Live Migration Timing Optimization
Integration with VMware Environments

Mohamed Esam Elsaid1(✉), Mohamed Sameh2, Hazem M. Abbas2, and Christoph Meinel1

1 Internet Technologien und Systeme, Hasso-Plattner Institut, Potsdam University, Potsdam, Germany
{Mohamed.Elsaid,Meinel}@HPI.de
2 Department of Computer and Systems Engineering, Ain Shams University, Cairo, Egypt
mohamedsamehkhalil@gmail.com, Hazem.Abbas@eng.asu.edu.eg

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 overhead. Currently, virtual datacenters admins run live migrations without an idea about the migration cost prediction and without recommendations about the optimal timing for initiating a VM live migration especially for large memory VMs or for concurrently multiple VMs migration. Without cost prediction and timing optimization, live migration might face longer duration, network bottlenecks and migration failure in some cases. The previously proposed timing optimization approach is based on using machine learning for live migration cost prediction and the network utilization prediction of the cluster. In this paper, we show how to integrate our machine learning based timing optimization algorithm with VMware vSphere. This integration deployment proves the practicality of the proposed algorithm by presenting the building blocks of the tools and backend scripts that should run to implement this timing optimization feature. The paper shows also how the IT admins can make use of this novel cost prediction and timing optimization option as an integrated plug-in within VMware vSphere UI to be notified with the optimal timing recommendation in case of a having live migration request.

Keywords: Cloud computing · Virtual · Live migration · Timing · VMware · vMotion · Modeling · Overhead · Cost · Datacenter · Prediction · Machine learning

1 Introduction

Datacenter resource virtualization is commonly used by IT administrators during the last decade. Running virtual or software defined machines has shown...
higher availability, rapid scaling, better resource utilization and more cost efficiency. Live migration of virtual machines is a key feature in virtual environments and cloud computing datacenters. Using live migration, virtual machines can be moved from a physical host to another while the applications are running online. This is because live migration causes negligible service interruption during the migration process. Servers load balance, power saving, fault tolerance and dynamic virtual machines allocation are all dependent on live migration. During the live migration process, the VM CPU cache, memory pages and IO buffers contents are migrated. However the storage content is shared between the source and the target servers, so storage content is not migrated. Live migration traffic is sent over the TCP/IP network that interconnects the virtualized cluster. During the live migration process, the memory, the CPU cache and system buffers content. However the memory content is the major content to be migrated.

Virtualized clusters load balance, power saving, dynamic resource management and fault tolerance depend on live migration feature.

- 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 switching 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.

Live migration has three different types; Pre-copy, Post-copy and Hybrid-copy [16]. As discussed in [16], Pre-copy Live migration is the commonly used type due to its robustness against VMs crash during the migration. So, Pre-copy migration is used by almost all hypervisors in the market; VMware ESXi, Microsoft Hyper-V, Xen and KVM.

As discussed before, VMs live migration is an essential feature in cloud computing and virtual datacenter environments, however live migration cost can not be ignored. The cost includes migration time, down time, network, CPU and power consumption overhead. The definition and the root cause of each cost parameter are as following:

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 [17].

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 [27]. 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 [30].

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 [34]. Meanwhile, the more available CPU resources, the less migration time in case of having available network bandwidth.

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 [14] 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 regression based models for these migration cost parameters. In Reference [19], 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 [28] with more investigation on the parameters that control the live migration time and downtime duration. The authors [9] 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 [33]. This is important for environments with SLA requirements. For this purpose, a test-bed is built in [33] where the running Web 2.0 workload is Olio application, combined with Faban load
Table 1. Summary of related work.

| Paper | Research scope | Mig. time | Down time | CPU | Network | Power | Testing env. |
|-------|----------------|-----------|-----------|-----|---------|-------|-------------|
| [14]  | Regression Modeling | X         | –         | –   | X       | X     | VMware      |
| [19]  | Perf. Comparison   | X         | X         | –   | –       | –     | All         |
| [28]  | Analysis and Comparison | X         | X         | –   | –       | –     | Xen, VMware |
| [33]  | Perf. Evaluation   | X         | X         | –   | –       | –     | Xen         |
| [22]  | Analysis on Apps Perf. | X         | X         | –   | –       | –     | Xen         |
| [12]  | Multi-VMs Scheduling | X         | –         | –   | –       | –     | VMware      |
| [25]  | Analytical & Regression based Modeling | X         | X         | –   | X       | X     | Xen         |
| [24]  | Analysis and Model Checker | X         | –         | –   | –       | –     | Xen         |
| [10]  | Analytical Modeling | X         | X         | –   | –       | –     | All         |
| [13]  | Cost Analysis      | X         | –         | –   | –       | X     | KVM         |
| [31]  | Cost Prediction    | X         | X         | –   | –       | X     | Xen         |
| [32]  | Cost Prediction    | X         | –         | –   | –       | X     | KVM         |
| [6]   | Cost Prediction    | –         | –         | X   | X       | X     | KVM         |
| [35]  | Prediction         | X         | –         | –   | –       | –     | VMware but not vMotion |
| [8]   | Prediction         | –         | –         | X   | X       | X     | Oracle Virtual Box |
| [23]  | Prediction         | X         | X         | X   | X       | –     | All         |
| [5]   | Prediction         | X         | X         | –   | –       | –     | Xen         |
| [29]  | Prediction         | X         | X         | –   | –       | –     | VMware/KVM |
| [21]  | Prediction         | X         | –         | X   | –       | –     | Xen         |
| [18]  | Markov Model Prediction | –         | –         | X   | –       | –     | CloudSim    |
| [26]  | Prediction         | X         | –         | –   | –       | –     | Xen         |
| [15]  | ML based Prediction | X         | –         | –   | X       | X     | VMware      |
| [11]  | Analytical Modeling | X         | X         | –   | X       | –     | All         |
| [20]  | Cost modeling      | –         | –         | –   | X       | –     | Xen         |
generator that access the Apache 2.2.8 Web server with MySQL database. In [12] 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 [25] 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 [7] 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 [24], 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 [10], 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 [13], 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.

In [15], we proposed a machine learning based cost prediction technique for live migration in VMware environments and in [16], we have proposed a novel timing optimization algorithm for VMs live migration that is based on [15] and on using datacenter network utilization prediction technique [16]. In this paper, we integrate the algorithm proposed in [16] as a plug-in with VMware vSphere UI that helps the IT admins for VMware clusters to predict live migration cost and to get a recommendation for the optimal timing to run live migration for the specified VM. For integration with VMware vSphere, JAVA, HTML, VMware PowerCLI and Python tools are used.
The rest of this paper is organized as following; in Sect. 2 we discuss the background on networking in VMware clusters and the research challenge to predict the optimal time for live migration. In Sect. 3, we present the timing optimization algorithm proposed in [16] to be integrated with VMware vSphere. In Sect. 4, we show the contribution of this paper which is the integration details of the timing optimization algorithm with VMware. This integration helps the IT admins to use the timing optimization feature for VMware environment as an example of a commonly used platform for virtual datacenter and cloud environments management portals. The integration results will be presented in Sect. 5 and then we conclude the paper in Sect. 6.

2 Background

As proposed in [16], live migration timing optimization depends on two main techniques; the first one is the cost prediction for live migration of the VMs [15], and the second approach is the datacenter network utilization prediction [16]. In this section, we present in depth the technical details of live migration configurations and virtual networking in VMware as a required background to show how live migration can impact the LAN or WAN scale networks throughput.

2.1 Live Migration Configurations

Live migration can be used also by different configurations; as shown in Fig. 1 that show the flexibility to live migrate a VM in LAN or WAN scale, with or without a share storage.

- 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 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 VMware Virtual Networking Configuration

Virtual networking in VMware is structured such that, each VM has one or more virtual Network Interface Cards (vNICs). Each vNIC at least one virtual port (vPort) and each vPort is assigned to a vSwitch. There are two types of vSwitch; a local switch inside the physical host only and a virtual Distributed Switch (vDS). The local vSwitch connects the VMs within this host, however the vDS connects between the VMs within this cluster. Each vSwitch has one or more uplink port which forwards and receives the traffic to and from a physical switch port.

Figure 2 shows an example of two physical hosts that are interconnected to a shared storage through a FC-SAN network and connected to the an IP network through an Ethernet switch. The solid lines show the physical connections and the dotted lines show the virtual connections for the virtual distributed switch. As shown in Fig. 2 a storage array is shared between the cluster servers using FC-SAN network; which is a common configuration in datacenters. Live migration utilizes the TCP/IP protocol and so it uses the IP network. From best practice point of view, each VM should have at least 2 NICs and each physical host should have at least 2 physical NICs [2]. A virtual distributed switch connects between the VMs in the cluster are connected. Using port groups, the IO traffic of the VMs can be isolated. There are two types of port groups in VMware; VM network distributed port group and VMkernel distributed port group. VM network port group manages the production traffic of the applications. VMkernel port group manages the special classes of traffic such as management, vMotion, NFS, Fault tolerance, iSCSI, replication traffic and VMware vSAN as a Software Defined Storage (SDS) traffic [3].

Physical machines NICs ports are represented to the distributed switch as uplink ports; which 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. An uplink port can be shared between one or more port groups. For vMotion traffic, it is recommended to create a separate VMkernel port group across the VMs in the cluster. This vMotion port group must have one or more uplink ports from each physical host [2]. This uplink port function is not only for vMotion port group, but can be also for other VMkernel port group. From physical port point of view, vMotion traffic is physically isolated on the server port level abort from the applications traffic. However, it depends on the back-end networking topology, as vMotion and workload traffic might share and compete on the back-end network bandwidth.

Fig. 2. Network topology for VMware vMotion [16].

3 Cost Prediction Algorithm and Timing Optimization

3.1 The Research Problem

For the best of our knowledge, the IT admins are currently running the live migrations of the VMs without an estimation about the migration cost and with no idea if the migration start time that they use is actually the optimal time or not. Missing this information might lead to resources bottlenecks, more migration cost and migration failures. So, the problem that we solve in this paper is how to help the IT admins to know the optimal timing for a VM live migration and to have a prediction about this migration cost.

In order to resolve this research problem, we have proposed the machine learning based cost prediction approach in [15] and the live migration timing optimization algorithm in [16].

In this paper, we extend on the previously achieved work and show how to integrate these algorithms with VMware vSphere UI as an added plug-in that
can be optionally used before a live migration request. This is to prove that the cost prediction and the timing optimizing techniques proposed in [15] and [16] can be practically implemented and used by VMware clusters’ admins.

3.2 Cost Prediction Algorithm

From these papers, the following empirical models could be proposed for live migration time, data rate and power consumption after applying the regression techniques:

- The relation between the network rate and the active memory size can be modelled as an exponential relation; as shown in Eq. (1).

\[ R_s = \alpha e^{V_{mem}} + \beta \]  

\( R_s \): is the source host network overhead in kBps.  
\( V_{mem} \): is the active memory size in kB of the source host when the live migration should start.  
\( \alpha \) and \( \beta \): are the equation constants. From Eq. (1).

- Migration Time: A linear relationship is obtained between the migration time and the division of the memory size over the transmission rate; as represented in Eq. (2).

\[ T_{mig} = a \left( \frac{V_{mem}}{R_s} \right) + b \]  

\( T_{mig} \) is the duration of the migration time in seconds.  
a and \( b \) are the equation constants.

- Peak power consumption overhead has linear relation with the transmission rate; as represented in Eq. (3).

\[ P_{mig} = \frac{dE_{mig}}{dt} = c \frac{dV_{mig}}{dt} = c R_s \]  

\( P_{mig} \) is the peak power overhead in Watt, and \( c \) is constant. From Eq. (3).

In our previous papers, the above models could be used for cost analysis but not for cost prediction. This is because of the equations constants. These constants depend on the cluster hardware configuration like CPU specs, so they change from a cluster environment to another. So in order to determine these constants and achieve higher accuracy in cost prediction, we propose a machine learning framework to predict the live migration cost. the above models could be used for cost analysis but not for cost prediction. This is because of the equations constants. These constants depend on the cluster hardware configuration like CPU specs, so they change from a cluster environment to another. So in order to determine these constants and achieve higher accuracy in cost prediction, we propose a machine learning framework to predict the live migration cost.

The cost prediction algorithm is as proposed in Fig. 3. The training phase starts when the VMware PowerCLI script connects to the cluster vCenter Server.
Appliance (vCSA). Then data collection starts with listing all the events happened in the cluster during the last 12 h. This 12 h cycle can be changed based on the cluster admin preference. From the collected events, vMotion events are filtered out. These vMotion events details like the source host, target host and time stamp are captured. Then the script calculates the complete and start time differences in order to get the migration time of each vMotion request. The performance logs of vCSA are collected at the start and the completion times at the vMotion events in order to get the active memory size of the migrated VMs in kB, the network overhead in kBps and the peak power change in Watt.

From the above data of each vMotion event, we use the regression models in Eqs. (1–3) to calculate the equations constants after doing several substitution and considering the minimum Root Mean Square Error (RMSE); Eq. (4).

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (d_i - f_i)^2}$$  \hspace{1cm} (4)
where $N$ is the number of sample points collected during the last 12 h. $d_i$ is the measured performance value and $f_i$ is the regression equation value.

If the change in all the constants value became greater than 10% of the last 12 h cycle, the script waits for more 12 h and run again to continue in the training phase. If these changes became less than 10% of the last 12 h, so we consider the training phase of this cluster is finished, and the script then moves to the prediction phase. The time consumed until reaching this 10% convergence depends on the changes that happen in the VMs active memory size; which depends on the running workload. This sequence of data collection and models training makes the algorithm can fit at any vCenter Server cluster and adapt its models based on the cluster configuration in order to provide cost prediction.

In the prediction phase when a vMotion request is sent by the cluster admin, the active memory size is captured by the script before proceeding with live migration. Once the active memory size is known, Eq. (1) is used to predict the source host network throughput. Then Eq. (2) is used to predict the migration time, and finally Eq. (3) is used to predict the peak power consumption. The prediction data is exported to a .csv file that the cluster admin can read, and decide to proceed with this migration or not.

### 3.3 Timing Optimization Algorithm

In [16], we proposed a live migration timing optimization technique that is presented in Fig. 4 flowchart. The algorithm starts with establishing the connection with VMware vCenter Server Appliance (vCSA) [4] using PowerCLI client [1]. The developed PowerCLI script runs on the VMware cluster that is management by this vCSA. The next step is to train the live migration cost model based on the past 12 h events; as discussed in Fig. 3. Network traffic utilization prediction model uses the Hidden Markov Model algorithm proposed in [36] such that every 30 s a sample is captured of the VMware VMkernel network traffic history of the past day. These 2880 points are used as the dataset training. By finishing this step, the training phase should be finished and the algorithm is ready to predict.

In the prediction phase, When a vMotion request is issued for a specific VM or for Multi-VMs migration, the migration traffic rate and migration time are predicted by calling the machine learning technique proposed in Fig. 3. In this step, the migration time and network rate are estimated. Then the cluster network utilization prediction; proposed in [36] is used to estimate the network traffic volume of the VMkernel network for every 30 s during the next 1 h. By finishing this step, the prediction phase of the network traffic volume, the live migration time and the migration transmission rate is finished and timing optimization check should start.
Fig. 4. Timing optimization approach for VM migration [16].

For optimal migration timing, the algorithm checks if the current time; when the vMotion request is received is a good time for initiating the vMotion process. For this check, the script runs Eq. (3) that estimates the network utilization during the estimated migration time interval.

\[ R_{busy} = \sum_{n=0}^{N_{mig}} \frac{V_n}{T_{mig}} \]  

\[ R_{Avail.} = BW - R_{busy} \]  

\( R_{busy} \) is the network traffic volume in bps that will be utilized by other VMkernel network traffic such as vSAN, management, etc.

\( n \) is the 30 s based sample number in integer.

\( N_{mig} \) is the last sample that ends with the estimated migration time.

\( V_n \) is the traffic volume prediction in bytes for each sample.

\( R_{Avail.} \) is the available throughput in bps for vMotion traffic.

\( BW \) is the VMkernel network bandwidth in bps.

The success check point in the flow chart verifies basically checks the below condition in Eq. (4)

\[ R_{Avail.} > R_s \times (1 + P_{Acc.}) \]  

Where \( P_{Acc.} \) is the prediction accuracy for the live migration network throughput. So Eq. (4) checks if the available network rate for VMkernel network \( R_{Avail.} \) can meet the estimated migration transmission rate requirement.
with considering the prediction accuracy that is mentioned in [15]. If this check-
point result is (Yes), live migration will start immediately. If the result is (No),
the algorithm starts a new phase which is finding the optimal time for the VMs
migration process initiation. In the case of (No), the algorithm checks for the
optimal timing during the next hour from network availability stand point. So
with 30s interval, Eq. (4) is applied for the next hour prediction samples. If
another optimal time is found, the network admin will be notified. If the admin
accepts, the VMs migration will be postponed to the new time automatically.
If the admin rejects the proposed new time recommendation, the migration will
be start immediately. In case of not finding a better time during the next hour,
the admin will be notified as well to request the migration again after 1 h. If the
admins accepts, the algorithm stops. If the admin rejects that, the migration
starts momentarily.

Figure 5 shows an example for a live migration process with and without
timing optimization. As shown, the migration time can be minimized by using
the timing optimization technique due to the higher transmission rate.

![Graph showing migration time decrease and rate increase due to optimal timing of live migration](image)

**Fig. 5.** Migration time decrease and rate increase due to optimal timing of live migra-
tion [16].

## 4 Integration with VMware vSphere

In this section, we present how the timing optimization algorithm proposed in
[16] could be integrated with VMware User Interface (UI) using the testing lab
discussed in Sect. 6. The integration with VMware UI as an extension work
on [16] shows that the proposed timing optimization feature can be practically
implemented and used by the IT admins.
VMware provides a software development kit for building plugins for the VMware vSphere client. The structure of the plugin is shown in Fig. 6 and it consists of the following components:

- HTML UI layer: this layer handles the way the plugin looks in the Web client. It allows adding menu options and navigation items.
- JAVA service layer: this layer is based on the Spring MVC and the OSGI framework, it represents the backend of the plugin. This layer communicates with the PowerCLI and Python modules to perform the coefficients calculation and provide the estimation.

**Fig. 6.** Integration with VMware plug-in components structure.

**Fig. 7.** Integration with VMware plugin data-flow.
Live Migration Timing Optimization Integration

- PowerCLI module: This component handles the live data collection through the PowerShell PowerCLI APIs. It collects data about the migrations that occurred in the last 12h. The collected data is then processed by the Python module and returned to the JAVA service layer.

- Python module: This component processes the data using the Algorithm described by this paper and outputs the value for the coefficients $\alpha$, $\beta$, $a$, $b$, and $c$ to predict the expected duration for the migration, the expected power consumption, and the expected network usage.

Based on the integration of the above platforms and tools with VMware and the algorithms charts in Fig. 3 and Fig. 4, the steps we have used to integrate the proposed timing optimization algorithm with VMware vSphere are:

1. A shown in Fig. 7 The user interacts with the vSphere client UI, which contains the HTML layer added for the plugin, and requests the predicted migration overhead for a VM. This step is possible because the vSphere client software development kit allows developers to alter the user interface of the vSphere client to add new custom features.

2. The JAVA service layer receives the request from the UI layer and then forwards the request to the PowerCLI module.

3. The PowerCLI module gathers all the events run in the vCenter Server during the last 12h and filter the vMotion events. The script identifies the source and target hosts of each vMotion event and the start and end time stamp of each migration. Based on that, the performance statistics of each migration is collected and passed to the next step in CSV format.

4. The Python module then takes in the collected data set and for each pair of data items it calculates the coefficients. Once the script finds Two successive coefficients with less than 10% difference in value it stops the search and returns this value in a CSV format to the PowerCLI module.

5. The equations from (1), (2), (3) are then used to predict the duration, power consumption, and required network bandwidth of the migration.

6. These results are returned to the JAVA service layer as a string of characters and then returned to The HTML UI layer after formatting the string and shown to the user as in Fig. 8.

In the next section, we show the result of using these scripting tools and following the above steps to integrate the cost prediction and timing optimization algorithms with VMware vSphere UI.
Fig. 8. Added icon: cost prediction plug-in.

Fig. 9. Added icon: optimal timing plug-in.

5 Testing Results

The result of the integration with VMware vSphere UI is as shown in Fig. 8 and Fig. 9. As shown in Fig. 8, this newly added plugin in the UI of the VMware vSphere allow getting insights about the migration time, network rate and power consumption before initiating a VM migration and by using the data collected from the live migration history within the VMware cluster and with monitoring the VM active memory size. This allow the datacenter admins to make educated decisions about VMware vMotion events before committing the migration.

After determining the expected network rate of a VM live migration and using this integration method, it is also possible to create a more sophisticated migration system using the Hidden Markov model [36] in Fig. 4. Such a system
would predict how the network bandwidth will change in the future and thus inform the datacenter administrator of the optimal time to do the migration in order to decrease the load on the network infrastructure. Figure 9 shows how this system would look to the administrator. To implement such a system the following changes would be made:

- The PowerCLI module needs to periodically collect network data to be used by the Python module for training.
- the Hidden Markov model needs be implemented in the Python module. This model will be able to predict the network state in the future and thus guess the best time for a certain VM to migrate.
- When the user requests a migration, the UI will show the user the best predicted time to do the migration after running the model from the previous step.

As a result of using live migration cost prediction and timing optimization techniques, the live migration time of the VMs can be saved by just shifting the migration start time of the same VM to the recommended optimal time. Table 2 show average and the maximum migration time increase without using timing optimization as normalized values versus using the timing optimization techniques. As shown in Table 2, for memory stress benchmark the average increase in the migration time without timing optimization is 1.45 versus with using timing optimization. This average increase is 1.26 for network stress applications. From the peak increase in the migration time point of view, the maximum migration time increase for memory stress benchmark without timing optimization is 2.05 versus with the timing optimization. This maximum increase is 1.36 for network stress benchmark. The results mentioned in Table 2 show how significantly the timing optimization can save the migration time cost for VMs live migrations.

|                  | Mem stress | Net stress |
|------------------|------------|------------|
| Average mig. time | 145        | 126        |
| Max. mig. time   | 205        | 136        |

6 Conclusion

Virtual datacenters and cloud computing environments depend on the live migration feature for dynamic management of the infrastructure compute resources, power saving and system load balance. It is common to find tens or may be hundreds of live migration events per day in enterprises’ datacenters. The cost of the live migration process includes the migration time, the network throughput overhead and the power consumption cost.
The challenge that the IT admins face is running the live migration sessions without an idea about the expected cost of each migration request of a VM. This might lead to resources bottlenecks, increase in the migration cost and higher failure probability of the VM copy, especially for large memory VM migration or for multiple concurrent VMs migrations.

In our last papers, we have proposed a machine learning based live migration cost prediction approach and a migration timing optimization technique that can resolve this challenge. In this paper, we provide an extension work on the previously proposed techniques in [16] and [15] and show a practical solution for this problem that could integrate the cost prediction and timing optimization algorithms with VMware vSphere UI; as an added plug-in to the UI. We show in this paper the building blocks of the software tools and platforms that should be used to achieve this integration and explain how this new feature work in the system back-end to facilitate this cost prediction and timing optimization as optional service for VMware clusters’ IT admins.

References

1. https://code.vmware.com/tool/vmware-powercli/6.5
2. 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
3. VMware VMkernel. https://docs.vmware.com/en/VMware-vSphere/6.7/com.vmware.vsphere.networking.doc/GUID-D4191320-209E-4CB5-A709-C8741E713348.html
4. www.vmware.com/products/vcenter-server.html
5. 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 2010, pp. 37–46. IEEE Computer Society, Washington, DC (2010). https://doi.org/10.1109/MASCOTS.2010.13
6. 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, 19–21 March 2018, pp. 384–391 (2018). https://doi.org/10.5220/0006682803840391
7. 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)
8. Berral, J.L., Gavalda, R., Torres, J.: Power-aware multi-data center management using machine learning. In: Proceedings of the 2013 42nd International Conference on Parallel Processing, ICPP 2013, pp. 858–867. IEEE Computer Society, Washington, DC (2013). https://doi.org/10.1109/ICPP.2013.102
9. Bezerra, P., Martins, G., Gomes, R., Cavalcante, F., Costa, A.: Evaluating live virtual machine migration overhead on client’s application perspective. In: 2017 International Conference on Information Networking (ICOIN), pp. 503–508, January 2017. https://doi.org/10.1109/ICOIN.2017.7899536
10. Cerroni, W.: Multiple virtual machine live migration in federated cloud systems. In: 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 25–30 (2014)

11. 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

12. 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)

13. Dargie, W.: Estimation of the cost of VM migration. In: 23rd International Conference on Computer Communication and Networks, ICCCN 2014, Shanghai, China, 4–7 August 2014, pp. 1–8. IEEE (2014). https://doi.org/10.1109/ICCCN.2014.6911756

14. 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, August 2014. https://doi.org/10.1109/FiCloud.2014.42

15. 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, 2–4 May 2019, pp. 456–463 (2019). https://doi.org/10.5220/0007749204560463

16. Elsaid, M.E., Abbas, H.M., Meinel, C.: Live migration timing optimization for VMware environments using machine learning techniques. In: Proceedings of the 10th International Conference on Cloud Computing and Services Science - Volume 1: CLOSER, pp. 91–102. INSTICC, SciTePress (2020). https://doi.org/10.5220/0009397300910102

17. Salfner, F., Tröger, T., Polze, A.: Downtime analysis of virtual machine live migration. In: The Fourth International Conference on Dependability (DEPEND 2011), France, pp. 100–105 (2011). ISBN 978-1-61208-149-6

18. 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, September 2013. https://doi.org/10.1109/seaa.2013.23

19. Hu, W., et al.: A quantitative study of virtual machine live migration. In: Proceedings of the 2013 ACM Cloud and Autonomic Computing Conference, CAC 2013, pp. 11:1–11:10. ACM, New York (2013). https://doi.org/10.1145/2494621.2494622

20. Huang, Q., Gao, F., Wang, R., Qi, Z.: Power consumption of virtual machine live migration in clouds. In: 2011 Third International Conference on Communications and Mobile Computing, pp. 122–125 (2011)

21. 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)

22. Jiang, X., Yan, F., Ye, K.: Performance influence of live migration on multi-tier workloads in virtualization environments. In: CLOUD 2012 (2012)

23. Jo, C., Cho, Y., Egger, B.: A machine learning approach to live migration modeling. In: Proceedings of the 2017 Symposium on Cloud Computing, SoCC 2017, pp. 351–364. ACM, New York (2017). https://doi.org/10.1145/3127479.3129262

24. 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)
25. Liu, H., Xu, C.Z., Jin, H., Gong, J., Liao, X.: Performance and energy modeling for live migration of virtual machines. In: HPDC (2011)
26. 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)
27. Rybina, K., Schill, A.: Estimating energy consumption during live migration of virtual machines. In: 2016 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), pp. 1–5 (2016)
28. Salfner, F., Tröger, P., Polze, A.: Downtime analysis of virtual machine live migration. In: The Fourth International Conference on Dependability, pp. 100–105. IARIA (2011)
29. Salfner, F., Tröger, P., Richly, M.: Dependable estimation of downtime for virtual machine live migration. Int. J. Adv. Syst. Meas. 5, 70–88 (2012). http://www. iariajournals.org/systems_and_measurements/tocv5n12.html
30. 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)
31. Strunk, A.: Costs of virtual machine live migration: a survey. In: Proceedings of the 2012 IEEE Eighth World Congress on Services, SERVICES 2012, pp. 323–329. IEEE Computer Society, Washington, DC (2012). https://doi.org/10.1109/SERVICES.2012.23
32. 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)
33. Voorsluys, W., Broberg, J., Venugopal, S., Buyya, R.: Cost of virtual machine live migration in clouds: a performance evaluation. In: Jaatun, M.G., Zhao, G., Rong, C. (eds.) CloudCom 2009. LNCS, vol. 5931, pp. 254–265. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10665-1_23
34. Wu, Y., Zhao, M.: Performance modeling of virtual machine live migration. In: 2011 IEEE 4th International Conference on Cloud Computing, pp. 492–499(2011)
35. 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 2007, pp. 5:1–5:8. ACM, New York (2007). https://doi.org/10.1145/1408654.1408659
36. Chen, Z., Wen, J., Geng, Y.: Predicting future traffic using hidden Markov models. In: 2016 IEEE 24th International Conference on Network Protocols (ICNP), pp. 1–6, November 2016. https://doi.org/10.1109/ICNP.2016.7785328