Engineering with Computers
DOI 10.1007/s00366-016-0434-5

ORIGINAL ARTICLE

Prediction-based proactive load balancing approach through VM migration

Anju Bala[1] · Inderveer Chana[1]

Received: 7 September 2015 / Accepted: 5 January 2016

Abstract The ever-growing intricacy and dynamicity of Cloud Computing Systems has created a need for Proactive Load Balancing which is an effective approach to improve the scalability of today's Cloud services. In order to manage the load proactively on the Cloud system during application execution, load should be predicted through machine learning approaches and handled through VM migration approaches. Thus, this paper formulates an effort to focus on the research problem of designing a prediction-based approach for facilitating proactive load balancing through the prediction of multiple resource utilization parameters in Cloud. The involvement of this paper is twofold. Firstly, various machine learning approaches have been tested and compared for predicting host overutilization as well as underutilization. Secondly, the load prediction model having maximum accuracy from the tested models has been utilized for implementing the proactive VM migration using multiple resource utilization parameters. Further, the proposed technique has been validated through performance evaluation parameters using CloudSim and Weka toolkits. The simulation results clearly demonstrate that the proposed approach is effective for handling VM migration, reducing SLA Violations, VM migrations, execution mean and standard deviation time.

Keywords Cloud Computing · VM migration · Fault tolerance · Machine learning

✉ Anju Bala
anjubala@thapar.edu

Inderveer Chana
inderveer@thapar.edu

[1] Computer Science and Engineering Department, Thapar University, Patiala, Punjab, India

1 Introduction

Cloud Computing is a technology that uses the internet and central remote servers to maintain data and applications. This technology allows consumers and businesses to use applications without installation and access their personal files at any computer with internet access. Today, the computing world is swiftly renovating towards developing software using Cloud services with the required features like elasticity, virtualization, low cost, pay per usage etc. As Cloud provide various services to the user like Software as a Service (SaaS), Hardware as a Service (HaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) [1]. Therefore, Cloud requires reliability, scalability, availability, robustness and high performance features.

Although Cloud has large amount of resources such as RAM, CPU, Bw, Disk storage etc., still, this paradigm has various open key challenges that need to be resolved such as autonomic load balancing, fault tolerance, dynamic scheduling, energy efficiency and big data analytics etc [2]. Autonomic load balancing and fault tolerance are one of the major issues in Cloud and have been addressed in this work. Although, various load balancing techniques have proposed by many of the authors to increase the scalability and availability of Cloud applications, but none of the authors have implemented prediction-based proactive load balancing technique which can autonomically handle the overutilized and underutilized hosts. Thus, proactive prediction-based load balancing approach is immensely required to boost the performance of today's Cloud services.

In order to predict a host for overutilization and underutilization of hardware resources such as RAM utilization, Bandwidth utilization, Disk storage and CPU utilization, it is usually adequate to simply employ a threshold on these parameters. The prediction of overloaded and underloaded host requires more sophisticated algorithms such as machine learning to predict the load from various hosts proactively. Further, after the prediction of underloaded and overloaded host, an effective virtual machine migration policy is also entailed to proactively shift the computation away from overloaded or underloaded suspicious machines to other working machines.

Hence, the main objective of the research work is to implement a prediction-based load balancing technique after examining the results of machine learning-based approaches along with VM migration policy. The machine learning algorithms such as K-Nearest Neighbors, Artificial Neural Network, Random Forest and Support Vector Machines have been tested to implement a prediction-based model for identifying overloading or underloading subscription of hosts. The machine learning approach with maximum accuracy and minimum error has been used to predict the load after deploying PlanetLab application in simulation environment. The data of resource utilization parameters has been generated for training and testing the prediction model by running Planetlab workload application at fixed intervals in CloudSim [3]. Proactive VM migration has also been implemented to migrate the overloaded as well as underloaded virtual machines automatically using inter-correlation coefficient approach through multiple resource utilization parameters such as CPU, RAM, Bw, Network I/O. Further, the effectiveness of the proposed approach has been validated using various evaluated metrics. We also show that our proposed approach is not only intelligent to make precise predictions but also skilled enough to balance the load proactively by considerably reducing the number of SLA violations, VM migrations, mean execution time and standard deviation time. The research motivation of our work staunched from the following.

1.1 Motivation for the work

- Most of the load balancing techniques are reactive and do not handle load proactively. Therefore, our research motivated towards proactive load balancing techniques.
- The current research work also aims at analyzing the problem of load prediction so that Cloud systems would be capable of making decisions intelligently by predicting the utilization of various parameters such as CPU utilization, RAM, Disk Storage and Bandwidth utilization.
- To detect host overloading and underloading, statistical techniques have been used till now, hence, through the implementation of machine learning approaches, prediction model can be implemented.
- Most of the existing works have been implemented VM migration approaches by considering one parameter only, therefore, our proposed approach will be enhanced for multiple parameters in Cloud Environment which has not been implemented by any of the researchers till now.
- The current research work also aims at analyzing the problem of load prediction so that Cloud systems would be capable of making decisions intelligently by predicting the utilization of various parameters such as CPU utilization, RAM, Disk Storage and Bandwidth utilization. Our proposed approach will increase the performance of Cloud services by minimizing SLA Violations, execution mean, number of VM migrations and standard deviation time.
- The paper has been organized as follows: Sect. 2 discusses the related work. Sect. 3 devotes to methodology and load prediction approaches used for proposed approach. Section 4 details the evaluation of various parameters. Section 5 presents the experimental results. Section 6 comprises a discussion on conclusion.

2 Related work

A very few of the authors have implemented prediction-based load balancing approaches in Cloud to handle the load proactively but some of the authors have utilized machine learning for fault prediction. Therefore, in this section, we make an extensive survey of the work related to predicting and detecting load through load balancing and VM migration approaches.

2.1 Load balancing approaches

Beloglazov et al. [4] have utilized linear regression for host overload and underload detection . Silva et al. [5] have proposed various heuristics for dynamic scaling which is useful to acquire the speedups in line. But the proposed heuristics have not being utilized to scale the load proactively. Then, Lim et al. [6] have proposed proportional thresholding which performs better than static threshold values and Caron et al. [7] also presented a report where they have proposed pattern matching algorithms which are useful for forecasting the resource utilization but the historical data is inefficient. Kupferman et al. [8] commended a set of scoring metrics to measure the effectiveness and efficiency of dynamic scaling algorithms in terms of availability and cost. The resource utilization prediction based on the past values would be useful to handle the load effectively. Catal and Diri [9] have presented a review that machine learning models have better prediction results than statistical models. Islam et al. [10] demonstrated in their work that the accuracy of ANN is better for the prediction of resource utilization in Cloud environment. Then, the similar work of

Kousiouris et al. [11] approximated resource provisioning with Artificial Neural Networks (ANN) in Cloud platforms and validated the enhanced accuracy for prediction through ANN. Ren et al. [12] have also utilized exponential smoothing forecasting methods for dynamic load balancing by resource utilization parameters. Further, the work of Aniello et al. [13] also presented various auto scaling techniques such as static threshold based, reinforcement learning, queuing theory and time series-based approach. They have suggested in their work that time series-based approaches such as regression, neural networks, SVM and KNN can be used for balancing the load proactively. Beloglazov et al. [4] have utilized linear regression for host overload and underload detection, but they have not employed machine learning approaches for prediction. Bala and Chana [14] have implemented failure prediction models through ANN, Naive Bayes, logistic regression and random forest. But they not employed for proactive load balancing through VM migration. Therefore, to the best of authors knowledge, none of the above discussed approaches have implementing prediction-based load balancing after testing the experimental results of machine learning approaches such as ANN, SVM, KNN and Random Forest and none of the authors have compared the performance of these approaches to find the optimal approach for prediction along with VM migration approaches which are discussed below.

2.2 VM migration approaches

Beloglazov et al. [4] have proposed VM selection policies like Minimum Migration Time (MMT), Maximum Correlation Coefficient (MC)and Random Choice (RC) policy. MMT has utilized to migrate the VM that takes minimum time for migration Whereas, Random Choice (RC) policy has used to select a VM according randomly and Maximum correlation policy [15] has used to find the correlation between virtual machines and CPU utilization. But they not utilize the inter-correlation between the overloaded machines on the same host. Further, the work of Bala and Chana [16] has extended MC for fault tolerance through autonomic VM migration and extended their work in [17] for multiple parameters with underloadedcoefficient approach. But they have not optimized the performance of Cloud host for underlaoded hosts in their work. Therefore, VM migration approach is the part of our previous work that has been extended for underutilized hosts also.

2.3 Our contributions

The general conclusion extracted from this study is that there is a need to develop an efficient load balancing approach which would be able to cope with balancing the load proactively. To persist this work, we propose prediction-based proactive load balancing approach using machine learning approaches which have been found suitable from the literature review discussed above. As proactive load balancing approach requires a prior knowledge of the resource utilization parameters, therefore, we have gathered the data of resource utilization parameters after executing PlanetLab application at different intervals in CloudSim. After preprocessing the data, machine learning approaches such as KNN, ANN, SVM and RF have been compared based on the evaluated performance metrics. The approach having maximum accuracy has been utilized as prediction-based approach. Secondly, the VM migration approach has been proposed to migrate the VMs on another hosts proactively by predicting overutilized and underutilized host by prediction-based approach. Furthermore, the experimental results clearly demonstrate the effective results by improving the performance in terms of accuracy and reducing execution time, Number of migrations and SLA violations.

3 Methodology

Figure 1 depicts the steps used to implement the proposed approach, these steps are defined below:

- Deploy the planetload workload application in Cloud environment and read the resource utilization parameters such as RAM, BW, CPU and Disk I/O values, store these values to a data file.
- Test various machine learning approaches such as ANN, KNN, SVM and RF using the stored data files in WEKA [18].
- Compare the accuracy of these models in terms of maximizing the accuracy and minimizing the error.
- Implement the prediction model which has maximum accuracy for prediction of load in Cloud environment.
- If the resource utilization is more than threshold value then it is being classified as overutilized host.
- If the host is overutilized then inter-correlation coefficient-based approach is used find which VM is suitable to migrate from overloaded host. if the inter-correlation factor is zero then VM is listed to Migrate list, otherwise put it into Ignore List.
- If the resource utilization is below than the threshold value, then it is being classified as underloaded host. The VM which has correlation factor equal to zero with resource utilization that is currently migrated to other host by keeping it in sleep mode. All the machines are being migrated one by one and hosts are switched off to reduce the power consumption.
- If current resource utilization is more than the minimum threshold and less than the maximum threshold then class is identified as none of these.

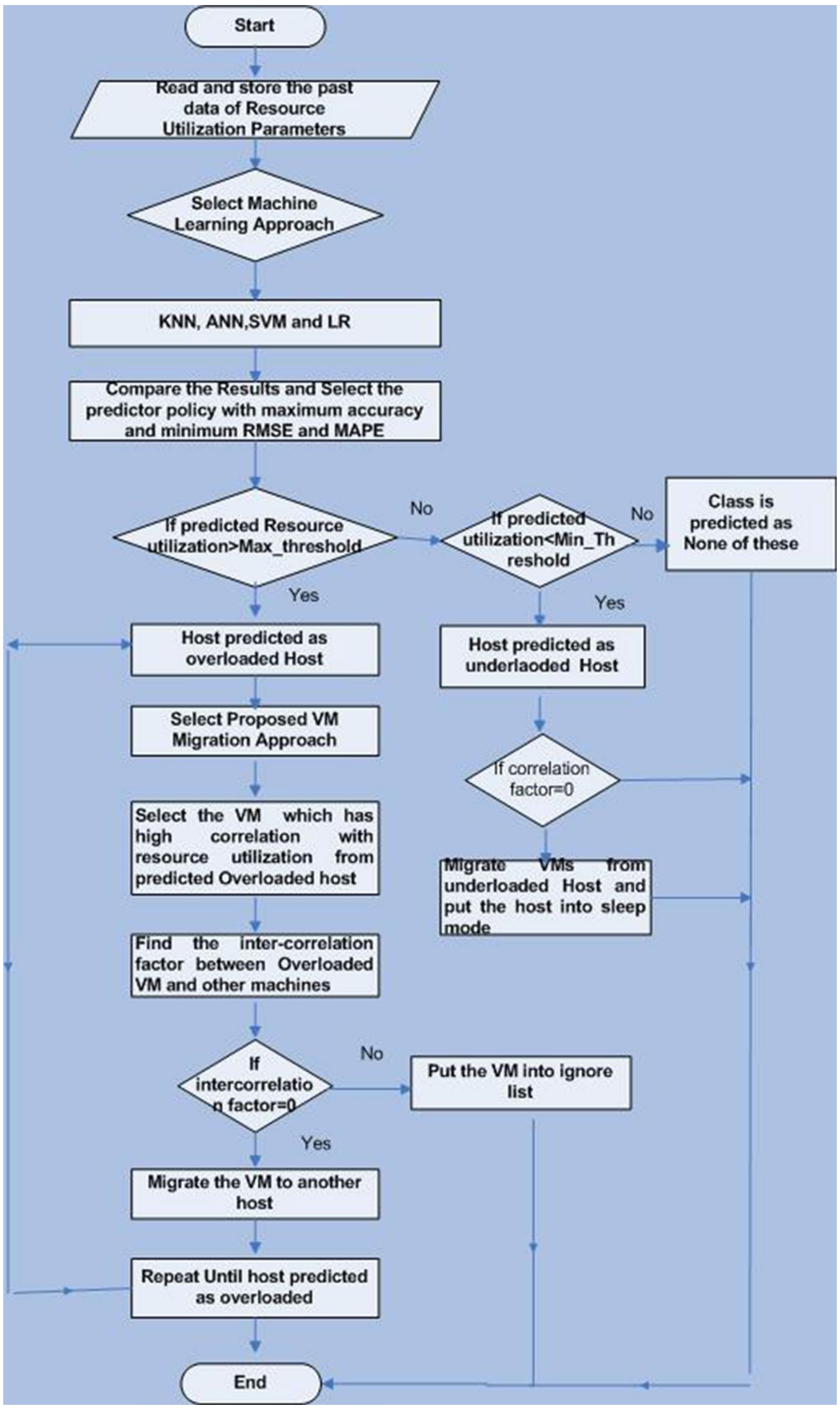

Start
Read and store the past data of Resource Utilization Parameters
Select Machine Learning Approach
KNN, ANN,SVM and LR
Compare the Results and Select the predictor policy with maximum accuracy and minimum RMSE and MAPE
If predicted Resource utilization>Max_threshold
No
If predicted utilization<Min_Threshold
No
Class is predicted as None of these
Yes
Yes
Host predicted as overloaded Host
Host predicted as underlaoded Host
Select Proposed VM Migration Approach
If correlation factor=0
Select the VM which has high correlation with resource utilization from predicted Overloaded host
Migrate VMs from underloaded Host and put the host into sleep mode
Find the inter-correlation factor between Overloaded VM and other machines
If intercorrelation factor=0
No
Put the VM into ignore list
Yes
Migrate the VM to another host
Repeat Until host predicted as overloaded
End

Fig. 1 Flowchart for implementing the prediction-based approach

3.1 Approaches used for load prediction

As to predict the load intelligently, the following machine learning approaches have been employed.

- K-nearest neighbors (KNN)
- Load prediction neural network (LPNN)
- Random forest (RF)
- Support vector machines (SVM)

3.1.1 K-nearest neighbors(KNN)

K-nearest neighbor (KNN) is the most frequently used supervised machine learning technique and utilizes training data to classify unknown instances . The instances which are included in the training data set include the known class labels based on their similar properties. KNN is a non parametric method used for classification and the output is a class membership of various class labels. K-NN does not have any training data to do any classification, hence it is being deemed as a lazy learning algorithm. The pseudo code for KNN is written in Algorithm 1.

Algorithm 1: LP through KNN.

```
– Input{$(X_1, Y_1), \ldots, (X_m, Y_n)$},where $X_1, X_2, X_3, \ldots, X_m$ are data points and $Y_1, \ldots, Y_n$ are the classes where $Y_i \varepsilon \{-1, 0, 1\}$
– For each unknown instance $X_j$ from $j = 1$ to $m$
– Find $X_1, X_2, X_3, \ldots, X_m$ the $m$ most nearest instances using Euclidean distance metric from training instances $Y$ nearest to $X_i$
– If the data values > Max-threshold, Set the class label = 1 as Overutilized
– Else if the data values < Min-threshold, Set the class label = −1 as Underutilized
– Else 0 as Ok class
– End If
– End For
– Return class label
– The distance metric using Euclidean distance is shown as follows in Eq. (1).
```

$$d(X_i, X_j)^2 = \|X_i - X_j\|^2 \tag{1}$$

3.1.2 Load prediction neural network

Load Prediction Neural Network includes multiple layers such as one input layer, two hidden layers and one output layer as Fig. 2. Input layer has four input neurons, $A = [A1, A2, A3, A4]$ and output layer having three output neurons, $B = [B1, B2, B3]$ and two hidden layers having three hidden neurons, $H = [H1, H2, H3]$ in between them.

The neurons at each layer are associated to the neurons of the next layer including a weight W_i that is to be evaluated during training. With each training point A_i and W_i network computes the resultant output B_j. If B (actual output) and B_j (predicted output) are different, then the network weights have been modified using learning rate (p) = 0.4 and momentum (m) = 0.2. The sum of squared errors i.e. Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) is the performance measure that have used a gradient-descent technique to minimize the error and to reach the local optima. When the calculated error during all training data is adequately small then the network has reached at local optima. Thus, the resultant Artificial Neural Network with backpropogation assured to generate a best prediction output after observing the input data in real time. The proposed algorithm has been shown in Algorithm 2.

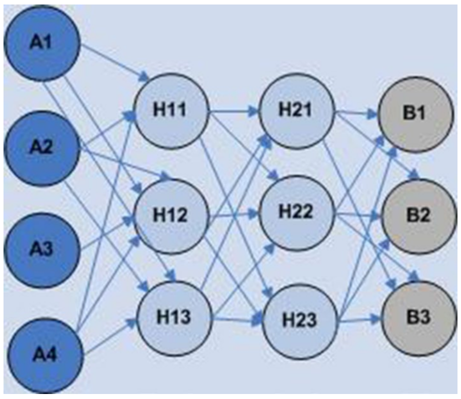


Fig. 2 Four layer neural network

Algorithm 2: LP with Backpropogation Learning

```
– Initialize network with random weight vector $W_i$
– For all training examples from $j = 1$to$m$, do
– For $i = 1$ to $n$ do
– Evaluate output $B_j = \sum_{i=1}^{n} A_i W_i$
– Compare network output with correct output $B$ with $B_j$
– Evaluate the error $E = B - B_j$
– Use gradient descent to minimize the error
– Adapt weights in current layer using $p = 0.4$ and $m = 0.2$
– Repeat until the RMSE and MAPE error has been minimized.
– EndFor
– EndFor
```

3.1.3 Random forest (RF)

Random Forest (RF) approach is being utilized to generate various trees and the random forest categorizes a new object from an input vector by penetrating the input vector on every tree in the forest. Each tree is used to cast a unit vote at the input vector with classification and forest selects the classification which have the maximum votes over all the trees in the forest [19] . Each Random tree is constructed with the following steps [14] :

- For N number of cases in training set,sample N cases randomly.
- For every node Y load predictors are randomly selected out of X input variables and $X << Y$ where $X = \sqrt{Y}$.

- Each Tree is created to the large extent with no pruning.

3.1.4 Support vector machines (SVM)

SVM is an efficient training algorithm used by many of authors for prediction. As the complexity of SVM is less, thus, it can accurately categorize the data which is linearly separable and also utilized for complex decision boundaries. The hyper-planes are identifies according to the classes as in Algorithm 3.

Algorithm 3: LP with SVM

- $\{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_N)\}$ where $X = (X_1, X_2 \ldots X_n)$ is the input vector and $Y_i \varepsilon \{1, 0, -1\}$ where 1 is for overutilized class, -1 for underutilized and 0 for None of these.
- SVM finds the linear separable function in the form of $f(x) = \langle WX + b \rangle$
- $Y_i = \{1 if \langle WX + b > 0 \rangle$
- -1 if $\langle WX + b < 0 \rangle$
- 0 if $\langle WX + b = 0 \rangle\}$
- Return class labels as Y_i

3.2 Proactive load balancing through proposed VM migration approach

Once, the host has predicted as overutilized or underutilized proactively before the occurrence of any fault from list of hosts, and then VMs need to be migrated from overloaded and underloaded host to other hosts. VM migration policy has been proposed to optimize the performance of data centers. Inter-correlation coefficient approach has been identified to enhance the performance of overloaded and underloaded hosts. If the host is predicted as overloaded then again the proposed approach has been used to migrate the VM from the overloaded host until the host is considered as being not overloaded. If the host is predicted as underutilized then all the machines migrates from the host and sleep down the host to reduce energy consumption.

3.2.1 Correlation coefficient approach

Inter-correlation coefficient approach has been used to find the association between resource utilization parameters (RAM, CPU, Bw, Disk) and hosts. If the host have resource utilization more than threshold then host would be predicted as over utilized and if the overall resource utilization is less than minimum threshold value then host would be predicted as underutilized through machine learning approaches. After the prediction of overutilized or underutilized hosts, VM migration approach has been extended by using inter-correlation with multiple parameters as shown Algorithm 4 and Algorithm 5.

- If the host is predicted as overloaded, then proposed algorithm selects VMs that have to be migrated from overloaded host.
- This algorithm first finds the correlation between the virtual machine VM and resource utilization, the VM which have maximum correlation with Resource utilization that indicates that VM can be consider for migration.
- Secondly, our proposed algorithms find the inter-correlation factor between the overloaded VM machines with other machines on the same host. The overloaded VM can work with other machines in a group on the same host. Therefore, if the inter-correlation factor is one then put it into the ignore list because these machines have strong association along with each other, otherwise, if the inter-correlation factor is zero that indicates these overloaded machines are independent of each other, then put that machine into the migrate list.
- If the host is predicted as underloaded, then VMs from the underloaded host migrates to another host if the correlation factor between VM and resource utilization is approximately zero.

Algorithm 4: Proactive Load Balancing Approach for Overutilized Hosts

```
Input: Host List, VM list
Output: Migration List
Foreach Host in HostList(j),j= 1 to m do
If the host is predicted as overloaded
Then Foreach VM in VMList(i), i=1 to n do
Find the correlation between VM(i) and Resource Utilization parameters
If correlation factor=1 then find inter- correlation of VM(i) with other VMs
If inter- correlation factor= 1 then put VM(i) into ignore list
Else put into migrate list
End If
End If
End For
End If
End For
Return Migration List of VM
```

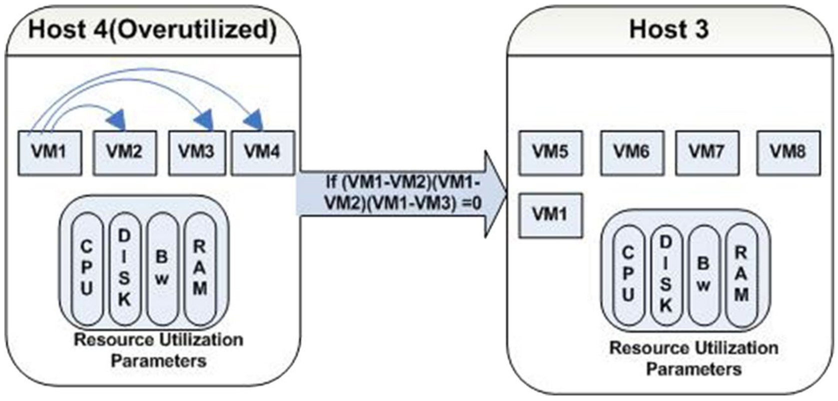

Host 4(Overutilized)
VM1
VM2
VM3
VM4
CPU
DISK
Bw
RAM
Resource Utilization Parameters
If (VM1-VM2)(VM1-VM2)(VM1-VM3) =0
Host 3
VM5
VM6
VM7
VM8
VM1
CPU
DISK
Bw
RAM
Resource Utilization Parameters

Fig. 3 Optimization of Cloud Host during resource overutilization

VM migration during host underutilization has been presented in Algorithm 5 that place the underloaded VMs to another host which is not underutilized.

- If the utilization of host is below than minimum threshold value, then host is predicted as underloaded then find the correlation between VMs and resource utilization parameters.
- Iif the correlation factor =0 then migrate the underloaded VM to another host which has predicted as overloaded, as the VMs have already migrated from overloaded host.
- Else migrate to another hosts which are not underutilized.
- If all the VMs have been migrated to another hosts, then switch off the underutilized host as to reduce the power consumption and SLAV violations.

```
Algorithm 5: **Proactive Load Balancing Approach for underutilized Hosts**
**Input**: Host List, VM list, **Output**: Migration List
**Foreach** Host from the HostList do
**If** the host is predicted as undreloaded
**Then** foreach VM in VMList(i), i=1 to n do
**Find** the correlation using Multiple parameters between VM(i) and Resource Utilization
**If** correlation factor=0 then place VM(i) into overloaded Host(j)
**Else** migrate to other hosts which are not predicted as overloaded
**End** if
**End** if
**Endfor**
**Switch** off the underloaded host
**Return** Migration List of VMs and Hosts
```

3.2.2 *Optimization of Cloud resources*

VM migration approach has been used to optimize various Cloud resources by proactively predicting the host as overloaded or underloaded. Therefore, VM should be migrated based on the utilization of various Cloud resources such as CPU, RAM, Bw and Disk Storage utilizations. If the Host 4 is predicted as overloaded, then VMs can be migrated from overloaded host using proposed approach. For example, there are four virtual machines $VM1$, $VM2$, $VM3$ and $VM4$ on the overloaded Host 4. Let $VM1$ have highest correlation factor with overall resource utilization which is calculated by the Eq. (2) where r is correlation coefficient, hence, $VM1$ is being considered as overloaded virtual machine from overloaded host. Then, inter-correlation factor can be calculated by using Eq. (3) between $VM1$ and other machines such as $VM2$, $VM3$ and $VM4$ on the same host.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2}$$

Then, the value can be computed using R^2_{1234} which is squared coefficient of correlation between predicted and practical variable. For example, VM1 is denoted as 1 supposed to be highly overloaded VM and finds the inter-correlation r with other machines VM2, VM3, VM4 ,here r_{12} indicates the correlation between VM1 and VM2 and r_{13} indicates the correlation between $VM1$ and $VM3$, similarly r_{14} indicates the correlation between $VM1$ and $VM4$,therefore R_2 has been calculated by using Eq. (3) correlation of $VM1$ with other machines such as $VM2$, $VM3$ and $VM4$ on the same host.

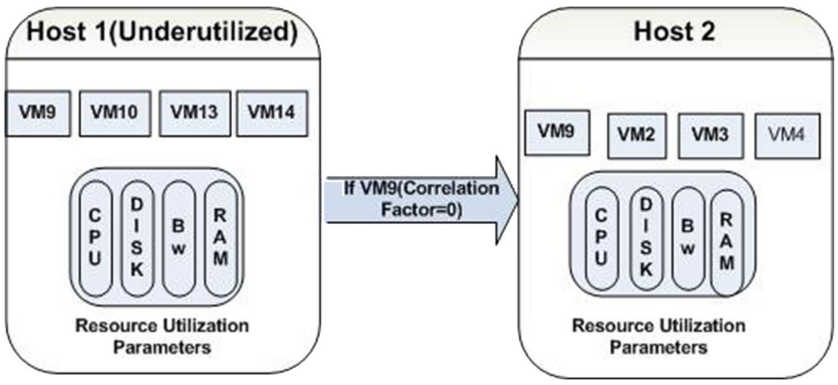

Host 1(Underutilized)
VM9
VM10
VM13
VM14
CPU
DISK
BW
RAM
Resource Utilization Parameters
If VM9(Correlation Factor=0)
Host 2
VM9
VM2
VM3
VM4
CPU
DISK
BW
RAM
Resource Utilization Parameters

Fig. 4 Optimization of Cloud Host during underutilization

$$R^2_{1234} = \frac{r^2_{12} + r^2_{13} + r^2_{14} - 2(r_{12})(r_{13})(r_{14})(r_{234})}{1 - r^2_{234}} \tag{3}$$

If $VM1$ is not associated with other machines , then it could be migrated to other host as shown in Fig. 3. It illustrates that Host 4 is predicted as overutilized as the overall resource utilization is more than maximum threshold value, therefore, proposed VM migration approach migrates the virtual machine $VM1$ which is suitable for migration to another Host 3 which has the utilization below the maximum threshold value.

Optimization of Cloud Host during Resource Underutilization: Figure 4 shows that host 1 is underutilized because the host has overall resource utilization is less than the minimum threshold. The correlation factor has been calculated between virtual machines and resource utilization, if it is zero VM has been migrated and placed to the overutilized hosts which can reduces the overall utilization of overutilized VMs. If $VM9$ is predicted as underutilized virtual machine, then $VM9$ needs to be migrated from Host 1 to another Host 2 which is not overutilized. After migrating all other machines, host 1 has changed to the sleep mode as to reduce the idle power consumption.

4 Evaluation criteria

The objectives of the paper are twofold. Firstly, it needs to correctly identify which hosts are in the classes of Overutilized , Underutilized or Ok state which intends to predicts the overloaded as well as underloaded hosts proactively through machine learning approaches. After predicting the overutilized and underutiized hosts, VM migration policy is also proposed. Hence, different performance measures are used to calculate the performance of load predictor and the VM migration approach.

4.1 Predictor performance

For predicting the performance of load prediction model, data of Resource utilization parameters have collected after every 10 minutes in Planet Lab's VMs by changing the threshold limits of resource utilization parameters. CloudSim classes have been used to read and trace these values after a fixed interval those workload traces. Then, prediction accuracy of ANN, KNN, SVM and RF has been compared to evaluate the best prediction model using the performance metrics which have been defined below:

4.1.1 *Sensitivity and specificity*

Sensitivity specifies the percentage of actual overloaded or underloaded host which are correctly classified whereas Specificity is the amount of hosts which are correctly identified as not overutilized as well as underutilized [9]. The measures are calculated using the formulas as given below in Eq. (4) where FNR is false negative rate and FPR is the false positive rate [20].

$$Sensitivity = 1 - FNR, Specificity = 1 - FPR \tag{4}$$

4.1.2 *RMSE and MAPE*

In our experiments MAPE and RMSE are evaluated to measure error in percentage. MAPE is used for evaluating the prediction accuracy [21] as percentage. Where y_j is the actual output, $\widehat{y}_j$ is the predicted output and m indicates total number of observations in the dataset and MAPE is defined in Eq. (5) and lower value of MAPE indicates a more precise prediction technique [14] .

$$MAPE = \frac{1}{n}\sum_{j=1}^{n} \frac{|\widehat{y}_j - y_j|}{y_j} \tag{5}$$

The metric RMSE [22] is described by the following formula in the Eq. (6) and smaller value of RMSE signifies a more effective prediction scheme.

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(\widehat{y}_j - y_j)^2} \tag{6}$$

4.1.3 F-Measure and ROC

The F-Measure values are computed as a harmonic mean of the precision and sensitivity as shown in Eq. (7).The highest value of F-Measure would be considered as the best case whereas lowest value is evaluated as worst case. TP is the true positive rate, FN is the false negative rate and FP is the false positive rate.

$$\mathrm{F-Measure} = \frac{2 * TP}{2TP + FN + FP} \tag{7}$$

The accuracy of prediction approaches can also estimated by comparing their ROC curves in graphical approach. The area under curve method is used to access the ROC curve which is shown in Eq. (8) where tpr specifies the true positive rate and fpr indicates false positive rate [22]. The maximum value of area under curve signifies the best predictor.

$$\text{Area Under Curve} = \int_0^1 tpr * (fpr)\mathrm{d}f \quad pr \in 0, 1 \tag{8}$$

4.1.4 Cross-validation

Cross-validation technique with tenfold has been used for predicting the accuracy of the load prediction models. The dataset is divided into 10 equal partitions whereas one partition is used for testing the proposed model and nine other partitions are utilized for training. The similar process repeats for all the 10 partitions.

4.2 Evaluation measures for autonomic VM migration approach

These are the evaluation measures which have been used to measure the performance of proposed prediction-based load balancing approach.

(1) *VM Execution Mean Time*: Reallocation Mean Time is average time which is denoted by x bar has been used to find the average time or reselecting or migrating the Virtual Machines from one host to another as shown in Eq. (9). Here n is the total number of calculations.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{9}$$

(2) *VM Execution Standard Deviation Time*: Standard Deviation measures how much Reallocation time deviates from the mean is denoted by s as shown in Eq. (10).

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{10}$$

(3) *Number of VM Migrations*: Number of VM migrations stipulates the count for how many times various VMs have migrated to other hosts. For the efficient VM migration approach there should be minimum number of VM migrations.

(4) *SLAV*: The SLAV metric represents the CPU performance that has not been distributed to an application when requested that further resulting in performance degradation. For the efficient VM migration policy, SLAV should be minimum. It shows the agreement between the resource provider and consumers are violated due to requested resource utilization has not given to the provider. The SLA violation Time per Active Host (SLAVTH) shows the percentage of time during active hosts that are overutilized and the Performance Degradation due to Migrations (PDDTM) describes the overall performance degradation by VMs due to migrations. SLAVTH is defined by the Eq. (11) where n represents total number of hosts and T_{sj} indicates total time during host j has the utilization more than maximum threshold value that leads to further SLAV. T_{aj} describes total number of hosts j being currently in active state. PDDTM is defined in the Eq. (12) where m is the total number of virtual machines and P_{di} represents approximation of the performance degradation of VM$_i$ due to migrations. P_{ci} is the total capacity requested by VM$_i$ during runtime. Thus, the metric SLAV depends on the performance degradation due to overutilization of host and number of VM migrations which is described in the Eq. (13) [4].

$$SLAVTH = \frac{1}{n}\sum_{j=1}^{n}\frac{T_{sj}}{T_{aj}} \tag{11}$$

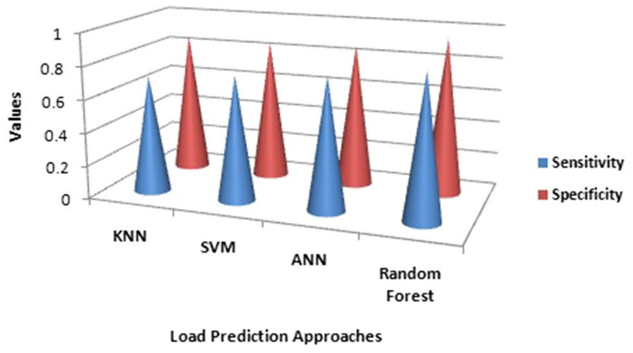
$$PDDTM = \frac{1}{m}\sum_{i=1}^{m}\frac{P_{di}}{P_{ci}} \tag{12}$$

$$SLAV = SLAVTH * PDDTM \tag{13}$$

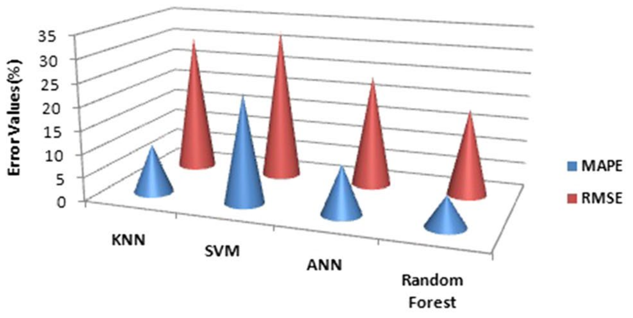
5 Experimental results and discussion

Experimental Results of performance prediction in the Cloud have been evaluated by implementing all of the approaches which have mentioned in Sects. 3.1 and 3.2. Our experimental setup has been categorized into two steps:

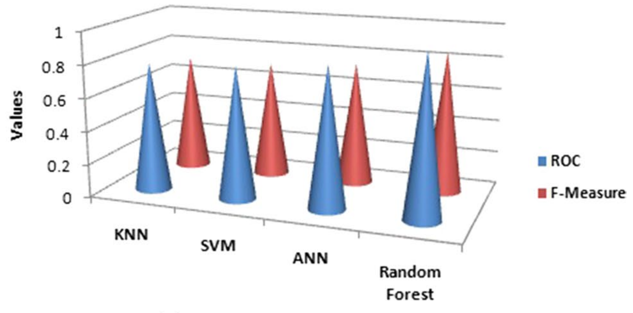
- Performance Evaluation of Load Prediction
- Performance Evaluation of Proactive Load Balancing through VM migration



(a) Sensitivity and Specificity



(b) MAPE and RMSE



(c) ROC and F-Measure

Fig. 5 Performance metrics for prediction-based load balancing approach through machine learning

5.1 Performance evaluation of load prediction

Figure 5 evaluates the accuracy of machine learning algorithms through various performance metrics as shown above in Sect. 4.1. As the results of Fig. 5a illustrates that the sensitivity and specificity of random forest is maximum (0.866, 0.95) where LPNN with two hidden layers has sensitivity and specificity values are (0.798,0.873) and KNN has minimum values among all the prediction approaches. Figure 5b depicts that the MAPE and RMSE using SVM is maximum whereas random forest has minimum values of error (6.58, 18.67 %). It can be seen in Fig. 5c that F-Measure and

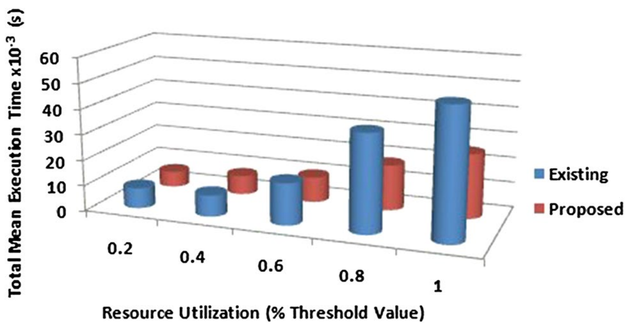


Fig. 6 Total mean execution time

ROC has utmost values (0.86 and 0.971) in Random Forest whereas LPNN with two hidden layers has lower values than random forest and higher than KNN and SVM. Hence, the experimental results evidently confirmed that Random Forest has highest prediction accuracy and minimum error rate and LPNN with two hidden layers renounce higher prediction accuracy of prediction than SVM and KNN. Hence, the load prediction model using random forest has been further implemented along with VM migration approach.

5.2 Performance evaluation of proactive load balancing through VM migration

To calculate the performance of the proposed algorithms on a large-scale virtualized datacenter is extremely difficult. Therefore, to evaluate the usefulness of the proposed approach, proposed proactive load balancing approach has been implemented and tested using CloudSim 4.0 which supports modeling and simulating one or more VMs on a data center [3]. We have simulated the data center that comprises 100 heterogeneous hosts and 100 Vms on each node. Half of the nodes are HP ProLiant ML110 G4 servers, and the half consists of HP ProLiant ML110 G5 servers.Every node have single core CPU with 16 GB of RAM, 10 GB/s network bandwidth and 1 TB of storage. We have evaluated our results by using different threshold values of resource utilizations. The experimental results have been compared with maximum correlation coefficient approach (existing) [4] and proposed approach through multiple parameters. The evaluated results show that the proposed proactive load balancing technique is better in terms of number of VM migrations, SLA violations, execution mean time and standard deviation time.

5.2.1 Result analysis

Several metrics have been used to evaluate the performance and efficiency of the proposed approach. The selection of performance parameters should be done in such a manner that must be able to find the efficiency of the algorithm in terms of time as to reduce the number of migrations as well SLA violations. The descriptive parameters have been used like mean, standard deviation that helps to understand how much average reallocation time it takes for the algorithm to handle load autonomically. Furthermore, to evaluate the performance of each VM migration metric, we have amended parameters threshold values from 0.2, 0.4, 0.6 and 0.8 as detailed below.

(i) *Total VM Reallocation Mean Time*: Figure 6 shows how much time it takes on an average for reallocating and migarting VMs. It can be inferred that the average time has improved due to the proposed approach as it is used for reallocating the machines based on finding the inter-correlation factor using multiple resource utilization parameters. For the maximum threshold value 1.0, the mean of execution time for the existing approach is 0.05093728 seconds whereas in case of the proposed approach, the value of the mean is 0.02531359 seconds which is appreciably smaller that the existing ones. Thus, the results of mean execution time corroborated that the proposed approach is better for reducing the mean execution time.

(ii) *Total Standard Deviation Mean Time*: For all the threshold values, the standard deviation is minimum for the proposed algorithm as shown in Fig. 7, which signifies that the proposed approach performs better during reallocation of VMs.

(iii) *SLA Violations*: Figure 8 shows that the SLA violations are significantly decreasing in the proposed approach as compared to the existing ones. There are 0.000157 % violations in the case of maximum correlation coefficient approach for threshold value 0.2 whereas for proposed approach there are 0.000113 violations. The SLA violations are appreciably increasing as threshold value raised and for the threshold value 0.8, it gives the best results for proposed approach with SLAV (0.000672) and for existing, the value of SLAV is 0.000514.

(iv) *Number of VM migrations*: Figure 9 depicts that there are minimum number of migrations in the proposed VM migration approach. For the threshold value of 0.2, the total number of VM migrations for the existing correlation coefficient approach is 515 whereas in the proposed approach the count is 427. Similarly, for all other threshold values, the number of migrations has reduced as the threshold value is increased by 0.2. For threshold value 1.0, there are maximum number of (5193) VM migrations in the existing approach while for the proposed approach, the number of migrations are 3860. Hence, the results depicted through the graph, demonstrate that the proposed approach is effective during overutilization of hosts.

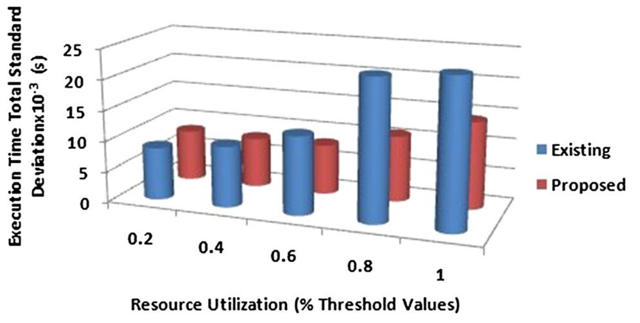


Fig. 7 Total standard deviation mean execution time

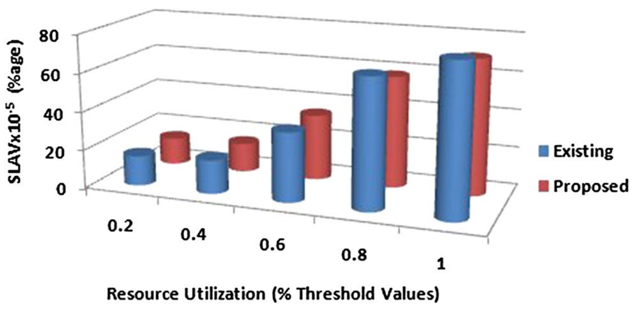


Fig. 8 SLA violations

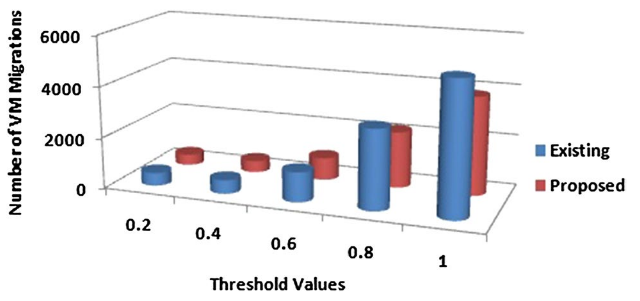


Fig. 9 Number of VM migrations

6 Conclusion

This paper provides an effective approach of load prediction model for resource utilization in order to ease load balancing autonomically in Cloud environment. The experimental results have compared for implementing load prediction model using machine learning approaches such as KNN, ANN, SVM and Random Forest and also proved that the accuracy of Random Forest is maximum as comparison to KNN, ANN and SVM. Then, the proposed model using Random Forest has implemented to predict the underutilized or overutilized hosts in Cloud environment. Proactive Load balancing approach through VM migration has also validated through CloudSim toolkit by considering multiple resource utilization parameters as to enhance the performance of VM migration policy. The experimental results have validated the superior effectiveness of the proposed approach by minimizing execution mean time, standard deviation time, SLAV and number of VM migrations which has not been implemented by any of the authors till now.

6.1 Future directions

The following future directions would be useful for researchers.

– Other real life scientific applications such as flood prediction, genomics and cyclone predictions would be considered in future as to increase the performance and prediction accuracy of these applications.
– The proposed prediction-based approach can be tested on Amazon EC2 and workload generators can also be used for generating the workload to test various applications such as Rain, SPEC, Faban, Rubis and Olio etc.
– The incorporation of prediction approaches with fault tolerant techniques would be able to enhance the efficiency of today's Cloud services.
– In future, our prediction approach can be implemented for workflow applications which can further be validated through real test bed such as Pegasus and hadoop.

References

1. Armbrust M, Fox A, Griffith R, Joseph AD, Katz R (2009) Above the Clouds: a Berkeley view of Cloud Computing. Commun ACM 53(4):1–25
2. Zhang Q, Cheng L, Boutaba R (2010) Cloud Computing: state-of-the-art and research challenges. J Internet Serv Appl 1:7–18
3. Calheiros RN, Ranjan R, Beloglazov A, Rose CAFD, Buyya R (2011) CloudSim: a toolkit for modelling and simulation of Cloud Computing environments and evaluation of resource provisioning algorithms. Softw Pract Exp 41:23–50
4. Beloglazov A, Buyya R (2012) Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers. J Concurr Comput Pract Exp 24:1397–1420
5. Silva JN, Veiga L, Ferreira P (2008) Heuristic for resources allocation on utility computing infrastructures. In: MGC08 proceedings of the 6th international workshop on middleware for grid computing. ACM, New York, NY, USA, p 16
6. Lim HC, Babu S, Chase JS, Parekh SS (2009) Automated control in Cloud Computing: challenges and opportunities. In: ACDC09: proceedings of the 1st workshop on automated control for datacenters and Clouds. ACM, New York, NY, USA, p 1318
7. Caron E, Desprez F, Muresan A (2010) Forecasting for Cloud Computing on-demand resources based on pattern matching. Technical Report, INRIA
8. Kupferman J, Silverman J, Jara P, Browne J (2010) Scaling into the Cloud. Advanced Operating System, p 1–10. http://cs.ucsb.edu/jkupferman/docs/ScalingIntoTheClouds.pdf
9. Catal C, Diri B (2009) A systematic review of software fault prediction studies. Expert Syst Appl 36:7346–7354
10. Islam S, Keunga J, Lee K, Liu A (2012) Empirical prediction models for adaptive resource provisioning in the cloud. Future Gener Comput Syst 28:155–162
11. Kousiouris G, Menychtasa A, Kyriazis D, Gogouvitis S, Varvarigou T (2014) Dynamic, behavioural-based estimation of resource provisioning based on high-level application terms in Cloud platforms. Future Gener Comput Syst 32:27–40
12. Ren X, Lin R, Zou H (2011) A dynamic load balancing strategy for Cloud Computing platform based on exponential smoothing forecast. In: Proceedings of IEEE CCIS2011, pp 220–224
13. Aniello L, Bonomi S, Lombardi F, Zelli A, Baldoni R (2014) An architecture for automatic scaling of replicated services. In: Noubir G, Raynal M (eds) Networked Systems. Springer, Lecture Notes in Computer Science, pp 122–137
14. Bala A, Chana I (2014) Intelligent failure prediction models for scientific workflows. Expert Syst Appl 42(3):980–989
15. Abdi H (2007) Multiple correlation coefficients. In: Salkind NJ (ed) Encyclopedia of measurement and statistics. Sage, Thousand Oaks, CA, USA, pp 648–651
16. Bala A, Chana I (2013) VM migration approach for autonomic fault tolerance in Cloud Computing. In: International conference of grid and Cloud applications GCA13. Las Vegas, USA, pp 3–10
17. Bala A, Chana I (2014) Autonomic fault tolerant scheduling approach for scientific workflows in Cloud Computing. Concurr Eng Res Appl Sage Publ (Accepted)
18. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. SIGKDDExplorations 11:10–18
19. Guo L, Ma Y, Cukic B, Singh H (2004) Robust prediction of fault-proneness by random forests. In: Proceedings of the 15th international symposium on software reliability engineering, pp 417–428
20. Salfner F, Lenk M, Malek M (2010) A survey of online failure prediction methods. ACM Comput Surv 42:1–42
21. Malhotra R, Jain A (2012) Fault prediction using statistical and machine learning methods for improving software quality. J Inf Process Syst 8:241–262
22. Aggarwal KK, Singh Y, Kaur A, Malhotra R (2009) Empirical analysis for investigating the effect of object-oriented metrics on fault proneness: a replicated case study. Softw Process Improv Pract 16:39–62