Check for updates

Hindawi Publishing Corporation
Journal of Engineering
Volume 2016, Article ID 3061674, 9 pages
http://dx.doi.org/10.1155/2016/3061674

Hindawi

Research Article

Machine Learning Based Statistical Prediction Model for Improving Performance of Live Virtual Machine Migration

Minal Patel,[1] Sanjay Chaudhary,[2] and Sanjay Garg[3]

[1]*Computer Engineering Department, A. D. Patel Institute of Technology, New Vallabh Vidhyanagar, Post Box 52, Vitthal Udyognagar, Anand District, Gujarat 388121, India*
[2]*Institute of Engineering & Technology (IET), Ahmedabad University, Ahmedabad, Gujarat, India*
[3]*Institute of Technology, Nirma University, Ahmedabad, Gujarat, India*

Correspondence should be addressed to Minal Patel; mppatel.adit@gmail.com

Received 20 November 2015; Revised 9 February 2016; Accepted 21 March 2016

Academic Editor: Christos Bouras

Copyright © 2016 Minal Patel et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Service can be delivered anywhere and anytime in cloud computing using virtualization. The main issue to handle virtualized resources is to balance ongoing workloads. The migration of virtual machines has two major techniques: (i) reducing dirty pages using CPU scheduling and (ii) compressing memory pages. The available techniques for live migration are not able to predict dirty pages in advance. In the proposed framework, time series based prediction techniques are developed using historical analysis of past data. The time series is generated with transferring of memory pages iteratively. Here, two different regression based models of time series are proposed. The first model is developed using statistical probability based regression model and it is based on ARIMA (autoregressive integrated moving average) model. The second one is developed using statistical learning based regression model and it uses SVR (support vector regression) model. These models are tested on real data set of Xen to compute downtime, total number of pages transferred, and total migration time. The ARIMA model is able to predict dirty pages with 91.74% accuracy and the SVR model is able to predict dirty pages with 94.61% accuracy that is higher than ARIMA.

1. Introduction

The market of cloud service is globally increasing that includes infrastructure services too. In the computer technology, infrastructure based computing is considered as abstractions of the hardware like server, storage and networks, and so forth. The user is able to use the infrastructure by renting instances of a virtual machine. The user is not allowed to manage underneath cloud infrastructure but has the control over operating system (OS) and storage and user can also deploy various applications. Virtualization is the key technology for cloud to run a virtual set of resources. In Xen, two major types of virtualization are available [1, 2]: (i) full virtualization and (ii) paravirtualization. Live migration is the procedure to migrate VM based on computation algorithm using either precopy approach or postcopy approach. Virtual machine monitor (VMM) program is able to perform live migration on the organization's public, private, and hybrid cloud [1].

In this paper, live migration of VMs is configured using memory based migration that can transfer states of memory during iterative process. The precopy is the most feasible and robust approach used for migration of VM in different environments [3, 4]. Performance of live migration is measured by two main key parameters: downtime (time for which a service is interrupted) and total migration period (time used to copy VM from a source to the desired destination).

The precopy based live migration has been successfully applied using various techniques including compression, CPU scheduling, and time series based analysis. In our proposed work, precopy algorithm is adopted to handle migration of dirty pages using time series predictions models. Forecasting analysis is performed based on past and future values using time series. Time series modeling and forecasting are essential for practical applications. Stochastic based models are the basic models, which are used with time series analysis [5]. These models are being applied to calculate the accuracy of time series for forecasting analysis

to solve real world problems. Hidden Markov and ARIMA models are those statistical models used to solve problems based on probability. A hidden Markov model (HMM) is a statistical Markov model which uses a Markov process with unobserved states [5]. ARIMA model is produced from a modified version of an autoregressive moving average (ARMA) model [6, 7]. This model is classified by ARIMA (p, d, q), where p is known for autoregressive components, d is used for integrated components, and q is known for moving average components. ARIMA has extensive scope and the performance of this model is efficient compared to HMM [5]. In this paper, migration of VM is proposed using modified precopy algorithm based on two different regression models.

The learning based regression models are mainly based on linear discriminant analysis (LDA), neural networks (NNs), and support vector machine (SVM), which have their mechanism to work on time series based analysis [8]. LDA is a generalization of Fisher's linear discriminant. This method is used in various applications of statistics, pattern recognition, and machine learning to find a linear combination of features based on classification. Another regression based model, artificial neural networks (ANNs), has few models which are very promising to work on time series models. It is used for classification and regression based modeling. Most common models of ANNs in time series are multilayer perceptrons (MLP), time lagged neural networks (TLNNs), and seasonal artificial neural network (SANN) [5]. The other regression based model named SVM [9], developed by Vapnik, is a powerful technique for classification, regression, and outlier detection with an intuitive model representation. SVM has a library called LIBSVM, which is being used to get optimal solution based on machine learning techniques. SVM is successfully applied to solve various real world problems [10–13]. It has many new features and empirical performance compared to neural networks and LDA [5]. The proposed ARIMA model and proposed SVR model are evaluated on same date set with comparative analysis that has been discussed in Section 4.

The main objective of this paper is to forecast memory pages which get dirtied during ongoing iterations. The regressive analyses using statistical prediction model and learning based prediction model have the ability to get optimal accuracy for live migration system. We need to observe that statistical learning model has higher accuracy than statistical probability model.

The major contributions of our proposed live migration system are stated below:

(i) Live migration system is estimated with various performance metrics like migration time, service time, and total pages transferred. The migration cost will be estimated based on these parameters.

(ii) The framework is tested using following different types of VM workloads: (i) idle and kernel compilation workloads; (ii) web server, dynamic web server, and file system workloads; and (iii) stress test based resource allocation workload.

Section 2 provides the detailed analysis of live migration techniques. In Section 3, the framework of improved precopy algorithm of Xen is proposed. ARIMA model and SVR model are discussed briefly in Section 4. Finally, in Section 5, experiments and results are shown using these models.

2. Literature Survey

Live migration algorithms are surveyed in this section. Time series based live migration algorithms are also recently applied in this area. Statistical, probabilistic, and learning based regression models are new dimension to enhance live migration systems.

The analysis of different workloads is given in [14], which has been dealing with the methods of reducing dirty pages. Live migration is evaluated based on stop-and-copy condition of Xen. Xen's shadow page table mechanism is useful to manage WWS size that is used to track dirty pages statistics on different workloads. Downtime and total migration time are evaluated based on methods of stunning rouge processes and freeing page cache pages.

In [15], performance and energy modeling for live migration of virtual machines are discussed. The base model is derived to evaluate network traffic and downtime. Experimental results show that more than 90% prediction accuracy is achieved and migration cost is also reduced using this base model.

LRU algorithm [16] is performed on the process migration based writable working set prediction algorithm. The algorithm is evaluated on different workloads to compare results. The proposed algorithm is able to reduce total data transferred during migration and total migration time over Xen. Jin et al. [17] presented the MEMCOM algorithm, which provides efficient, stable virtual machine migration using dictionary based memory pages compression. The MEMCOM is able to minimize network traffic, service time, and total migration time compared to Xen.

Reuse distance algorithm [18] keeps the track of modified pages using two different arrays: to_reuse which is used to get the information regarding modified pages and reuse_d which is used for counting the reuse distance of VM's pages. Two types of memory based migration system are used for iterative process analysis: (i) generic process and (ii) memory intensive process. In generic process, workloads have been able to deal with those pages which are not being modified frequently. This algorithm also works efficiently with memory intensive applications. Cui and Song presented matrix bitmap algorithm [19], which works on the policy to predict next page based on bitmap structure. This structure will give the statistics for pages that are to be either sent or not sent in the current iteration. Modified pages can have large reuse distance, if VM's pages are changed sequentially. The survey of live virtual machine migration is shown in [20]. A new technique for efficient live migration of multiple virtual machines is developed based on queuing models [21].

Time series based precopy approach [22] and Kalman filter [23] are designed on past observations and prediction of future data. In these approaches, frequently updated pages are sent in the last round and data are being computed for specific time interval to check the state of memory pages.

3962, 2016, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2016/3061674 by Ahmedabad University, Wiley Online Library on [01/05/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

It has been observed that dirty pages analysis is the key issue to make effective migration. For this, following are the issues to handle dirty pages effectively at the time of migration process:

(i) Reducing dirty pages based on CPU scheduling.

(ii) Predicting dirty pages which are to be dirtied for consecutive iterations.

(iii) Compressing pages which are able to reduce network traffic.

Basic methods are based on reduction of dirty pages using CPU scheduling [14]. These methods have their bottleneck for the performance of live migration systems [16, 19].

Other methods of improved precopy algorithms [4, 19] are based on either LRU or compression algorithms which have their limitations to handle dirty pages.

Our proposed work is based on prediction using time series analysis given in [23–26]. These techniques are successfully applied to VM migration, including context based prediction (CBP) algorithm, hidden Markov model, and Kalman filter technique. The statistical prediction model (ARIMA model) and learning based prediction model (SVR model) are discussed in Section 4.

3. Improved Precopy Algorithm

In this section, the working of precopy algorithm has been described in detail. The precopy algorithm is modified by forecasting of pages that is known as improved precopy algorithm.

3.1. Proposed Framework. The migration of virtual machine can be considered to be consisting of two phases [14]. During the first phase, a certain number of iterations are taking place. The second phase is called service dead phase that suspends the virtual machine to copy all remaining dirty pages in the last iteration. After this phase, the migration process will be completed to start activities on migrated VM:

$$T_i = \frac{P_i}{D_i}, \tag{1}$$

where T_i represents time taken by iteration i, P_i represents pages dirtied in iteration i, and D_i represents page dirty rate:

$$T_m = \sum_{i=1}^{n} T_i + T_d, \tag{2}$$

where T_m represents total migration time, T_i represents time taken by iteration i, and T_d represents downtime, which is time taken during stop-and-copy phase [14].

Equations (1) and (2) represent migration time of single iteration and total migration time, respectively.

The proposed framework based on Xen 4.2 [2] is configured for the migration of virtual machines on Intel-VT architecture. This framework is extended with additional module called prediction module. As the pages are to be predicted iteratively, both the number of pages to transfer and network traffic can be reduced.

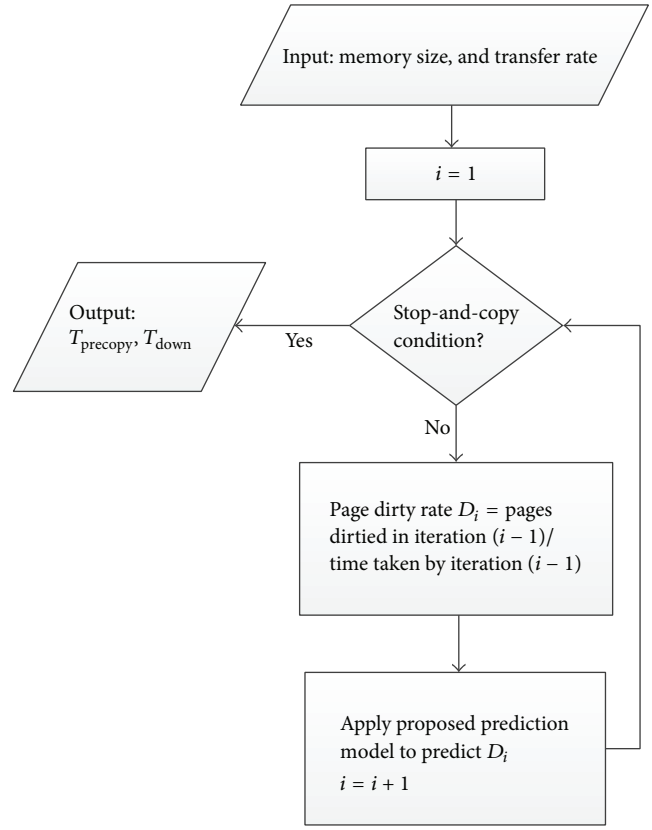


Figure 1: Improved precopy algorithm.

Stop-and-copy condition shown in Figure 1 is described below [27]:

(i) A maximum limit of iterations is 29.

(ii) A maximum limit on the amount of data to be migrated is three times the size of RAM.

(iii) The number of pages dirtied is less than a threshold value at the time of current iteration, which is given the value of 50.

(iv) The page dirty rate of the last iteration is greater than a threshold (in Mbps).

The precopy algorithm represented by [14] works as follows.

Until stop-and-copy condition is satisfied, migration process will calculate precopy time of each iteration given by (1). When stop-and-copy condition is false, total migration time will be calculated by (2). The performance of live virtual machine migration depends on the following factors: size of virtual machine, network bandwidth, and dirty rate of the application. Dirty rate and data rate are the two main parameters to improve the performance of any live migration system. In existing system of Xen, dirty pages are being transferred iteratively using simple LRU based technique which has its limitation to manage large reuse distance. Due

3962, 2016, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2016/3061674 by Ahmedabad University, Wiley Online Library on [01/05/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

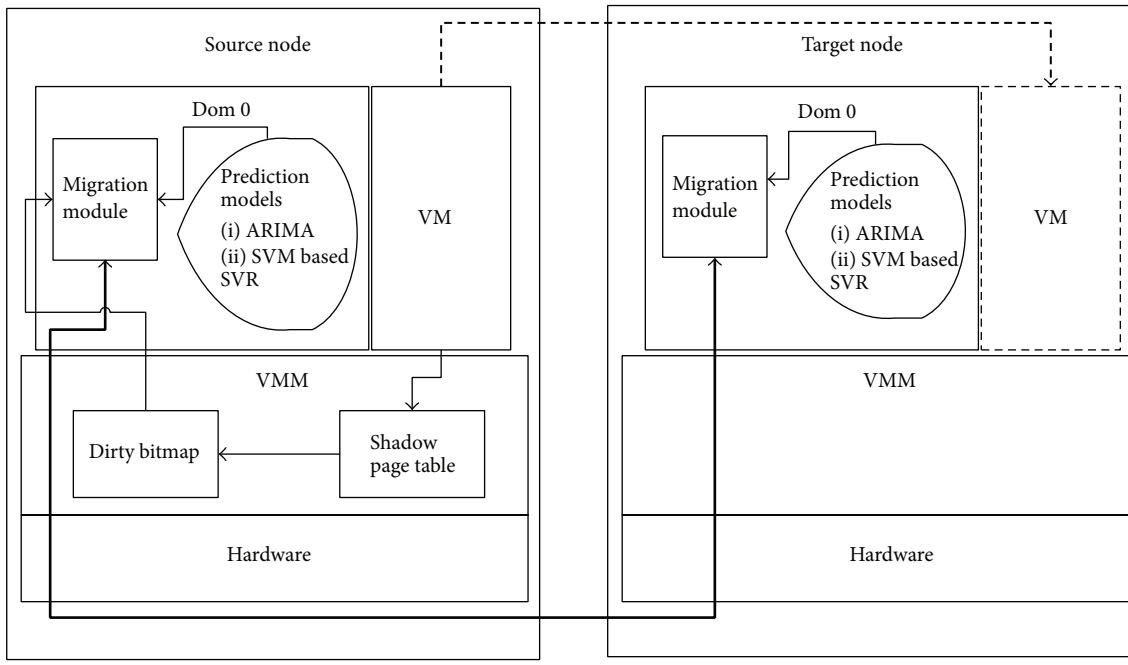


Figure 2: Framework of improved precopy algorithm.

to this mechanism, dirty pages are being sent repeatedly instead of sending them at last with their updated copy.

Figure 2 has been improved by us and the modification is described as follows.

As per precopy algorithm, updated memory pages of VM are identified by shadow page table and this process sets a flag in dirty bitmap [17]. At the beginning of each iteration, the related bit value is sent to the migration module (as shown in Figure 2). Bitmaps and shadow page table entries are cleared in each next round. In the modified precopy algorithm, characteristics of pages are being measured using historical analysis. Pages are predicted using regression based model. Those predicted pages, which are likely to become dirty pages, are not being sent during current iteration and, this way, remaining pages are iteratively sent to the destination until stop-and-copy condition becomes false as shown in Figure 1. In Figure 2, migration module presented in Dom 0 is the main component that is used to perform a live migration of VMs.

Prediction module is proposed in existing framework shown in Figure 2 with the following models: (i) ARIMA model and (ii) SVM based SVR model. The accuracy of these models are tested based on the prediction of dirty pages mechanisms.

4. Proposed Models Based on Statistical and Regression Techniques

Time series has attracted a research community for several decades due to its dynamic nature into modeling of data. It collects data based on past observations, which is able to show the real world working model of a series [6, 25]. Successful time series forecasting is the act of predicting future by analysis of past values. Due to the deterministic nature of Markov models and also limitations with regression models of LDA and neural networks (NNs), the new regression based models have been proposed here.

One of the most best-known stochastic time series models is ARIMA [6, 8, 25]. ARIMA is built by three submodels: autoregressive (AR), moving average (MA), and ARMA. It is also very useful for seasonal time series based forecasting. The other model which is SVM developed at AT&T Bell Laboratories in 1995 [9, 26, 28, 29]. It is also applied in many fields named regression, signal processing, estimation, and time series analysis. SVM is a practical approach for the structural risk minimization (SRM) principle, which has been shown to be superior to the conventional NN based learning algorithms [5].

4.1. ARIMA Model. ARIMA model is performed with time series having stationary data and it is also used to forecast the training data. Real data set [27] is used to evaluate performance parameters using ARIMA model.

4.1.1. Stationary Series Analysis and Autocorrelations. Time series [8] is defined as $x(t)$, where $x(t)$ is a random variable over a set of data points and t represents the time elapsed having values 0, 1, 2, . . ., and so forth. It can be designed based on single variable or multiple variables and also based on continuous or discrete signal. There are basic four components to design any ARIMA model: trend, cyclical, seasonal, and irregular patterns. In live migration, pages are iteratively copied so, based on cyclic nature, ARIMA model is designed. ARIMA has three phases [8]: (i) identification phase, (ii) estimation and testing phase, and (iii) forecast phase. In identification phase, stationary series is identified first. The parameters of time series are estimated using the suitable

3962, 2016, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2016/3061674 by Ahmedabad University, Wiley Online Library on [01/05/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

order of ARIMA (p, d, q) in second phase of model. ACF (autocorrelation function) and PACF (partial autocorrelation function) are prepared based on p, d, and q parameters.

The basic equations are as follows for ARIMA [8]:

$$\mathrm{AR}(p) : x_t = a_0 + a_1 x_{t-1} + a_2 x_{t-2} + \cdots + a_p x_{t-p} + e_t, \quad (3)$$

$$\mathrm{MA}(q) : x_t - a_0 = e_t + b_1 e_{t-1} + b_2 e_{t-2} + \cdots + b_q e_{t-q}, \quad (4)$$

where (3) suggests the basic AR term of ARIMA model and (4) represents computation of MA term.

4.1.2. Forecasting Analysis. Time series data are forecasted in third phase of ARIMA model. Forecast function is used for prediction of various model fitting functions. During ARIMA procedure, a number of models are selected based on lag values of AR and MA terms. These models are evaluated and final model is decided based on Akaike's Information Criterion (AIC) [8] parameter. The lowest AIC is considered to be closer to a real data. It is used to quantify the goodness of the statistical data.

Bitmaps to_send and to_skip of Xen are used to identify dirty pages. Dirty pages which have changed their state to be nondirty are being sent to the destination once they are found in to_send bitmap. The pages, which are frequently dirtied, called high dirty pages have to stay in WWS. AIC values of different ARIMA models are evaluated using R language [6]. Results are shown in Experiment 1.

4.1.3. Working of ARIMA Based Algorithm on Real Data Set. Live migration data are stored in matrix of 2400 (pages) × 30 (iterations). The program initially takes an input file in the CSV format to generate time series. The tseries and forecast packages of R language [30] are developed to assist the time series data. At the end, forecasting of time series is being evaluated using predict function of forecast collection. The algorithm based on ARIMA model is shown in the following part.

Dirty pages prediction algorithm based on ARIMA model is as follows:

Input: time series data of dirty pages.

Output: accuracy.

Read library (forecast).

Read library (tseries).

Reading real data set using read.csv.

Plotting time series and verifying whether the series is homogenous or not.

Taking diff(series) function for nonhomogenous series to get difference.

Applying adf.test to check stationary series.

Plotting acf(series) and pacf(series) using AR and MA functions.

Examining acf and pacf plots.

Applying suitable ARIMA models using arima() function to calculate AIC values.

Forecasting series using predict() function.

4.2. SVR Model. In this section, SVR model is applied for live migration of virtual machines. The model is used to evaluate accuracy on real data set [27]. ARIMA model has been designed based on universal approximation so it may have been lacking to generate higher accuracy on testing data set. For complex models, ARIMA model is quite difficult to design. The SVR can overcome these issues with introducing support vector based analysis.

4.2.1. Classification and Regression Analysis. The analytical process of SVM [9, 28] is identified based on the following parameters: linearly separable data, linearly nonseparable data, generalized optimal separating hyperplane, generalization in high dimensional space and kernel functions.

The SVM is used to find the globally optimal solution. It has a feature of generating independent results with irrespective input patterns and their size. In the design of SVM optimization problem, the distance is maximized between two boundaries to separate the classes. This distance is called the margin, which generates support vectors. The aim of SVM classification is to obtain a maximum margin. In nonlinearity, when it cannot find a linear separator, data points are projected into a higher-dimensional space where the data points effectively become linearly separable. Thus, the whole task can be formulated as a quadratic optimization problem, which has ability to solve the problem using known techniques [26, 28].

The labeling scheme of SVM is generated based on $f(x) = \mathrm{sign}(w^T x + b)$ since $f(x) = +1$ for all x above the boundary and $f(x) = -1$ for all x below the boundary. It can have two classes with lables 1 and −1. If any arbitrary point x_1 is picked up to the line $w^T x + b = -1$, then the closest point on line $w^T x + b = 1$ to x_1 can generate distance. Following are equations to represent SVM and SVR models, respectively:

$$\begin{aligned} &\min_{w,b,\xi,\rho} \quad \frac{1}{2} w^T w - \nu\rho + \frac{1}{l}\sum_{i=1}^{l} \xi_i \\ &\qquad y_i \left(w^T \phi(x_i) + b\right) \geq \rho - \xi_i, \\ &\qquad \xi_i \geq 0,\ i = 1, \ldots, l,\ \rho \geq 0, \\ &\min_{w,b,\xi,\xi^*} \quad \frac{1}{2} w^T w + C\sum_{i=1}^{l} \xi_i + C\sum_{i=1}^{l} \xi_i^* \\ &\qquad z_i - w^T \phi(x_i) - b \leq \epsilon + \xi_i, \\ &\qquad w^T \phi(x_i) + b - z_i \leq \epsilon + \xi_i^*, \\ &\qquad \xi_i, \xi_i^* \geq 0,\ i = 1, \ldots, l. \end{aligned} \quad (5)$$

4.2.2. SVM Kernels and Cross-Validation. SVM is treated with a vector of real numbers so if there are categorical attributes,

3962, 2016, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2016/3061674 by Ahmedabad University, Wiley Online Library on [01/05/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

they must be converted into numeric data. Selection of kernel is the key idea of solving any classification and regression problem. There are main four kernels: linear and polynomial and radial basis function (RBF) and sigmoid. The RBF kernel is the first choice because nonlinear behaviour of RBF maps samples into a higher-dimensional space. The linear kernel is a special case of RBF kernel. The sigmoid kernel behaves like RBF for certain parameters. The polynomial kernel has more hyperparameters than the RBF kernel.

There are two parameters for cross-validation in given kernels: C (cost) and γ. The values of C and γ are not known for a given problem. The goal is to identify optimal values for C and γ so that the classifier can accurately predict data for testing set. Using cross-validation, high training accuracy is being achieved based on parameters selection criteria [9].

4.2.3. Working of SVR Based Algorithm on Real Data Set. SVM is able to classify data into fixed region classes. The limitation of this model is that if data lies on the mixture point of two classes, then it is very difficult to classify a particular data point. In this case, regression based analysis is the better choice on classified regions to separate two classes. This concept of SVM is known as SVR that is statistical learning approach. The objective of this model is to make learning on real data to predict effectively on overlap regions. This model is also able to give accuracy in terms of weight of values if it falls nearer to any of the class. In this analysis, it is assumed that data falls in a particular class only. Our problem is focused with linear regressive analysis using this model. The algorithm based on SVR model is shown in the following part.

Dirty pages prediction algorithm based on SVR model is as follows:

Input: time series data of dirty pages.

Output: accuracy.

Reading e1071 package.

Reading real data set using read.csv.

Using data.frame function using input and output series data.

Using cross-validation to find best values of cost, gamma, and epsilon.

Applying RBF kernel.

Applying regression model using eps-regression type of SVM.

Forecasting series using predict() function.

It has also been seen that constant values of γ, cost, and epsilon are major challenges to generate accurate SVR based analysis. Real data contains the set of dirty pages and, based on this input data, the data frame is generated, which shares collection of variables as a data structure. After this, the model is designed and radial function is applied to it. R interface to LIBSVM is given in package e1071 [9, 28].

5. Experiments and Results

The hardware setup of our proposed work is given below.

The proposed framework is set up on Xen 4.2 for performing live migration of virtual machines. Distributed replicated block device (DRBD) has been used for storage on one of the two PCs. Physical resources of each PC has Intel Core i3 CPU 550 @ 3.20 GHz, 3.19 GHz, and 3.80 GB of RAM with 1 GBPS Ethernet switch. Host OS and guest OS are Ubuntu 12.04% LTS. The proposed framework is shown in Figure 2.

Total migration time, downtime, and forecasting dirty pages are performance parameters of our live migration system. Experiments 1 and 2 are shown using ARIMA model and SVR model, respectively, using real data set given in [27].

As per what is given in Section 3, the downtime is evaluated at stop-and-copy phase during last iteration and the total migration period is calculated on the total of precopy rounds and downtime.

The proposed framework is tested on the following application scenarios (VM workloads):

(1) Basic workloads: idle system and kernel compilation based workloads.

(2) Web server and file system workloads: web server workload, dynamic web server workload, and dbench workload.

(3) Resource allocation workloads: stress test based workload.

Service time and total migration time are calculated for the above three types of workload described in Experiments 3 and 4. The experiments are categorized into normal workloads and advanced workloads, which contain web server, file system, and resource allocation based workloads.

Experiment 1 (ARIMA model). The comparison of actual dirty pages and predicted dirty pages is shown in Figure 3. As per Section 4, ARIMA model can be estimated using p, d, and r values. The following models are evaluated: ARIMA(1, 1, 1), ARIMA(2, 1, 1), ARIMA(2, 1, 2), and ARIMA(5, 1, 1). ARIMA(5, 1, 1) model is optimal so the predicted values are taken from that model which is shown in Figure 3. Improved precopy algorithm using ARIMA is able to predict dirty pages with 91.74% accuracy.

Experiment 2 (SVR model). The prediction of dirty pages is generated using hybrid model. In this model, support vector based regression analysis is performed on real data set. Figure 4 shows actual pages with predicted pages using this model. The confusion matrix is given based on 200 instances of test data set which is shown in Table 1. This matrix is able to mitigate false prediction problem of ARIMA model, which has its limitation due to approximate model on given data. The accuracy of SVR model is 94.61% which is better than ARIMA.

Experiment 3 (total migration time for workloads). Normal and advanced workloads are considered here to evaluate the performance of improved_Xen. These workloads are

3962, 2016, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2016/3061674 by Ahmedabad University, Wiley Online Library on [01/05/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Table 1: Confusion matrix based on SVR model (200 instances of test data set).

	Total population	True condition	
		Condition positive	Condition negative
Predicted condition	Predicted condition positive	138	8
	Predicted condition negative	2	52

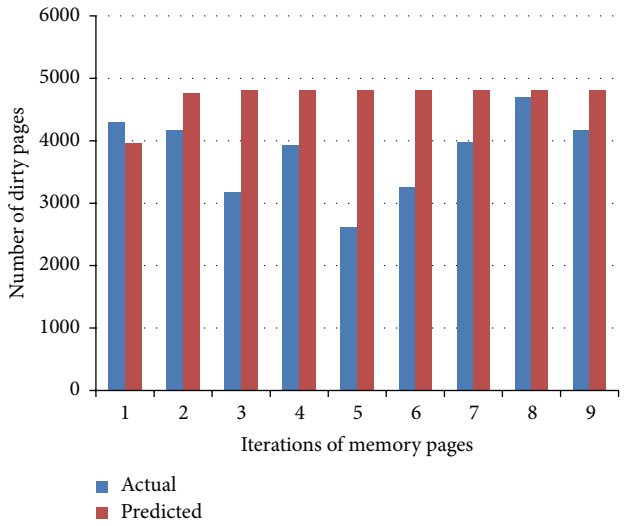


Figure 3: Comparison of dirty pages prediction using ARIMA model.

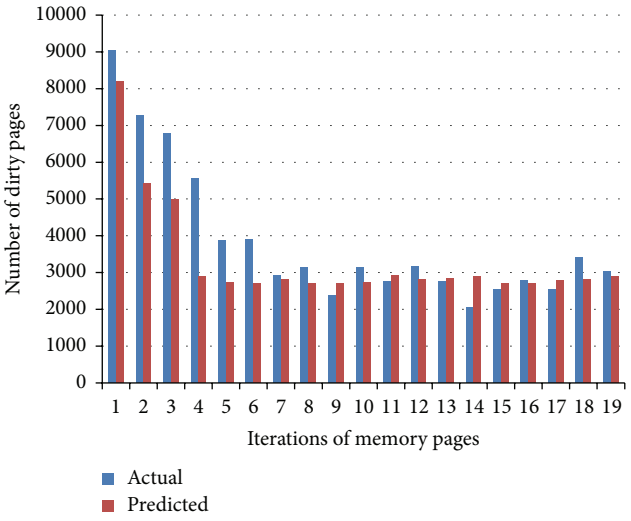


Figure 4: Comparison of dirty pages prediction using SVR model.

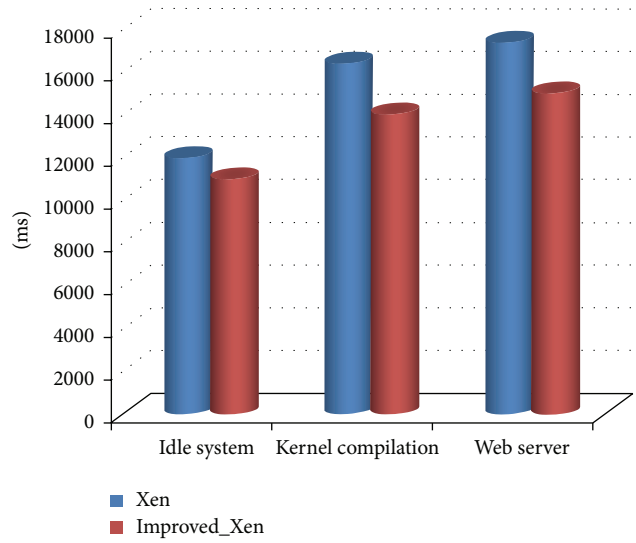


Figure 5: Total migration time for normal workloads.

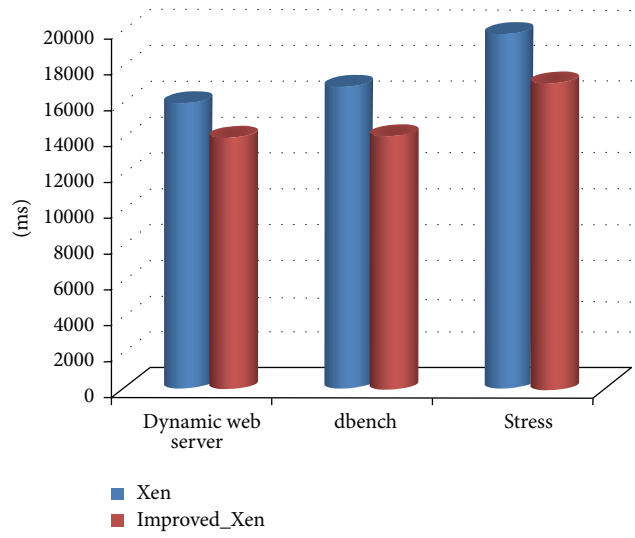


Figure 6: Total migration time for advanced workloads.

compared to existing Xen algorithm. According to the result of experiments described in Figures 5 and 6, when the VM is running with these workloads, the performance of total migration time for different workloads is improved in comparison to existing Xen.

Experiment 4 (downtime for workloads). Normal and advanced workloads are compared to existing Xen algorithm. According to the results of experiment in Figures 7 and 8, when the VM is running with these workloads, the performance of downtime for different workloads is improved compared to existing Xen.

In Table 2, proposed models are compared with existing models. The proposed SVR model is shown with the highest accuracy compared to other models.

3962, 2016, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2016/3061674 by Ahmedabad University, Wiley Online Library on [01/05/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

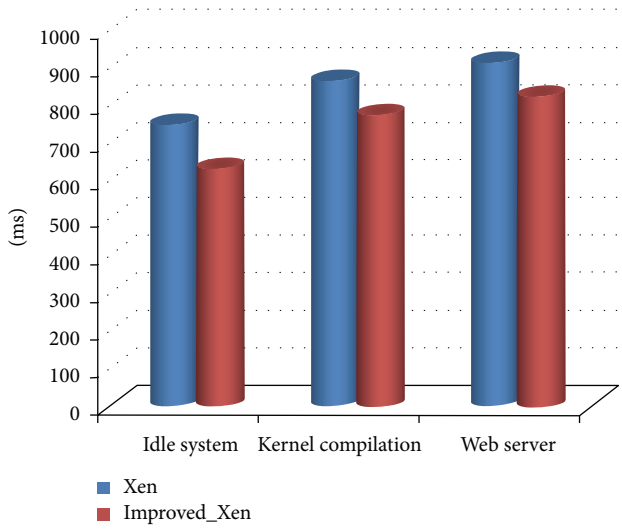


Figure 7: Downtime for normal workloads.

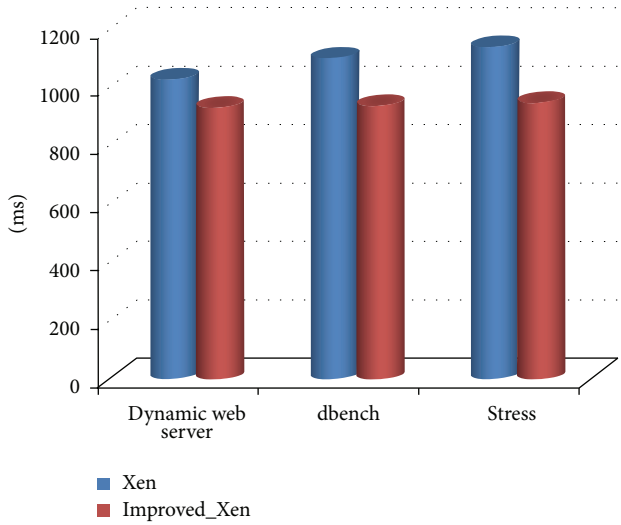


Figure 8: Downtime for advanced workloads.

Table 2: Accuracy comparison of different models.

Name of method	Accuracy
Base model [15]	81%
Simulation model [31]	90%
Refined model [15]	90.50%
Our ARIMA model	91.74%
Our SVR model	94.61%

The state-of-the-art time series models, ARIMA model and SVR model, have been presented.

(i) The improved precopy algorithm using ARIMA is able to predict dirty pages with 91.74% accuracy. At the same time, SVR model is able to predict dirty pages with 94.61% accuracy which is higher than ARIMA model. The confusion matrix is calculated based on 200 instances of test data set that is able to mitigate false prediction issues for dirty pages. Our evaluation process depends on various parameters to improve prediction accuracy. For ARIMA model, parameters are MSE, RMSE, and Box and Jenkins test and, for SVR model, parameters are cost, γ, and epsilon values.

(ii) The different performance metrics like service downtime, total migration time, and predicting dirty pages are compared, and these metrics are evaluated on different types of VM workloads: (i) normal workloads and (ii) advanced workloads.

6. Conclusion and Future Work

Two types of regression models have been evaluated: (i) ARIMA and (ii) SVR. ARIMA model is useful in wide variety of applications for prediction of data. It has strength to produce optimal results based on estimation of time series data. The limitation of this model is that it has no defined rules to justify fitted model for solving optimization problem. SVR model has its wide scope in almost all the fields for forecasting and classification based optimization problems. It is able to design regressive model to classify accurately the data based on learning mechanism. The overall idea of SVM based SVR model is very effective compared to other existing available models of machine learning and soft computing, but sometimes this model can be overtrained or undertrained, and for this reason it is very much essential to tune the specific parameters before applying this model.

Normal workloads and advanced workloads are evaluated based on ARIMA and SVR model. In the future work, optimization of live migration can be extended for other regression based models. Different prediction techniques can be tested on hypervisors to improve the performance of live migration system.

Competing Interests

The authors declare that they have no competing interests.

References

[1] C. Min, I. Kim, T. Kim, and Y. I. Eom, "VMMB: virtual machine memory balancing for unmodified operating systems," *Journal of Grid Computing*, vol. 10, no. 1, pp. 69–84, 2012.

[2] P. Barham, B. Dragovic, K. Fraser et al., "Xen and the art of virtualization," in *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, pp. 164–177, ACM Press, New York, NY, USA, 2003.

[3] A. Shribman and B. Hudzia, "Pre-copy and post-copy VM live migration for memory intensive applications," in *Euro-Par 2012: Parallel Processing Workshops*, vol. 7640 of *Lecture Notes in Computer Science*, pp. 539–547, Springer, Berlin, Germany, 2013.

[4] C.-H. Hsu, S.-J. Peng, T.-Y. Chan, K. Slagter, and Y.-C. Chung, "An adaptive pre-copy strategy for virtual machine live migration," in *Internet of Vehicles—Technologies and Services*, R. C.-H. Hsu and S. Wang, Eds., vol. 8662 of *Lecture Notes in Computer Science*, pp. 396–406, 2014.

3962, 2016, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2016/3061674 by Ahmedabad University, Wiley Online Library on [01/05/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

[5] R. Adhikari and R. K. Agrawal, *An Introductory Study on Time Series Modeling and Forecasting*, Lambert Academic Publishing (LAP), Saarbrücken, Germany, 2013.

[6] R. J. Hyndman and Y. Khandakar, "Automatic time series forecasting: the forecast package for R," *Journal of Statistical Software*, vol. 27, no. 3, pp. 1–22, 2008.

[7] K. Gairaa, A. Khellaf, Y. Messlem, and F. Chellali, "Estimation of the daily global solar radiation based on Box-Jenkins and ANN models: a combined approach," *Renewable and Sustainable Energy Reviews*, vol. 57, pp. 238–249, 2016.

[8] D. C. Montgomery, C. L. Jennings, and M. Kulahci, *Introduction to Time Series Analysis and Forecasting*, John Wiley & Sons, New York, NY, USA, 2008.

[9] D. Meyer, "Support vector machines. The Interface to libsvm in package e1071," *R News*, vol. 1, no. 3, 2014.

[10] P. J. García Nieto, E. García-Gonzalo, J. R. Alonso Fernández, and C. Díaz Muñiz, "A hybrid PSO optimized SVM-based model for predicting a successful growth cycle of the *Spirulina platensis* from raceway experiments data," *Journal of Computational and Applied Mathematics*, vol. 291, pp. 293–303, 2016.

[11] S. Raghavendra and P. C. Deka, "Support vector machine applications in the field of hydrology: a review," *Applied Soft Computing Journal*, vol. 19, pp. 372–386, 2014.

[12] B. Zheng, S. W. Myint, P. S. Thenkabail, and R. M. Aggarwal, "A support vector machine to identify irrigated crop types using time-series Landsat NDVI data," *International Journal of Applied Earth Observation and Geoinformation*, vol. 34, no. 1, pp. 103–112, 2015.

[13] W. Zhuo, Z. Cao, and Y. Xiao, "Cloud classification of ground-based images using texture-structure features," *Journal of Atmospheric and Oceanic Technology*, vol. 31, no. 1, pp. 79–92, 2014.

[14] C. Clark, K. Fraser, S. Hand et al., "Live migration of virtual machines," in *Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation (NSDI '05)*, vol. 2, pp. 273–286, USENIX Association, Boston, Mass, USA, 2005.

[15] H. Liu, C.-Z. Xu, H. Jin, J. Gong, and X. Liao, "Performance and energy modeling for live migration of virtual machines," in *Proceedings of the 20th ACM International Symposium on High-Performance Parallel and Distributed Computing (HPDC '11)*, pp. 171–181, San Jose, Calif, USA, June 2011.

[16] E. Phyu Zaw and N. Tar, "Improved live VM migration using LRU and splay tree algorithm," *International Journal of Computer Science and Telecommunicaitons*, vol. 3, no. 3, pp. 1–7, 2012.

[17] H. Jin, L. Deng, S. Wu, X. Shi, and X. Pan, "Live virtual machine migration with adaptive, memory compression," in *Proceedings of the IEEE International Conference on Cluster Computing and Workshops (CLUSTER '09)*, pp. 1–10, New Orleans, La, USA, August-September 2009.

[18] J. F. Alamdari and K. Zamanifar, "A reuse distance based pre-copy approach to improve live migration of virtual machines," in *Proceedings of the 2nd IEEE International Conference on Parallel, Distributed and Grid Computing (PDGC '12)*, pp. 551–556, IEEE, Solan, India, December 2012.

[19] W. Cui and M. Song, "Live memory migration with matrix bitmap algorithm," in *Proceedings of the IEEE 2nd Symposium on Web Society (SWS '10)*, pp. 277–281, Beijing, China, August 2010.

[20] R. W. Ahmad, A. Gani, S. H. A. Hamid, M. Shiraz, A. Yousafzai, and F. Xia, "A survey on virtual machine migration and server consolidation frameworks for cloud data centers," *Journal of Network and Computer Applications*, vol. 52, article no. 1345, pp. 11–25, 2015.

[21] G. Sun, D. Liao, V. Anand, D. Zhao, and H. Yu, "A new technique for efficient live migration of multiple virtual machines," *Future Generation Computer Systems*, vol. 55, pp. 74–86, 2016.

[22] B. Hu, Z. Lei, Y. Lei, D. Xu, and J. Li, "A time-series based precopy approach for live migration of virtual machines," in *Proceedings of the 17th IEEE International Conference on Parallel and Distributed Systems (ICPADS '11)*, pp. 947–952, Tainan, Taiwan, December 2011.

[23] Z. H. Sun and X. L. Hu, "Live migration for virtual machine based on kalman prediction of dirty pages," *Applied Mechanics and Materials*, vol. 668-669, pp. 1363–1367, 2014.

[24] C. Yong, L. Yusong, Y. Guo, L. Runzhi, and W. Zongmin, "Optimizing live migration of virtual machines with context based prediction algorithm," in *Proceedings of the International Workshop on Cloud Computing and Information Security (CCIS '13)*, October 2013.

[25] B. G. Tabachnick and L. S. Fidell, *Using Multivariate Statistics*, CourseSmart eTextbook, Pearson Publication, New York, NY, USA, 6th edition, 2013.

[26] P. Filzmoser, *Linear and Nonlinear Methods for Regression and Classication and Applications in R*, Department of Statistics and Probability Theory, Vienna University of Technology, Vienna, Austria, 2008.

[27] S. Nathan, U. Bellur, and P. Kulkarni, "Towards a comprehensive performance model of virtual machine live migration," in *Proceedings of the 6th ACM Symposium on Cloud Computing (SoCC '15)*, pp. 288–301, Kohala Coast, Hawaii, August 2015.

[28] C. W. Hsu, C. C. Chang, and C. J. Lin, "A practical guide to support vector classification," Tech. Rep., Department of Computer Science, National Taiwan University, 2003.

[29] A. Farag and R. M. Mohamed, "Regression using support vector machines: basic foundations," Tech. Rep., Computer Vision and Image Processing Laboratory, Electrical and Computer Engineering Department, University of Louisville, Louisville, Ky, USA, 2004.

[30] http://www.rstudio.com/products/rpackages/.

[31] S. Akoush, R. Sohan, A. Rice, A. W. Moore, and A. Hopper, "Predicting the performance of virtual machine migration," in *Proceedings of the 18th Annual IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS '10)*, pp. 37–46, Miami Beach, Fla, USA, August 2010.

3962, 2016, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2016/3061674 by Ahmedabad University, Wiley Online Library on [01/05/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License