riviera, France
28. Rybina, K., Schill, A.: Estimating energy consumption during live migration of virtual machines. In: IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), pp. 1–5 (2016)
29. Strunk, A., Dargie, W.: Does live migration of virtual machines cost energy?. In: 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA), pp. 514–521 (2013)
30. Wu, Y., Zhao, M.: Performance modeling of virtual machine live migration. In: 2011 IEEE 4th International Conference on Cloud Computing, pp. 492–499 (2011)
31. Noshy, M., Ibrahim, A., Ali, H.A.: Optimization of live virtual machine migration in cloud computing: a survey and future directions. J. Netw. Comput. Appl. **110**, 1–10 (2018). https://doi.org/10.1016/j.jnca.2018.03.002
32. Jiang, X., Yan, F., Ye, K.: Performance influence of live migration on multi-tier workloads in virtualization environments. In: CLOUD (2012)
33. Chen, Y., Liu, I., Chou, C., Li, J., Liu, C.: Multiple virtual machines live migration scheduling method study on vmware vmotion. In: 2018 3rd International Conference on Computer and Communication Systems (ICCCS), pp. 113–116 (2018)

34. Liu, H., Xu, C.-Z., Jin, H., Gong, J., Liao, X.: Performance and energy modeling for live migration of virtual machines. In: HPDC (2011)
35. Kikuchi, S., Matsumoto, Y.: Performance modeling of concurrent live migration operations in cloud computing systems using prism probabilistic model checker. In: 2011 IEEE 4th International Conference on Cloud Computing, pp. 49–56 (2011)
36. Cerroni, W.: Multiple virtual machine live migration in federated cloud systems. In: IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 25–30 (2014)
37. Dargie, W.: Estimation of the cost of VM migration. In: 23rd International Conference on Computer Communication and Networks, ICCCN 2014, Shanghai, China, August 4–7, IEEE, pp. 1–8 (2014). https://doi.org/10.1109/ICCCN.2014.6911756
38. Strunk, A.: A lightweight model for estimating energy cost of live migration of virtual machines. In: 2013 IEEE Sixth International Conference on Cloud Computing, pp. 510–517 (2013)
39. Zhao, M., Figueiredo, R.J.: Experimental study of virtual machine migration in support of reservation of cluster resources. In: Proceedings of the 2nd International Workshop on Virtualization Technology in Distributed Computing, VTDC '07, ACM, New York, NY, USA, pp. 5:1–5:8 (2007). https://doi.org/10.1145/1408654.1408659
40. Berral, J.L., Gavaldà, R., Torres, J.: Power-aware multi-data center management using machine learning. In: Proceedings of the 2013 42nd International Conference on Parallel Processing, ICPP '13, IEEE Computer Society, Washington, DC, USA, pp. 858–867 (2013). https://doi.org/10.1109/ICPP.2013.102
41. Jo, C., Cho, Y., Egger, B.: A machine learning approach to live migration modeling. In: Proceedings of the 2017 Symposium on Cloud Computing, SoCC '17, ACM, New York, NY, USA, pp. 351–364 (2017). https://doi.org/10.1145/3127479.3129262
42. Akoush, S., Sohan, R., Rice, A., Moore, A.W., Hopper, A.: Predicting the performance of virtual machine migration. In: Proceedings of the 2010 IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, MASCOTS '10, IEEE Computer Society, Washington, DC, USA, pp. 37–46(2010). https://doi.org/10.1109/MASCOTS.2010.13
43. Salfner, F., Troger, P., Richly, M.: Dependable estimation of downtime for virtual machine live migration. Int. J. Adv. Syst. Meas. **5**, 70–88 (2012)
44. Huang, Q., Shuang, K., Xu, P., Liu, X., Su, S.: Prediction-based dynamic resource scheduling for virtualized cloud systems. J. Netw. **9**, 375–383 (2014)
45. Farahnakian, F., Liljeberg, P., Plosila, J.: LiRCUP: linear regression based CPU usage prediction algorithm for live migration of virtual machines in data centers. In: 2013 39th Euromicro Conference on Software Engineering and Advanced Applications, IEEE (2013). https://doi.org/10.1109/seaa.2013.23
46. Patel, M., Chaudhary, S.: Survey on a combined approach using prediction and compression to improve pre-copy for efficient live memory migration on xen. In: 2014 International Conference on Parallel, Distributed and Grid Computing, pp. 445–450 (2014)
47. Cerroni, W.: Network performance of multiple virtual machine live migration in cloud federations. J. Internet Serv. Appl. **6**(1), 6:1-6:20 (2015). https://doi.org/10.1186/s13174-015-0020-x
48. Huang, Q., Gao, F., Wang, R., Qi, Z.: Power consumption of virtual machine live migration in clouds. In: Third International Conference on Communications and Mobile Computing, pp. 122–125 (2011)
49. Bezerra, P., Martins, G., Gomes, R., Cavalcante, F., Costa, A.: Evaluating live virtual machine migration overhead on client's application perspective. In: International Conference on Information Networking (ICOIN), pp. 503–508 (2017). https://doi.org/10.1109/ICOIN.2017.7899536
50. Bashar, A., Mohammad, N., Muhammed, S.: Modeling and evaluation of pre-copy live vm migration using probabilistic model checking. In: 2018 12th International Conference on Signal Processing and Communication Systems (ICSPCS), pp. 1–7 (2018)
51. Melhem, S.B., Agarwal, A., Goel, N., Zaman, M.: Markov prediction model for host load detection and VM placement in live migration. IEEE Access **6**, 7190–7205 (2018). https://doi.org/10.1109/ACCESS.2017.2785280
52. Melhem, S.B., Agarwal, A., Goel, N., Zaman, M.: A markov-based prediction model for host load detection in live vm migration. In: 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud), pp. 32–38 (2017)

53. Elghamrawy, K., Franklin, D., Chong, F.T.: Predicting memory page stability and its application to memory deduplication and live migration. In: IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), pp. 125–126 (2017)
54. Shi, B., Shen, H.: Memory/disk operation aware lightweight vm live migration across data-centers with low performance impact. In: IEEE INFOCOM 2019—IEEE Conference on Computer Communications, pp. 334–342 (2019)
55. SNIA Persistent Memory Introduction: https://www.snia.org/education/what-is-persistent-memory
56. Programming with Intel Persistent Memory: https://software.intel.com/content/www/us/en/develop/articles/introduction-to-programming-with-persistent-memory-from-intel.html
57. Intel Persistent Memory Technology Partners: http://www.intel.com/content/www/us/en/architecture-and-technology/optane-dc-persistent-memory.html
58. Micron Persistent Memory Technology: www.micron.com/campaigns/persistent-memory
59. Intel Cascadelake CPU: https://www.intel.com/content/www/us/en/design/products-and-solutions/processors-and-chipsets/cascade-lake/2nd-gen-intel-xeon-scalable-processors.html
60. Dell R740 server with NVDIMM: https://topicscdn.dell.com/pdf/poweredge-r740_users-guide3$_$en-us.pdf
61. Dell R940 server with NVDIMM: www.dell.com/support/manuals/us/en/04/poweredge-r940/nvdimmn_ug_pub/introduction?guid=guid-8884370c-5553-4089-b613-a3c570b56f0e&lang=en-us
62. HPE NVDIMM: www.hpe.com/us/en/servers/persistent-memory.html
63. NVDIMMs Support for HPE Servers: https://support.hpe.com/hpesc/public/docDisplay?docId=a00038965en_us&docLocale=en_US
64. Lee, C., Shin, W., Kim, D.J., Yu, Y., Kim, S., Ko, T., Seo, D., Park, J., Lee, K., Choi, S., Kim, N., G, V., George, A., V.V, Lee, D., Choi, K., Song, C., Kim, D., Choi, I., Jung, I., Song, Y.H., Han, J.: Nvdimm-c: A byte-addressable non-volatile memory module for compatibility with standard ddr memory interfaces. In: 2020 IEEE International Symposium on High Performance Computer Architecture (HPCA), pp. 502–514 (2020)
65. Katzburg, N., Golander, A., Weiss, S.: Nvdimm-n persistent memory and its impact on two relational databases. In: IEEE International Conference on the Science of Electrical Engineering in Israel (ICSEE), pp. 1–5 (2018)
66. NVDIMMs Support for VMware vSphere: https://storagehub.vmware.com/t/vsphere-6-7-core-storage-1/pmem-persistant-memory-nvdimm-support-in-vsphere/
67. Waddington, D., Kunitomi, M., Dickey, C., Rao, S., Abboud, A., Tran, J.: Evaluation of intel 3d-xpoint nvdimm technology for memory-intensive genomic workloads. In: Proceedings of the International Symposium on Memory Systems, MEMSYS '19, Association for Computing Machinery, New York, NY, USA, pp. 277–287 (2019). https://doi.org/10.1145/3357526.3357528
68. Persistent Memory Technology value proposition for Database Applications: https://itpeernetwork.intel.com/give-your-database-apps-a-performance-boost-with-a-new-memory-tier/#gs.8fwfmr
69. SNIA Persistent Memory Summit 2020: https://www.snia.org/pm-summit
70. vSphere Support for Persistent Memory: https://storagehub.vmware.com/t/vsphere-6-7-core-storage-1/pmem-persistant-memory-nvdimm-support-in-vsphere/
71. Elsaid, M.E., Abbas, H.M., Meinel, C.: Machine learning approach for live migration cost prediction in vmware environments. In: Proceedings of the 9th International Conference on Cloud Computing and Services Science, CLOSER 2019, Heraklion, Crete, Greece, May 2–4, pp. 456–463 (2019). https://doi.org/10.5220/0007749204560463
72. Kumar, P., Kumar, R.: Issues and challenges of load balancing techniques in cloud computing: a survey. ACM Comput. Surv. **51**(6), 1–35 (2019)
73. Xie, R., Jia, X., Yang, K., Zhang, B.: Energy saving virtual machine allocation in cloud computing. In: 2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops, pp. 132–137 (2013)
74. Graubner, P., Schmidt, M., Freisleben, B.: Energy-efficient virtual machine consolidation. IT Prof. **15**(2), 28–34 (2013). https://doi.org/10.1109/MITP.2012.48
75. Akhter, N., Othman, M.: Energy efficient virtual machine provisioning in cloud data centers. In: 2014 IEEE 2nd International Symposium on Telecommunication Technologies (ISTT), pp. 330–334 (2014)
76. Travostino, F., Daspit, P., Gommans, L., Jog, C., de Laat, C., Mambretti, J., Monga, I., van Oudenaarde, B., Raghunath, S., Wang, P.Y.: Seamless live migration of virtual machines over the man/wan. Future Gener. Comput. Syst. **22**(8), 901–907 (2006). https://doi.org/10.1016/j.future.2006.03.007

77. Wood, T., Ramakrishnan, K.K., Shenoy, P., van der Merwe, J.: Cloudnet: dynamic pooling of cloud resources by live wan migration of virtual machines. In: Proceedings of the 7th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, VEE '11, Association for Computing Machinery, New York, NY, USA, pp. 121–132 (2011). https://doi.org/10.1145/1952682.1952699
78. Apache Spark: https://spark.apache.org/
79. Redis Database: https://redis.io/

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.