Modeling and Analysis of Dependability Attributes for Services Computing Systems

Jiwei Huang; Chuang Lin, *Senior Member, IEEE*; Xiangzhen Kong; Bing Wei;
Xuemin (Sherman) Shen, *Fellow, IEEE*

Abstract—Dependability is an important consideration during the design and development of IT systems and services. But in services computing systems, the traditional definition and evaluation methods from the systems' and components' point of view meet challenges. In this paper, we veer from the angle of view, and study the dependability and their attributes from the service-oriented perspective. A stochastic model using semi-Markov process is put forward, and the quantitative analysis of the dependability attributes is carried out. By extending and transforming this model, the mean time to dependability attributes failure is calculated. Based on the analysis and calculations, some theorems are proposed and proved, to show the inter-relationships and comparisons of the different dependability attributes. Furthermore, we model the service composition and conduct workflow analysis to show how this model could deal with complex services. In addition, LANL service systems are analyzed as a case study to show how the proposed model and calculation methods could apply to real systems, and sensitivity analysis is also performed to identify the bottlenecks and find effective ways for dependability optimization.

Index Terms—Dependability, evaluation, semi-Markov processes, modeling, services computing.

1 Introduction

SERVICES represent a type of relationships-based interactions or activities between at least one service provider and one service consumer to achieve a certain business goal or solution objective [37]. Services computing, which intends to create, operate, manage and optimize the service processes in a well-defined architecture for high flexibility facing future business dynamics [38], has become a cross-discipline subject that covers the science and technology of bridging the gap between Business Services and IT Services [37]. Nowadays, the service is becoming the basic building block of IT systems with the rapid development of services computing [1].

As services computing become more and more popular and widely used, dependability, which is the ability of a system to keep on reliable service and avoid failures, becomes an important requirement, especially for critical applications such as traffic control, 24 hours/7 days healthcare and military applications. The dependability is an integrating concept that encompasses several attributes, which are availability, reliability, safety, integrity and maintainability. Different attributes represent different aspects of the system to perform accurate, continuous and trusted functions, and different types of systems have different requirements of different attributes. The evaluation and optimization of dependability in services computing become a hot topic in both industry and academe.

For large-scale complicated services computing systems such as Services-Oriented Architecture (SOA) or Cloud Computing systems widely used today, the dependability from the system's and component's point of view is hard to analyze because of the characteristics of massive-scale service sharing, complex service composition, wide-area network, heterogeneous components and complicated interactions among them. The general dependability models for pure software/hardware or conventional networks are mostly done from the standpoint of components and systems, which require full information about the composition, structure and behavior of every part of the system. Due to the complexity of services computing systems, these models cannot be simply applied to study the dependability of services computing systems such as SOA or Cloud [2].

Therefore, instead of traditional dependability analysis based on components and systems, we veer from the angle of view, and study the dependability definition and evaluation from the service-oriented perspective by taking a top-down approach. A general stochastic model is put forward and the formal method for dependability attributes evaluation is introduced. The comparison and interrelationships of the dependability attributes are discussed. In addition, considering the intrinsic characteristics of services computing, we model the basic service composition and selection patterns to show how to deal with complex services. Furthermore, we apply this model to real computing systems, and perform sensitivity analysis to find the effective ways for system optimization.

The remainder of this paper is organized as follows. In Section 2, the existing definitions and models of dependability and their challenges in services computing are

- *Jiwei Huang, Chuang Lin, Xiangzhen Kong and Bing Wei are with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China. E-mail: huangjw05@gmail.com; chlin@tsinghua.edu.cn; xiangzhen1985@gmail.com; weisoldier@gmail.com*
- *Xuemin (Sherman) Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada. E-mail: sshen@uwaterloo.ca.*

Part of this work was presented in IEEE SCC'11.

Manuscript received 17 Feb. 2012; revised 30 Oct, 2012; accepted 7 Feb, 2013.

Digital Object Indentifier 10.1109/TSC.2013.8

summarized. In Section 3, a semi-Markov process (SMP) model for a services computing system from the service view point is developed. Based on the model, the steady-state probabilities which lead to the calculation of the dependability and the mean time to failure (MTTF) of the dependability attributes are analyzed in Section 4. Moreover, some theorems are proposed and proved to show the internal relations of the attributes in Section 5. In Section 6, we model the service selection and composition, and show how to analyze the dependability attributes of the complex services. In Section 7, the LANL service systems are analyzed using the proposed models and computation methods as a case study to show how the evaluation could be applied in real systems, and parameter sensitivity analysis is taken to illuminate the different effects of the system parameters on different dependability attributes, which is helpful for system optimization. At last, we conclude the paper in Section 8.

2 Related Work

2.1 General definitions and analysis of dependability

The widely accepted definition of dependability is the ability to avoid service failures that are more frequent and more severe than is acceptable [3], [4]. According to the specification of IFIP WG 10.4 and related literature [3], [5], [6], dependability is an aggregated concept which consists of many attributes. As shown as Fig.1, the attributes include the following five "abilities". *Availability*: readiness for correct service; *Reliability*: continuity of correct service. *Safety*: to provide service without the occurrence of catastrophic failures; *Maintainability*: ability to undergo modifications and repairs; *Integrity*: the absence of improper alterations of user information and system state. The system is under threats including faults, errors and failures that may affect the system during its entire life [3]. The means to attain dependability have been summarized and grouped into four major categories, namely fault prevention, fault tolerance, fault removal and fault forecasting.

Another important consideration is the security of the services or the systems. Besides integrity and availability, *confidentiality*, which represents the ability of the system/service to prevent unauthorized access to and/or handling of information [3], [5], is the third attribute of the security shown as Fig. 1. Recently, the research community has started focusing on security of the services. One of the aspects of these research work is to analyze or evaluate the security of the services [39], and the other one is to propose security certification schemes to guarantee the services' security [40]. However, as the confidentiality is usually classified as an attribute of security excluded from pure dependability [3], and the traditional dependability fault assumption was that of non-malicious and stochastic faults, such as those resulting from a component failure, rather than deliberate, malicious security faults (attacks) [3], [28], this attribute is omitted in the following discussions in this paper.

Fig. 1. The dependability attributes, from [3], [5], [6].

The existing dependability analysis methods are commonly based on the components and systems, which model each part of the system and analyze the dependability attributes of the whole system. The methods are usually separated into two categories: combinatorial model methods and state-based stochastic model methods [7]. The combinatorial models are based on the structure of system, and reduce the system structure into the basic series element and parallel element, whose typical examples include the Reliability Block Diagrams (RBD) [8], Fault Trees (FT) [9], Attack Trees (AT) [10], etc. State-space model methods, which are usually based on the stochastic model of the system states, are good at dealing with the dynamics, uncertain, and complex relationships [11]. The well-known state-space models for dependability analysis include Markov Models [8], stochastic Petri nets [12], etc.

2.2 Challenges for Dependability Analysis of Services Computing Systems

Classical definition and evaluation methods of dependability based on component and systems meet difficulties and challenges in services computing systems. It is mainly manifested in the following aspects.

First, the upper-layer service in the services computing systems has high dynamic characteristics. Service aggregation or composition [13] provides composite applications and maximizes reuse. The service aggregators may reuse services that have already been created, or offer new services formed by choreographing interactions with available services offered by other providers. Service decomposition develops the parallelism in finer grain, to improve the efficiency of the service system. Service collaboration [14] enables the interaction and conjunction of multiple services to implement a more complex function. The dynamics and cooperativity of services themselves lead to the situation of the mapping relations between the services and the underlying system components which provide services are uncertain and dynamically changing. It is difficult to ensure services functionality even if the components and systems are fault-free, which causes difficulty for the classical dependability analysis and evaluation from the component and system perspective.

Moreover, virtualization is one of the common features of services computing such as Cloud, and virtualized infrastructure is viewed as the trend of IT infrastructure [15], [16]. Virtualized infrastructure has loosely coupled and resource sharing characteristics. Some mechanisms of virtualization such as server consolidation [17] and live migration [18] bring more flexibility and high efficiency. However, these advanced characteristics and mechanisms of virtualization also bring challenges to the dependability analysis. The services are not bound to a fixed hardware machine. And the high dynamism and complexity brought by the server consolidation, transparent resource

sharing, and live migration pose particular difficulties for the classical dependability analysis based on components and systems.

In addition, services are usually being deployed over unbounded large-scale infrastructures which are composed of a large number of computers and communication links. Failure in such large-scale systems is thought to be normality instead of abnormality [19], which brings complexity to classical dependability models and may lead to state explosion problems. Furthermore, not all resources in the services computing infrastructure need to be up at every point in time to fulfill a service, and a resource is required to be up only during the time periods when a service requests the resource [20]. It motivates us to focus on service instead of components and systems. In addition, both for service providers and users, they are more concerned with the dependability of the services instead of the underlying components and systems.

2.3 Existing Approaches on Dependability Analysis of Services Computing Systems

Because of the challenges for dependability analysis brought by the characteristics and complexities of services computing systems, researchers have studied the issue from different aspects.

User plays an important role in the services computing systems. Interacting with the system during the process of service, the user could be regarded as an active system communicating with the service provider. In several studies, the dependability or its related attributes are modeled and analyzed based on the behavior of user, called user-centered, user-perceived, etc. Heddaya and Heldal [19] revised the definitions of availability and reliability to account for network and service states that affect the end-user. Then they defined dependability as the product of the two metrics, and studied the relation between dependability and performance. Wang and Trivedi [20] proposed the definition and modeling of the user-perceived service availability based on user behavior. They derived formulas to compute service availability with the user behavior model called User Behavior Graphs (UBG) [21] and the system availability model using Stochastic Reward Nets (SRNs). The factors that influence service availability were evaluated in their work.

In addition, some research started with the characteristics of the system architecture of services computing systems. Zyla and Caban [22] used a generic model of SOA system composed of a number of web services to analyze the dependability of the system, and provided two measures of service dependability and performance, i.e. functional availability and throughput. The dependability model was implanted into multiple runs simulation with fault and failure models [23]. Dai et al. [24] presented a general model for centralized heterogeneous distributed system (CHDS) to study the service reliability and availability for distributed systems. Based on this model, the distributed service reliability which is defined as the probability of successfully providing the service in a distributed environment is investigated, and sensitivity analysis of the intrinsic parameters to affect the system availability was conducted. Oppenheimer and Patterson [25] made an analysis of the architectures and causes of failure at three representative large-scale Internet services which are Online, Content and ReadMostly, and discussed the principles for building highly available and maintainable large-scale Internet services. Sato and Trivedi [26] proposed a technique based on the Markov reward model (MRM) foundation for stochastic reliability analysis of composite services, which could reduce the computational complexity without sacrificing accuracy.

However, there are several limitations for these research introduced above. First, dependability is a comprehensive concept including many attributes, but the existing work usually focused on only one or two special attributes such as reliability and availability, which lacked a detailed and systematic study of the dependability. Second, some of the research limited to some special types of systems such as Web services, SOA and CHDS, and lacked a general abstract model for services computing systems.

Therefore, this paper makes an attempt to fill these gaps, models and analyzes the dependability with a service-oriented perspective. We will also discuss the interrelationships among the dependability attributes, which are the contents usually ignored by the related studies.

An early version of this work appeared in the 2011 IEEE International Conference on Services Computing [27]. This present paper adds a more complete description of the definition of the dependability attributes, and gives both transient and steady-state formulation of them. We make some modifications to the dependability state transition models to make it more appropriate to real systems, and add detailed mathematical analysis of the models. Furthermore, we model the complex service selection and composition, and conduct workflow analysis to show the impact brought by the intrinsic characteristics of the services computing systems to the dependability attributes. Moreover, we study some cases in real service systems to validate the theorems and conclusions, and perform sensitivity analysis to identify bottlenecks and show effective ways for system optimization in such systems in reality.

3 SMP Model for Dependability Quantification

3.1 SMP model

In the process of a service, facing the rich variety of possible events such as failure, attack, repair and recovery, it gives rise to a wide range of system states which are relevant to its ability to accept and successfully carry out its service. There is a tentative summary on the failure states of a distributed system in [19]. However, it focuses on only two of the attributes which are availability and reliability. Though according to the definitions, the maintainability and safety could be analyzed with the model, it lacks of the ability to model and evaluate the integrity of services computing systems. In this paper, in order to serve the definition of all the attributes of dependability of a services computing system, we make some amend-

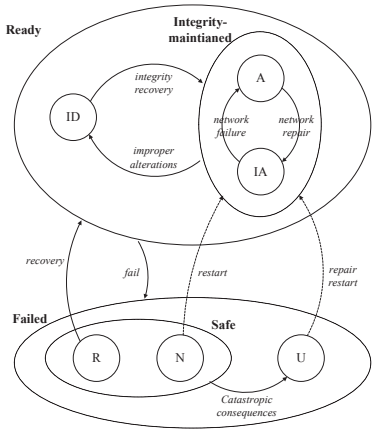


Fig. 2. A state transition graph for services computing systems

ments and supplements of the original system state graph.

As shown in Fig. 2, the states are hierarchically categorized as *Ready* or *Failed*. Whether the system can maintain the integrity divides the *Ready* state into two substates, i.e. *Integrity-maintained* and *Integrity-destroyed (ID)*. The former is classified into *Accessible (A)* and *Inaccessible (IA)*. For example, a system can be ready to accept service requests, but a network fault or exceeded capacity may lead to network failure which might cause the fact that the user's service requests cannot arrive at the system. The *Integrity-destroyed* state, which means that the system is undergoing improper system or information alterations which may be cause by software / hardware errors or improper system management by human, also contains *Accessible* and *Inaccessible* states that are omitted in Fig. 2.

When the services computing system is in the *Ready* state, an error or failure may occur, which might be caused by human error, software or hardware vulnerability, malicious attacks, virus, power outage, hardware failure, etc. The error or failure may be masked or tolerated in some situations, for example, in the virtualized infrastructure, once one of the hardware machines fails during the processing of a service, the upper virtual machines which contain the service programs could migrate to other operational hardware machine by live migration [18], which helps the system remain in *Ready* state. When the failure can't be masked or tolerated, however, the system will enter the *Failed* state. A subset of the *Failed* state is *Unsafe (U)* state, which means the failure may cause catastrophic consequences, and the "level of catastrophe" is to be defined by the user under different situations. An example of a catastrophic failure is a failure in the drive-by-wire system of a car that would lead to an accident with possible casualties [28]; another example is the failed cooling system in large-scale data centers which may cause hardware overheat and fire. A *Safe* state is further subdivided into *Recoverable (R)* and *Non-recoverable (N)*. In the former state, repair or replacement as well as recovery procedures are used to resume the state the system was in right before the failure; while in the *Non-recoverable* state, only restart is possible through repair or replacement [19].

From the dependability and security quantification point of view, some of the sojourn time distribution functions may be non-exponential [29]. For example, it is proved in [30] that the time between failures is not modeled well by an exponential distribution, and is fit well by a gamma or Weibull distribution. Therefore, the stochastic model based on the state graph is formulated in terms of a semi-Markov process (SMP), which is a more general model than continuous-time Markov chain (CTMC) commonly used. It is obvious that this SMP model is also feasible when all the distribution functions are exponential, for a Markov process is a special case of SMP.

Note that the system is not quite ready to process service requests as it performs recovery procedures when in *Recovering* state [19] of the preliminary model in [27], the *Recovering (RI)* state could be merged into the *Recoverable (R)* state, which means the time cost by the recovery procedures could be regarded as part of the sojourn time in the *Recoverable* state. Moreover, as the recovering state could not commonly be observed in the process of the service in the system log [31], the modified model in Fig. 2 is more operative than the preliminary one in real services computing systems.

According to the state graph in the Fig. 2, we will study the dependability of services computing systems, based on the transient and steady state probability. The notations and definitions used in this paper are shown in Table 1.

TABLE 1
NOTATIONS AND DEFINITIONS

Notation	Definition
$X(t)$	Event of system being in state X at the time point t.
$X(t,t')$	Event of system being continuously in state X from time t to t'.
$Pr(X(t))$	The transient-state probability that system is in state X at the time t.
$Pr_S(X)$	The steady-state probability in state X, and $Pr_S(X)=\lim_{T\to\infty}\frac{1}{T}\int_0^T Pr(X(t))dt$.
$Pr(X(t)\|Y(t))$	The conditional probability that system is in state X at the time t given that system is in state Y at the time t.
$Pr_S(X\|Y)$	The steady-state conditional probability, and $Pr_S(X\|Y)=\lim_{T\to\infty}\frac{1}{T}\int_0^T Pr(X(t)\|Y(t))dt$
h_i	The average sojourn time in state i.
v_i	The embedded DTMC steady-state probability in state i.
π_i	The SMP steady-state probability in state i.
V_i	The visit count of state i in MTTF analysis.
$D_i(t)$	One of the dependability attributes of service/system i.
$S_\lambda(M)$	The sensitivity of value M to the parameter λ.

3.2 Dependability quantification

Avizienis et al. [3], [6] have made a nice summary about the several attributes of dependability such as availability, reliability, maintainability, safety, integrity and confidentiality, of which confidentiality is also usually classified as an attribute of security excluded from pure dependability.

Their definitions were at the macroscopic level and were general enough to cover the entire range of computing and communication systems. So the specific characteristics of a certain system were omitted. Therefore, the mathematical descriptions of the attributes of dependability were not given in their work, especially for some certain types of systems, which weakens the specific guidance function in practice. According to characteristics of the increasingly popular services computing systems, this paper studies the specific formal definitions and detailed calculations of the attributes of dependability from the service-oriented perspective.

Reliability: Reliability is the ability of the system to offer service continuously without interruption. Reliability of a system could be defined as $R(t,\tau)$ which is the probability that the system functions properly and continuously in the time interval $[t,t+\tau]$, assuming that it was in a normal state at time *t* [32], [33], expressed as:

$$R(t,\tau)=\Pr(\text{Ready}(t,t+\tau)|\text{Ready}(t)). \quad (1)$$

Reliability is also commonly measured by the *mean time to failure (MTTF)*, which will be discussed in the following section.

Availability: Availability is defined as readiness for usage, representing the accessibility of the service at the time that users submit the requests for service. With the SMP model proposed above, the availability of a services computing system at a certain time *t* could be expressed by the probability of the state *Accessible*, denoted as

$$A(t)=\Pr(\text{Accessible}(t)). \quad (2)$$

Integrity: Integrity reflects the ability to make the service and data of a system absent of improper alterations or destructions. In the process of service, the users often concern more on integrity when they are using running service instead of failed service. Hence, in the SMP model, the integrity could be defined as the conditional probability of the system being in *Integrity-maintained* state given that it is in the *Ready* state, donated as

$$I(t)=\Pr(\text{Integrity-maintained}(t)|\text{Ready}(t)). \quad (3)$$

Safety: Safety of the services computing system expresses the ability to make the system absence of catastrophic consequences on the users or the environment. It reflects the ability of the system to keep the failure from causing catastrophic consequences once it fails. Hence, the safety could be calculated using the following expression:

$$S(t)=\Pr(\text{Safe}(t)|\text{Failed}(t)). \quad (4)$$

Maintainability: Maintainability is the ability to undergo repairs and modifications. The maintainability is defined as the conditional probability of the system being in *Recoverable* state once it fails, expressed as

$$M(t)=\Pr(\text{Recoverable}(t)|\text{Failed}(t)). \quad (5)$$

4 Model Analysis

4.1 DTMC steady-state probability computations

To evaluate the dependability attributes, we need to analyze the SMP model of the system described by its state transition diagram. The complete description and calculation of this SMP model require the knowledge of various parameters such as the mean sojourn time in each state h_i where $i\in\{A,IA,ID,R,N,U\}$ and the branching probabilities shown in the transition probability matrix in (6) as below. Note that we focus more on developing a methodology for model and analyzing the dependability of services computing systems than model parameterization, we will not discuss the measurement and estimation issue of the parameters in this paper, although it is important to the model accuracy. An example is to get the parameters from the set of data in the area of failure via some statistical methods, which will be shown in the case study in Section 7.

The SMP corresponding to Fig. 2 can be described in terms of its embedded discrete-time Markov chain (DTMC). According to the model and parameters shown above, the DTMC transition probability matrix could be written as:

$$P=\begin{array}{c} \\ A \\ IA \\ ID \\ R \\ N \\ U \end{array}\begin{array}{c} \begin{array}{cccccc} A & IA & ID & R & N & U \end{array} \\ \begin{bmatrix} 0 & p_{IA} & p_{ID} & p_R & p_N & p_U \\ p_A & 0 & p_{ID} & p_R & p_N & p_U \\ p_{IDA} & p_{IDIA} & 0 & p_R & p_N & p_U \\ p_{RA} & p_{RIA} & p_{RID} & 0 & 0 & p_{RU} \\ p_{NA} & p_{NIA} & 0 & 0 & 0 & p_{NU} \\ p_{UA} & p_{UIA} & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}. \quad (6)$$

Since the probability of transition from state *i* to some state must be 1, we have

$$p_{IA}+p_{ID}+p_R+p_N+p_U=1;$$
$$p_A+p_{ID}+p_R+p_N+p_U=1;$$
$$p_{IDA}+p_{IDIA}+p_R+p_N+p_U=1;$$
$$p_{RA}+p_{RIA}+p_{RID}+p_{RU}=1;$$
$$p_{NA}+p_{NIA}+p_{NU}=1;$$
$$p_{UA}+p_{UIA}=1.$$

The DTMC steady-state probabilities in each state can be computed by:

$$\nu=\nu\cdot P, \quad (7)$$

where $\nu=[\nu_A,\nu_{IA},\nu_{ID},\nu_R,\nu_N,\nu_U]$, in addition with

$$\sum_i \nu_i=1,\ i\in\{A,IA,ID,R,N,U\}. \quad (8)$$

Then with the embedded DTMC steady-state probabilities ν_i, the steady-state probabilities π_i of the SMP states could be calculated using the following expression:

$$\pi_i=\frac{\nu_i h_i}{\sum_j \nu_j h_j}, \text{ for each } i\in\{A,IA,ID,R,N,U\}. \quad (9)$$

In the point of system design and optimization, instead of the probabilities at a certain time *t*, the steady-state dependability attributes evaluation is usually concerned. According to the definitions, the steady-state dependability attributes could be analyzed using this model and the results we already have.

Availability: According to (2), the availability of a services computing system could be expressed as the steady-state probability of the *Accessible* state, denoted as

$$A=\Pr_S(A)=\lim_{T\to\infty}\frac{1}{T}\int_0^T \Pr(\text{Accessible}(t))=\pi_A. \quad (10)$$

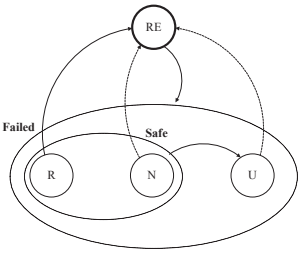


Fig. 3. Simplified SMP model

Integrity: In the steady-state dependability analysis, the integrity could be denoted as the steady-state conditional probability in (3) as

$$I = \Pr_S(\text{Integrity-maintained}|\text{Ready}) = \frac{\pi_A + \pi_{IA}}{\pi_A + \pi_{IA} + \pi_{ID}}. \quad (11)$$

Safety: According to (4), the safety of the services computing system in steady-state could be calculated using the following expression:

$$S = \Pr_S(\text{Safe}|\text{Failed}) = \frac{\pi_R + \pi_N}{\pi_R + \pi_N + \pi_U}. \quad (12)$$

Maintainability: The maintainability could be defined as the conditional probability as

$$M = \Pr_S(\text{Recoverable}|\text{Failed}) = \frac{\pi_R}{\pi_R + \pi_N + \pi_U}. \quad (13)$$

Reliability: In steady-state, the reliability could be analyzed by the probability that the system functions properly and continuously in the time interval $[0,\tau]$, assuming that it was in a normal state at time 0, expressed as:

$$R(\tau) = \Pr(\text{Ready}(0,\tau)|\text{Ready}(0)). \quad (14)$$

The detailed calculation of reliability along with the *MTTF* analysis will be given in the following subsection.

4.2 MTTF analysis

Mean time to failure (MTTF) is a common measure for reliability. It provides the mean time it takes for the system to reach one of the designated failure states, given that the system starts in a good or working state [29]. Using and extending the MTTF analogy, we can evaluate the dependability attributes with a different point of view.

When analyzing one of the dependability attributes, the failed states are made absorbing states. Classification of the SMP states into absorbing and transient categories depends on the actual nature of the attributes being analyzed. Once the system reaches any of the absorbing states, the probability of moving out of such state is 0. Hence, from a normal initial state *q*, the MTTF could be calculated with the visit count and sojourn time of each transient states, with the approaches in [8], [29], and [34].

To make the evaluation easy to operate, a simplified SMP model is proposed, where the states in the set of *Ready* in Fig. 2 is regarded as one macro state. It is convenient to analyze some attributes such as reliability, safety and maintainability which are less related to the state transition in *Ready* states, and this model is practicable when some states in *Ready* such as *Integrity-destroyed* or *Inaccessible* is hard to recorded or analyzed in some certain types of systems. The simplified SMP model is shown as Fig. 3.

Its transition probability matrix is expressed as

$$P = \begin{array}{c} \\ RE \\ R \\ N \\ U \end{array} \begin{array}{c} \begin{array}{cccc} RE & R & N & U \end{array} \\ \begin{bmatrix} 0 & p_R' & p_N' & 1-p_R'-p_N' \\ 1-p_{RU} & 0 & 0 & p_{RU} \\ 1-p_{NU} & 0 & 0 & p_{NU} \\ 1 & 0 & 0 & 0 \end{bmatrix} \end{array}, \quad (15)$$

where

$$p_R' = \frac{p_R}{p_R + p_N + p_U} \text{ and } p_N' = \frac{p_N}{p_R + p_N + p_U}.$$

With equation (7), the DTMC steady-state probabilities in each state could be

$$[\nu_{RE}, \nu_R, \nu_N, \nu_U] = \frac{1}{2 + p_R' p_{RU} + p_N' p_{NU}} \left[1, p_R', p_N', 1 - p_R' + p_R' p_{RU} - p_N' + p_N' p_{NU}\right] \quad (16)$$

As we already know that $\pi_i = \nu_i h_i / \sum_j \nu_j h_j$ for each $i \in \{RE, R, N, U\}$, then we have the steady-state probabilities of the SMP states as

$$[\pi_{RE}, \pi_R, \pi_N, \pi_U] = \alpha\left[h_{RE}, p_R' h_R, p_N' h_N, \left(1 - p_R' + p_R' p_{RU} - p_N' + p_N' p_{NU}\right) h_U\right], \quad (17)$$

where,

$$\alpha = \frac{\nu_{RE}}{\sum_j \nu_j h_j} = \frac{1}{\left(2 + p_R' p_{RU} + p_N' p_{NU}\right) \sum_j \nu_j h_j}.$$

4.2.1 MTTRF

We define the concept of *mean time to reliability failure (MTTRF)* as a measurement of reliability of a system. To analyze *MTTRF* with our model, the states $X_a = \{R, N, U\}$ will form the set of absorbing states, while $X_t = \{A, IA, ID\} = \{RE\}$ is the set of transient states. The resulting transition probability matrix *P* has the general form as

$$P = \begin{bmatrix} Q & C \\ 0 & I \end{bmatrix}, \quad (18)$$

where, the transition probabilities between transient states are expressed as

$$Q = \begin{bmatrix} 0 & p_{IA} & p_{ID} \\ p_A & 0 & p_{ID} \\ p_{IDA} & p_{IDIA} & 0 \end{bmatrix}. \quad (19)$$

We assume the initial state here is *Accessible (A)*, which gives $q = [1, 0, 0]$. In steady state, the visit count of each transient state has the following equation:

$$V_i = q_i + \sum_{j \in X_t} V_j Q_{ji}. \quad (20)$$

With (19) and (20), we have

$$V_A = \frac{1 - p_{ID} p_{IDIA}}{1 - p_A p_{IA} - p_{ID} p_{IDA} - p_{IA} p_{ID} p_{IDA} - p_{ID} p_{IDIA} - p_A p_{ID} p_{IDIA}},$$

$$V_{IA} = \frac{p_{IA} + p_{ID} p_{IDIA}}{1 - p_A p_{IA} - p_{ID} p_{IDA} - p_{IA} p_{ID} p_{IDA} - p_{ID} p_{IDIA} - p_A p_{ID} p_{IDIA}}, \quad (21)$$

$$V_{ID} = \frac{p_{ID} + p_{IA} p_{ID}}{1 - p_A p_{IA} - p_{ID} p_{IDA} - p_{IA} p_{ID} p_{IDA} - p_{ID} p_{IDIA} - p_A p_{ID} p_{IDIA}}.$$

Hence, we have the *MTTRF* as:

$$MTTRF = \sum_{i \in X_t} V_i h_i = V_A h_A + V_{IA} h_{IA} + V_{ID} h_{ID}. \quad (22)$$

When the states of $\{A, IA, ID\}$ are regarded as one macro state *RE* as in Fig. 3, it is clear that $Q = [1]$, and $MTTRF = h_{RE}$. Hence we get the expression of the sojourn time in *Ready* state as (23), which is with value in the following calculations.

$$MTTRF = V_A h_A + V_{IA} h_{IA} + V_{ID} h_{ID} = h_{RE}. \quad (23)$$

Besides *MTTRF*, reliability of a system could also be defined as $R(\tau)$ as (14). Because of the variety of the distribution of the sojourn time in each state, it is difficult to give a general and normal expression of $R(\tau)$. However, when the sojourn time conforms to exponential distribution, or for analysis simplification the sojourn time in the *Ready* states is assumed to be exponential, with the parameter $\lambda = 1/h_{RE}$, we have the probability density function of failure occurrence $f(x) = \lambda e^{-\lambda x}$ and the probability function $F(x) = 1 - e^{-\lambda x}$. Therefore, the reliability of the system could be calculated as follows:

$$R(\tau) = 1 - F(\tau) = e^{-\lambda \tau} = e^{-\tau / h_{RE}}. \quad (24)$$

4.2.2 MTTAF

The calculation process of *mean time to availability failure (MTTAF)* is simple. It is clear that there is only one transient state $X_t = \{A\}$, with the transition probability matrix $Q = [1]$, resulting in $V_A = 1$. The *MTTAF* could be expressed as

$$MTTAF = V_A h_A = h_A. \quad (25)$$

4.2.3 MTTMF

When the maintainability of a system is failed, the system should be in the *Non-recoverable* or *Unsafe* states. Hence, using the simplified model, $X_t = \{RE, R\}$, and $X_a = \{N, U\}$, the transition probability matrix is

$$Q = \begin{bmatrix} 0 & p_R' \\ 1 - p_{RU} & 0 \end{bmatrix}. \quad (26)$$

With initial state $q = [1, 0]$, we have

$$V_{RE} = \frac{1}{1 - p_R' + p_R' p_{RU}},\ V_R = \frac{p_R'}{1 - p_R' + p_R' p_{RU}}. \quad (27)$$

Then the *mean time to maintainability failure (MTTMF)* could be expressed as

$$MTTMF = \frac{1}{1 - p_R' + p_R' p_{RU}} \cdot h_{RE} + \frac{p_R'}{1 - p_R' + p_R' p_{RU}} \cdot h_R. \quad (28)$$

4.2.4 MTTR

When evaluating maintainability of a recoverable system, besides MTTMF, *mean time to repair (MTTR)* is also an important reference attribute, which gives the time it takes by the repair process after the system failure. Using the SMP model proposed above, the *MTTR* of the services computing system could be calculated in the similar way as MTTF. Here, the transient states $X_t = \{R, N, U\}$, and absorbing states $X_a = \{RE\}$. Then we have the transition probability matrix:

$$Q = \begin{bmatrix} 0 & 0 & p_{RU} \\ 0 & 0 & p_{NU} \\ 0 & 0 & 0 \end{bmatrix}. \quad (29)$$

The initial state of the system should be a failed state, which is no longer a working or normal state as before. We define the initial state using the steady state probabilities of the SMP model in (17), as $q = [\pi_R', \pi_N', \pi_U']$, where $\pi_R' = \pi_R / (\pi_R + \pi_N + \pi_U)$, $\pi_N' = \pi_N / (\pi_R + \pi_N + \pi_U)$ and $\pi_U' = \pi_U / (\pi_R + \pi_N + \pi_U)$.

Using equation (20), we get the visit counts as

$$V_R = \pi_R',\ V_N = \pi_N',\ V_U = p_{RU} \pi_R' + p_{NU} \pi_N' + \pi_U'. \quad (30)$$

Hence,

$$MTTR = \frac{\pi_R}{\pi_R + \pi_N + \pi_U} \cdot h_R + \frac{\pi_N}{\pi_R + \pi_N + \pi_U} \cdot h_N + \left(p_{RU} \frac{\pi_R}{\pi_R + \pi_N + \pi_U} + p_{NU} \frac{\pi_N}{\pi_R + \pi_N + \pi_U} + \frac{\pi_U}{\pi_R + \pi_N + \pi_U} \right) \cdot h_U. \quad (31)$$

4.2.5 MTTSF

To analyze the safety in time domain, the set of absorbing states contains only *Unsafe* state. Here, $X_t = \{RE, R, N\}$ is the set of transient states, and the transition matrix is shown as:

$$Q = \begin{bmatrix} 0 & p_R' & p_N' \\ 1 - p_{RU} & 0 & 0 \\ 1 - p_{NU} & 0 & 0 \end{bmatrix}. \quad (32)$$

Using (20), we have the mean visit count of each state in X_t as:

$$V_{RE} = \frac{1}{1 - p_R' + p_R' p_{RU} - p_N' + p_N' p_{NU}},$$
$$V_R = \frac{p_R'}{1 - p_R' + p_R' p_{RU} - p_N' + p_N' p_{NU}},$$
$$V_N = \frac{p_N'}{1 - p_R' + p_R' p_{RU} - p_N' + p_N' p_{NU}}. \quad (33)$$

Then the mean time it takes for the system to reach safety failure state (*MTTSF*) could be expressed as

$$MTTSF = \frac{1}{1 - p_R' + p_R' p_{RU} - p_N' + p_N' p_{NU}} \cdot h_{RE} + \frac{p_R'}{1 - p_R' + p_R' p_{RU} - p_N' + p_N' p_{NU}} \cdot h_R + \frac{p_N'}{1 - p_R' + p_R' p_{RU} - p_N' + p_N' p_{NU}} \cdot h_N. \quad (34)$$

4.2.6 MTTIF

In addition, the *mean time to integrity failure (MTTIF)* could be analyzed with the same methodology. With the transient states $X_t = \{A, IA\}$ and transition probability matrix

$$Q = \begin{bmatrix} 0 & p_{IA} \\ p_A & 0 \end{bmatrix}. \quad (35)$$

We obtain the mean visit count of each state as

$$V_A = \frac{1}{1-p_A p_{IA}},\ V_{IA} = \frac{p_{IA}}{1-p_A p_{IA}}. \quad (36)$$

Therefore, the *MTTIF* could be calculated using the following expression:

$$MTTIF = \frac{1}{1-p_A p_{IA}} \cdot h_A + \frac{p_{IA}}{1-p_A p_{IA}} \cdot h_{IA}. \quad (37)$$

5 Relationships and Comparisons of Dependability Attributes

In this section, we will make further discussions on the attributes of dependability of services computing systems. Some theorems and their proofs will be given, which helps us to have an insight into the attributes and their relationship and comparison.

Theorem 1. *The MTTFs for dependability of a system have the following equation:*

$$MTTSF \geq MTTMF \geq MTTRF \geq MTTIF \geq MTTAF.$$

Proof.

As $0 \leq p_{NU} \leq 1$, $0 \leq p_R' \leq 1$ and $0 \leq p_{RU} \leq 1$, it is obvious that $1-p_R'+p_R'p_{RU} \geq 1-p_R'+p_R'p_{RU}-p_N'(1-p_{NU}) \geq 0$.

Hence, $\frac{1}{1-p_R'+p_R'p_{RU}-p_N'(1-p_{NU})} \geq \frac{1}{1-p_R'+p_R'p_{RU}} \geq 0$.

Also we have $\frac{p_N'}{1-p_R'+p_R'p_{RU}-p_N'(1-p_{NU})} \cdot h_N \geq 0$. Hence,

$$MTTSF = \frac{1}{1-p_R'+p_R'p_{RU}-p_N'(1-p_{NU})} \cdot h_{RE} + \frac{p_R'}{1-p_R'+p_R'p_{RU}-p_N'(1-p_{NU})} \cdot h_R + \frac{p_N'}{1-p_R'+p_R'p_{RU}-p_N'(1-p_{NU})} \cdot h_N \geq \frac{1}{1-p_R'+p_R'p_{RU}} \cdot h_{RE} + \frac{p_R'}{1-p_R'+p_R'p_{RU}} \cdot h_R = MTTMF.$$

Next, we prove $MTTMF \geq MTTRF$. As we know that $\frac{1}{1-p_R'(1-p_{RU})} \geq 1$ and $\frac{p_R'}{1-p_R'+p_R'p_{RU}} \cdot h_R \geq 0$, hence,

$$MTTMF = \frac{1}{1-p_R'(1-p_{RU})} \cdot h_{RE} + \frac{p_R'}{1-p_R'(1-p_{RU})} \cdot h_R \geq h_{RE} = MTTRF.$$

As $0 \leq p_{ID}p_{IDIA} < 1$, we have

$$\frac{1-p_{ID}p_{IDIA}}{1-p_A p_{IA}-p_{ID}p_{IDA}-p_{IA}p_{ID}p_{IDA}-p_{ID}p_{IDIA}-p_A p_{ID}p_{IDIA}} = \frac{1}{1-\frac{p_A p_{IA}}{1-p_{ID}p_{IDIA}}-\frac{p_{ID}p_{IDA}+p_{IA}p_{ID}p_{IDA}+p_A p_{ID}p_{IDIA}}{1-p_{ID}p_{IDIA}}} \geq \frac{1}{1-p_A p_{IA}}.$$

For $p_{IA}+p_{ID}p_{IDIA} \geq p_{IA}$ and $1-p_A p_{IA}-p_{ID}p_{IDA}-p_{IA}p_{ID}p_{IDA}-p_{ID}p_{IDIA}-p_A p_{ID}p_{IDIA} \leq 1-p_{IA}p_{IDIA}$, we know that

$$\frac{p_{IA}+p_{ID}p_{IDIA}}{1-p_A p_{IA}-p_{ID}p_{IDA}-p_{IA}p_{ID}p_{IDA}-p_{ID}p_{IDIA}-p_A p_{ID}p_{IDIA}} \geq \frac{p_{IA}}{1-p_A p_{IA}}.$$

Hence, we have $MTTRF \geq MTTIF$.

For $\frac{1}{1-p_A p_{IA}} \geq 1$ and $\frac{p_{IA}}{1-p_A p_{IA}} \cdot h_{IA} \geq 0$, we have

$$MTTIF = \frac{1}{1-p_A p_{IA}} \cdot h_A + \frac{p_{IA}}{1-p_A p_{IA}} \cdot h_{IA} \geq h_A = MTTAF.$$

In summary, we have the conclusion that $MTTSF \geq MTTMF \geq MTTRF \geq MTTIF \geq MTTAF$. □

Theorem 1 gives a quantitative comparison among the MTTFs of the different dependability attributes. It shows that in a normal services computing system, the availability fails most often on average. Safety often maintains for the longest time in the five attributes of dependability.

For steady-state probability based dependability analysis, the dependability attributes could be mapped to some of the states in the model. The composition and aggregation of the states could help better understand the relations between the different attributes, seeing the following two theorems.

Theorem 2. *In the steady-state point of view, the improvement of the maintainability can contribute to better the safety of the system. Furthermore, the set of states of maintainability is a subset of that of safety.*

Proof.

According to the definition of maintainability, we have $M = \Pr(R \mid Failed)$.

And, as the expression of safety, we have

$$S = \Pr(Safe \mid Failed) = \Pr(R \mid Failed) + \Pr(N \mid Failed) = M + \Pr(N \mid Failed).$$

Hence, we get the conclusion that when the maintainability is enhanced, the safety of the system will increase.

For the SMP model shown in Fig. 2, the *Safe* state is the aggregation of two states which are *Recoverable (R)* and *Non-recoverable (N)*. Along with the expression above, it is obvious that the set of states of maintainability is a subset of that of safety. □

Theorem 3. *Ceteris paribus, increasing the availability leads to the enhancement of integrity.*

Proof.

From the expression of integrity as in (6), we have

$$I = \frac{\pi_A + \pi_{IA}}{\pi_A + \pi_{IA} + \pi_D} = 1 \Big/ \left(1 + \frac{\pi_{ID}}{\pi_A + \pi_{IA}}\right).$$

Therefore, ceteris paribus, when availability $A = \pi_A$ is enhanced, the integrity of the system will go up. □

These theorems could also be validated by a numerical solution to the SMP model. Fig. 4 shows the variation of the dependability attributes according to p_R, which

mainly affects the maintainability of the system. We could see that as p_R goes up, the maintainability increases rapidly, making the safety enhance at the same time. Fig. 5 shows that with the increase of availability caused by h_A being enhanced, the integrity also goes up. These two experiments could validate the Theorem 2 and 3 proposed above. In addition, the MTTFs of dependability with different h_A is shown in Fig. 6, which shows the comparison among the dependability attributes, corresponding with the same conclusion as Theorem 1.

These theorems show the interrelationship between the dependability attributes, and are with reference value in system design and optimization. For instance, the more likely the system turns to *Recoverable* state when failed, the higher maintainability and safety it is. And the integrity would be enhanced when introducing better mechanism for availability. The relations are directive for efficient system optimization.

6 Service Selection and Composition Models

Considering the intrinsic characteristics of services computing such as dynamic selection and search of services, service composition, etc., we will propose dependability models for service selection and composition in services computing systems in this section. We will show the influence brought by these characteristics to the dependability attributes of services computing systems.

6.1 Service composition

A service could be classified as a simple service or a complex one, which depends on the request of the service task. Simple services have certain special functions, and they can be combined together to build up different complex services with different complicated functions. A comprehensive business process commonly requires supports from multiple services, and service composition could refer to a process of adaptively composing a set of available services into a business process flow according to predefined business requirements [37]. Service composition is an important consideration in services computing.

From service-oriented point of view, the dependability of the composite service is an important metric of the services computing system. Based on the models and analysis proposed above, we could model the service composition and evaluate the dependability attributes of the comprehensive business process.

6.2 Basic Block Diagram

We borrow the idea from Reliability Block Diagram (RBD) to model different types of service composition. There are two basic block diagrams which are series and parallel. The services which are necessary for the required function are connected in series, while services which can fail with no effect on the required function (redundancy) are connected in parallel.

For series block diagram, the two services/systems must be on to complete the whole task. We define that $D_i(t)$ is one of the dependability attributes of the i^{th} basic block at time t, and then the dependability of the whole system, expressed as $D(t)$, could be obtained by (38). Furthermore, the composite service's MTTF of such dependability attribute could be calculated using (39). As $0 \leqslant D_i(t) \leqslant 1$, it is obvious that the complex series service has less dependability and MTTF than any of the single ones.

$$D(t) = \prod_i D_i(t) \le \min_i \{D_i(t)\}, \tag{38}$$

$$MTTF = 1/\sum_i \frac{1}{MTTF_i} \le \min_i \{MTTF_i\}. \tag{39}$$

On the other hand, parallel composited services have higher dependability attributes and MTTFs than the single ones, shown as the following expressions.

$$D(t) = 1 - \prod_i (1 - D_i(t)) \ge \max_i \{D_i(t)\}, \tag{40}$$

$$\begin{aligned} MTTF = {} & \sum_i MTTF_i - \sum_{i \ne j} \frac{1}{\frac{1}{MTTF_i} + \frac{1}{MTTF_j}} \\ & + \sum_{i \ne j \ne k} \frac{1}{\frac{1}{MTTF_i} + \frac{1}{MTTF_j} + \frac{1}{MTTF_k}} - \cdots \ge \max_i \{MTTF_i\}. \end{aligned} \tag{41}$$

6.3 Basic Service Composition Patterns

For complex service composition, there are several workflow patterns to model the service composition [36]. In this paper, we discuss the basic patterns capturing elementary aspects of process control. Furthermore, the more complex service compositions could be modeled with these basic blocks using similar methodology.

Sequence pattern: An activity in a service process is enabled after the completion of another activity in the same process. The sequence composition and its related block diagram are shown as Fig. 7(a). The dependability and MTTF of the composite service could be expressed as:

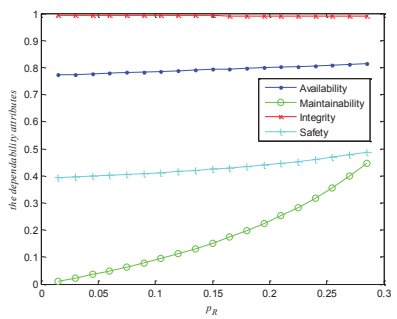


Fig. 4. Dependability with different p_R

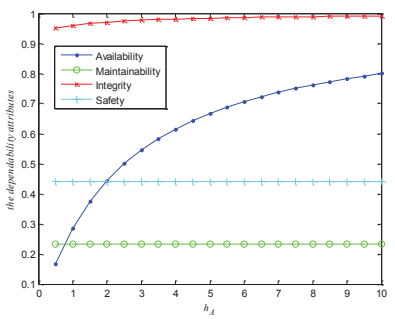


Fig. 5. Dependability with different h_A

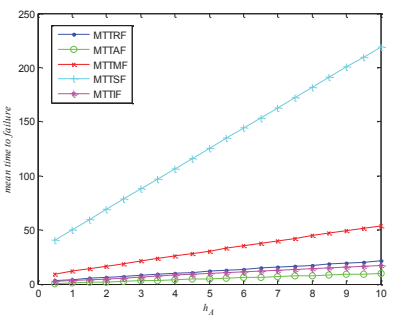


Fig. 6. MTTFs with different h_A

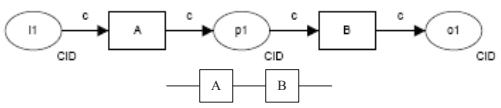


Fig. 7(a). Sequence pattern

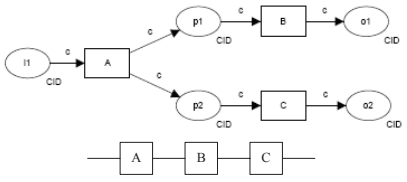


Fig. 7(b). Parallel split pattern

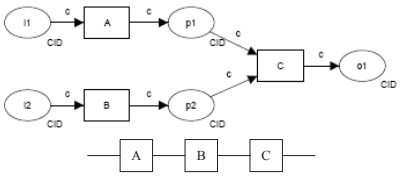


Fig. 7(c). Synchronization pattern

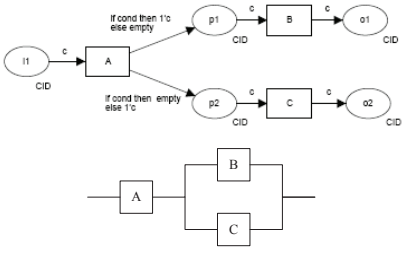


Fig. 7(d). Exclusive choice pattern

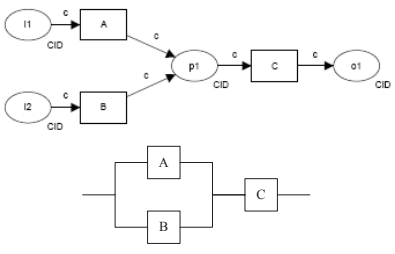


Fig. 7(e). Simple merge pattern

$$D(t) = D_A(t) \cdot D_B(t), \tag{42}$$

$$MTTF = \frac{MTTF_A \cdot MTTF_B}{MTTF_A + MTTF_B}. \tag{43}$$

Parallel split pattern: Parallel split pattern is a point in the service process where a single thread of control splits into multiple threads which can be executed in parallel, thus allowing activities to be executed simultaneously or in any order, shown as Fig. 7(b). In its block diagram, although the parallel services could be executed at the same time or in any order, the relationship between them is also series. Therefore, its dependability attributes and MTTF should be expressed as follows.

$$D(t) = D_A(t) \cdot D_B(t) \cdot D_C(t), \tag{44}$$

$$MTTF = 1/\left(\frac{1}{MTTF_A} + \frac{1}{MTTF_B} + \frac{1}{MTTF_C}\right). \tag{45}$$

Synchronization pattern: In this service composition pattern, multiple parallel subprocesses/activities converge into one single thread, thus synchronizing multiple services, shown as Fig. 7(c). The relationship among the services is the same as the parallel pattern, and its dependability could be calculated using (44) and (45).

Exclusive choice: The exclusive choice pattern, also known as service selection, is the composite service where one of several branches is chosen based on a decision, shown as Fig. 7(d). For this type of service composition, the dependability attributes could be obtained using (46), while the MTTF could be calculated by (47).

$$D(t) = D_A(t) \cdot \left[1 - \left(1 - D_B(t)\right)\left(1 - D_C(t)\right)\right], \tag{46}$$

$$MTTF = 1/\left(\frac{1}{MTTF_A} + \frac{MTTF_B + MTTF_C}{MTTF_B{}^2 + MTTF_B MTTF_C + MTTF_C{}^2}\right). \tag{47}$$

Simple merge: two or more alternative branches come together without synchronization, shown as Fig. 7(e). Hence, the service A and B are parallel, and service C is series. As a result, D(t) and MTTF are shown as follows:

$$D(t) = \left[1 - \left(1 - D_A(t)\right)\left(1 - D_B(t)\right)\right] \cdot D_C(t), \tag{48}$$

$$MTTF = 1/\left(\frac{MTTF_A + MTTF_B}{MTTF_A{}^2 + MTTF_A MTTF_B + MTTF_B{}^2} + \frac{1}{MTTF_C}\right). \tag{49}$$

Using the models proposed above, we could give mathematical analysis of the composited services. The analytical results could help guide the service selections and system optimization when compositions occur.

7 Case Study and Sensitivity Analysis

In this section, a case study in reality is provided to show how this approach could be applied in real systems. Furthermore, sensitivity analysis is proposed, which is helpful for bottleneck identify and system optimization.

7.1 System introduction

One can obtain a practical solution to the SMP model described in this paper using system usage and failure data in real services computing systems. The failure data at Los Alamos National Laboratory (LANL) [31] is analyzed to show how this model can apply to real systems. The whole data set is collected over the past nine years from 1996 to 2005, and covers 23 HPC systems including a total of 4,750 machines and 24,101 processors [30]. 19 systems are chosen from them which provide computing, Grid or web services to the users. They have the same characteristics of the services computing systems such as large-scale infrastructure, massive-scale service sharing, complex service composition, and heterogeneous components.

Failures in LANL are assigned to six categories, namely Facilities, Hardware, Software, Network, Human Error and Undetermined. The software failures and failures caused by human error could be classified into recovera-

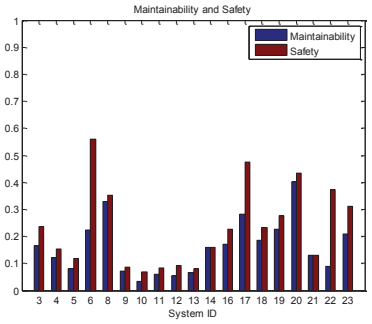


Fig. 8. Maintainability and safety of LANL systems

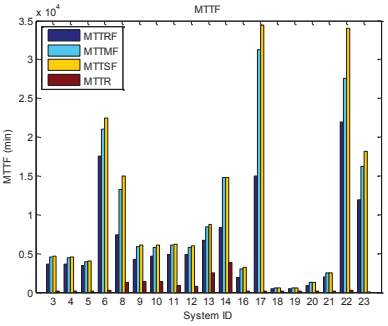


Fig. 9. MTTFs of LANL systems

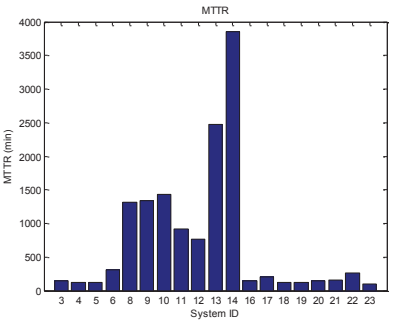


Fig. 10. MTTR of LANL systems

ble failures, which could be automatically recovered by virtual machine live migration or restarting. Facility failures are caused by the environments, including power outages or A/C failures, and could be regarded as non-recoverable failures that cannot be recovered by the system itself but is usually harmless to the key components and functions of the system. Network failures make the system turn into the *Inaccessible* state. As the "level of catastrophe" is to be defined by the user, we assume the hardware failures are catastrophic, which could cut down the service immediately and take much time and money to repair and restart the whole system.

7.2 Dependability analysis

It has been shown in [30] that, with the statistics of the data in [31], the time between failures well conforms to Weibull distribution and the repair times are well modeled by a lognormal distribution. Therefore, some of the existing approaches such as [12], [20] and [26] to model and analyze dependability or some of its attributes might not work properly, because in such approaches it was assumed that the sojourn times in the states were exponentially distributed. Instead, the model proposed in this paper could fill this gap and could evaluate the dependability without losing accuracy.

Note that as the data set does not contain the records that show whether improper alterations or destructions have occurred in the system, hence the integrity could not be evaluated precisely with this data set. Therefore, the simplified model shown in Fig. 3 would be used to evaluate part of the attributes of dependability. In some certain types of service systems, however, where the integrity is one of the crucial attributes in system design, the events of improper alterations or destructions should be recorded, which could help to analyze the integrity of the system using the comprehensive model shown in Fig. 2.

With the data set, the parameters in the model could be obtained using statistical methods. Then, we have the steady-state probabilities of the states in the SMP model and its embedded DTMC, with which some of the dependability attributes could be calculated using the analysis methodology proposed in Section 4. The maintainability and safety of the 19 systems are shown as Fig. 8.

The MTTFs and MTTR of the system could also be calculated using the proposed models and methods with the limited data source, shown as Fig. 9 and 10. The results could also validate the conclusion drawn in Theorem 1.

The figures show the dependability attributes vary significantly across systems. This fact might be caused by the high variation of the system parameters such as failure rates, repair times, etc. across different types of systems and different intensities of the workload running on them [30]. Hence, the correlations between the dependability attributes and the system parameters are with reference value for system design and optimization, which will be discussed in the following subsection.

7.3 Sensitivity analysis of Dependability

Sensitivity analysis is often performed on dependability and performance models so that the system can be optimized, parts of the system model sensitive to error or failures can be identified, and bottlenecks in the system can be found [35]. We perform parametric sensitivity analysis on the SMP model and examine the sensitivity of the dependability and MTTFs of the LANL service systems based on data sets got from reality.

Assume M is the derivative of a measure to compute, with respect to various system parameters λ_i. The sensitivity of M to parameter λ_i could be expressed as (50). In practice, the sensitivity could be calculated with the definition of partial differential coefficient, expressed as (51) where $\Delta\lambda_i$ is infinitely small to λ_i. Refining the parameter λ_i that results in the maximum value of $S_{\lambda_i}(M)$ is the most effective way to optimize the system.

$$S_{\lambda_i}(M) = \left| \frac{\partial M}{\partial \lambda_i} \right|. \tag{50}$$

$$S_{\lambda_i}(M) = \left| \frac{M(\lambda_i + \Delta\lambda_i) - M(\lambda_i)}{\Delta\lambda_i} \right|. \tag{51}$$

Fig. 11 shows the values of sensitivities of the dependability attributes of the systems to the sojourn time parameters, while Fig. 12 illuminates the sensitivities to transition probabilities in the stochastic model. The values on the Y axis indicate different effects of different parameters upon different dependability attributes and the MTTFs in different systems, which could help to identify the system bottlenecks and find the efficient way for system optimization under different requirements for dependability. For example, according to Fig. 11, the sojourn time in *Ready* state i.e. h_{RE} is the most effective parameter to MTTRF, MTTMF or MTTSF, which means the advanced technique of hardware reliability enhancement

making it more likely to maintain the system in *Ready* state will help a lot to enhance these attributes. However, h_{RE} has much less effect than the other parameters to MTTR in most of the systems. We could also see that h_R and h_U are commonly two important factors for maintainability and safety optimization. Shown as Fig. 12, the increase of the probability of system entering *Recoverable* state instead of the other two *Failed* state when failure occurs, saying p_R' is one of the key factors for maintainability, MTTMF or MTTSF enhancement; while decreasing p_N' by some effective means to prevent the systems from entering *Non-recoverable* state will help a lot for the enhancement of the MTTR, because the sojourn time in *Non-recoverable* state is the longest in most of the systems.

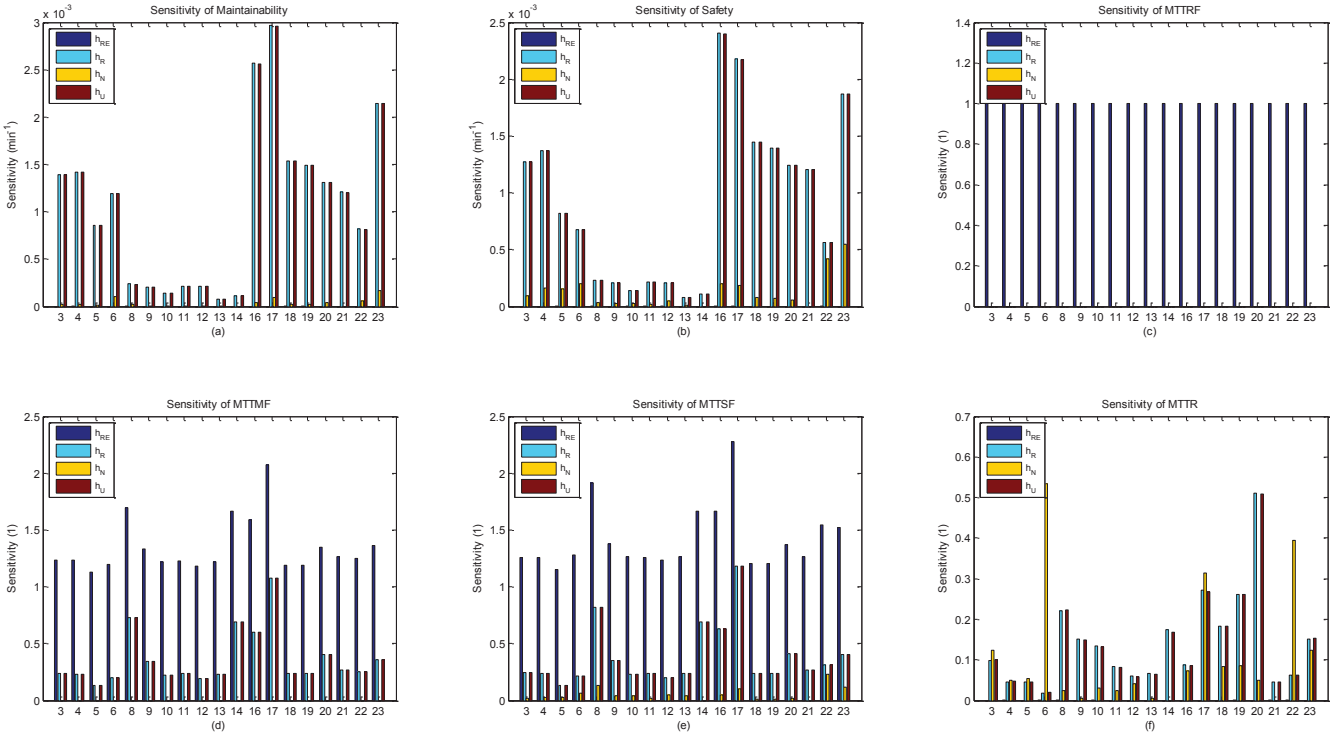


Fig. 11. Sojourn time parameter sensitivity analysis

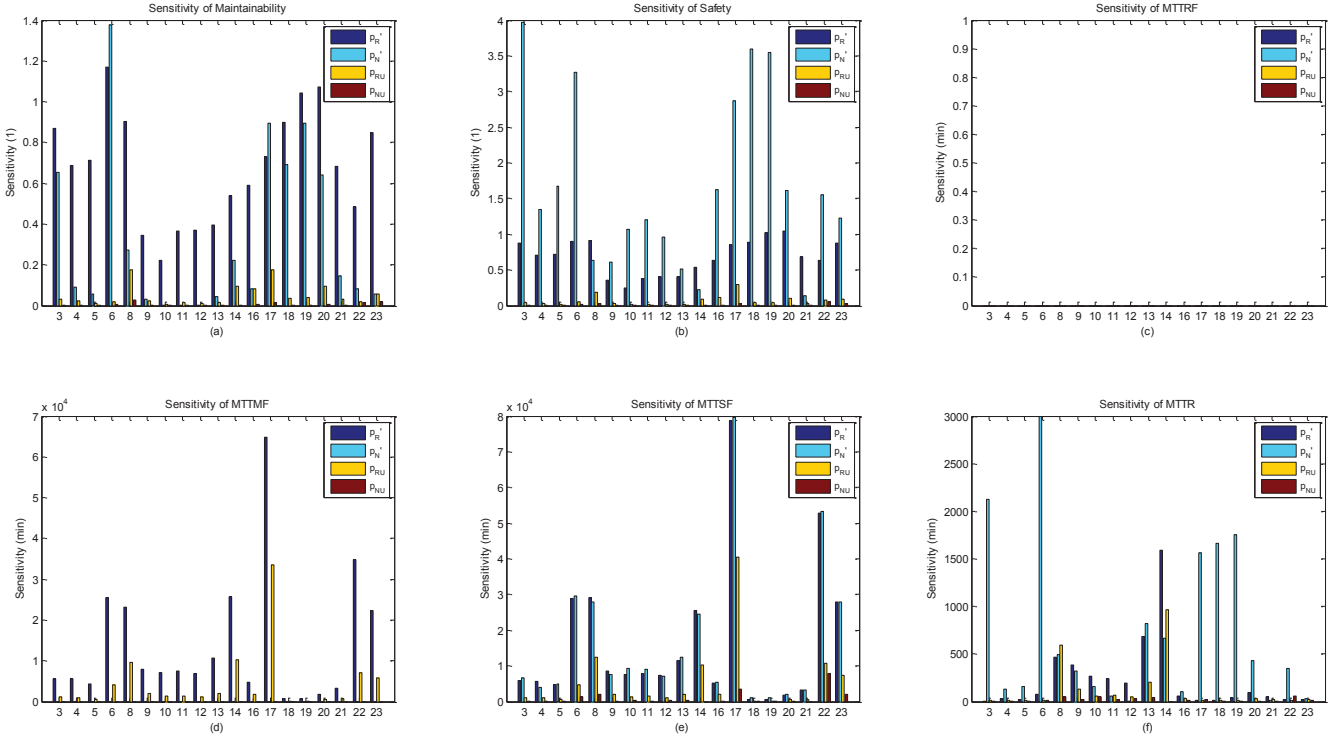


Fig. 12. Transition probability parameter sensitivity analysis

Note that in the simplified SMP model, $MTTRF = h_{RE}$ shown as the expression in (23), the sensitivity of MTTRF to h_{RE} should be 1, and the sensitivity to all the other parameters should be 0, which is validated both by Fig. 11(c) and Fig. 12(c). As we know that the MTTRF represents the average time it takes to reach one of the failure states given that the system starts in *Ready* state, it is clear that the MTTRF should closely rated to how long the system maintains in *Ready* states which is evaluated by h_{RE}, instead of how the system reaches the *Ready* state from the other states, which is the meaning of the other parameters such as h_R, h_U, h_N, p_{RU} or p_{NU}. Either, the MTTRF should be irrelevant to which type of failure would occur once the system fails, which is expressed by p_R' or p_N'. Hence, the result of sensitivity analysis could be partially validated according to the definitions of the metrics.

8 Conclusion

Dependability has always been an important issue for design and optimization of computing systems. In services computing systems, the classical dependability and evaluation methods, which are based on the components and systems, are no longer feasible. This paper veers from the angle of view, and studies the dependability and its attributes from the service-oriented perspective by taking a top-down approach. We summarize the relevant research on dependability of services computing systems, and propose a stochastic state transition model that describes the dynamic behavior of the systems and users. Based on the model, we give the formal calculation of the dependability attributes, including transient-state evaluation, steady-state evaluation and the mean time to failure analysis. Furthermore, the relationships and comparisons of the dependability attributes are studied by proposing and proving some theorems. In addition, the service selection and composition models are proposed to evaluate the dependability of the complex services. Moreover, we study a case in reality i.e. the LANL service systems to show how this model and analysis work in real systems. The dependability attributes and MTTFs are analyzed, and sensitivity analysis is performed to find the bottlenecks and show the most efficient ways of system parameter optimization for the enhancement of certain attributes. This work is expected to offer a useful reference for design and optimization of services computing systems.

One of the goals of our future work is to propose a real-time dependability evaluation and optimization framework in services computing systems, especially for some high dependability required systems. Some efficient parameter measurements to handle high dynamics, real-time dependability guarantee and dynamic optimization mechanisms based on the state transition model and its related analysis could be studied. Also, the data set in real systems with the framework and mechanisms could provide us with a better understanding of the model and system behavior, and help to find effective ways for dependable system design and optimization.

In addition, we believe that the idea of the proposed models and analysis methods could have reference value in security evaluation of services computing systems. The confidentiality and authentication, which are also closely related to user and system behavior, could be analyzed using such type of state transition model and its related methods. The analysis of the multiple attributes and their relations and tradeoffs will yield insights into different services computing systems' strengths and weaknesses and offer better reference for the system development and optimization.

Acknowledgment

The authors would like to express their thanks to the associate editor and the reviewers for their constructive and useful comments. This work is supported by the National Grand Fundamental Research 973 Program of China (No. 2010CB328105, No. 2009CB320504), and the National Natural Science Foundation of China (No. 60932003, No. 61020106002).

References

[1] I. Foster, "Service-Oriented Science," *Science Magazine*, vol. 308, no. 5723, pp. 814-817, May 2005, DOI: 10.1126/science.1110411.

[2] Y. S. Dai, B. Yang, J. Dongarra, and G. Zhang, "Cloud Service Reliability: Modeling and Analysis," in PRDC, 2009.

[3] A. Avizienis, J. C. Laprie, B. Randell, and C. Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing," *IEEE Trans. Dependable and Secure Computing*, vol. 1, no. 1, pp. 11-33, 2004, doi: 10.1109/TDSC.2004.2.

[4] J. C. Laprie, *Dependability: Basic Concepts and Terminology*, Springer-Verlag, 1992.

[5] B. E. Helvik, "Perspective on the Dependability of Networks and Services", Telektronikk 3, pp. 27-44, 2004.

[6] A. Avizienis, J. C. Laprie, B. Randell, and C. Landwehr, "Fundamental Concepts of Dependability," UCLA CSD Report no.010028, LAAS Report no.01-145, Newcastle University Report no. CS-TR-739, 2001.

[7] D. M. Nicol, W. H. Sanders, and K. S. Trivedi, "Model-Based Evaluation: From Dependability to Security," *IEEE Trans. Dependable and Secure Computing*, vol. 1, no. 1, pp. 48-65, 2004, doi: 10.1109/TDSC.2004.11.

[8] K. S. Trivedi, *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*, second ed, New York: John Wiley and Sons, 2001.

[9] J. B. Dugan and M. R. Lyu, "Dependability Modeling for Fault-Tolerant Software and Systems," *Software Fault Tolerance*, M.R. Lyu, ed., Chichester: John Wiley & Sons, pp. 109-138, 1995.

[10] B. Schneier, *Secrets and Lies: Digital Security in a Networked World*, John Wiley & Sons, Aug. 2000.

[11] J. K. Muppala, M. Malhotra, and K. S. Trivedi, "Markov Dependability Models of Complex Systems: Analysis Techniques," *Reliability and Maintenance of Complex Systems*, S. Ozekici, ed., Germany: Springer, pp. 442-486, 1996.

[12] M. Malhotra and K. Trivedi, "Dependability Modeling Using Petri Nets," *IEEE Trans. Reliability*, vol. 44, no. 3, pp. 428-440, Sep. 1995, doi: 10.1109/24.406578.

[13] S. Majithia, D. W. Walker, and W. A. Gray, "A Framework for Automated Service Composition in Service-Oriented Architectures," *Lecture Notes in Computer Science*, vol. 3053, pp. 269-283, 2004, doi: 10.1007/978-3-540-25956-5_19.

[14] L. Zhang and M. Jeckle, "The Next Big Thing: Web Services Collaboration," *Lecture Notes in Computer Science*, vol. 2853, pp. 123-130, 2003, doi: 10.1007/978-3-540-39872-1_1.

[15] P. Ruth, X. Jiang, D. Xu, and S. Goasguen, "Virtual Distributed Environments in a Shared Infrastructure," *Computer*, vol. 38, no. 5, pp. 63-69, May 2005, doi: 10.1109/MC.2005.175 .

[16] D. A. Menasce, "Virtualization: Concepts, Applications, and Performance Modeling," *Proc. 31th International Computer Measurement Group Conference (CMG '05)*, pp. 407-414, Dec. 2005.

[17] M. R. Marty and M. D. Hill, "Virtual hierarchies to Support Server Consolidation," *Proc. 34th annual International Symposium on Computer Architecture (ISCA '07)*, pp. 46-56, 2007, doi: 10.1145/1250662.1250670.

[18] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live Migration of Virtual Machines," *Proc. 2nd ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI '05)*, pp. 273-286, May 2005.

[19] A. Heddaya and A. Heldal, "Reliability, Availability, Dependability and Performability: A User-Centered View". Technical Report. Boston University. Computer Science Department. 1996.

[20] D. Wang and K. S. Trivedi, "Modeling User-Perceived Service Availability," *Proc. 2nd International Service Availability Symposium (ISAS '05)*, pp. 107-122, Apr. 2005, doi: 10.1007/11560333_10.

[21] M. Calzarossa, R. A. Marie, and K. S. Trivedi, "System Performance with User Behavior Graphs," *Performance Evaluation*, vol. 11, no. 3, pp.155-164, 1990, doi: 10.1016/0166-5316(90)90008-7.

[22] M. Zyla, and D. Caban, "Dependability Analysis of SOA Systems," *Proc. 3rd International Conference on Dependability of Computer Systems (DepCoS-RELCOMEX '08)*, pp. 301-306, 2008, doi: 10.1109/DepCoS-RELCOMEX.2008.20.

[23] N. Looker, M. Munro, and J. Xu, "Simulating Errors in Web Services," *International Journal of Simulation*, vol. 5, no. 5, pp. 29-37, 2004.

[24] Y. S. Dai, M. Xie, K. L. Poh, and G. Q. Liu, "A Study of Service Reliability and Availability for Distributed Systems," *Reliability Engineering and System Safety*, vol. 79, no. 1, pp. 103-112, 2003, doi: 10.1016/S0951-8320(02)00200-4.

[25] D. Oppenheimer and D. A. Patterson, "Architecture and Dependability of Large-Scale Internet Services," *IEEE Internet Computing*, vol. 6, no. 5, pp. 41-49, 2002, doi: 10.1109/MIC.2002.1036037.

[26] N. Sato and K. S. Trivedi, "Accurate and Efficient Stochastic Reliability Analysis of Composite Services Using Their Compact Markov Reward Model Representations," *Proc. IEEE International Conference on Services Computing (SCC '07)*, pp. 114-121, 2007, doi: 10.1109/SCC.2007.21.

[27] J. Huang, C. Lin, X. Kong, and Y. Zhu, "Modeling and Analysis of Dependability Attributes of Service Computing Systems," *Proc. IEEE International Conference on Services Computing (SCC '11)*, pp. 184-191, 2011, doi: 10.1109/SCC.2011.88.

[28] E. Jonsson. "Towards an Integrated Conceptual Model of Security and Dependability," *Proc. 1st International Conference on Availability, Reliability and Security (ARES '06)*, pp. 646-653, 2006, doi: 10.1109/ARES.2006.138.

[29] B. B. Madan, K. Gogeva-Popstojanova, K. Vaidyanathan, and K. S. Trivedi, "Modeling and Quantification of Security Attributes of Software Systems,". *Proc. International Conference on Dependable Systems and Networks (DSN '02)*, pp. 505-604, 2002, doi: 10.1109/DSN.2002.1028941.

[30] B. Schroeder and G.A. Gibson, "A Large-Scale Study of Failures in High-Performance Computing Systems," *IEEE Trans. Dependable and Secure Computing*, vol. 7, no. 4, pp. 337-351, 2010, doi: 10.1109/TDSC.2009.4.

[31] Los Alamos National Laboratory, "All Systems Failure/Interrupt Data 1996-2005," http://institute.lanl.gov/data/fdata/.

[32] A. Birolini, *Reliability Engineering Theory and Practice*, Fifth edition. Springer, pp. 2-10, 2007.

[33] I. Koren and C. ManiKrishna, *Fault-tolerant Systems*, Morgan Kaufmann Publishers, pp. 4-6, 2007.

[34] U. Bhat, *Elements of Stochastic Processes*, 2Ed., John Wiley, New York, 1984.

[35] J. T. Blake, A. L. Reibman, and K. S. Trivedi, "Sensitivity Analysis of Reliability and Performability Measures for Multiprocessor Systems," *Proc. 1988 ACM SIGMETRICS conference on Measurement and modeling of computer systems*, pp. 177-186, 1988, doi: 10.1145/55595.55616.

[36] W.M.P van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros, "Workflow Patterns," Distributed and Parallel Databases, 14(3), pages 5-51, July 2003, doi: 10.1023/A:1022883727209.

[37] L. Zhang, J. Zhang, and H. Cai, "Services Computing," Springer, 2007.

[38] L. Zhang, "Services Computing: a New Discipline," Editorial Preface, International Journal on Web Services Research (JWSR) vol.2, no.1, 2005.

[39] G. Serme, A. S. Oliveira, J. Massiera, and Y. Roudier, "Enabling Message Security for RESTful Services," *Proc. IEEE International Conference on Web Services (ICWS'12)*, pp. 114-121, 2012, doi: 10.1109/ICWS.2012.94.

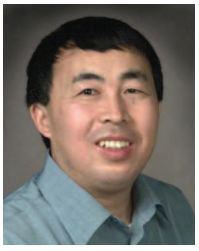
[40] M. Anisetti, C. A. Ardagna, and E. Damiani, "A Low-Cost Security Certification Scheme for Evolving Services," *Proc. IEEE International Conference on Web Services (ICWS '12)*, pp. 122-129, 2012, doi: 10.1109/ICWS.2012.53.

Jiwei Huang received the B.Eng. degree in computer science and technology from Tsinghua University, China, in 2009. He is currently working toward the Ph.D. degree with the Department of Computer Science and Technology, Tsinghua University, China. His research interests are in modeling, simulation and performance analysis of computer systems, data centers and cloud computing.

Chuang Lin (IEEE SM'04) is a professor of the Department of Computer Science and Technology, Tsinghua University, Beijing, China. He received the Ph.D. degree in Computer Science from the Tsinghua University in 1994. His current research interests include computer networks, performance evaluation, network security analysis, and Petri net theory and its applications. He has published more than 300 papers in research journals and IEEE conference proceedings and has published three books. He is a member of ACM Council, a senior member of the IEEE and the Chinese Delegate in TC6 of IFIP. He serves as the Associate Editor of IEEE Transactions on Vehicular Technology, the Area Editor of Journal of Computer Networks and the Area Editor of Journal of Parallel and Distributed Computing.

Xiangzhen Kong received his B.S. degree in computer science from Xi'an Jiaotong University, China, in 2007. He is currently a Ph.D. candidate in the Department of computer science and technology, Tsinghua University, China. His research interests include system virtualization, performance and dependability evaluation on computing system, Grid and Cloud computing.

Bing Wei received his B.S. degree in computer science from Xi'an Jiaotong University, China, in 2010. He is currently a M.S. student in the Department of computer science and technology, Tsinghua University, China. His research interests are in performance and dependability evaluation of computing system and Cloud computing.



Xuemin (Sherman) Shen (IEEE M'97-SM'02-F'09) received the B.Sc. (1982) degree from Dalian Maritime University (China) and the M.Sc. (1987) and Ph.D. degrees (1990) from Rutgers University, New Jersey (USA), all in electrical engineering. He is a University Research Chair Professor, Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on mobility and resource management in interconnected wireless/wired networks, UWB wireless communications networks, wireless network security, wireless body area networks and vehicular ad hoc and sensor networks. He is a coauthor of three books, and has published more than 400 papers and book chapters in wireless communications and networks, control and filtering. He is a Distinguished Lecturer of IEEE Communications Society. He received the Excellent Graduate Supervision Award in 2006, and the Outstanding Performance Award in 2004 and 2008 from the University of Waterloo, the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada, and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo. He is a registered Professional Engineer of Ontario, Canada.