International Journal of GEOMATE, Aug, 2017, Vol.13, Issue 36, pp.108-115
Geotec., Const. Mat. & Env., ISSN:2186-2990, Japan, DOI: http://dx.doi.org/10.21660/2017.36.2855

LIFE DATA ANALYSIS OF SERVER VIRTUALIZED SYSTEM

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

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

*Corresponding Author, Received: 15 August 2016, Revised: 29 Nov. 2016, Accepted: 26 Dec. 2016

ABSTRACT: The use of reliability metrics and life data analysis has received considerable attention recently in the software engineering literature. Life data analysis under the actual operational profile can, however, be expensive, time consuming or even infeasible. In this paper, a systematic approach has been adopted in order to reduce the experimentation time for estimating time to failure of a server virtualized system. The study of time to failure (TTF) is very essential in server virtualized system, because it is the crux of the cloud computing infrastructure. In order to meet service-level agreements (SLAs) like availability, reliability and response time, prediction of reliability metrics like mean time to failure (MTTF), life distribution etc are indispensable. The most important contributions of this paper are the reduction of experimental time, and the life data analysis of the server virtualized systems which were not addressed so far. Experimental results demonstrate that there is only four percentage deviation from the observed results from the Normalized Root Mean Square Error and resulting in 96% accuracy of predicting MTTF.

Keywords: Accelerated testing; Software reliability; Cloud computing; Server virtualised system; Life data Analysis;

1. INTRODUCTION

Life data analysis involves analyzing times-to-failure (TTF) data obtained under the normal operating conditions in order to quantify the life characteristics of a product, system or component. For many reasons, obtaining such life data (or times-to-failure data) may be very difficult or impossible. The reasons for this difficulty can include the long life times of today's products, the small time period for testing products. In order to overcome this difficulty, reliability practitioners have attempted to devise methods to force these products to fail more quickly than they would under normal operational/use conditions. In other words, they have attempted to accelerate their failures. Over the years, the phrase accelerated life testing has been used to describe all such practices. Accelerated life testing involves the acceleration of failures with the purpose of quantifying the life characteristics of the product at normal operational/use conditions. Accelerated life testing can be classified as: qualitative accelerated testing and quantitative accelerated life testing. Qualitative accelerated testing focuses on identifying failures and failure modes while quantitative accelerated life testing concentrates on predicting the life of the product at normal operational/use conditions. This paper concentrates on quantitative accelerated life testing of server virtualized system. To the best of our knowledge, this is the first work on life data analysis of server consolidated systems.

In order to accelerate the failure time generally two methods of acceleration are employed: usage rate acceleration and overstress acceleration methods. Usage rate acceleration is employed for the product that does not operate continuously; one can accelerate the time it takes to induce failures by continuously testing these products, this is called usage rate acceleration. For products for which the usage rate acceleration is impractical, one can apply stress at levels which exceed the levels that a product will encounter under normal use conditions; this is called overstress acceleration. The time-to-failure data obtained in this manner is used to extrapolate the times-to-failure at operational/use conditions. Server consolidation systems are built to run for a longer period of time so overstress acceleration is used in this case

In life data analysis, life distribution of the system under test (SUT) has to be determined from the failure data. Once this probability density function has been obtained, all other desired reliability results can be easily determined. However, we face the challenge of determining the use level probability density function from accelerated failure data, rather than from failure data obtained under use conditions. To accomplish this there should be mechanism that allows us to extrapolate from data collected at accelerated conditions to arrive at an estimation of use level characteristics. In order to extrapolate the data life stress models are used.

Life data analysis has been done on many platforms like Web Server, Operating Systems and Embedded Systems. Recently, virtualized platforms are getting more popular due to cloud computing applications. Virtualization [10] can be viewed as part of an overall trend in enterprise information technology that refers to the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, operating systems, storage devices, and computer network resources. The main goal of virtualization is to centralize administrative tasks while improving the scalability [3]. Server virtualization is a technique by which multiple servers are hosted on one physical machine. There are many advantages for server virtualization as this technique increases the resource utilization, makes the system highly available, easy to administer, reduces the power usage, reduces the man power and it also reduces the spending on physical infrastructure.

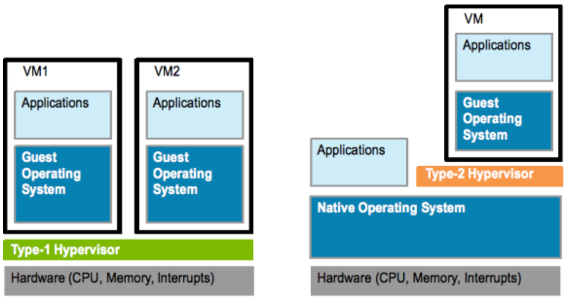

Fig.1 (a) Fig.1 (b)

Fig. 1(a),(b) Different types VMM ; Fig. 1 (a) Type 1 Hypervisor ; Fig.1(b) Type 2 Hypervisor

Server virtualization is developed on the software architecture known as virtual machine. Virtual machine is concept by which underlying physical architecture is hidden from upper layers by a layer of abstraction. In server virtualization system a hypervisor or virtual machine monitor (VMM) will be used to hide the underlying physical architecture from upper layers. The hypervisor is of two types, Type 1 and Type 2. Type 1 (Bare Metal) VMM runs directly on the physical machine to control the hardware and to manage guest operating systems. In Type 2 (hosted) VMM runs within a host operating system. Fig. 1(a),(b) shows different types of VMMs.

The life data analysis is very essential as it helps in making predictions about product life by "fitting" a statistical distribution to life data from a representative sample. This distribution can then be used to estimate important life characteristics of your product such as reliability or probability of failure at a specific time, the mean life for the product and failure rate. This kind of study is very essential in server consolidated systems which helps the system administrators to avoid the unnecessary outages. Due to the long operational life of the server consolidated such a study is difficult, hence in this paper, an accelerated life testing has been adopted to reduce the experimentation time. This is the first time such an approach has been adopted in server consolidation or server virtualized system. The prediction of reliability metric like mean time to failure helps the system administrator in turn to predict the time to rejuvenate and time for migrating the virtual machines in the server consolidated system.

The rest of the paper is organized as follows. Section 2 of this paper deals with the related work, here similar works in the related domains are compared. Section 3 discusses about how the acceleration in stress can be applied, what are the appropriate distributions for life data analysis and it discuss about the different life stress relationships. The experimental setup, the results of goodness of fit of different distributions and prediction of the mean time to failure at different stress levels are topics of discussion in Section 4. Finally we conclude the paper in Section 5.

2. RELATED WORK

Pengcheng Yin et al. [1] used support vector machine (SVM) to predict the life of an electric motor. Ten motors are chosen to perform accelerated life testing at three different temperature levels. A practical model is proposed in this paper to predict the life of the items in accelerated life testing based on support vector machine. Authors claimed that reliability can be predicted accurately for small size of sample data, without using life stress models and the specific life distribution types. This method is easy to use and it has no prior assumptions of distributions are major advantages; but it can be used only for small sample size.

Shuzhen Li et al. [2] adopted accelerated degradation testing (ADT) to verify the reliability and life of high-reliable, long-life product. In this paper, a new degradation prediction method based on Support Vector Machine (SVM) is proposed and developed to predict time-to-failure of product. This prediction method is also compared with Back Propagation Artificial Neural Networks (BPANN) and regression methods to validate its effectiveness. The paper concludes that, SVM gives better result compare to BPANN. This paper compared three models and suggested that SVM is better for given data set. This paper never discusses about the scalability of the results in comparison with observed results.

Matias et al. [3] applied accelerated degradation test (ADT) on apache web server. The memory consumed by *httpd* process has been assumed as performance degradation factor. Design of Experiment (DOE) factors used was page size, page type and request rate. Page type signifies whether it is static page or dynamic page. They considered regular and high load factor for each parameters. According to analysis, Page size and page type influenced degradation, not the request rate. Experiments are based on accelerated degradation tests (ADT), which do not look for failure times; instead degradation measure of a product's performance has taken over time. Tests containing dynamic requests forced the amount of memory used by *httpd* processes to reach the limit of the total main memory available. It forced the Linux kernel to thrashing. The main advantage is the reduction of the experimental time, but the results are not compared with observed ones.Further Matias et al [4] proposed and evaluated the use of quantitative accelerated life tests (QALT) to reduce the time to obtain the lifetime distribution of systems that fail due to software aging. This approach has experimentally estimated the lifetime distribution of a real web server system. The accuracy of the estimated distribution is evaluated by comparing its reliability estimates with a sample of failure times observed from the real system under test. Major advantages of this method are the reduction of experimental time and the selection of the stress variable based on the aging related failures. The lacuna of this method lies in the lack of lucidity in the explanation of estimating pseudo failure time.

Jing Zhao et. al. [5] created a test bed of a web server, a database server, and a set of clients. The experiments were conducted on Tomcat web container, and all *html* pages were dynamically generated by the server. TPC-W standard bench mark has been used. The web traffic is generated by a Remote Browser Emulator, which emulates users of the website. Memory leaks are injected artificially to accelerate degradation. Inverse Power Law-Lognormal distribution is used to calculate Life characteristic relationship. Major advantages of their approach are the reduction of experimental time and the use of semi-Markov process, to optimize the software rejuvenation trigger interval. The tests are conducted at application level so the readings used for estimation may not be accurate.

Tingting Huang et. al. [6] presented an optimum design of constant stress accelerated life testing based on proportional hazards-proportional odds using penalized local D-optimality. It established the objective function as the product of the Fisher information matrix as defined in proportional hazards-proportional odds model and penalty functions which describe the closeness of the probability density functions of two specified stress levels. This optimum method avoids obtaining limit stress levels of test planning using D-optimality for some cases. The comparison of the optimization results by D-optimality and penalized local D-optimality shows that optimization results using penalized local D-optimality is more reasonable. It is non parametric method, hence it does not require the prior knowledge about the data. Proportional hazards-proportional odds model assumes data follow weibull or lognormal distribution and this assumptions may lead to erroneous results in some cases.

Javier Alonso et. al. [7] proposed a framework that monitors the system level metrics and predicts the time until the system crashes. The authors evaluated two different families of Machine learning algorithms: Linear Regression and Decision Trees. They have considered M5P and REPTree algorithms for Decision Trees. They have captured the system snapshot and evaluated the variation of resource consumption rate. The metrics used are: Throughput, Response time, workload, System load, disc usage, swap used, number of processes, number of threads, free system memory, memory occupied by the running application, number of http connections received, and number of connections to the data base. M5P algorithm gives comparatively better result. The authors did not assume the distributions of the data. The reasons for selecting the metrics for prediction is not explained and the models failed to predict the crashes accurately.

Tao Yuan et. al. [8] developed a method for planning optimal step-stress accelerated life testing. Most of the studies in this area used Maximum Likelihood Estimators for the reliability metrics of interest; Authors used Bayesian method for parameter estimation. Maximum Likelihood Estimation requires precise data and it cannot be applied if there is uncertainty. This study applied the method to design a simple step-stress accelerated life test with Type-I censoring and the Weibull life distribution. The Bayesian optimal plans are compared with the plan obtained by maximum likelihood method. Influence of sample size and prior distribution on the optimal plan is also investigated. Results indicate that the Bayesian approach has promising potential in the planning of reliability life testing when there is uncertainty in the precise values of the model parameters. This model can be applied even there are uncertainties exist in the data. The disadvantage of this method is that it requires prior knowledge about the product life characteristics.

Table 1. shows the merits and demerits of different existing approaches. It is clear that none of these studies are focused on life data analysis of server virtualized system. Most of these studies concentrated on the hardware systems where the stresses are temperature, vibration, humidity, voltage, and thermal cycling. In [3],[4] the system under test was web server.

Table 1. Merits and Demerits of Existing Works

Paper	Merits	Demerits
Pengcheng Yin et al. [1]	Easy to use and it has no prior assumptions about the probability distributions	Tested only for small sample sets
Matias et al. [3]	Reduction of the experimental time	Results are never compared with observed ones
Matias et al [4]	The selection of the stress variable based aging related failures	Lack of clarity in the estimation of pseudo failure time
Jing Zhao et. al. [5]	Use of semi-Markov process for the optimization of rejuvenation interval	Application level metrics reading are prone to errors
Tingting Huang et. al. [6]	Does not require prior knowledge about the data	Proportional hazards-proportional odds assumes data follows weibull or lognormal distribution
Shuzhen Li et al. [2]	Proposed new degradation prediction model based on SVM	Silent about the scalability of the results
Tao Yuan et. al. [8]	Model can be applied even when data is not certain.	Require prior knowledge about the product

In this paper the system under test is server virtualized system which consists of a Hypervisor/Virtual Machine Monitor and a set of Virtual Machines. The motivation behind taking server virtualized system as our system under test is that it forms the crux of any datacenter. It is important to study the life data analysis of server virtualized setup because it avoids unnecessary outages which help the system administrator of any server virtualized system to migrate the Virtual Machines to a fresh Hypervisor to reduce the down time. That in turn helps the enterprise to meet its service level agreements.

3. PROPOSED METHODLOGY

Figure 2 shows the overall architecture of the proposed methodology of accelerated life testing on server virtualized or server consolidated system.

As shown in Figure 2, the primary step in the test planning is the definition of accelerating stress variable and its levels of utilization (load). Commonly used accelerating stresses are temperature, vibration, humidity, voltage, and thermal cycling [9]. These stresses are appropriate for many engineering applications, where tests are applied to physical or chemical components that are governed by well-known physical laws. However, for software components, we cannot adopt the above mentioned accelerating stresses. From [3],[4],[5], it is clear that memory exhaustion is the main reason for system failure, so memory usage has been used as the stress variable.

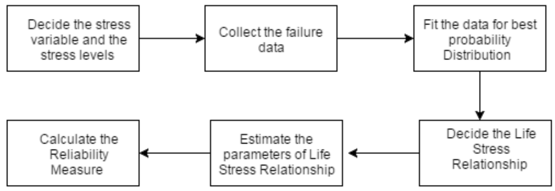

Fig 2. The Overall Architecture of the Proposed Methodology

The failure data at the selected stress levels were collected, the collected failure times has to be fitted for the best probability distribution. Log-normal, Weibull and exponential distributions have been used quite effectively in analyzing positively skewed data, which play important roles in the reliability analysis. In order to find the best fit for the failure data three goodness-of-fit tests are conducted: *Kolmogorov–Smirnov test (K-S Test), Akaike information criterion (AIC) and Bayesian information criterion (BIC).*

The collected failure times are for System Under Test (SUT) operating under stress, not under its normal operational/use condition. Hence a model is required, that relates the failure times observed at the tested stress levels to the underlying lifetime distribution of the SUT operating in its normal use condition. This model is called life-stress relationship [9], [10]. Several life-stress relationship models have been developed for different engineering fields. Examples of such well-known models are Arrhenius, Eyiring, Coffin-Manson, Peck, and Zhurkov [9], [10], [11]. Based on the SUT's physical/chemical properties, the underlying theories used to build these models assume specific stress types. For this reason, traditional models applied to physical systems cannot justifiably be employed to build life-stress relationship models for software systems. An exception is the Inverse Power Law (IPL) relationship model, the IPL is applicable to any type of positive stress, unlike the above mentioned models that are used for specific types of stress variables. The Algorithm 1 shows overall steps involved in the life data analysis of server consolidated systems.

Algorithm 1:Overall steps involved in the Life Data Analysis of

Server Virtualized System

Algorithm 1
Step1 : Decide the stress variables and the stress levels Step 2: Collect the failure data Step 3: Find the best fit for the failure data from the positively skewed distributions Step 4: Decide the Life stress relationships Step 5: Estimate the parameters of Life Stress Relationship Step 6: Calculate the reliability Measures

The IPL life-stress relationship can be expressed in Eq.(1) , Where *L* represents a quantifiable life measure. *V* represents stress levels, *K* and *n* represents model parameters to be determined from the life distribution. Based on the life distribution, the following are the Life stress probability distribution function.

$$L = 1/(KV)^n \tag{1}$$

The life stress probability distribution function of IPL-Exponential is given by Eq. (2)

$$\boldsymbol{f(t,V) = KV^n e^{-KV^n t}} \tag{2}$$

where *t* is the time in hours, *K*, *n* is model parameters to determined and *V* is the stress level.

The life stress probability distribution function of IPL-Weibull is given by Eq. (3)

$$\boldsymbol{f(t,V) = \beta KV^n (KV^n t)^{\beta-1} e^{-(KV^n t)^{\beta}}} \tag{3}$$

where *t* is time in hours, β is shape parameter, *K and n* are model parameters to be determined and *V* is the stress level.

The life stress probability distribution function of IPL-Log-Normal is given by Eq. (4)

$$\boldsymbol{f(T,V) = \frac{1}{T\sigma_{T'}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{T'+\ln(K)+n\ln(V)}{\sigma_{T'}}\right)^2}} \tag{4}$$

where $\boldsymbol{\sigma_{T'}}$ is the standard deviation of the natural logarithms of the Time to Failure, *K*, *n* are the model parameters to determined and *V* is the stress level.

Once an underlying life distribution and the life-stress relationship model to fit the accelerated data have been decided, the next step is to decide the parameter estimation method. Here Maximum Likelihood Estimation method is employed as it is more robust than probability plotting and least square estimators. The required reliability information is Mean Time to Failure (MTTF). MTTF give the rejuvenation time of the Virtual Machines. MTTF for IPL- Exponential life stress probability distribution is given by Eq.(5)

$$\boldsymbol{\overline{T} = \frac{1}{KV^n}} \tag{5}$$

where *K*, *n* are the model parameters to determined, and *V* is the stress level. MTTF for IPL- Weibull life stress probability distribution is given by Eq.(6).

$$\boldsymbol{\overline{T} = \frac{1}{KV^n} \cdot \Gamma\left(\frac{1}{\beta}+1\right)} \tag{6}$$

Where *K*, *n* are the model parameters to determined, and *V* is the stress level. And $\boldsymbol{\Gamma\left(\frac{1}{\beta}+1\right)}$ is the gamma function to be evaluated at $\left(\frac{1}{\beta}+1\right)$, where β is the shape factor of the Weibull distribution. MTTF for IPL-Lognormal life stress probability distribution is given by Eq.(7)

$$\boldsymbol{\overline{T} = e^{-\ln(K)-n\ln(V)+\frac{1}{2}\sigma^2_{T'}}} \tag{7}$$

Where $\boldsymbol{\sigma_{T'}}$ is the standard deviation of the natural logarithms of the Time to Failure, *K*, *n* are the model parameters to determined, and *V* is the stress level.

4. RESULTS ANALYSIS

The system under test used in this experiment is VMWare ESXi 5.5 on LENOVO 5498-PR1, 4 Cores, 500GB hard disk, 2.93 GHz processor with Hyperthreading enabled and 8GB RAM. 8 Virtual Machines (VMs) were created on this machine and installed Ubuntu 14.04 operating system on all of them. Apache 2.0 and PHP 5.0 were installed on these VMs. In these setup 8 VMs are acting as servers. In the same virtualised environment, a client VM is created with Ubuntu 14.04 operating system and *httpref* [12] installed on it. Shell program is written to call the *test.php* page using *httperf* continuously with the rate of 500 requests per second. Server VMs used 1vCPU, 4GB RAM and 100GB hard disk with thin provisioning. The Client VM used 1GB RAM, 1 vCPU, 16GB of Hard disk. 8VMs using 4GB RAM results in 32GB of logical RAM plus client VM is using 1 GB, but physically only 8 GB RAM is present. Hence main memory is over-committed to approximately four times.

The basic assumption behind the workload is that a memory leak is injected randomly after "N"clients requests. This number is taken as the stress factor in this experiment. In this paper three stress values N=5, 10, 20 are considered and it means that a memory leak is deliberately injected within that number of client requests. To make it clear let us suppose N=5, then a memory leak is injected between 0 to 5 client requests. So N=5 is the overstressed condition among the three stress levels used. The flowchart for the memory leak injection is shown as in the Figure 3.

After collecting the failure times, the best probability distribution has to identified from the failure data of Virtual Machines at each stress levels N=5, N=10 and N=20. For each stress level, three sets of failure data has been collected. Since there are 8 Virtual Machines 24 failure data has been collected for each stress level.

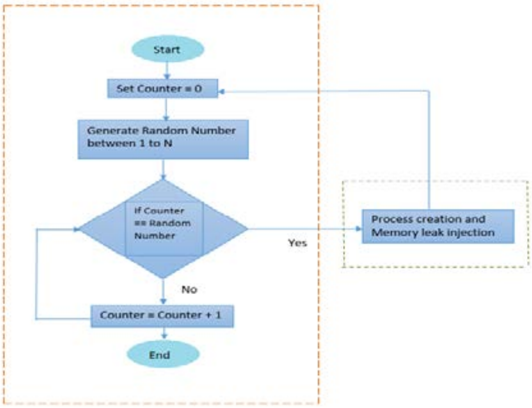

Fig 3. Flowchart of the memory injection algorithm

The twenty four times failure data were collected at each stress level is shown as below. For the first stress i.e. at N=5, 24 failure data are collected as shown below in Table 2.

Table 2. Failure times of the Virtual Machines collected for the first stress level at N = 5

Virtual Machine	Experiment 1	Experiment 2	Experiment 3
1	37	33	37
2	23.5	24	39
3	14	27.5	15.5
4	37	37	34
5	28.5	23.5	41
6	37	28	40
7	32.5	30	38.5
8	25.5	37.5	26

For the second stress i.e. at N=10; twenty four failure data are collected as shown below in Table 3.

Table 3. Failure times of the Virtual Machines collected for the second stress level at N = 10

Virtual Machine	Experiment 1	Experiment 2	Experiment 3
1	40.5	38.5	34.5
2	40.5	32.5	43
3	40	32.5	33.5
4	45.5	15	39.5
5	45.5	39	37
6	21.5	36.5	35
7	21	18	35.5
8	67.5	35.5	35.5

For the second stress i.e. at N=20; twenty four failure data are collected as shown below in Table 4.

Table 4. Failure times of the Virtual Machines collected for the third stress level at N = 20

Virtual Machine	Experiment 1	Experiment 2	Experiment 3
1	48.5	71	38
2	60.5	54.5	47.5
3	70	67	40
4	62.5	72.5	44
5	47.5	41	45
6	50	53	52.5
7	48.5	54.5	39.5
8	50.5	58	40.5

The failure data at the selected stress levels were collected, the collected failure times has to be fitted for the best probability distribution. Three distribution were chosen Log-normal, Weibull and exponential distributions because they have been used quite effectively in analyzing positively skewed data, which play important roles in the reliability analysis. In order to find the best fit for the failure data three goodness-of-fit tests are conducted. The Kolmogorov–Smirnov test (K-S Test), Akaike information criterion (AIC), Bayesian information criterion (BIC) were used to find the best fit among the three probability distributions chosen. For the stress level N=5, N=10, N=20 the results are shown as below. Table 5 gives the goodness-of-fit scores of the first data set ; i.e. The experiment conducted at stress level N=5.

Table 5. Goodness-of-fit scores of failure data for the first stress level at N = 5

Distributions	Kolmogorov–Smirnov test (K-S Test)	Akaike information criterion (AIC)	Bayesian information criterion(BIC)
Weibull	0.535	166.1789	168.535
Log Normal	0.3557	174.0621	176.4182
Exponential	0.5385	175.777	178.1331

Table 6. gives the goodness-of-fit scores of experiments conducted at stress level N=10. Table 7 gives the goodness-of-fit scores of experiments conducted at stress level N=20.

Table 6. Goodness-of-fit scores of failure data for the second stress level at N = 10

Distributions	Kolmogorov–Smirnov test (K-S Test)	Akaike information criterion (AIC)	Bayesian information criterion(BIC)
Weibull	0.2033	185.3682	187.4354
Log Normal	0.07237	186.3865	188.7426
Exponential	7.11E-15	586.8487	589.2048

Table 7. Goodness-of-fit scores of failure data for the second stress level at N = 20

Distributions	Kolmogorov–Smirnov test (K-S Test)	Akaike information criterion (AIC)	Bayesian information criterion(BIC)
Weibull	0.995	178.3682	182.7243
Log Normal	0.9888	181.5825	183.9386
Exponential	0.9676	180.986	183.3421

The goodness-of-fit scores shows two-parameter Weibull fits the data well. So the IPL-Weibull stress relationship has been used to identify the MTTF of the Server Virtualized system. The parameters are estimated using Maximum Likelihood Estimation. The shape parameter, β = 4.462529, the model parameters K = 0.054748 and n = -0.367017.

Table 8 gives the details of the observed failure time and calculated mean time to failure.

Table 8. Comparison of the observed failure time and calculated mean time to failure

Stress level	Calculated using MTTF IPL-Weibull	Observed Failure Time
100	91.3	110
200	130.4	124
500	182.6	167
1000	228.5	211
2000	304.3	310
4000	456.5	467

The MTTF has been calculated for six different stress levels and it is compared with the observed ones. The root mean square error (RMSE) is calculated from the observed and predicted values. The RMSE is given Eq.(8)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{8}$$

The root mean square is 13.43, which is reasonably good for the values. The normalized root mean square error shows only 4% deviation from the observed results. i.e. 96% accuracy in terms of MTTF.

5. CONCLUSION

The approach considerably reduced the experimental time to predict the mean time to failure. It has introduced a new approach to predict the rejuvenation time. Another contribution worth mentioning here is that to the best of our knowledge this is first time this kind of experiment strategy has performed on the server virtualized systems. This approach may be effectively utilized for predicting the migration time of a Virtual Machine. The results obtained are reasonably good at the experimented level and it could be tested at lower stress levels. The experimental results presented above confirm that the tested experimental plan is a good starting point for future efforts. The life stress relationship IPL-Weibull model showed very good accuracy for predicting the failures which helps system administrators to avoid unnecessary outages. It may motivate other researchers to apply it to similar software systems.

The paper could be extended by comparing the existing technique with some machine learning techniques or Regression models. This model could be used to predict the rejuvenation time in physical routers/networking devices or embedded systems when there are signs of performance degradation.

6. REFERENCES

[1] Yin, Pengcheng, and ChengdongWang. "Life-prediction of accelerated life testing based on support vector machine." Quality, Reliability, Risk, Maintenance, and Safety Engineering (ICQR2MSE), 2011 International Conference on. IEEE, 2011.

[2] Shuzhen Li; Xiaoyang Li; Tongmin Jiang, "Life and reliability forecasting of the CSADT using Support Vector Machines," Reliability and Maintainability Symposium (RAMS), 2010 Proceedings - Annual , vol., no., pp.1,6, 25-28 Jan. 2010.

[3] 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.

[4] 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.

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

[6] Huang, Tingting, and Tongmin Jiang. "Design of accelerated life testing using proportional hazards-proportional odds." Reliability and Maintainability Symposium (RAMS), 2010 Proceedings-Annual. IEEE, 2010.

[7] Alonso, Javier, Jordi Torres, and Ricard Gavalda. "Predicting web server crashes: A case study in comparing prediction algorithms." Autonomic and Autonomous Systems, 2009. ICAS'09. Fifth International Conference on. IEEE, 2009.

[8] Yuan, Tao, and Xi Liu. "Bayesian planning of optimal step-stress accelerated life test." Reliability and Maintainability Symposium (RAMS), 2011 Proceedings Annual. IEEE, 2011.
[9] B. N. Nelson, Accelerated testing: statistical method, test plans, and data analysis, New Jersey: Wiley, 2004.
[10] A. Mettas, “Understanding Accelerated Life Testing Analysis,” In Proc. of International Reliability Symposium, 1-16, 2003.
[11] W. Q. Meeker, and L. A. Escobar, Statistical Methods for Reliability Data. New York: Wiley, 1998.
[12] http://www.hpl.hp.com/research/linux/httperf/