International Journal of GEOMATE, May, 2017, Vol. 12, Issue 33, pp. 139-146
Geotec., Const. Mat. & Env., ISSN:2186-2990, Japan, DOI: http://dx.doi.org/10.21660/2017.33.2854

RESOURCE USAGE PREDICTION BASED ON ARIMA-ARCH MODEL FOR VIRTUALIZED SERVER SYSTEM

*Biju R Mohan[1] and G Ram Mohana Reddy[2]

[1,2]Department of Information Technology,National Institute of Technology Karnataka.
Mangalore,Karnataka 575025, India

*Corresponding Author, Received: 13 Aug. 2016, Revised:29 Nov. 2016, Accepted: 25 Dec. 2016

ABSTRACT: Performance degradation is unavoidable in server systems and this is because of factors such as shrinkage of system resources, data corruption, and numerical error accumulation. The resource shrinkage leads to the system failure due to the error propagation. Thus the resource prediction is useful to the administrator of the system so that an accidental outage can be avoided. It has been observed in past that most of the failures occur due to the exhaustion of free physical memory, so here free physical memory of a server consolidation setup is observed. It is also found that most of the studies in this direction were using the measurement-based approach with time series models for prediction. This paper reviews the effectiveness of such models and it examines whether volatility is present in the data or not. It checks whether Gauss-Markov assumptions about homoscedasticity holds good for the ordinary least square estimators of such models or not. This paper applies a combination of AutoRegressive Integrated Moving Average - AutoRegressive Conditional Heteroskedastic (ARIMA-ARCH) model to predict resource usage. Experimental results demonstrate that the goodness of fit of the ARIMA-ARCH Model has improved when compared to the linear ARIMA model.

Keywords: Cloud Computing; Performance Degradation; Resource Exhaustion; Server Consolidation; ARIMA-ARCH Model

1. INTRODUCTION

Resource usage prediction is essential in a server virtualized system because Virtual Machines (VMs) use resources on demand. The other reason why resource prediction is important because of the progressive performance degradation of long running server systems. Performance degradation is due to operating system's resource shrinkage [4] .The most common causes for performance degradation include memory resource leakages, file descriptors which are not released and errors which occur during numerical approximation. From [1],[2],[3], it is found that memory exhaustion contributes majorly to the system failure due to resource shortage. This is the reason why the free available memory has been chosen in the proposed work for the detailed analysis of resource prediction.

Here a virtualized server system is used to collect the free available memory. The reason for selecting server virtualized system is due to the increased popularity of cloud computing, and the resource allocation in such system is dynamic in nature. The dynamic nature of allocating resources in server virtualized system makes resource prediction more challenging. There are two types of server virtualization which are most commonly used, the first one is the full virtualization and the second one is the paravirtualization. The full virtualization is popular among both because it offers better isolation and security for VMs, and simplifies migration and portability. Another advantage is that full virtualization avoids the extra layer of abstraction as in the case of paravirtualisation.

It is common to use time series models to predict the dynamic behavior of the resource usage. The leading choices are linear models like AutoRegressive (AR) models, Moving Average (MA) models, AutoRegressive Moving Average (ARMA) and AutoRegressive Integrated Moving Average (ARIMA) models. These models are successful in places where resource usage do not show nonlinear dynamic patterns like asymmetry, frequency amplitude dependence and volatility clustering. In this paper, resource usage data shows volatility clustering both at the beginning and at the beginning and at the end of system's lifetime. This dynamic nature of the data prompted us to use volatility model for resource prediction.

The major contribution of the paper lies in the fact that it is the first work on using AutoRegressive Integrated Moving Average - AutoRegressive Conditional Heteroskedastic (ARIMA-ARCH) model to analyze the resource usage data of any system. Most of the previous work in this area using the time series model never considered the heteroscedastic nature of data or the clustered volatility of the data which is prevalent. Another worth point to be noted is that no such study has happened before on the server consolidated system. The proposed model has reduced prediction error 55 times when compared to ARIMA model.

2. RELATED WORK

Next paragraphs discuss the merits and demerits of the previous work in this area. It is clear that none of the works in this area studied about the volatility and structural change of the data. Moreover, the Gauss-Markov assumptions about Ordinary Least Square (OLS) estimators have violated in many cases.

Lei Li et al. [5] used time-series analysis methods for identifying software performance deterioration. Apache web server system, a Linux system status monitoring tool, and a web server workload generator were used in the Experiment. Linear Regression and Sen's slope estimation methods were used to detect an aging trend. An autoregressive exogenous (ARX) model was used to estimate the usage of system resources and the result is compared with a linear regression model's estimation. The non-linear nature of the data was not taken into consideration. The goodness of fit of the ARX model was not discussed in this paper.

Hoffmann et al. [6] integrated the best practices from the experiences of two different studies; the first study was on how the selection of variables contributes more to model quality than selecting a particular modeling technique. In the second study, they compared five linear and non-linear models and found that the superiority of non-linear model was not always significant while comparing the model complexity. They proposed a coherent approach by integrating the goodness of the above studies. The main lacuna in this paper is that it stands as a guideline rather than as an effective methodology. The results for call availability prediction of an industrial telecommunication system were presented at the end of the paper. The paper did not discuss the adequateness of the model for the results presented.

Yongquan Yan et al.[7] proposed a hybrid model that combines autoregressive integrated moving average (ARIMA) and artificial neural network (ANN) to improve the prediction accuracy of resource consumption of an IIS web server which suffered from software aging problems. Their assumption was that the error term was nonlinear in nature and they used the ANN to model it. This paper did not discuss the goodness of the fit for the linear component using ARIMA model and further authors did not throw light on the residual analysis.

Araujo et al. [8] proposed a method which depends on multiple thresholds for ensuring a safe scheduling of rejuvenation actions. This method ensured that there will be minimal interference on system's performance due to time series computation. The experiments were conducted using the Eucalyptus cloud computing framework and thus they proved the efficacy of the method. The time series trend analysis was done and compared actual virtual memory utilization with the predicted results. But their study did not consider the non-linearity in the data.

Simeonov and Avresky [9] presented a framework, which is used for identifying anomalies that lead servers to crash. Further authors theoretically justified their proposed framework. The experimental validation did not consider the model adequacy. Matias [10] presented a full factorial design of experiment to predict the factors which were most influential on apache web servers aging. They found that page type and page size are the primary causes for the memory size variation in httpd processes. Further, no specific model was been discussed by the authors.

It is clear from the related work that the most popular model among the researchers studying the resource exhaustion is time series models. The reason is quite obvious because the resource utilization is collected against the time which forms a time series. It is surprising that none of the authors considered the clustered volatility generally prevalent in such data.

3. PROPOSED METHODOLOGY

The major problem identified during the literature survey is that none of the previous works try to address the volatility in the system resource data. This study focuses on modeling volatility in the time series data. The structure of the volatility model can be described as

$$x_t = \mu_t(\theta) + \varepsilon_t \quad (1)$$

$$\boldsymbol{\varepsilon_t = \sigma_t z_t} \quad (\mathbf{2})$$

Where

$\mu_t(\theta) = E[x_t \mid F_{t-1}]$,

$\sigma_t^2(\theta) = E\left[\left(x_t - \mu_t(\theta)\right)^2 \mid F_{t-1}\right]$,

$z_t \sim N(0,1)$.

In Eq. (1), the time series x_t is decomposed into a conditional mean $\mu_t(\theta)$ and a residual term ε_t. The conditional mean $\mu_t(\theta)$ may be an *ARIMA(p,d,q)* where p is the order of the autoregressive terms, d is the order of differencing and q is the order of moving average terms. F_t is the set of information available at time *t*. It may include the current and past values of x_t, current and past values of the residuals or any other variable known at time *t*. According to Eq. (2) , the residual term ε_t has a volatility conditional on the information available at time *t-1* denoted by σ_t. θ is a vector of unknown parameters. The variable z_t

follows normal distribution with mean *zero* and variance *one*. The overall structure of the proposed methodology is shown in Fig. 1.

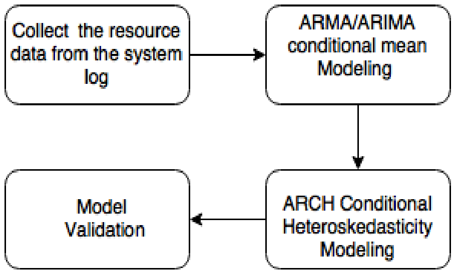

Fig. 1. Overall structure of Proposed Methodology

3.1 ARIMA Modeling

ARIMA(p,q,d) (Autoregressive Integrated Moving Average with orders *p,d,q*) is used as the mean process. *The ARIMA(p,q,d)* can be represented as in Eq. (3)

$$\left(\mathbf{1}-\sum_{k=1}^{p} \propto_k \boldsymbol{B}^k\right)(\mathbf{1}-\boldsymbol{B})^d \boldsymbol{X}_t = \left(\mathbf{1}+\sum_{k=1}^{q} \boldsymbol{\beta}_k \boldsymbol{B}^k\right)\boldsymbol{\varepsilon}_t \qquad (\mathbf{3})$$

Where X_t is the time series, $\propto$ and β are the parameters/coefficients of autoregressive and moving average terms with order p and q respectively. ε_t are error terms generally assumed to be independent, identically distributed variables sampled from a normal distribution with zero mean. B is the difference operator defined as follows by Eq. (4). Where d is the order of the difference operator.

$$\Delta \boldsymbol{X}_t = \boldsymbol{X}_t - \boldsymbol{X}_{t-1} = (\mathbf{1}-\boldsymbol{B})\boldsymbol{X}_t \qquad (\mathbf{4})$$

The overview of the statistical analysis is given by the flowchart as shown in Fig 2.

In order to find the series is stationary or not, the autocorrelation of the series has to be found out. Autocorrelation, also known as serial correlation, is the cross-correlation of a signal with itself. Informally, it is the similarity between observations as a function of the time lag between them. It is a mathematical tool for finding repeating patterns, such as the presence of a periodic signal obscured by noise, or identifying the missing fundamental frequency in a signal implied by its harmonic frequencies. It is often used in signal processing for analyzing functions or series of values, such as time domain signals.

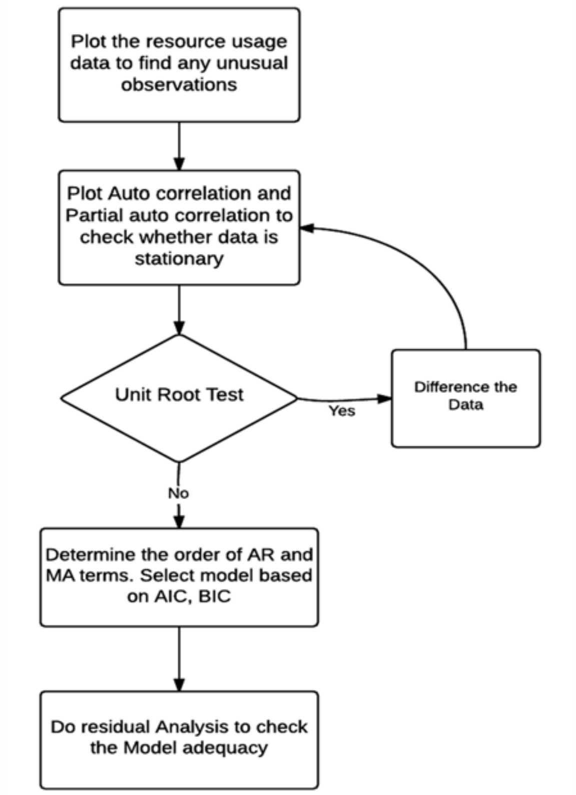

Fig. 2. The flowchart for ARIMA Conditional Mean Modeling .

The autocorrelation between time X_t and $X_{t+\tau}$ is given by the Eq. (5). Where X_t and $X_{t+\tau}$ are the time series with lag τ . μ is the mean of the population. $\sigma_{X_t}, \sigma_{X_{t+\tau}}$ are standard deviations at the time t and $t+\tau$. ρ_k is the autocorrelation at lag k

$$\rho_k = \frac{E[(X_t-\mu)(X_{t+\tau}-\mu)]}{\sigma_{X_t}\sigma_{X_{t+\tau}}} \qquad (5).$$

The sample estimate $r_k$of the same is given by Eq. (6). $\bar{X}$ s the sample mean and N is the number samples

$$r_k = \frac{\frac{1}{N}\sum_{t=1}^{n-k}(X_t-\bar{X})(X_{t-\tau}-\bar{X})}{\frac{1}{N-1}\sum_{t=1}^{n}(X_t-\bar{X})^2} \qquad (6)$$

From Fig. 2. it is clear that the first step is to plot the resource variable against time, this is to find any unusual observations. The next step in the flowchart is to plot the autocorrelation function and partial autocorrelation function of the resource usage data. This is to check whether the resource usage data is stationary or not. If the plot shows that data is non-stationary, the non-stationarity of the data has to be confirmed. This is done by doing a unit root test. Here augmented dickey fuller test [11] has been used as unit root test. Once the series is proved to be non-stationary then the series is differenced. By looking at the autocorrelation function (ACF) and partial autocorrelation (PACF) plots of the differenced series, one can tentatively identify the numbers of AR and/or MA terms that are needed. Once the number of autoregressive terms, the number of differencing and moving average terms are decided, parameters of the model have to be estimated. The maximum likelihood or least square estimators are commonly used to estimate the parameters.

For Model selection, three measures of goodness are used Akaike information criterion (AIC), Akaike information criterion corrected(AICc), Bayesian information criterion (BIC) [12]. The Akaike information criterion (AIC) deals with the trade-off between the goodness of fit of the model and the complexity of the model. It offers a relative estimate of the information lost when a given model is used to represent the process that generates the data. AIC does not provide a test of a model in the sense of testing a null hypothesis; i.e. AIC can tell nothing about the quality of the model in an absolute sense. Given a set of candidate models for the data, the preferred model is the one with the minimum AIC value. The Eq. (7) gives the AIC measure.

$$AIC = 2k - 2\ln(L) \quad (7).$$

Where k is the number of parameters and L is the maximum value of the likely hood function. Given a set of candidate models for the data, the preferred model is the one with the minimum AIC value.

AICc is AIC with a correction for finite sample sizes given by the Eq. (8) where n is the sample size.

$$AIC_c = AIC + \frac{2k(k+1)}{n-k-1} \quad (8).$$

Bayesian information criterion (BIC) or Schwarz criterion (also SBC, SBIC) is a criterion for model selection among a finite set of models. It is based likelihood function and it is closely related to the Akaike information criterion (AIC). Both BIC and AIC resolve this problem by introducing a penalty term for the number of parameters in the model; the penalty term is larger in BIC than in AIC. BIC is given by the Eq. (9).

$$BIC = -2.\ln(\mathrm{L}) + k.\ln(n) \quad (9).$$

3.2 ARCH Modeling

Autoregressive conditional heteroskedasticity models are used to characterize and model time series. ARCH models assume the variance of the current error term or innovation to be a function of the actual sizes of the previous time periods' error terms: often the variance is related to the squares of the previous innovations. So an $ARCH\,(p)$ assumes that the conditional variance $\sigma_t^2(\theta)$ is a linear function of the past p squared innovations where θ is a vector of unknown parameters which could be estimated by maximum likelihood estimators. $ARCH\,(p)$ is given by Eq. (10).

$$\boldsymbol{\sigma_t^2(\theta) = \omega + \alpha_1\varepsilon_{t-1}^2 + \cdots + \alpha_p\varepsilon_{t-p}^2 = \omega + \sum_{i=1}^{p} \alpha_i\varepsilon_{t-i}^2} \quad \mathbf{(10)}.$$

According to Eq. 10., the conditional volatility is assumed as moving average of squared innovations $\boldsymbol{\varepsilon_t^2}$. For the model to be well defined and the conditional variance to be positive, the parameters must satisfy the following constraints: $\boldsymbol{\omega > 0}$, and $\boldsymbol{\alpha_i \geq 0,\ i = 1, \cdots, p}$. The unconditional variance of innovation, denoted by $\boldsymbol{\sigma^2}$, is the unconditional expectation of $\boldsymbol{\sigma_t^2}$: $\boldsymbol{\sigma^2 = E[\varepsilon_t^2] = E[\sigma_t^2]}$. It can be easily shown that $\boldsymbol{\sigma^2 = \frac{\omega}{1-\sum_{i=1}^{p}\alpha_i}}$.

This shows that the process ε_t is covariance stationary only when sum of the autoregressive terms less than one, i.e., $\sum_{i=1}^{p}\alpha_i < 1$.

4. RESULT ANALYSIS

The resource usage data collected from the experimental setup discussed below is analyzed here. The free physical memory and swap read and write rates are the main focus because these resources are major indicators of performance degradation.

4.1 Experimental Setup and Data Collection

Experiments were done on an HP ProLaint ML350 G6 machine, it has two processors with 6 cores in each processor. It has 24 logical processors (when hyper threading enabled). This Intel Xeon architecture processors work at the speed of 2.40 GHz. 1 TB hard disk and 16 GB physical memory capacity are available for the machine. The machine has 2 NIC cards to effectively manage the network traffic.

VMware ESXi 5.0.0 is used as the hypervisor/Virtual Machine Monitor (VMM) in this experimental setup. This VMM belongs to type 1 VMM or this is a bare metal Hypervisor. 25 virtual machines have loaded on top of the hypervisor. Virtual machines (VMs) use Ubuntu 14.10 as the operating system. Each VM is configured with 4 vCPUs, 1 GB physical memory space, and 10 GB Hard disk space.

The *esxtop* command has been used to collect the information about CPU usage, swap in and swap out rate, interrupts, context switches, network statistics and power usage. In this experiment, the memory statistics alone has been used for further analysis. In this experiment, *httpref* tool is used in order to generate requests with constant time intervals between two requests. Each request accesses one specified file of size 5kb from the server. httpref is used not only as a workload generator but also used as a performance measuring tool.

The memory reclamation techniques[13] in VMware ESXi 5.0.0 like Transparent page sharing (TPS), Ballooning, Hypervisor swapping, and Memory compression has been enabled to increase the dynamicity of the resource allocation, thus making resource prediction more challenging. Transparent page sharing reclaims memory by removing redundant pages with identical content while ballooning reclaims memory by artificially increasing the memory pressure inside the guest. With memory compression, ESXi stores pages, which would otherwise be swapped out to disk through host swapping, in a compression cache located in the main memory. Memory compression outperforms host swapping because the next access to the compressed page only causes a page decompression, which can be an order of magnitude faster than the disk access. In the cases where ballooning, transparent page sharing, and memory compression are not sufficient to reclaim memory, ESXi employs hypervisor swapping to reclaim memory. Hypervisor swapping is a guaranteed technique to reclaim a specific amount of memory within a specific amount of time. However, hypervisor swapping is used as a last resort to reclaim memory from the virtual machine due to limitations on performance.

4.2 Analysis of Resource Utilization

Fig. 3. shows the time series plot of free physical memory over a period of time. The data is collected over a period of 10 days with the time interval of 30 minutes.

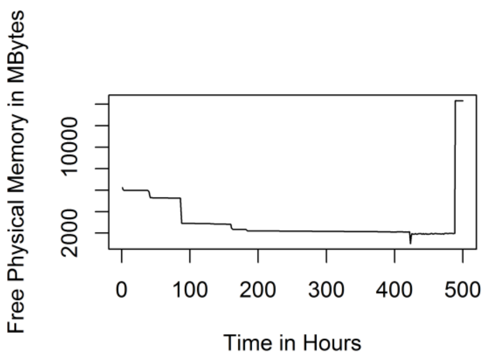

Fig. 3. Free Physical memory collected over a period of 250 hours.

The memory over commitment during this period is kept as 4.11. The Memory Overcommitment during this period is shown as in Fig. 4. The memory overcommit is given by the Eq. (11). Where N is the number of VMs switched on and $EsxMemory$ is the total host *VMware Esxi* memory.

$$memoryovercommit = \frac{\sum_{i=1}^{N} memsize}{EsxMemory}. (11)$$

It is clear from the Fig. 3. that at the 245th hour the Free Memory has increased from 1900 Mbytes to 14000 Mbytes. From Fig. 4., there is a clear indication that the Memory Overcommitment (15 Minutes Average) has decreased from 4.11 to zero, which clearly shows that some VMs or all VMs has been switched off and swapped into the secondary memory. It clearly gives an indication of thrashing. For further investigation swap read and write rates are plotted against time.

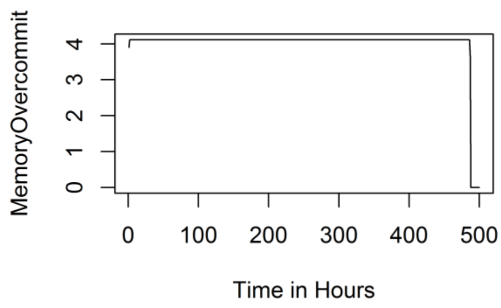

Fig. 4. Memory Overcommit (15 Minutes Average) plotted against time

Fig. 5. and Fig. 6. gives swap read and write per second respectively.

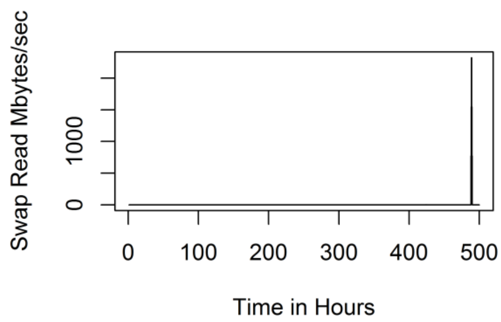

Fig. 5. Swap Mbytes Read per Second against time

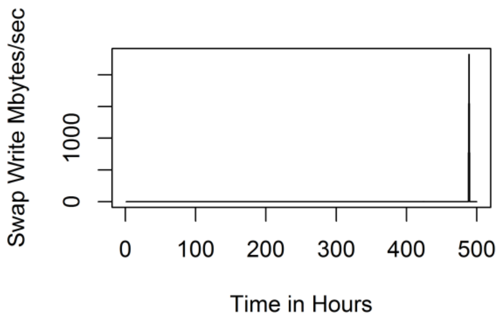

Fig. 6. Swap Mbytes Writes per Second against Time

The huge swap read and write rates at 245th hour shows the free physical memory has shrunk to a level, where the virtualized server system has forced one or more VMs to virtual space of the hypervisor. This forced the system to fail, so this point is considered as the failure point of the setup.

4.3. ARIMA Mean Modelling

The resource which is most vulnerable in terms of the crash is free physical memory because the system mainly crashes/hangs due to the memory leakage. So here the variable chosen for time series analysis is free physical memory. In order to find the series is stationary or not, the autocorrelation of the series has to be found out. Fig. 7. shows the Autocorrelogram plot of Free Physical Memory against Lag.

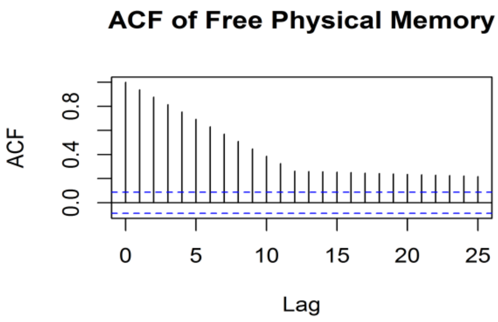

Fig. 7. Autocorrelogram of Free Physical Memory

It is clear from the Fig. 7. that all the values up to 25 lags are significant or above the significant band. This indicates a possibility of nonstationarity in the series which has been confirmed by the augmented dickey fuller test. The augmented dickey fuller test confirms there is no statistical evidence that series is stationary. That means the series is nonstationary.

Before differencing the series, the partial autocorrelation function has to be obtained. In time series analysis, the partial autocorrelation function (PACF) plays an important role in identifying the extent of the lag in an autoregressive model. The use of this function was introduced as part of the Box–Jenkins approach to time series modeling, whereby plotting the partial autocorrelative functions one could determine the appropriate lags p in an *AR (p)* model or in an extended *ARIMA (p,d,q)* model. Fig. 8. shows PACF of Free Physical memory.

ACF and PACF slowly die down which gives a clear indication of non-stationarity, So the time series has to be differenced once. The ACF and PACF of the differenced series show no clues of non-stationarity. The Augmented dickey fuller test was conducted again on the differenced series, to know whether the series is non-stationary or not. There is no statistical evidence of non-stationarity in the time series.

From the ACF and PACF, *ARIMA (0,1,1), ARIMA(2,1,0), ARIMA(1,1,0), ARIMA(1,1,1), ARIMA(1,0,0)* are to be verified. These models are arbitrarily selected because the ACF and PACF gives a clue that the series is random as there are no significant terms in both ACF and PACF.

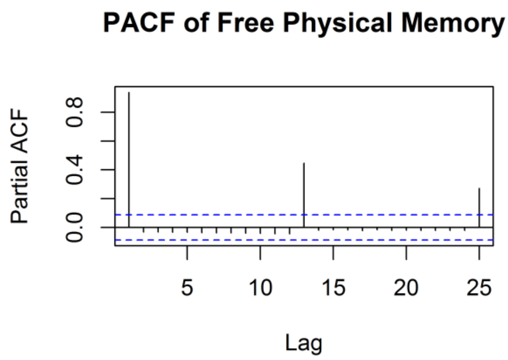

Fig. 8 Partial autocorrelation function of free physical memory

Table 1. The Goodness of Fit Measures ARIMA Models.

Model	AIC	BIC	AICc
ARIMA(0,1,1)	7740	7741	7752
ARIMA(2,1,0)	7742	7643	7543
ARIMA(1,1,0)	7739.	7132	7564.1
ARIMA(1,1,1)	7742	7432	7432
ARIMA(1,0,0)	7732	7432	7679
ARIMA(0,0,1)	8598	8432	8564

To analyze the goodness of fit, three measures of goodness are used Akaike information criterion (AIC), Akaike information criterion corrected (AICc), Bayesian information criterion (BIC). The values of this measures are given as below in

Table 1. It is clear from Table 1, that none of the models fit the data well so the residuals of the best available model i.e.; *ARIMA (1,1,0)* has to be analysed.

4.4 ARCH Modelling

Analysis of residuals from the fitted model gives a clue for modification. Here one model with less AIC values is selected for the analysis. *ARIMA(1,1,0)* which is the best among the analyzed models is selected for residual analysis. Standardized residuals of this model are plotted as shown in Fig. 9.

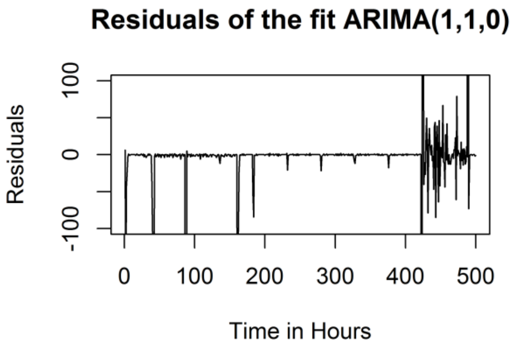

Fig. 9 Standardized Residual of ARIMA(1,1,0)

Fig. 9. shows that there is a huge variation in residuals, which is the indication of heteroskedasticity. In order to test the normality of the residual quantile-quantile plot is plotted from the residuals of *ARIMA(1,1,0)* model. Fig. 10. shows the quantile-quantile plot of the residuals of *ARIMA(1,1,0)* model.

Fig. 10 clearly shows that the residual series is not normal. It is clear from the Model diagnostics that the time series is heteroskedastic in nature.

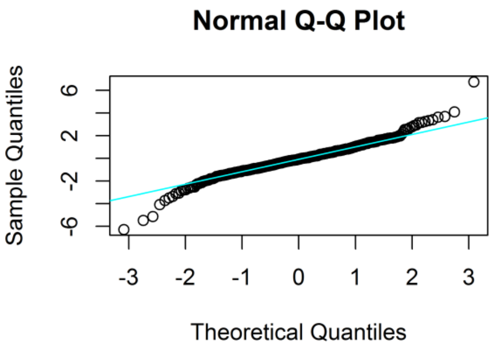

Fig. 10 The Q-Q Plot of Residuals from ARIMA(1,1,0)

The ARCH model has to be applied to the residuals after fitting with a linear mean model. In this case ARIMA(1,1,0) has been used as the mean model. The next step is to obtain the order of ARCH model. ARCH(p) model from p = 1 to 6 are considered to fit residual data, and the AIC and BIC values of these models are listed in the Table 2.

Table 2. The goodness of Fit Measures ARCH Models.

Model	AIC	BIC	AICc
ARCH(1)	5.08	5.13	5.08
ARCH(2)	5.08	5.15	5.08
ARCH(3)	5.48	5.23	5.48
ARCH(4)	5.56	5.51	5.56
ARCH(5)	5.58	5.52	5.58
ARCH(6)	5.59	5.63	5.59

It is clear from the Table 2 that the *ARCH(1)* has the best goodness of fit value. The goodness of measure has improved 1548 fold roughly when compared to the *ARIMA(1,1,0)* model. In order to check the accuracy of prediction, Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) of *ARIMA(1,1,0)* and *ARCH(1)* Models are compared as shown in the Table 3.

Table 3. Comparison of ARIMA and ARCH Models.

Model	RMSE	MAE
RIMA(1,1,0)	2051.6	1427.4
ARCH(1)	37.69	37.02

ARCH(1) model has reduced the error by 55 times roughly when RMSE measure is taken into consideration.

5. CONCLUSION

The primary objective of the paper was to analyze prediction models of the resource usage in the server virtualized systems. Resource prediction is essential in order to predict the Time to Exhaustion (TTE) in any system so that the administrator of the system could avoid an accidental outage. It is also found that most of the studies in this direction were using the measurement-based approach with time series models for prediction. This paper reviewed the effectiveness of ARIMA model and it examined whether Gauss-Markov assumptions about homoscedasticity holds good for the Ordinary Least Square estimators of such models. It is clear from the model diagnostics that the time series is heteroscedastic in nature. ARIMA models are not suitable for predicting the resource usage and to predict time to exhaustion of the system. AutoRegressive Conditional Heteroscedastic model has been used to model the residuals of ARIMA mean process. Results show that the goodness of fit has improved roughly 1500 times.

The clustered volatility at the end and beginning of the time series shows that a regime switching model is much apt for predicting resources

6. REFERENCES

[1] Matias Rivalino, Pedro Alberto Barbetta, and Kishor S. Trivedi. "Accelerated degradation tests applied to software aging experiments." Reliability, IEEE Transactions on 59.1 (2010): 102-114.

[2] Matias Rivalino, Kishor S. Trivedi, and Paulo Romero Martins Maciel. "Using accelerated life tests to estimate time to software aging failure." Software Reliability Engineering (ISSRE), 2010 IEEE 21st International Symposium on. IEEE,2010.

[3] Zhao, Jing, et al. "Injecting memory leaks to accelerate software failures." Software Reliability Engineering (ISSRE), 2011 IEEE 22nd International Symposium on. IEEE, 2011.

[4] Y. Huang, C. Kintala, N. Kolettis and N.D. Fulton, "Software Rejuvenation: Analysis, Module and Applications," Proc. of FTCS-25, Pasadena, CA, Jun. 1995.

[5] Lei Li; Vaidyanathan, K.; Trivedi, K.S., "An approach for estimation of software aging in a Web server," in Empirical Software Engineering, 2002. Proceedings. 2002 International Symposium, vol., no., pp.91-100, 2002

[6] Hoffmann, G.A.; Trivedi, K.S.; Malek, M., "A Best Practice Guide to Resource Forecasting for Computing Systems," in Reliability, IEEE Transactions on, vol.56, no.4, pp.615-628, Dec. 2007

[7] Yongquan Yan; Ping Guo; Lifeng Liu, "A Novel Hybridization of Artificial Neural Networks and ARIMA Models for Forecasting Resource Consumption in an IIS Web Server," in Software Reliability Engineering Workshops (ISSREW), 2014 IEEE International Symposium on, vol., no., pp.437-442, 3-6 Nov. 2014

[8] Araujo, J.; Matos, R.; Maciel, P.; Vieira, F.; Matias, R.; Trivedi, K.S., "Software Rejuvenation in Eucalyptus Cloud Computing Infrastructure: A Method Based on Time Series Forecasting and Multiple Thresholds," in Software Aging and Rejuvenation (WoSAR), 2011 IEEE Third International Workshop on , vol., no., pp.38-43, Nov. 29 2011-Dec. 2 2011

[9] D. Simeonov and D.R. Avresky. "Proactive Software Rejuvenation Based on Machine Learning Techniques," Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 34, pp. 186-200, 2010.

[10] Matias, R.; Filho, P.J.F., "An Experimental Study on Software Aging and Rejuvenation in Web Servers," in Computer Software and Applications Conference, 2006. COMPSAC '06. 30th Annual International, vol.1, no., pp.189-196, 17-21 Sept. 2006

[11] Elliott, G.; Rothenberg, T. J.; Stock, J. H. (1996). "Efficient Tests for an Autoregressive Unit Root". Econometrica 64 (4): 813–836. JSTOR 2171846.

[12] Aho, K.; Derryberry, D.; Peterson, T. (2014), "Model selection for ecologists: the worldviews of AIC and BIC", Ecology 95: 631–636, doi:10.1890/13-1452.1

[13] Understanding Memory Resource Management in VMware ESX Server https://www.vmware.com/files/pdf/perf-vsphere-memory_management.pdf