Original Article

Institution of MECHANICAL ENGINEERS
Journal of RISK AND RELIABILITY

Proc IMechE Part O:
J Risk and Reliability
0(0) 1–8

DOI: 10.1177/1748006X12475110
pio.sagepub.com
SAGE

Reliability modeling and analysis for open source cloud computing

Yoshinobu Tamura[1], Masaru Kawakami[2] and Shigeru Yamada[2]

Abstract

At present, cloud computing is attracting attention as a network service to share the computing resources, such as networks, servers, storage, applications, and services. Also, many open source softwares are developed in all parts of the world, e.g. Firefox, Apache HTTP server, Linux, Android, etc. This article focuses on a cloud computing environment by using open source software such as OpenStack and Eucalyptus because of the unify management of data and low cost. This article proposes a new approach to software reliability assessment based on the stochastic differential equations and hierarchical Bayesian modeling in order to consider the interesting aspect of the network status of a cloud computing environment. Also, actual software fault-count data to show numerical examples of software reliability assessment considering the characteristics of network environment is analyzed.

Keywords

Cloud computing, reliability modeling, open source software, stochastic differential equation, hierarchical Bayesian modeling

Date received: 4 September 2012; accepted: 28 December 2012

Introduction

The successful experience of adopting the distributed development model in open source projects includes GNU/Linux operating system, Apache HTTP server, and so on.[1] However, the poor handling of quality and customer support prohibits the progress of open source software (OSS). This article focuses on the problem of software quality, which prohibits the progress of OSS. In particular, cloud OSS is now attracting attention as the next-generation software service paradigm because of the cost reduction, quick delivery, and work saving. Also, the cloud OSS has such unique features as a network service to share the computing resources. In particular, the cloud OSS has no testing phase. Then, it is important to assess the reliability during the operational phase of the cloud OSS. At present, there is a growing interest in the next-generation software development paradigm by using network computing technologies such as cloud computing. However, the source code of a cloud system based on OSS is open to all over the world. Therefore, the weakness of reliability and security for the cloud system is becoming a significant problem.

For example, the traditional OSSs include the application software, server one, embedded one, operating system, i.e. Firefox Web browser, Apache HTTP server, Android, Linux distributions, etc. These OSSs differ substantially from the cloud OSS because of the provisioning control in the operational form. Also, the difference between cloud computing and grid computing is that cloud computing can make the provisioning process to the resources depending on the usage situation. Therefore, it is important for the software managers to consider these changes of operational form in a cloud environment. Based on the above mentioned points, this article focuses on the environmental factor of the operational phase in cloud computing by using OSS. The network status, the change in the number of users, and the change in the number of software components as the environmental factor in cloud computing are considered in this article.

Many software reliability growth models (SRGMs)[2] have been applied to assess the reliability for quality

[1]Department of Electronic and Information System Engineering, Yamaguchi University, Yamaguchi, Japan
[2]Department of Social Management Engineering, Tottori University, Tottori, Japan

Corresponding author:
Department of Electronic and Information System Engineering, Graduate School of Science and Engineering, Yamaguchi University, Tokiwadai 2-16-1, Ube-shi, Yamaguchi, 755-8611, Japan.
Email: tamura@yamaguchi-u.ac.jp

management and testing progress control of software development. On the other hand, the effective method of dynamic testing management for a new distributed development paradigm, as typified by the open source project, has only a few presentations.[3–6] Also, there are some interesting research articles in terms of the cloud hardware, cloud service, and cloud performance evaluation.[7,8] However, most of them have focused on the case study of cloud service, and cloud data storage technology. The effective method of dynamic reliability assessment for cloud OSS has only a few presentations. Several methods of reliability assessment for the cloud system have been proposed.[9,10] However, these methods of reliability assessment for cloud systems are not focused on the change in operating environment of cloud computing, such as provisioning process. Moreover, the existing reliability assessment method for the cloud system does not considered the characteristics about OSS. Also, the cloud systems, by using OSS, have problems in terms of the security and reliability, because the source code is open. It is important for software managers to consider the change in the specification of OSS. Also, it is difficult for the existing SRGMs to describe the characteristics of the environmental factors of operational phase in cloud computing from the aspect of the model structure. In particular, it is important to consider the change of specification in OSS. Also, it is important to consider the characteristics of the cloud computing environment, i.e. the network status, the change in the number of users, and the change in software components. This article proposes a new stochastic differential equation model considering the above-mentioned points in terms of the provisioning process.

This article proposes a new approach to software reliability assessment based on the stochastic differential equations in order to consider the interesting aspect of network status of the cloud computing environment. Also, several reliability assessment measures from the proposed model are derived. Also, actual software fault-count data to show numerical examples of software reliability assessment considering the characteristics of network environment is analyzed. This article shows that the proposed reliability analysis can assist improvement of quality for the operational phase of the cloud OSS.

Stochastic differential equation model

Let $N(t)$ be the number of detected faults in the cloud OSS by operating time t ($t \geqslant 0$). Suppose that $N(t)$ takes on continuous real values. Since latent faults in the cloud OSS are detected and eliminated during the operational phase, $N(t)$ gradually increases as the operation procedures go on. Thus, under common assumptions for software reliability growth modeling, the following linear differential equation is considered in this article

$$\frac{dN(t)}{dt} = b(t)\{D(t) - N(t)\} \tag{1}$$

where $b(t)$ is the software fault-detection rate at operating time t and a non-negative function, and $D(t)$ means the amount of changes of fault-prone specification. Also, $D(t)$ is defined as

$$D(t) = \alpha e^{-\beta t} \tag{2}$$

where α is the number of latent faults in the OSS, and β the changing rate of fault-prone specification. This article assumes that the fault-prone specification of OSS grows exponentially in terms of t, as shown in Figure 1.[11,12] Thus, the OSS shows a reliability regression trend if β is a negative value. On the other hand, the OSS shows a reliability growth trend if β is a positive value.

This article focuses on the cloud computing environment. Considering the characteristic of such cloud computing, the software fault-reporting phenomena keeps an irregular state in the operational phase, because the network access of several users and the addition and deletion of software components are repeated under the operational environment, i.e. this article assumes that the frequency of use and OSS fault-prone specification depends on the operational time. Therefore, equation (1) can be extend to the following stochastic differential equation[13,14]

$$\frac{dN(t)}{dt} = \{b(t) + c(t)\sigma\gamma(t)\}\{D(t) - N(t)\} \tag{3}$$

where σ is a positive constant representing a magnitude of the irregular fluctuation, $\gamma(t)$ a standardized Gaussian white noise, and $c(t)$ a environmental function for cloud network.

Equation (3) can be extended to the following stochastic differential equation of an Itô type

$$\begin{aligned} dN(t) = &\left\{b(t) + \frac{1}{2}c(t)^2\sigma^2\right\}\{D(t) - N(t)\}dt \\ &+ c(t)\sigma\{D(t) - N(t)\}dW(t) \end{aligned} \tag{4}$$

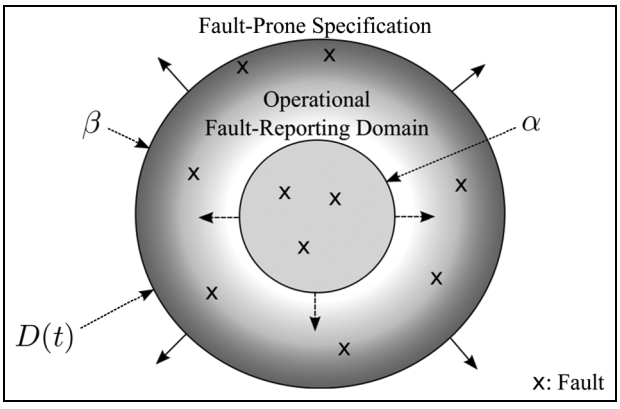

Figure 1. The basic concept of OSS fault-prone specification and fault-reporting domain.

where $W(t)$ is a one-dimensional Wiener process that is formally defined as an integration of the white noise $\gamma(t)$ with respect to time t. The Wiener process is a Gaussian process and it has the following properties

$$\Pr[W(0) = 0] = 1 \tag{5}$$
$$\mathrm{E}[W(t)] = 0 \tag{6}$$
$$\mathrm{E}[W(t)W(t')] = Min[t, t'] \tag{7}$$

where $\Pr[\cdot]$ and $\mathrm{E}[\cdot]$ represent the probability and expectation, respectively.

By using Itô's formula,[13,14] we can obtain the solution of equation (4) under the initial condition $N(0) = 0$ as follows[15]

$$N(t) = D(t)\left[1 - \exp\left\{-\int_0^t b(s)ds - c(t)\sigma W(t)\right\}\right] \tag{8}$$

Using solution process $N(t)$ in Eq.(8), we can derive several software reliability measures.

Moreover, the software fault detection rates per fault in case of $b(t) \equiv b_1(t)$ and $b(t) \equiv b_2(t)$ are defined as

$$\int_0^t b_1(s)ds = \frac{\frac{dN_e(t)}{dt}}{a - N_e(t)} \doteq \frac{\frac{dH_e(t)}{dt}}{a - H_e(t)} = b \tag{9}$$
$$\int_0^t b_2(s)ds = \frac{\frac{dN_s(t)}{dt}}{a - N_s(t)} \doteq \frac{\frac{dH_s(t)}{dt}}{a - H_s(t)} = \frac{b^2 t}{1 + bt} \tag{10}$$
$$c(t) = \frac{1 - e^{-\mu t}}{c_n} \tag{11}$$

where $H_e(t)$ and $H_s(t)$ mean the exponential SRGM and the delayed S-shaped SRGM, respectively, based on the nonhomogeneous Poisson process (NHPP), c_n the number of cloud software components, and μ the parameter of cloud environmental factor. Equation (11) means that $c(t)$ increases exponentially. However, $c(t)$ decreases with the increase of cloud software components because of the increase in system complexity.

Therefore, the cumulative number of detected faults of these two models are obtained as

$$N_e(t) = D(t)[1 - \exp\{-bt - c(t)\sigma W(t)\}] \tag{12}$$
$$N_s(t) = D(t)[1 - (1 + bt)\exp\{-bt - c(t)\sigma W(t)\}] \tag{13}$$

Parameter estimation

The estimation of environmental factor

This article applies the hierarchical Bayesian modeling to the estimation of the parameter $\mu(= \mu_0)$ in terms of the cloud environmental factor included in the proposed stochastic differential equation model, because the stability of the cloud computing depends on several cloud application components. The software managers can understand the confidence interval of stability for the cloud computing environment by using the hierarchical Bayesian model, i.e. the software managers can identify the unstable cloud component of the cloud computing environment., This article can effectively approach the above mentioned points with a probabilistic method by using the stochastic differential equations and the hierarchical Bayesian model. Then, the fault detection rate for each cloud component is assumed as the normal distribution, because the unstable status for each cloud component keeps the specified steady state.

Generally, the fault-detection rate of the cloud OSS changes according to the change of operating environment. Therefore, this article assumes that the fault detection rate at time t is the normal distribution as follows

$$f(t) = \frac{1}{\sqrt{2\pi}\sigma_0}\exp\left(-\frac{(t - \mu_0)^2}{2\sigma_0^2}\right) \tag{14}$$

where μ_0 is the mean value and σ_0 the standard deviation in terms of the fault detection rate at operating time t. This article assumes the hierarchical Bayesian modeling based on the prior distribution and hyper prior distribution composed of μ_0 and σ_0. Also, the prior distribution of μ_0 is defined as

$$q(t) = \frac{1}{\sqrt{2\pi}\sigma_1}\exp\left(-\frac{(t - \mu_1)^2}{2\sigma_1^2}\right) \tag{15}$$

where μ_1 is the mean value and σ_1 the standard deviation in terms of the occurrence probability for the data communication traffic on the cloud network.

Similarly, the prior distribution of σ_0 can be defined as

$$r(t) = \frac{\lambda^\alpha}{\Gamma(\alpha)}x^{\alpha-1}e^{-\frac{\lambda}{t}} \tag{16}$$

where α is the shape parameter and λ the scale parameter. The mean μ_2 and the standard deviation σ_2 of inverse gamma distribution as the occurrence probability for the access frequency on cloud system are given by

$$\frac{\lambda}{\alpha - 1}(\alpha > 1)$$

and

$$\frac{\lambda^2}{(\alpha - 1)^2(\alpha - 2)}(\alpha > 2)$$

This article assumes that μ_1, σ_1, μ_2, and σ_2 are given by the data recorded on the access log of the cloud system.

Let $D\ (d_k; k = 1, 2, \cdots, K)$ be the fault-detection rate in the cloud OSS by operation time t. Also, $\boldsymbol{\theta}(= \mu_0, \sigma_0)$ is the parameters at operation time t. $\boldsymbol{\theta}$ can be estimated by using the fault detection rate d_t up to time t. The prior distribution up to time $(t - 1)$ can use in this case. As an example, the updated data is given by the following equation based on Bayesian theory in the case that

the software developers have knowledge of the prior information $\boldsymbol{\theta}$ independently of data D

$$p(\boldsymbol{\theta}|D) \propto p(D|\boldsymbol{\theta})p(\boldsymbol{\theta}) \tag{17}$$

By using Markov chain Monte Carlo (MCMC) methods, we estimate μ_0 and σ_0 based on equation (17).

This article assumes that the target density $p(\theta_t)$ is $p(\mu_t, \sigma_t)$, D_t the $(y_1, y_2, \cdots, y_t)$. Therefore, $\alpha(\theta_t, \theta')$ is given by the following equation in this article

$$\alpha(\theta_t, \theta') = \frac{p(\theta'|D_t) f(\theta', \theta_t|D_t)}{p(\theta_t|D_t) f(\theta_t, \theta'|D_t)} \tag{18}$$

Then, it is assumed that the unknown parameter μ of the proposed stochastic differential equation model is the mean of standard deviation for cloud components. σ_t for each cloud component can be estimated by using above mentioned Metropolis-Hastings (MH) algorithm. It is especially important to consider the variation in each cloud component. This means that the stability of the cloud system can be determined by the variation in the cloud component. This article focuses on the standard deviation of proportion data of the fault-detection rate for the cloud component, because the software managers can compare the variation in cloud component as the non-dimensional parameter.

The method of maximum-likelihood

In this section, the estimation method of unknown parameters α, β, b, and σ in equation (8) is presented. Let us denote the joint probability distribution function of the process $N(t)$ as

$$\begin{aligned} &P(t_1,\ y_1;\ t_2,\ y_2;\ \cdots;\ t_K,\ y_K) \\ &\equiv \Pr[N(t_1) \leqslant y_1,\ \cdots,\ N(t_K) \leqslant y_K \\ &|\quad N(t_0) = 0] \end{aligned} \tag{19}$$

The density of equation (9) is shown as

$$\begin{aligned} &p(t_1,\ y_1;\ t_2,\ y_2;\ \cdots;\ t_K, y_K) \\ &\equiv \frac{\partial^K P(t_1,\ y_1;\ t_2, y_2;\ \cdots;\ t_K,\ y_K)}{\partial y_1 \partial y_2 \cdots \partial y_K} \end{aligned} \tag{20}$$

Since $N(t)$ takes on continuous values, the likelihood function, l, can be constructed for the observed data $(t_k, y_k)(k = 1, 2, \cdots, K)$ as

$$l = p(t_1,\ y_1;\ t_2,\ y_2;\ \cdots;\ t_K,\ y_K) \tag{21}$$

For convenience in mathematical manipulations, the following logarithmic likelihood function can use

$$L = \log l. \tag{22}$$

The maximum-likelihood estimates α^*, β^*, b^*, and σ^* are the values making L in equation (22) maximal. These can be obtained as the solutions of the following simultaneous likelihood equations[15]

$$\frac{\partial L}{\partial \alpha} = \frac{\partial L}{\partial \beta} = \frac{\partial L}{\partial b} = \frac{\partial L}{\partial \sigma} = 0 \tag{23}$$

Cloud OSS reliability assessment measures

Information on the cumulative number of detected faults in the system is important to estimate the situation of the progress on the debugging procedures. Since $N(t)$ is a random variable in the proposed model, its expected value can be a useful measure. These can be calculated from equation (8) as follows[15]

$$\mathrm{E}[N(t)] = D(t)\left[1 - \exp\left\{-\int_0^t b(s)ds + \frac{c(t)^2\sigma^2}{2}t\right\}\right] \tag{24}$$

where $E[N(t)]$ is the expected number of faults detected up to time t. Therefore, the expected number of detected faults, $E[N_e(t)]$ and $E[N_s(t)]$, are given by the following equations, respectively

$$\mathrm{E}[N_e(t)] = \alpha e^{-\beta}\left[1 - \exp\left\{-bt + \frac{c(t)^2\sigma^2}{2}t\right\}\right] \tag{25}$$

$$\mathrm{E}[N_s(t)] = \alpha e^{-\beta}\left[1 - (1 + bt)\cdot\exp\left\{-bt + \frac{c(t)^2\sigma^2}{2}t\right\}\right] \tag{26}$$

It is important for software managers to assess the number of latent faults according to the change of specification of OSS. Considering the change of fault-prone specification, this article assumes that the number of latent faults is given by equation (2).

Also, the number of remaining faults considering the change of fault prone specification can be obtained as

$$\mathrm{E}[N_r(t)] = \mathrm{E}[D(t) - N(t)] \tag{27}$$

Moreover, the mean time between software failures (MTBF) is useful to measure the property of the frequency of software failure-occurrence (or fault-detection). Then, the cumulative MTBF (denoted by $MTBF_C$) is approximately given by

$$MTBF_C(t) = \frac{t}{\mathrm{E}[N(t)]} \tag{28}$$

Procedures of reliability analysis

The procedures of reliability analysis based on the proposed method for cloud computing environment are shown as follows.

1. The software managers process the fault data in terms of the cumulative number of detected faults for the cloud OSS and data of the fault-detection rate in the operational phase of the cloud system for reliability analysis. Also, the software managers

collect the data in terms of the data communication traffic and access frequency from the cloud operational environment.
2. The software managers estimate the unknown parameters μ included our hierarchical Bayesian modeling by using MCMC method and the data of communication traffic and access frequency. Thereby, it becomes possible to comprehend the characteristics of the cloud network environment.
3. The software managers estimate the unknown parameters α, β, b, and σ included in our stochastic differential equation model by using the method of maximum likelihood.
4. It is useful for the software managers to understand the debugging progress in the operational phase of the cloud computing environment by using the expected number of detected faults, the expected number of remaining faults, and MTBF as software reliability assessment measures considering the characteristics of the cloud network environment.

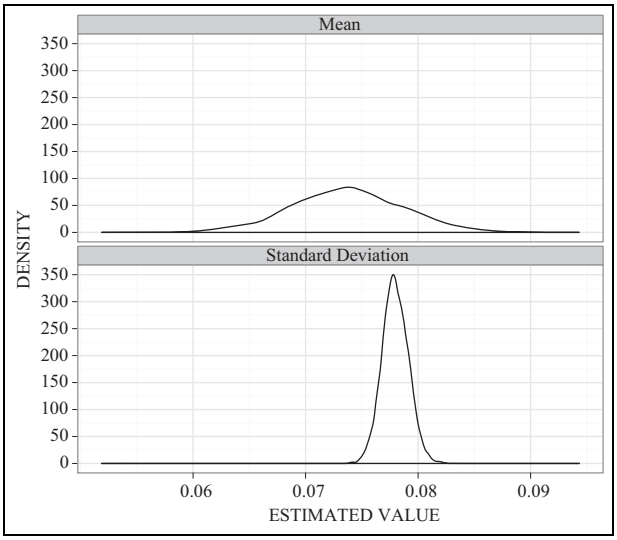

Figure 2. The estimated environmental factor in case of ($\mu_1 = \mu_2 = 0.25$, $\sigma_1 = \sigma_2 = 0.125$).

Numerical illustrations

Data for numerical illustrations

The OSS is closely watched from the point of view of cost reduction and quick delivery. There are several open source projects in the area of cloud computing. In particular, this article focuses on OpenStack[16] in order to evaluate the performance of our method. This article shows numerical examples by using the data sets for OpenStack of the cloud OSS. The fault-count data used in this article are collected in the bug tracking system on the website of the OpenStack open source project. These data sets are in the comma separated value (CSV) format from bug tracking system. Moreover, the CSV data obtained from bug tracking system can be recounted as the fault-count data.

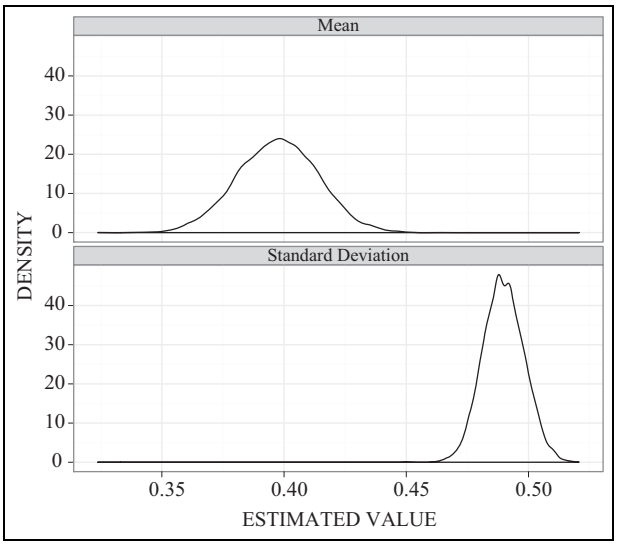

Figure 3. The estimated environmental factor in case of ($\mu_1 = \mu_2 = 0.5$, $\sigma_1 = \sigma_2 = 0.25$).

Reliability assessment results

The estimated environmental factors in the case of ($\mu_1 = \mu_2 = 0.25$, $\sigma_1 = \sigma_2 = 0.125$) and ($\mu_1 = \mu_2 = 0.5$, $\sigma_1 = \sigma_2 = 0.25$) are shown in Figures 2 and 3, respectively. Figure 2 shows that the environmental factor is small, if the data communication traffic and access frequency on the cloud system are small. On the other hand, the environmental factor is large, if the data communication traffic and access frequency on the cloud system are large.

The estimated expected numbers of detected faults in equations (25) and (26), $\widehat{\mathrm{E}}[N_e(t)]$ and $\widehat{\mathrm{E}}[N_s(t)]$, in case of $b(t) \equiv b_1(t)$ and $b(t) \equiv b_2(t)$ are respectively shown in Figures 4 and 5. From Figure 4, the case of $b_2(t)$ has the reliability growth trend. Also, the sensitivity analysis for the estimated number of detected faults in cases of $\mu = 0.0737$ and $b_2(t)$ is shown in Figure 6. Figure 6 means that the number of detected faults increases as the number of cloud software components increases.

Moreover, the sample paths of the estimated numbers of detected faults in equations (12) and (13), $N_e(t)$ and $N_s(t)$, in the case of $b(t) \equiv b_1(t)$ and $b(t) \equiv b_2(t)$, are shown respectively in Figures 7 and 8, approximately. From Figures 7 and 8, the noise of the sample path in the case of $b(t) \equiv b_2(t)$ and $\mu = 0.0737$ decreases as operation procedures go on, i.e. $\mu = 0.0737$ means that the data communication traffic and access frequency on the cloud system are small. On the other hand, the noises of the sample path in the case of $b_1(t)$, $b_2(t)$, and $\mu = 0.3974$ increase as operation procedures go on, i.e.

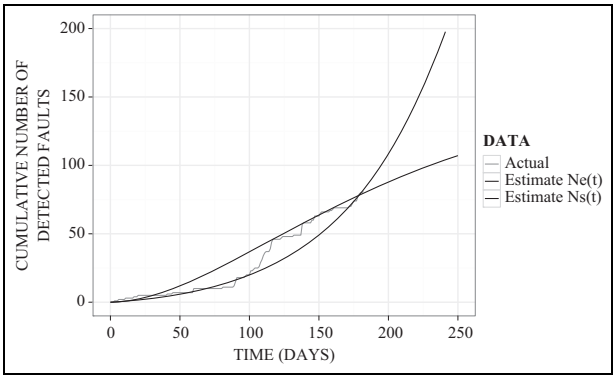

Figure 4. The estimated number of detected faults in case of $\mu = 0.0737$.

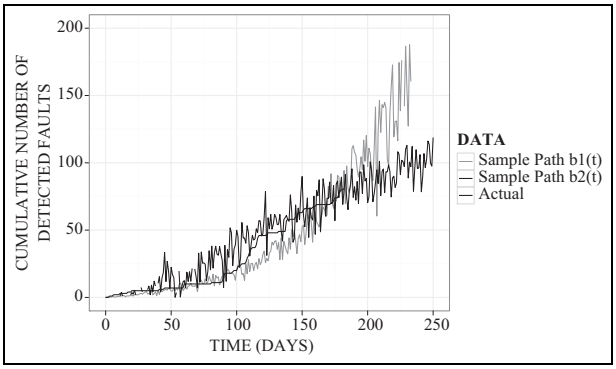

Figure 7. The sample path of the estimated number of detected faults in case of $\mu = 0.0737$.

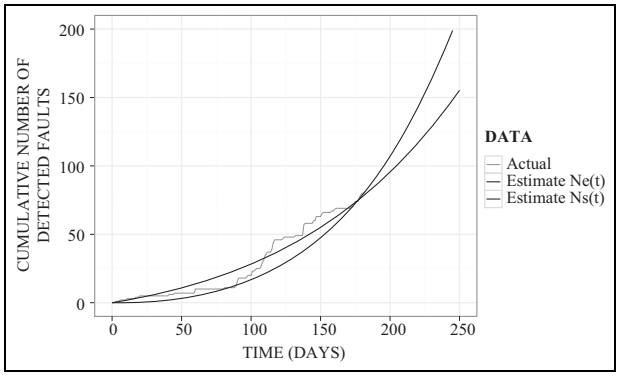

Figure 5. The estimated number of detected faults in case of $\mu = 0.3974$.

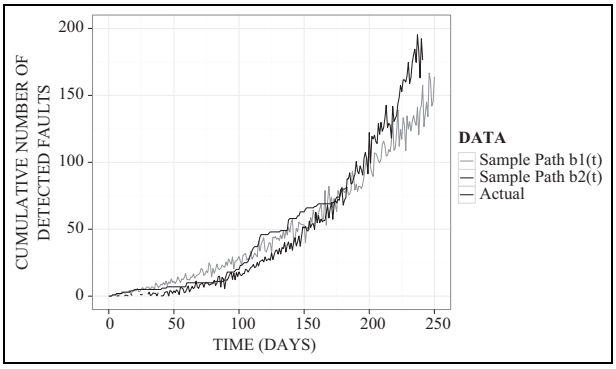

Figure 8. The sample path of the estimated number of detected faults in case of $\mu = 0.3974$.

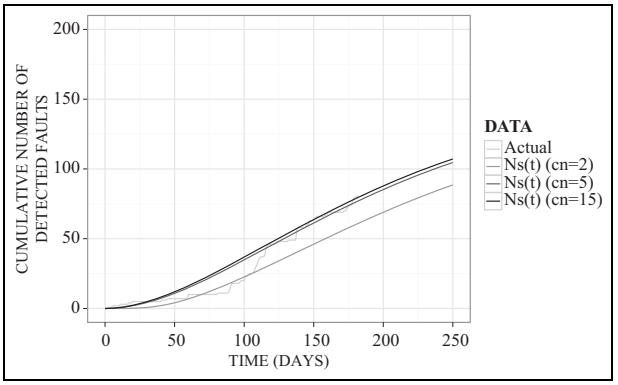

Figure 6. The sensitivity analysis for the estimated number of detected faults in cases of $\mu = 0.0737$ and $b_2(t)$.

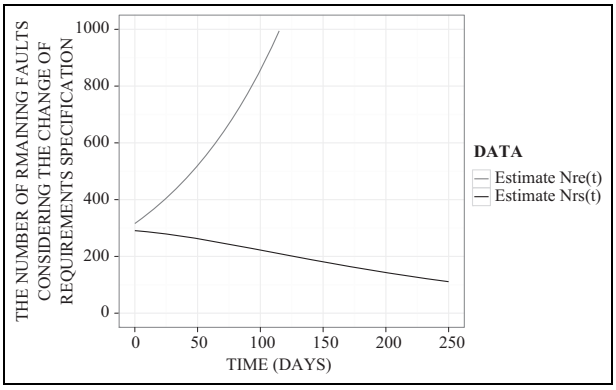

Figure 9. The number of remaining faults considering the change of requirements specification in case of $\mu = 0.0737$.

$\mu = 0.3974$ means that the data communication traffic and access frequency on the cloud system are large. From these results, the data communication traffic and access frequency on the cloud system affect the irregular fluctuation for the fault-detection rate of the cloud OSS. Moreover, the estimated number of remaining faults considering the change of requirements specification are shown in Figures 9 and 10. From Figure 9, the estimated number of remaining faults in the case of $b(t) \equiv b_2(t)$ decreases as the operation procedures go on. These figures are useful for software project managers to assess the potential of convergence.

The estimated cumulative MTBFs in equation (28), $MTBF_C(t)$ in the case of $b(t) \equiv b_1(t)$ and $b(t) \equiv b_2(t)$ are shown in Figures 11 and 12, respectively. From Figure 11, we can confirm that the estimated cumulative MTBFs in the case of $b(t) \equiv b_2(t)$ increases as the operating procedures go on. On the other hand,

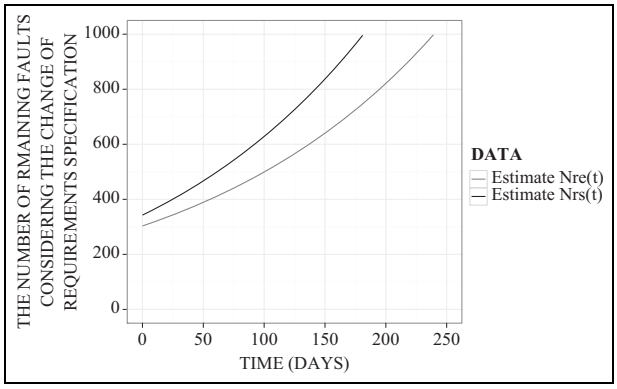

Figure 10. The number of remaining faults considering the change of requirements specification in case of $\mu = 0.3974$.

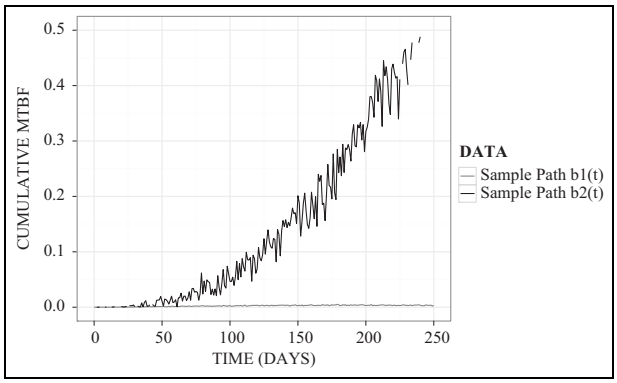

Figure 11. The estimated cumulative MTBF in case of $\mu = 0.0737$.

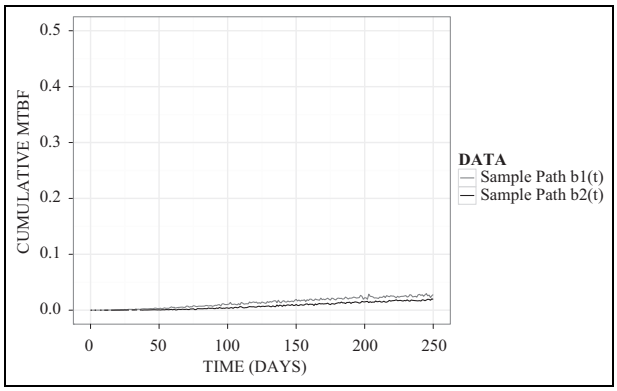

Figure 12. The estimated cumulative MTBF in case of $\mu = 0.3974$.

Figure 12 shows that the estimated cumulative MTBFs increase very little as the operating procedures go on.

The above-mentioned results show that the proposed model can describe the characteristics of the cloud environment according to the changes of the data communication traffic and access frequency on the cloud OSS. The proposed method will be useful to assess the reliability of the cloud computing environment considering the characteristics of network service.

Concluding remarks

This article has proposed a software reliability analysis based on stochastic differential equations in order to consider the data communication traffic and access frequency on the cloud system. This article has assumed that the software fault-detection rate depends on the time, and the software fault-reporting phenomena keep an irregular state in the operating phase. In particular, the hierarchical Bayesian modeling has been applied in order to consider the unique feature of the cloud OSS such as a network service to share the computing resources. Also, actual software fault-count data has been analyzed in order to show numerical examples of the software reliability assessment for the cloud OSS. Moreover, several reliability assessment measures have been derived from the proposed model.

At present, a new paradigm of distributed development typified by such an open source project will evolve at a rapid pace in the future. In particular, it is difficult to assess the reliability and security for the cloud system as a typical case of next-generation distributed software service paradigm. The proposed method may be useful as the method of reliability assessment for the cloud computing environment.

Funding

This work was supported in part by the Grant-in-Aid for Scientific Research (C) [Grant Nos 24500066 and 22510150] from the Ministry of Education, Culture, Sports, Science, and Technology of Japan.

References

1. E-Soft Inc. Internet research reports. Available from http://www.securityspace.com/s_survey/data/
2. Yamada S. *Elements of software reliability: modeling approach* (in Japanese). Kyoritsu–Shuppan, Tokyo, 2011.
3. MacCormack A, Rusnak J and Baldwin CY. Exploring the structure of complex software designs: an empirical study of open source and proprietary code. *Informs J Mgmt Sci* 2006; 52(7): 1015–1030.
4. Kuk G. Strategic interaction and knowledge sharing in the KDE developer mailing list. *Informs J Mgmt Sci* 2006; 52(7): 1031–1042.
5. Zhoum Y and Davis J. Open source software reliability model: an empirical approach. In: *Proceedings of the workshop on open source software engineering* (WOSSE) St Louis, MO, USA, 17 May 2005, 30(4): pp.67–72 ACM, NY.
6. Li P, Shaw M, Herbsleb J, et al. Empirical evaluation of defect projection models for widely-deployed production software systems. In: *Proceeding of the 12th international symposium on the foundations of software engineering* (FSE-12) Cary, North Carolina, USA, 11-16 November 2012, pp.263–272 ACM, NY.
7. Yang B, Tan F and Dai YS. Performance evaluation of cloud service considering fault recovery. *J Supercomputing*. Epub ahead of print, 23 February 2011. DOI number: 10.1007/978-3-642-10665-1_54.

8. Iosup A, Ostermann S, Yigitbasi MN, et al. Performance analysis of cloud computing services for many-tasks scientific computing. *IEEE Trans Parallel and Distributed Sys* 2011; 22(6): 931–945.
9. Fu Q, Cai ZH and Wu YQ. A new method for reliability evaluation based on cloud model. In: *Proceedings of the 2nd international conference on information technology and computer science* Kiev, Ukraine, 24–25 July 2010, pp.118–121. IEEE Computer Society, CA.
10. Huang J, Lin C, Kong X, et al. Modeling and analysis of dependability attributes of service computing systems. In: *Proceedings of the IEEE international conference on services computing* Cary, Washington, DC, USA, 4–9 July 2011, pp.184–191. IEEE Computer Society, CA.
11. Fujiwara T and Yamada S. A testing-domain dependent software reliability growth model for practical application. In: *Proceedings of the 2nd world congress for software quality* Yokohama, Japan, 25–29, September 2000, 2000, pp.821–826. Union of Japanese Scientists and Engineers, Tokyo, Japan.
12. Yamada S and Fujiwara T. Testing-domain dependent software reliability growth models and their comparisons of goodness-of-fit. *Int J Rel, Qual Saf Engng* 2001; 8(3): 205–218.
13. Arnold L. *Stochastic differential equations–theory and applications*. New York: John Wiley & Sons, 1974.
14. Wong E. *Stochastic processes in information and systems*. New York: McGraw-Hill, 1971.
15. Yamada S, Kimura M, Tanaka H, et al. Software reliability measurement and assessment with stochastic differential equations. *IEICE Trans Fundamentals* 1994; E77–A(1): 109–116.
16. The OpenStack project, OpenStack. [Online]. Available from http://www.openstack.org/