International Transactions on Systems Science and Applications
Volume 8 ▪ December 2012 ▪ pp. 1-11
sai: itssa.0008.2012.015

ISSN 1751-1461 (print)
ISSN 2051-5642 (online)

Performance Evaluation and Dependability Analysis for Open Source Cloud Computing

Y. Tamura [a] and **S. Yamada** [b]

[a] Yamaguchi University, Japan
[b] Tottori University, Japan
Email: tamura@yamaguchi-u.ac.jp; yamada@sse.tottori-u.ac.jp

Abstract: At present, many open source software are developed in all parts of the world, i.e., Firefox, Apache HTTP server, Linux, Android, etc. In particular, a cloud computing is an attracting attention as network service to share the computing resources, i.e., networks, servers, storage, applications, and services. This paper focuses on a cloud computing environment by using open source software such as OpenStack and Eucalyptus because of the unification management of data, and low cost. A new approach to software dependability assessment based on a jump diffusion model and stochastic differential equations in order to consider the interesting aspect of the numbers of components and users is proposed in this paper. In particular, several sensitivity analyses in terms of the proposed model parameters are shown in this paper. Also, sets of actual data are analyzed to show numerical examples of software dependability assessment considering such characteristics of cloud computing environment.

Keywords: cloud computing, dependability, modeling, open source software.

1. Introduction

The successful experience of adopting the distributed development model in such open source projects includes GNU/Linux operating system, Apache HTTP server, etc[1]. However, the poor handling of the quality and customer support prohibits the progress of OSS. This paper focuses on the problems of software quality, which prohibit the progress of OSS. Many software reliability growth models (SRGM's)[2] have been applied to assess the reliability for quality management and testing-progress control in the software development. On the other hand, the effective method of dynamic testing management for new distributed development paradigm as typified by open source projects has only a few presented[3,4,5,6]. Also, there are some interesting research papers in terms of the cloud hardware, cloud service, and cloud performance evaluation[7,8]. However, most of them have focused on the case studies of cloud service and cloud data storage technology. The effective method of dynamic dependability assessment considering the network environment such as cloud computing has only a few presented.

A 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 a unique feature such as a network service to share the computing resources. Especially, the cloud OSS has no testing phase. Then, it is important to assess the dynamic 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

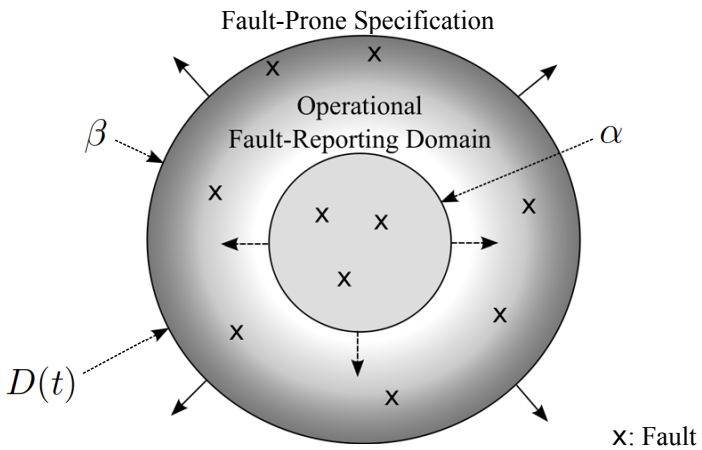

Figure 1. The basic concept of OSS fault-prone specifications and fault-reporting domain.

source code of cloud systems based on OSS's is opened around the world. Therefore, the weakness of reliability and security for the cloud system is becoming a significant problem.

In this paper, a new approach to software dependability assessment based on a stochastic differential equation model in order to consider the interesting aspect of the numbers of components and users is presented. Also, several dependability assessment measures from the proposed model are derived. Moreover, sets of actual data are analyzed to show numerical examples of software dependability assessment considering the characteristics of environment of cloud computing. Especially, several sensitivity analysis in terms of the proposed model parameters are shown in this paper. Moreover, this paper compares the jump-diffusion models with the stochastic differential equation ones in terms of goodness-of-fit for actual data. Then, the proposed dependability analysis can assist improvement of quality for the operational phase of cloud computing environment is shown in this paper.

2. Dependability Modeling for Open Source Cloud Computing

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

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

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 OSS fault-prone specifications. Also, $D(t)$ is defined as follows:

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

where α is the number of faults latent in the OSS, and β the changing rate of fault-prone specifications. This paper assumes that the fault-prone specifications of OSS grow exponentially in terms of t as shown in Fig. 1[9,10]. Thus, the OSS shows a reliability regression trend if β is negative. On the other hand, the OSS shows a reliability growth trend if β is positive.

This paper focuses on the cloud computing environment. Considering the characteristic of such cloud computing, the software fault-reporting phenomena keep an irregular state in the operation

phase, because the network access of several users and the addition/deletion of software components are repeated under the operation environment, i.e., the software managers have to consider that the OSS usage frequency and OSS fault-prone specification depends on the operating time. Therefore, Eq.(2) is extended to the following stochastic differential equation[11,12]:

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

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

Eq.(3) can extend to the following stochastic differential equation of an Itô type:

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

where $W(t)$ is a one-dimensional Wiener process which 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:

$$\Pr[W(0) = 0] = 1, \tag{5}$$

$$\mathrm{E}[W(t)] = 0, \tag{6}$$

$$\mathrm{E}[W(t)W(t')] = \mathrm{Min}[t, t'], \tag{7}$$

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

By using Itô's formula[11,12], the solution of Eq.(4) is obtained under the initial condition $N(0) = 0$ as follows[13]:

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

Using solution process $N(t)$ in Eq.(8), several software reliability measures are derived.

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

$$\int_0^t b_1(s)ds = \frac{\frac{dN_e(t)}{dt}}{a - N_e(t)} \doteq \frac{\frac{dH_e(t)}{dt}}{a - H_e(t)} = b, \tag{9}$$

$$\int_0^t b_2(s)ds = \frac{\frac{dN_s(t)}{dt}}{a - N_s(t)} \doteq \frac{\frac{dH_s(t)}{dt}}{a - H_s(t)} = \frac{b^2 t}{1 + bt}, \tag{10}$$

$$c(t) = \exp\left[-\frac{t}{c_n}\right], \tag{11}$$

where $H_e(t)$ and $H_s(t)$ mean the mean value functions for the exponential SRGM and the delayed S-shaped SRGM, respectively, based on nonhomogeneous Poisson process (NHPP), a the expected number of latent faults for each SRGM, and c_n the number of cloud software components. Eq.(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 for these two models are obtained as follows, respectively:

$$N_e(t) = D(t)\left[1 - \exp\left\{-bt - c(t)\sigma W(t)\right\}\right], \tag{12}$$

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

The jump term can be added to the proposed stochastic differential equation models in order to incorporate the irregular state around the time t by a change in the number of login users. Then, the jump-diffusion process[14] is given as follows.

$$dN_j(t) = \left\{b(t) - \frac{1}{2}\sigma^2\right\}\{D(t) - N_j(t)\}dt$$
$$+ c(t)\sigma\{D(t) - N_j(t)\}dW(t) + d\left\{\sum_{i=1}^{M_t(\lambda)}(V_i - 1)\right\}, \tag{14}$$

where $M_t(\lambda)$ is a Poisson point process with parameter λ at operation time t. Also, $M_t(\lambda)$ means the number of occurred jumps, and λ is the jump rate. $M_t(\lambda)$, $W(t)$, and V_i are assumed to be mutually independent. Moreover, V_i is i-th jump range.

By using Itô's formula[11,12], the solution of the former equation can be obtained as follows:

$$N_j(t) = D(t)\left[1 - \exp\left\{ - \int_0^t b(s)ds - c(t)\sigma W(t) - \sum_{i=1}^{M_t(\lambda)}\log V_i\right\}\right]. \tag{15}$$

The unknown parameters α, β, b, and σ of the proposed stochastic differential equation model are estimated by using the method of maximum-likelihood, respectively.

3. Gaussian Jump-Diffusion Process

As the types of jump-diffusion process, the following distributions are frequently used. The Gaussian Jump-diffusion process in which the i-th jump range V_i follows normal distribution is assumed in this paper. The density function of normal distribution is given as follows:

$$V_i \equiv f_n(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{(x-\mu)^2}{2\tau^2}\right], \tag{16}$$

where μ is the mean value and τ the standard deviation in terms of i-th jump.

The unknown parameters λ, μ, and τ of the proposed jump-diffusion model are estimated by using the genetic algorithm, respectively.

4. Cloud OSS Dependability Assessment Measures

4.1. Stochastic differential equation model

Since $N(t)$ is a random variable in the proposed model, its expected value can be a useful measure. It can be calculated from Eq.(8) as follows[13]:

$$\mathrm{E}[N(t)] = D(t)\left[1 - \exp\left\{ - \int_0^t b(s)ds + \frac{c(t)^2\sigma^2}{2}t\right\}\right], \tag{17}$$

where $\mathrm{E}[N(t)]$ is the expected number of faults detected up to time t.

It is important for software managers to assess the number of faults latent in the system according to the changes of specification of OSS. Considering the changes of requirements specification, the number of latent faults is given as Eq. (2).

Also, the number of remaining faults considering the changes of requirements specification can be obtained as follows:

$$\mathrm{E}[N_r(t)] = \mathrm{E}[D(t) - N(t)]. \tag{18}$$

Further, the mean time between software failures is useful to measure the property of the frequency of software failure-occurrences (or fault-detections). Then, the cumulative MTBF(denoted by $\mathrm{MTBF_C}$) is approximately given by:

$$MTBF_C(t) = \frac{t}{\mathrm{E}[N(t)]}. \tag{19}$$

4.2. Jump-diffusion model

Similarly, the cumulative number of detected faults in the system is important to estimate the situation of the progress on the software debugging procedures. Since $N_j(t)$ is a random variable in the proposed model, it is calculated as Eq.(15)[13].

Also, it is important for software managers to assess the number of latent faults according to the changes of specification of OSS. The number of remaining faults based on the jump-diffusion model considering the changes of requirements specification can be obtained as follows:

$$\begin{aligned} N_{rj}(t) &= D(t) - N_j(t) \\ &= D(t) \cdot \exp\left\{ -\int_0^t b(s)ds - c(t)\sigma W(t) - \sum_{i=1}^{M_t(\lambda)} \log V_i \right\}. \end{aligned} \tag{20}$$

Also, the mean time between software failures is useful to measure the property of the frequency of software failure-occurrences. Then, the cumulative MTBF(denoted by $\mathrm{MTBF_C}$) is approximately given by:

$$MTBF_{Cj}(t) = \frac{t}{N_j(t)}. \tag{21}$$

5. Performability

5.1. Stochastic differential equation model

It is important for the project managers to understand the performability in operation phase for cloud computing environment. In this paper, the performability ratio representing on operation availability is given by the following equation as the performability measure for cloud computing:

$$PER(t) = \frac{t - MTTR(t) \cdot \mathrm{E}[N(t)]}{t}, \tag{22}$$

where $MTTR(t)$ is the down time of cloud system at time t. Generally, $MTTR(t)$ means the mean time to correction as follows:

$$MTTR(t) = \int_0^\infty t \cdot \tau(t)dt. \tag{23}$$

In this paper, we assume the density function types of correction time as follows:

$$\tau(t) = \lambda e^{-\lambda t}, \tag{24}$$

where λ is the rate parameter of the exponential distribution.

5.2. Jump-diffusion model

Similarly, the sample path of performability ratio for cloud computing environment is given as follows:

$$PER_j(t) = \frac{\lambda t - N_j(t)}{\lambda t}, \tag{25}$$

where $N_j(t)$ is a random variable in the proposed jump-diffusion model.

6. Numerical Illustrations

6.1. Data for numerical illustrations

The OSS is closely watched from the point of view of the cost reduction and the quick delivery. There are several open source projects in area of cloud computing. In particular, this paper focuses on OpenStack[18] in order to evaluate the performance of the proposed methods. In this paper, numerical examples by using the data sets for OpenStack of cloud OSS are shown. The data used in this paper are collected in the bug tracking system on the website of OpenStack open source project.

6.2. Dependability assessment results

The estimated expected numbers of detected faults, $\widehat{\mathrm{E}}[N_e(t)]$ and $\widehat{\mathrm{E}}[N_s(t)]$, in case of $b(t) \equiv b_1(t)$ and $b(t) \equiv b_2(t)$ are shown in Fig. 2. Also, the sample paths of the estimated numbers of detected faults in Eqs. (12) and (13), $N_e(t)$ and $N_s(t)$, in case of $b(t) \equiv b_1(t)$ and $b(t) \equiv b_2(t)$ are shown in Fig. 3. From Fig. 3, the noise of sample path in case of $b(t) \equiv b_2(t)$ decreases as operation procedures go on. On the other hand, the noise of sample path in case of $b(t) \equiv b_1(t)$ is very small. From these results, the reliability trend in case of $b(t) \equiv b_1(t)$ means negative one.

Moreover, the sample paths of the estimated numbers of detected faults based on the proposed jump-diffusion model, $\widehat{N_j}(t)$, in case of $b(t) \equiv b_1(t)$ and $b(t) \equiv b_2(t)$ are shown in Figs. 4 and 5, approximately. From Figs. 4 and 5, the cloud OSS has the characteristic of reliability regression in case of $b(t) \equiv b_1(t)$. Moreover, it is important for software managers to assess the irregular fluctuation in terms of the number of detected faults according to the change in the number of login users in cloud computing. Then, Figs. 4 and 5 show that the jumps of the proposed model occur around the times by a changes in the number of login users in cloud computing, respectively. In particular, Fig. 5 shows that the jumps in spots at the operation time become small because of a decrease in jump diffusion state. On the other hand, the jumps in spots at the operation time become large because of the reliability regression.

7. Sensitivity Analysis for Jump-Diffusion parameters

From the results of the former sections, it is verified that the proposed model can be applied to assess quantitatively software dependability in the operating phase of the cloud computing. This section shows some behaviors of software dependability assessment measures in case of $b(t) = b_1(t)$ if the parameters λ, μ, and τ are changed. λ, μ, and τ are the jump rate of Gaussian Jump-diffusion process, the mean value of Gaussian Jump-diffusion process, and the standard deviation of Gaussian Jump-diffusion process, respectively.

The sample path of the estimated number of detected faults with changing the same values of the parameters λ, μ, and τ are illustrated in Figs. 6 and 7, respectively. Especially, the parameters λ, μ, and τ are related to the size of jump in the whole of operating phase. The frequency of jump, the jump size, the width of jump are growing with increasing the parameters λ, μ, and τ, respectively.

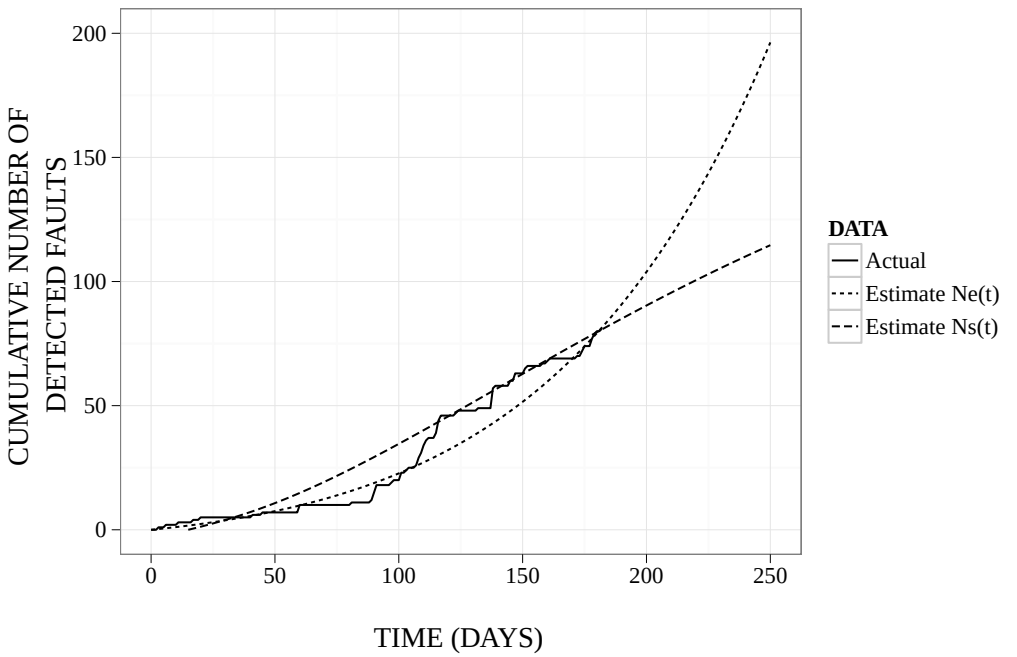

Figure 2. The estimated number of detected faults.

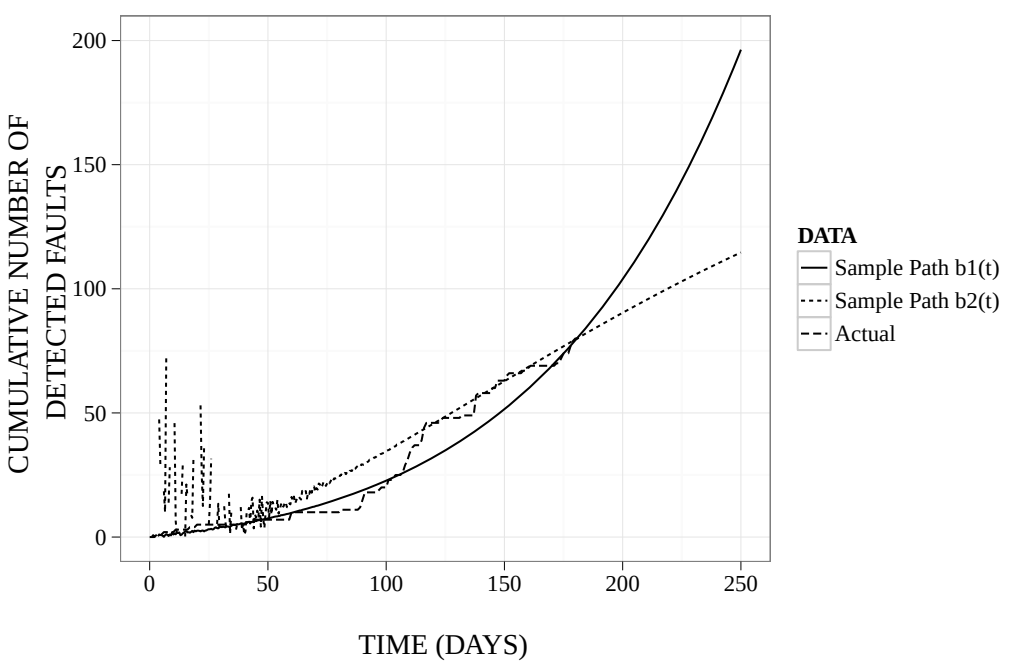

Figure 3. The sample path of estimated number of detected faults.

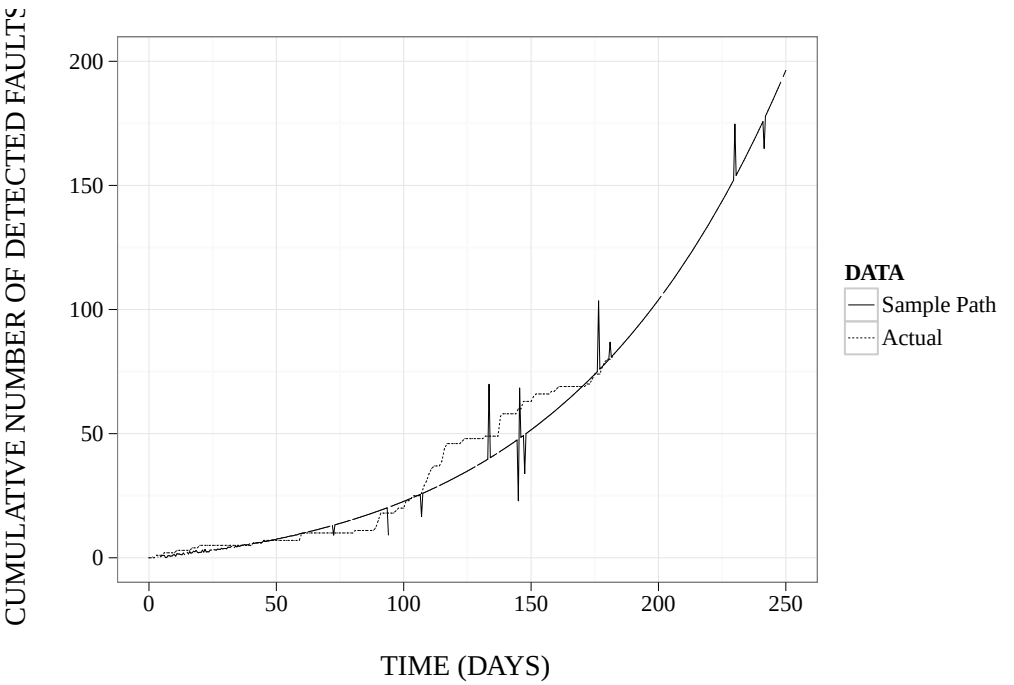

Figure 4. The sample path of the estimated number of detected faults, $\widehat{N_j}(t)$, in case of $b(t) \equiv b_1(t)$.

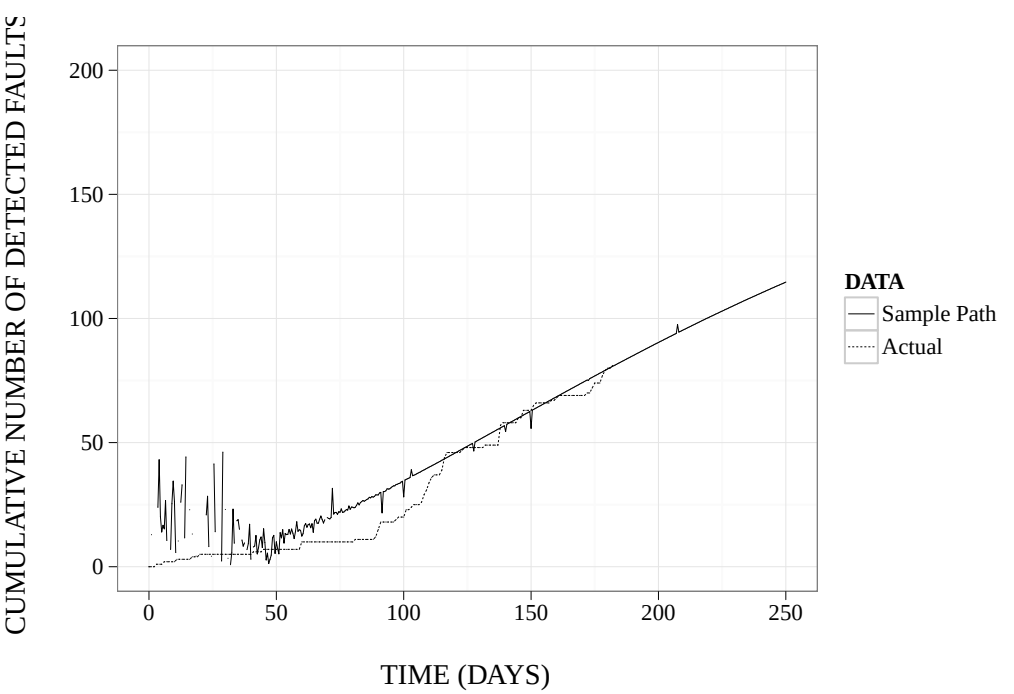

Figure 5. The sample path of the estimated number of detected faults, $\widehat{N_j}(t)$, in case of $b(t) \equiv b_2(t)$.

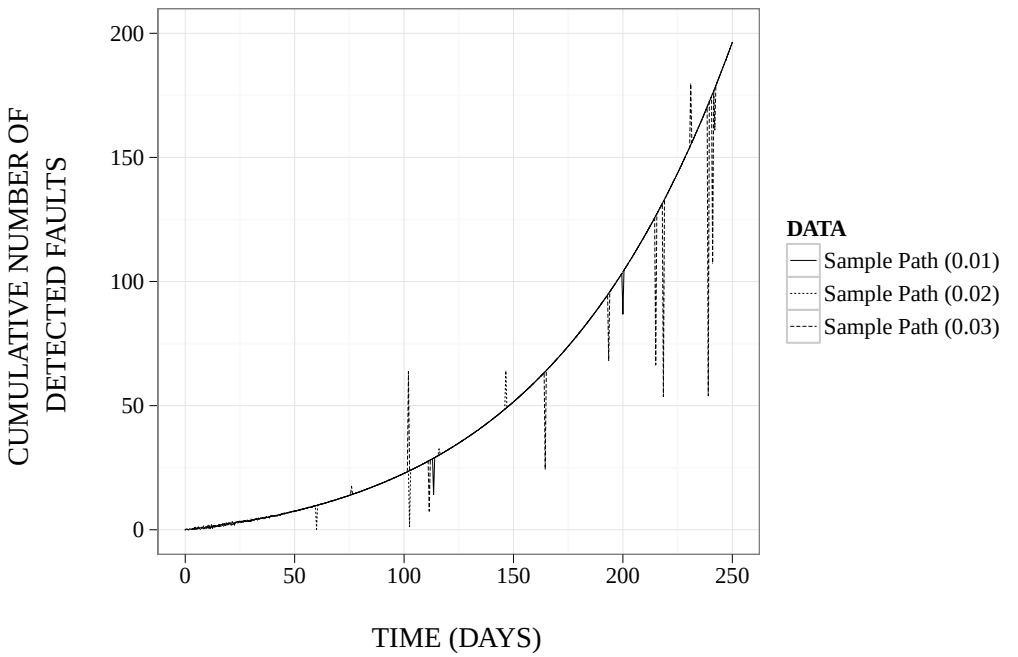

Figure 6. Dependence of model parameters λ, μ, and τ in case of $b(t) = b_1(t)$.

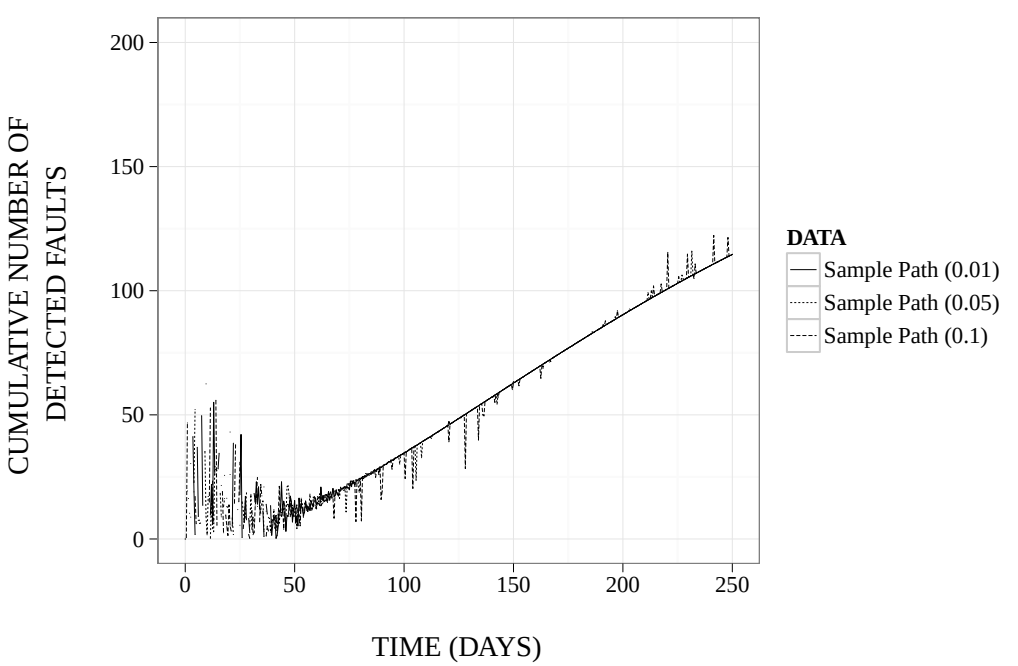

Figure 7. Dependence of model parameters λ, μ, and τ in case of $b(t) = b_2(t)$.

From above mentioned results, the proposed model can describe the characteristics of cloud environment according to the changes of the number of components on cloud OSS. The proposed method will be useful to assess the dynamic reliability of cloud computing environment.

8. Concluding Remarks

A cloud OSS is now attracting attention as the next-generation software service paradigm because of the cost reduction, quick delivery, and work saving. We have focused on the cloud OSS. Also, the method of dependability assessment for the cloud computing environment have discussed.

A software dependability analysis based on the proposed stochastic differential equation model in order to consider the environment of cloud computing have proposed in this paper. Then, this paper have 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. Especially, the jump diffusion model in order to consider around the times by the changes in the number of login users in cloud computing has been proposed. Also, this paper has focused on the requirements specification of OSS, and discussed the method of reliability assessment based on the stochastic differential equation model. Also, sets of actual data have been analyzed to show numerical examples of software dependability assessment for the cloud OSS. Moreover, several reliability and performability assessment measures have been derived from the proposed model.

In particular, several sensitivity analysis in terms of the parameters in the proposed jump-diffusion model have been shown in this paper. Then, this paper have compared the proposed jump-diffusion models with the proposed stochastic differential equation models in terms of goodness-of-fit for actual data.

At present, a new paradigm of distributed development typified by such open source project will evolve at a rapid pace in the future. Especially, it is difficult to assess the dynamic 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 methods of dependability assessment for the cloud computing environment.

Acknowledgments

This work was supported in part by the Grant-in-Aid for Scientific Research (C), Grant No. 24500066 and No. 22510150 from the Ministry of Education, Culture, Sports, Science, and Technology of Japan.

References

[1] E-Soft Inc., Internet Research Reports. http://www.securityspace.com/s_survey/data/

[2] S Yamada, *Elements of Software Reliability ∼Modeling Approach∼* (in Japanese), Kyoritsu–Shuppan, Tokyo, 2011.

[3] A MacCormack, J Rusnak, and C Y Baldwin, Exploring the structure of complex software designs: an empirical study of open source and proprietary code, *Informs Journal of Management Science*, Vol. 52, No. 7, 2006, pp. 1015–1030.

[4] G Kuk, Strategic interaction and knowledge sharing in the KDE developer mailing list, *Informs Journal of Management Science*, Vol. 52, No. 7, 2006, pp. 1031–1042.

[5] Y Zhoum and J Davis, Open source software reliability model: an empirical approach, *Proceedings of the Workshop on Open Source Software Engineering (WOSSE)*, Vol. 30, No. 4, 2005, pp. 67–72.

[6] P Li, M Shaw, J Herbsleb, B Ray, and P Santhanam, Empirical evaluation of defect projection models for widely-deployed production software systems, *Proceedings of the 12th International Symposium on the Foundations of Software Engineering (FSE-12)*, 2004, pp. 263–272.

[7] B Yang, F Tan and Y S Dai, Performance evaluation of cloud service considering fault recovery, *The Journal of Supercomputing*, Springer, published online: 23 Feb., 2011.

[8] A Iosup, S Ostermann, M N Yigitbasi, R Prodan, T Fahringer, and D H J Epema, Performance analysis of cloud computing services for many-tasks scientific computing, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 22, No. 6, 2011, pp. 931–945.

[9] T Fujiwara and S Yamada, A testing-domain dependent software reliability growth model for practical application, *Proceedings of the Second World Congress for Software Quality*, 2000, pp. 821–826.

[10] S Yamada and T Fujiwara, Testing-domain dependent software reliability growth models and their comparisons of goodness-of-fit, *International Journal of Reliability, Quality and Safety Engineering*, Vol. 8, No. 3, 2001, pp. 205–218.

[11] L Arnold, *Stochastic Differential Equations-Theory and Applications*, John Wiley & Sons, New York, 1974.

[12] E Wong, *Stochastic Processes in Information and Systems*, McGraw-Hill, New York, 1971.

[13] S Yamada, M Kimura, H Tanaka, and S Osaki, Software reliability measurement and assessment with stochastic differential equations, *IEICE Transactions on Fundamentals*, Vol. E77–A, No. 1, 1994, pp. 109–116.

[14] R C Merton, Option pricing when underlying stock returns are discontinous, *Journal of Financial Economics*, Vol. 3, 1976, pp. 125–144.

[15] P Honoré, Pitfalls in estimating jump-diffusion models, Working Paper Series 18, University of Aarhus, School of Business, 1998.

[16] S Yamada and S Osaki, Cost-reliability optimal software release policies for software systems, *IEEE Transactions on Reliability*, Vol. R-34, No. 5, 1985, pp. 422–424.

[17] S Yamada and S Osaki, Optimal software release policies with simultaneous cost and reliability requirements, *European Journal of Operational Research*, Vol. 31, No. 1, 1987, pp. 46–51.

[18] The OpenStack project, OpenStack. http://www.openstack.org/