International Journal of Computer Applications (0975 – 8887)
Volume 88 – No.14, February 2014

Generalized Reliability Model for Cloud Computing

Nikita Yadav
Research Scholar
Singhania University
Pacheri Bari, Rajasthan

V B Singh
Delhi College of Arts & Commerce
University of Delhi, Delhi

Madhu Kumari
Delhi College of Arts & Commerce
University of Delhi, Delhi

ABSTRACT

Performance of cloud computing depends on effective utilization of resources and reliability. With resource allocation algorithms such as banker's algorithm resource utilization can be done in an effective manner in cloud computing. With reliability we can estimate the fault tolerance capability of a system. Reliability improvement is largely dependent on the availability of operational profile that statistically models the pattern in which the system is more likely to be used in the operating environment. System is less reliable if it exhibits a degree of hardware and software dependency and more reliable if hardware and software failure occur independently. In Cloud computing environment, hundreds of thousands of systems are hosted that consume cloud computing services. These services have of lots of hardware, software platform and infrastructure support, each of which though carefully engineered are still capable of failure. These failure rates and complexity of database make cloud less reliable.

In this paper, we have proposed a reliability model that estimates the mean time to failure and failure rate based on delayed exponential distribution. Through this model, we study the effect of older and newer systems on cloud computing reliability that consumes the cloud computing services.

Keywords

Reliability, Cloud computing, Exponential distribution

1. INTRODUCTION

Cloud computing offer its services to hundreds and thousands of computers that allow massive scale sharing of resources and services. Cloud providers have to maintain and manage numbers of cloud computing resources and services that they provide to their customers. These services consist of lots of hardware like multiple hard disks, memory modules, network card, processor etc. each of which though carefully engineered are still capable of failures. The probability of failures rate in the lifetime of cloud computing magnifies with increase in number of systems in cloud computing environment. A single failure may interrupt entire active applications. Different systems have different hardware support that further increases the complexity and failure rate of services of cloud.

Complexity $\propto$ Failure rate $\propto$ 1/Reliability

For the past four decades, software reliability modeling has been one of the most active areas in software engineering. More than hundred models have been proposed, each with its own assumptions, applicability and limitations. To make cloud more reliable, failure rates and complexity of database need to be minimized. When a system fails, it affects the performance of cloud as lots of allocated resources wasted. On cloud computing lots of big web-sites depend and a minute of down-time of cloud can result in big penality as several big web-sites will be knocked out of services. To make cloud more reliable, Amazon engineered its avaiability zones that are designed to be insulated from failures in other availability zones. Application running in more than one availability zone can achieve higher availability [9].

Reliability is one of the key factor to be considered in cloud computing environment. Reliability is defined as the probability that a given item will perform its intended function for a given period of time under a given set of conditions. Cloud reliability means availability of cloud services even when several of its component fails. A cloud will be more reliable if it is more fault-tolerant and more adaptable to changing situations. It is impossible to have a cloud that is completely failure resistant. Various types of failures are interleaved in the cloud computing environment such as overflow failure, timeout failure, resource missing failure, network failure, hardware failure, software failure, and database failure [4]. In cloud computing environment if the service providers know the failure characteristics of cloud computing components, they can manage the computing resource in better way to tolerate failures and sustain better quality and performance.

2. THE EXPONENTIAL AND MULTIVARIATE WEIBULL DISTRIBUTION

The exponential distribution, another special case of the Weibull distribution family, is the simplest and perhaps most widely used distribution in reliability and survival studies of software. In the study of continous time stochastic processes, the exponential distribution is usually used to model the time until something happen in the process. The exponential model can be regarded as the basic form of the software reliability growth models. In this work, we will refer to applications, VM's, kernel and hypervisor as software as they all follow an exponential distribution time till failure distribution. A continous random variable X is said to have an exponential (λ) distribution if it has probability density function

$$fX(x|\lambda)=\begin{cases}\lambda e^{-\lambda x} & \text{for } x > 0\\ 0 & \text{for } x \leq 0\end{cases}$$

where $\lambda>0$ is called the rate of the distribution.

To study the correlation between multiple software components, Marshall-Olkin Multivariate Exponential Distribution (MOMED) based on fatal shock model is as follows[2].

$$\bar{F}_{Y1\ldots..Y_k}(\mathrm{y}_1\ldots\ldots\mathrm{y}_\mathrm{k}) = \exp\{\sum_{i=1}^{k}\gamma_i y_i - \sum_{i<j}\gamma_{ij}\max(y_i,y_j) - \sum_{i<j<l}\gamma_{ijl}\max(y_i,y_j,y_l) - \cdots. \quad (1)$$
$$-\gamma_{12..k}\max(y_1,y_2,\ldots\ldots.,y_k)\}$$

Depending upon the types of job software performing parallel or serial, there failures may be independent or dependent. Safety critical systems, video encoding are the case of independent software failure where the systems are much reliable and a single node failure don't effect working of whole system. Dependent failure may occur due to common components failure i.e. hardware or software and due to supply failure. Many studies have shown that time to failure of an individual node follows a Weibull distribution [2]. Nodes that follows a dependent failure ie more than one node failing at the same time multivariate Weibull distribution is used to study the hardware reliability model. Multivariate Weibull model using the transformation of variable technique used in [10], distribution is as follows:

$$\bar{F}_{Y1\ldots..Y_k}(\mathrm{y}_1\ldots\ldots\mathrm{y}_\mathrm{k}) = \exp\{\sum_{i=1}^{k}\lambda_i x_i^c - \sum_{i<j}\lambda_{ij}\max(x_i^c,x_j^c) - \sum_{i<j<l}\lambda_{ijl}\max(x_i^c,x_j^c,x_l^c) - \cdots. \quad (2)$$
$$-\lambda_{12..k}\max(x_i^c,\ldots\ldots.,x_k^c)\}$$

3. GENERALISED RELIABILITY MODEL

Thanadesh [1] has proposed four scenario for cloud system reliability model by considering Eqs.1 for software's and Eq.2 for hardware's. His four major scenarios is described as follows.

(i) Software failures occur independently. Also, hardware failures occur independently.

$$R_j(x) = \bar{F}(x) = P(X_1 > x, X_2 > x\ldots X_{k+n} > x) = \exp\{-\sum_{i=1}^{k}\lambda_i x_i - \sum_{v=1}^{n}\gamma_v x\}, \quad (3)$$

(ii) Correlated software failures and independent hardware failures

$$R_j(x) = \exp\{-\sum_{i=1}^{k}\lambda_i x_i - \sum_{v=1}^{n}\gamma_v x - \sum_{v,w=1}^{n}\gamma_{vw} x - \sum_{v,w,z=1}^{n}\gamma_{vwz} x - \ldots. - \gamma_{12..n} x\} \quad (4)$$

(iii) Software failures occur independently and hardware failures are correlated

$$R_j(x) = \exp\{\{-\sum_{i=1}^{k}\lambda_i x_i - \sum_{i,s=1}\lambda_{is}\max(x_i,x_s) - \sum_{i,s,l=1}\lambda_{isl}\max(x_i,x_s,x_l) - \cdots.$$
$$-\lambda_{12..k}\max(x_1,x_2\ldots,x_k)\} - \sum_{v=1}^{n}\gamma_v x\} \quad (5)$$

(iv) Both software and hardware correlated failures

$$R_j(x) = \exp\{-\sum_{i=1}^{k}\lambda_i x_i - \sum_{i,s=1}\lambda_{is}\max(x_i,x_s) - \sum_{i,s,l=1}\lambda_{isl}\max(x_i,x_s,x_l) - \cdots.$$
$$-\lambda_{12..k}\max(x_1,x_2\ldots,x_k) - \sum_{v=1}^{n}\gamma_v x - \sum_{v,w=1}^{n}\gamma_{vw} x - \sum_{v,w,z=1}^{n}\gamma_{vwz} x - \ldots. - \gamma_{12..n} x\} \quad (6)$$

The basic assumptions after proposed work was that the software failure follows an exponential distribution and the author used Marshall-Olkin Multivariate Exponential Distribution (MOMED) as a base model for their proposed derivations. The author stated in the model proposed for scenario 1 in Eq.3 that the cloud system reliability is governed by exponential distribution. The authors considered two parameters λ_i x_i to represent hardware reliability and γ_v x to represent software reliability. In the proposed work, the author assumed that all the n software components running on k physical nodes (i.e. Hardware) follows the same distribution. But this is not possible in practical scenario.

In this paper, it has been assumed that all the n software components will not follow the same distribution as different components have been developed in different environments. Bugs are inevitable in any software development life cycle. In cloud computing environment hundred and thousands of systems are involved. Out of these systems, some may be new and some may be reused software components. Software Reliability is hard to achieve, because the complexity of software tends to be high. The more severe the bug, the more it will affect the functioning of software and more complex the bug, the more time it will take for debugging and correction process. Most of the bugs are detected and removed in the testing phase. With time, software are well understood and extensively tested in different working environment. Older the software component means more reliable.

Figure1 [8] shows the reliability of software with time.

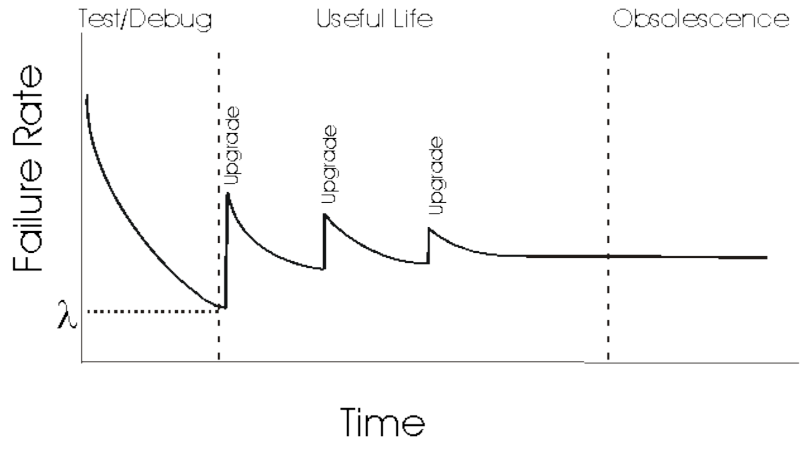

Fig1: Curve for software reliability

Here, we assumed that the components that are new in cloud computing environment will follow the exponential distribution. These new components will be more prone to failures as bugs will occur frequently here. Other reused components or older software components will follow delayed and more delayed exponential distribution depending upon their arrival time. We considered an n software components running on k number of physical nodes. Failure rate in any one of the n components will follow the below distribution [5] based on exponential and S-shaped curves.

$$m(t) = \sum_{i=1}^{n} ap_i \left[1 - \exp\left(-bq_i E(t)\right) \left[\sum_{j=0}^{i-1} \frac{(bq_i E(t))^j}{j!}\right]\right] \quad (7)$$

where : mean value function of the expected number of detected/removed bugs the time interval [0,t].

$\boldsymbol{i}$: Type of bug.
j : number of stages for removing a bug in the software .

p_i : proportion of type i in bugs
q_i : Proportion of type i accession
$a_i\left(= ap_i\right)$: Initial content of type i bug
b_i : bug removal rate per bug for type i

Khatri and singh [5] have developed a generalized software reliability growth model for object oriented software system by considering the accession of private, protected and public variables. It was considered that due to accession of private variable the bugs will occur which follow exponential distribution and due to accession of protected and public variables the bugs will occur and follow more delayed exponential distributions. The accession will determine the proportion of bugs of different complexity and severity depending on the testing environment. Following Eqs. (1) and (7) we can consider that reliability model of n software components can be expressed as follows:

$$\bar{F}_{Y1.....Y_k}(y_1......y_k) = \exp\{-\sum_{i=1}^{k}\sum_{j=0}^{i-1} {(\gamma_i x)^j}/{j!} - \sum_{i<m,m=1}^{k}\sum_{j=0}^{m-1} {(\gamma_{im} x)^j}/{j!} - \sum_{i<m<n,n=1}^{k}\sum_{j=0}^{i-1} {(\gamma_{imn} x)^j}/{j!} - \gamma_{12..k}\, {(x)^{k-1}}/{k-1!}\} \quad (8)$$

Here we have assume bqi = γ_i and E(t) = x. γ is representing parameter and x is representing maximum reliability of reused or older software components.

Now, taking four scenario of Thanadesh proposed work and fitting Eqs. (8) for software and Eqs. (2) for hardware we get reliability model as follows:

(i) Hardware failures occur independently. Also, software failures occur independently.

$$R_j(x) = \bar{F}(x) = P(X_1 > x, X_2 > x...X_{k+n} > x) = \exp\{-\sum_{i=1}^{k} \lambda_i x_i - \sum_{v=1}^{n}\sum_{j=0}^{v-1} {(\gamma_v x)^j}/{j!}\}, \quad (9)$$

(ii) Independent hardware failures and correlated software failures

$$R_j(x) = \exp\{\sum_{i=1}^{k} \lambda_i x_i - \sum_{v=1}^{n}\sum_{j=0}^{i-1} {(\gamma_v x)^j}/{j!} - \sum_{v<w,w=1}^{n}\sum_{j=0}^{w-1} {(\gamma_{vw} x)^j}/{j!} - \sum_{v<w<z,z=1}^{n}\sum_{j=0}^{z-1} {(\gamma_{vwz} x)^j}/{j!} - \gamma_{12..n}\, {(x)^{n-1}}/{n-1!}\} \quad (10)$$

(iii) Hardware failures are correlated and Software failures occur independently

$$R_j(x) = \exp\{\{-\sum_{i=1}^{k}\lambda_i x_i - \sum_{i,s=1}\lambda_{is}\max(x_i,x_s) - \sum_{i,s,l=1}\lambda_{isl}\max(x_i,x_s,x_l) - \cdots. -\lambda_{12..k}\max(x_1,x_2\ldots,x_k)\} - \sum_{v=1}^{n}\sum_{j=0}^{i-1}{(\gamma_v x)^j}/{j!}\ldots\} \quad (11)$$

(iv) Both hardware and software correlated failures

$$R_j(x) = \exp\{-\sum_{i=1}^{k}\lambda_i x_i - \sum_{i,s=1}\lambda_{is}\max(x_i,x_s) - \sum_{i,s,l=1}\lambda_{isl}\max(x_i,x_s,x_l) - \cdots. -\lambda_{12..k}\max(x_1,x_2\ldots,x_k) - \sum_{v=1}^{n}\sum_{j=0}^{i-1}{(\gamma_v x)^j}/{j!} - \sum_{v<w,w=1}^{n}\sum_{j=0}^{w-1}{(\gamma_{vw} x)^j}/{j!} - \sum_{v<w<z,z=1}^{n}\sum_{j=0}^{z-1}{(\gamma_{vwz} x)^j}/{j!} - \gamma_{12..n}{(x)^{n-1}}/{n-1!} \quad (12)$$

4. CONCLUSION

In cloud computing environment, reliability and survival of system is an important factor. The main purpose of analyzing system reliability is detecting weak components and upgrading quality and performance of all over the system. To study the system reliability it is important to understand the correlation between hardware and software components of the system. As known from the literature [6] and [7] hardware failure have correlations between them. Cloud computing support hundred and thousands of VM's on hypervisor for running client applications on physical servers. In this paper, we proposed reliability model that is based on four scenario proposed by Thanadesh [1]. The model that we have proposed is based on Weibull distribution for hardware and delayed exponential for software. Different software are developed and tested in different environment, have different reliability and therefore follows different exponential distribution.

From our work, we conclude that software who has minimum bug complexity and minimum operating time are much reliable. These software follows delayed and more delayed exponential distribution while newer software component follows exponential distribution that are more prone to failure's and less reliable. Our model supposed to reflect the improvement in cloud computing reliability as less reliable components are detected and redeveloped.

This study can also be further extended and can be applied in practical life to increase the confidence in the proposed models. Due to unavailability of data, numerical has not been done in this paper. In future, we will include numerical part based on generalized reliability model.

5. REFERENCES

[1] Thanadesh Thanakornworakij, Raja F. Nassar, Chokchai Leangsuksun, and Mihaela Păun: A Reliability Model for Cloud Computing for High Performance Computing Applications. In Springer-Verlag Berlin Heidelberg 2013, Euro-Par 2012 Workshops, LNCS 7640, pp. 474–483, 2013.

[2] Marshall Marshall, A.W., Olkin, I.: A multivariate exponential distribution. Journal of the American Statistical Association 62, 30–44 (1967).

[3] Schroeder, B., Gibson, G.A.: A large-scale study of failures in high-performance compu- ting systems. In: Proceedings of International Symposium on Dependable Systems and Networks, DSN, pp. 249–258. IEEE Computer Society (2006).

[4] Dai, Y.S., Yang, B., Dongarra, J., Zhang, G.: Cloud Service Reliability: Modeling and Analysis. In: The 15th IEEE Pacific Rim International Symposium on Dependable Com- puting (2009).

[5] Sujata Khatri, R.S.Chhillar, V.B.Singh: Measuring Bug Complexity in Object Oriented Software System. In ACM SIGSOFT Software Engineering Notes, volume 36 Issue 6, November 2011, pages 1-8.

[6] Xu, J., Kalbarczyk, Z., Iyer, R.K.: Networked Windows NT system field failure data analysis. In: Proceedings of the 1999 Pacific Rim International Symposium on Dependable Computing, pp. 178–185 (1999).

[7] Gottumukkala, N.R., Nassar, R., Paun, M., Leangsuksun, C.B., Scott, S.L.: Reliability of a System of k Nodes for High Performance Computing Applications. IEEE Transactions on Reliability 59(1), 162–169 (2010).

[8] Jiantao Pan, Carnegie Mellon University: software Reliability. In 18-849b Dependable Embedded Systems , Spring 1999.

[9] en.wikipedia.org/wiki/amazon-elastic-compute-cloud#reliability.

[10] Hanagal, D.D.: A multivariate Weibull distribution. Economic Quality Control 11, 193–200 (1996).

[11] Vishwanath, K.V., Nagappan, N.: Characterizing Cloud Computing Hardware Reliability. In: International Conference on Management of Data, pp. 193–204 (2010).

[12] Yi, S., Kondo, D., Andrzejak, A.: Reducing Costs of Spot Instances via Check pointing in the Amazon Elastic Compute Cloud. In: IEEE Cloud Computing, CLOUD, pp. 236–243 (2010).