Logistic Support Architecture with Petri Net Design in Cloud Environment for Services and Profit Optimization

Fuu-Cheng Jiang, *Member, IEEE,* Ching-Hsien Hsu*, Senior *Member, IEEE* and Shangguang Wang, *Member IEEE*

Abstract—Cloud computing refers to both the applications delivered as services over the Internet and hardware and system software in the cloud server farm that provides those services. The research on server backup of cloud server farm appears to be an important issue of cloud computing economics. Optimal logistic policy should be considered to be a profit-oriented framework simultaneously for providing qualified service to cloud users while the whole cloud center is under construction. The kernel point of the proposed approach is that a novel design pattern is developed for approaching optimal profit on logistics using the finite-source queuing theory. To model the proposed approach for qualitative analysis, a Petri Net model was developed to configure all relevant system aspects in a concise fashion. On quantitative work, a comprehensive mathematical analysis on profit pattern has been made in detail. Relevant simulations have also been conducted to validate the proposed optimization model. The design illustration is presented to demonstrate engineering application scenario in cloud environment, hence the proposed approach indeed provides a feasibly profit-oriented framework to meet logistic economy.

Index Terms—cloud computing, Petri Nets, logistic support, profit optimization.

1 Introduction

1.1 Background

CLOUD has emerged to be as alternative to conventional office-based computing environment. The promise of cloud computing (CC) is to deliver all the functionality of existing information technology (IT) services even as it dramatically reduces the upfront costs of computing that deter many organizations from deploying many cutting-edge IT services [1~3]. It implies an economical developing model that engineers or developers no longer require the large capital outlays in hardware to deploy their service or the administrational expense to keep the service steadily. With the ever-increasing popularity of the cloud platform, resource management attracts more attention from biology, medicine, engineering and social science [4] [5].

To mitigate unavoidable impact from the failures of servers in operation, a novel logistic framework to optimize the profit gained by cloud providers in terms of designing the spare profile for the server farm. On the balance of profit and cost, the logistic expense would drive the profit to the negative side although it can make the profit income more stably by maintaining service quality in good condition. Even more, any cloud provider with excellent reputation of service quality may attract more potential cloud users for profit. To explore the tradeoff study on them, the proposed optimization technique may provide the cloud expert with deployment scheme on the number of spare servers in case of server failure for profit optimization.

1.2 Contribution Profile

This novel idea in the proposed logistic support architecture (LSA) is originated from the theory of finite source queue (FSQ) model [6] [7]. At some pre-configured period, there exists a finite quantity of online (operating) servers to provide cloud service under contract-based commitment for cloud customers. On application modeling, such finite quantity of online servers can be regarded as the term: "finite source" in the FSQ model of queuing theory. The standby concept is the basic scheme to maintain operation with regulated service quality even when some components fail. The proposed system architecture consists of three subsystems: online subsystem, spares subsystem, and repair facility. The numbers of servers in the online subsystem are deployed according to service quality requirements. Whenever one of online servers fails, it is immediately replaced by a standby server in the spares subsystem and the failed server is delivered to the repair facility as well.

The design goal for this work is to explore the issue: On the profit-based logistics, how many standby servers in the spares subsystem would be optimal if a certain level of the

- *Fuu-Cheng Jiang is with the Department of Computer Science, Tunghai-Universitythe, No. 1727, Section 4, Taichung Boulevard, Taichung, Tawan. E-mail: admor@thu.edu.tw.*
- *Ching-Hsien Hsu is with the Department of Computer Science and Information Engineering, Chung Hua University, 707, Sec. 2, WuFu Rd., Hsinchu, Taiwan; Tianjin Key Laboratory of Intelligence Computing and Novel Software Technology, Tianjin University of Technology, 300191, Tianjin, China. E-mail: robertchh@gmail.com.*
- *Shangguang Wang is with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. E-mail: sgwang @bupt.edu.cn*

** The corresponding author*

xxxx-xxxx/0x/$xx.00 © 200x IEEE Published by the IEEE Computer Society

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2016.2514506, IEEE Transactions on Services Computing

server availability is kept? To explore the tradeoff study on them, the proposed optimization technique may provide the cloud expert with decision support on the number of spare servers. The key contributions of this paper are threefold: (1) this work provides cloud administrator with a feasible logistic framework to optimize the profit improvement. On management aspect, the proposed system can be adopted to be a decision-making methodology approaching predictive management other than reactive or chaotic management. (2) On qualitative study, a Petri Net is developed and designed to visualize the whole system operational flow. On quantitative aspect, M/M/1 FSQ with spares models has been newly derived and relevant system metrics has been established in a brand-new manner. (3) On verification aspect, relevant experimental results are conducted and obtained in terms of three different warm-standby configurations. The simulated results indicate that the proposed approach may provide a feasible decision support for deployment on quantities of standby servers.

The rest of the paper is organized as follows: Section 2 describes related work and the motivation behind this research. To demonstrate the logistic framework qualitatively, an FSQ theory is adopted and designed using Petri Nets for visualizing system flow in Section 3. On quantitative work in Section 4, the mathematical analysis has been conducted in detail and also relevant system performance measures like the expected number of online servers, the expected number of spares, etc. have been derived. Following this, in Section 5, the finite-source model is further addressed in terms of profit function, which simulations are conducted as well for the feasibility of the proposed scheme. Finally, some concluding remarks are made in Section 6.

2 Related Work

The service level agreement (SLA) [8] is the promised performance metrics regulated in the contract. The breakdown of operating servers would be experienced to be a crucial source of ruining the SLA. Hence, the backup policy on cloud logistics has emerged to be definitely indispensable for guaranteeing overall system performance in redundancy design of cloud platform. In [9], the combination of M/M/1 and M/M/m in sequence was proposed to model the cloud platform. Their work showed that to provide good quality of service in terms of response time, the cloud experts have to determine where the system has a bottleneck and then improve the corresponding parameter. On node failure in CC network, an algorithm was proposed to estimate the network performance under maintenance budget with nodes failure [10]. This work focused on the issue of maintenance reliability under the maintenance budget and time constraints. However it does not address the issue how to optimal the profit by logistic policy while confronted with breakdowns of servers.

Backup servers are effective for maintaining the high availability of cloud services against hardware failures and disasters. Hu et al. [11] proposed a backup sever sharing scheme in the Inter-cloud to reduce the cost of backup servers. Their numerical modeling was solely based upon the availability of computation without any of solid queuing materials. In the work of Chen et al. [12], the authors investigated and proposed a mobile cloud-based architecture for enrichment of existing logistics systems. Following the system architecture is the main issues including customer interaction, distributed storage and computing, real-time tracking, logistics vehicle scheduling and mobile payment, etc. In the research of Li et al. [13], the authors investigated how to optimize the monetary cost of purchasing cloud VMs for the hybrid cloud computing paradigm. They specifically tailored a theoretical model based on Lyapunov Optimization framework according to the real-world challenges of this problem.

Lee et al. [14] had developed a pricing model using processor-sharing for clouds with the consideration of queuing delay to be a component of the processing time. In addition to developing a pricing model using processor-sharing for clouds, the authors presented algorithms for scheduling service requests and prioritizing data access in the cloud with the main objective of maximizing profit. None of the abovementioned works addressed the issue on facing the scenario in case of failures of operating servers. It is noted that although an M/G/m/(m+r) queuing system has been considered [15], the M/M/m queuing model is the only model that accommodates an analytical and closed-form expression of the probability density function of the waiting time. None of backup scheme was presented in their work for profit consideration.

Virtualization technologies allow cloud datacenters to improve resource utilization and energy efficiency. Maliks et al. [21] compares three open source virtual machine based cloud management platforms: Eucalyptus, Open Nebula and Nimbus. High-level Petri-Net was adopted for modeling and analyzing the system architecture and performance. Ghosh et al. [22] studies the component failure problem in a large Infrastructure-as-a-Service (IaaS) cloud using a scalable, stochastic model-driven approach Stochastic Petri Net (SPN), which is an interacting Markov chain based approach to demonstrate the reduction in the complexity of analysis and the solution time. Based on virtualization techniques, dynamic server consolidation through live migration is an efficient way towards energy conservation in Cloud data centers. In general, server consolidation consists of three steps, estimation of instance resource demand, selection of released servers, and migration of instances on released servers. Most of previous work used a fixed value as instance resource demand profile [23]. Instead of server consolidation, Hsu et al. [24] present an energy-aware task consolidation (ETC) technique to minimize energy consumption in Cloud data center. The main idea of the ETC is to restrict CPU use below a specified peak threshold by consolidating tasks amongst virtual clusters. Similar to energy saving, Chiang et al. [25, 26] proposed power-saving methods for eliminating unnecessary idle power consumption in cloud systems. To address the conflict issue between performances and power saving, a tradeoff between power consumption cost and system congestion cost is conducted.

Motivated by the materials in [15], it compels us to expand the applicability into server backup scheme of cloud platform using queuing theory. Queuing theory has provided numerous applications on production systems, transportation

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2016.2514506, IEEE Transactions on Services Computing

systems, telecommunication networks and other scientific or engineering fields for its solid mathematical frameworks. When an online server fails, it should be moved to the repair subsystem and replaced by a standby server immediately. The number of spares in the standby subsystem is a decision parameter specified in the concept of profit model for the logistic policy. This threshold could be used to optimize the expected profit function for the cloud provides under the constraint of maintaining the service level agreement with their cloud users. The standby subsystem equipped with larger amount of standby servers implies higher guarantee for the service level agreement in terms of higher availability, but inevitably such approach would incur much more expenditure and the profit would be deteriorated accordingly. Instead of choosing this threshold in a haphazard way, an optimal scheme based on the finite-source queuing theory was proposed as a decision support for the cloud enterprise.

3 Logistic Framework and Petri Net Design

3.1 Logistic Support Architecture

The proposed framework of logistics can be modeled as an M/M/1 finite source queue (FSQ) with spares which is also termed as machine repair problem [6] [7]. The jargon "finite source" in FSQ theory can be regarded as the finite quantity of online serves provided to guarantee service quality by cloud providers. On the FSQ model with spares (or called standby servers), a cluster of identical servers (online and standby) are maintained by a repairman in the repair facility. The logistic support architecture (LSA) is considered to have N = M + S identical servers and one repairman. As many as M of these servers can operate simultaneously in parallel, the rest of the S servers are regarded as spares. Whenever one of these servers fails, it is immediately replaced by a spare if any is available and the failed server is delivered to the repair facility as well.

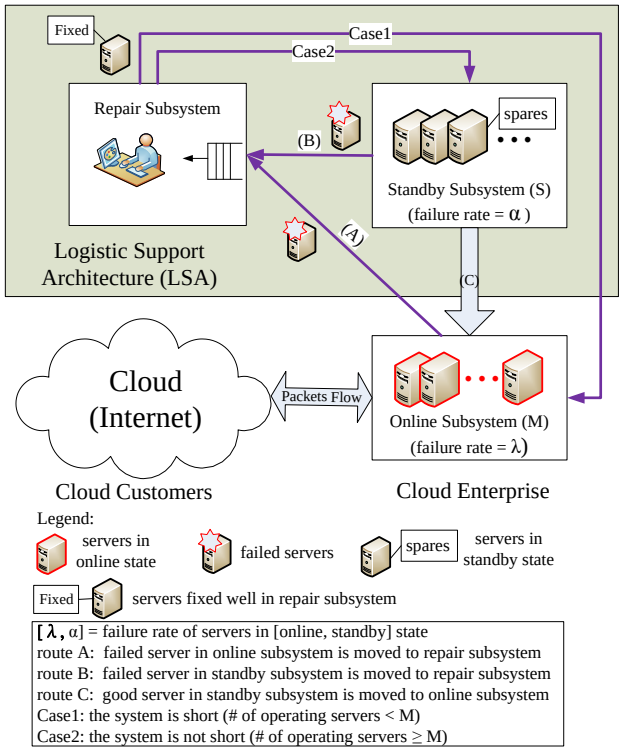

Fig. 1. Conceptual profile for the proposed LSA

Without the loss of generality, it is assumed that each of online servers fails independently of the others with failure rate λ. It is reasonably assumed that when any spare moves into an online state, its failure characteristics will be that of an online server. We study the behavior of finite-source queuing model, in which the operational flow for logistic pattern on cloud server farm can be modeled as shown in Fig. 1. In Fig. 1, whenever an online server or a spare fails, it is immediately sent to the repair facility and repaired by the repairman with identical repair rate μ. These two flows for an online server and spare are depicted by two arrows marked with symbols λ and α respectively. If no spare is available when failure occurs, then the system is short. Once a server is repaired, it becomes a spare, unless the system is short, in which case the repaired server goes immediately into service. Their corresponding flows are depicted by two arrows with symbols: Case1 and Case2 respectively in Fig. 1.

3.2 Petri Net concepts

Petri Nets (PNs) combine descriptive function and formal verification procedures with extensive possibilities for quantitative processing. They are a graphical mathematical modeling tool application to many systems [16]. They are a promising tool for describing and studying information processing systems that are characterized as being concurrent, asynchronous, distributed, parallel, nondeterministic, and/or stochastic. A PN is identified as a particular kind of bipartite directed graph populated by three types of objects. They are places, transitions, and directed arcs connecting places and transitions. A PN is a 5-tuple, PN ={P, T, I, O, M0} [17] where:

$P = \{p_1, p_2, \ldots, p_m\}$ is a finite set of places, where $m > 0$;

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2016.2514506, IEEE Transactions on Services Computing

$T = \{t_1, t_2, \ldots, t_n\}$ is a finite set of transitions with $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$, where $n > 0$;
I: $P \times T \rightarrow N$ is an input function that defines a set of directed arcs from P to T,
where $N=\{0, 1, 2\ldots\}$;
O: $T \times P \rightarrow N$ is an output function that defines a set of directed arcs from T to P.
M_0: $P \rightarrow N$ is the initial marking.

In graphical representation, places are drawn as circles, transitions as bars or boxes. A transition t is enabled if each input place p of t contains at least the number of tokens equal to the weight of the directed arc connecting p to t. When an enabled transition fires, it removes the tokens from its input places and deposits them on its output places. PN models are suitable to represent the system that characterizes event-driving, choice, concurrency, conflict, and synchronization. In modeling, using the concept of conditions and events, places represent conditions, and transitions represent events. A transition (an event) has a certain number of input and output places representing the pre-conditions and post-conditions of events, respectively. The behavior of many systems can be described in terms of system states and their changes. An event/condition PN model is developed for the system framework with the events causing the state transition for the propose approach. Other detailed properties, analysis and applications on PN can be found in [18].

3.3 Framework design using Petri Nets

Based on the prior system view in Fig. 1, the PN-based framework of the proposed system is composed of three subsystems: Online subsystem, Standby subsystem and Repairmen subsystem. The operational flow can be modeled using the command/response concept [19]. As shown in Fig. 2, each step of operational pattern can be regarded as an action item with a start transition, progressive place, end transition and completed place. The movement of server from one subsystem to another can be triggered with the transition depicted with dark symbol.

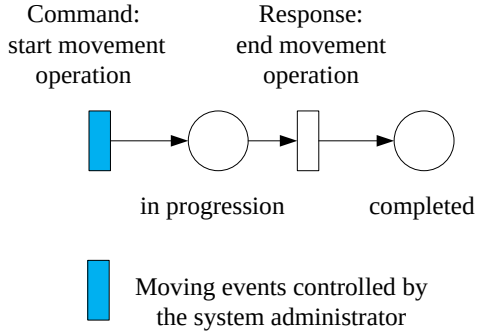

Fig. 2. Server movement behavior using the command/response concept

To shed light on the adoption of the command/response concept, the PN-based repair subsystem is constituted and illustrated in Fig. 3, where the operational flow: t2→P2→t4→P3 conveys the command/response concept to respond the event triggered by a server has failed in the online subsystem. The place P3 implies the completion of server movement and the server has entered the repair subsystem.

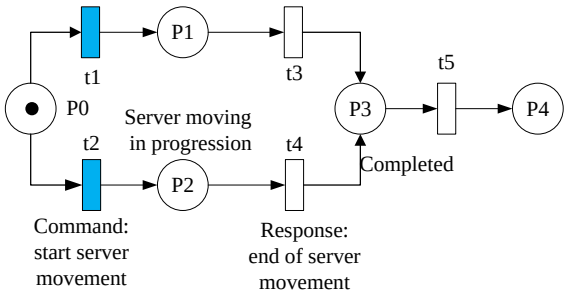

Fig. 3. Scenario of server movement in PN-based repair subsystem using the command/response concept

By adopting the command/response concept and implementing the system description, the PN-based framework is illustrated in Fig. 4, which consists of 10 places and 11 transitions. The corresponding PN notations for states (places) and transitions (events) are defined in PN_Notation 1 and PN_Notation 2 respectively. The place P4 represents the depository of servers fixed ready by the repairmen for reuse, which two transitions t6 and t7 would be triggered to conduct the server movement operations to the online subsystem and the standby subsystem respectively.

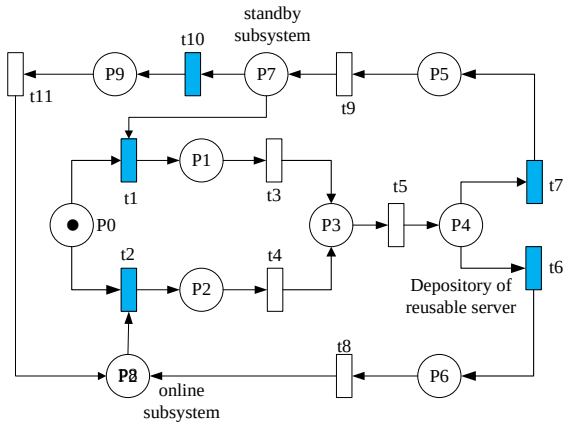

Fig. 4. PN-based framework design on the proposed system

PN_Notation_1 There are ten transitive states (places) on constituting the operational framework:
P0: The system is available
P1: Failed server moving to the repair facility from standby subsystem
P2: Failed server moving to the repair facility from online subsystem
P3: Servers in the repair facility for conducting repair operation
P4: Depository of servers fixed ready for reuse again
P5: Good server moving to the online subsystem
P6: Good server moving to the standby subsystem
P7: Server in the standby subsystem
P8: Server in the online subsystem
P9: Spare moving to the online subsystem.

PN_Notation_2 Given the set of transitive states (places)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2016.2514506, IEEE Transactions on Services Computing

B={P0, P1, … ,P9}, we define the state transitions (events) during modeling proposed system: (Cmd: Command; Re: Response)

t1:Cmd: start moving the failed server in the standby subsystem fails with failure rate λ

t2: Cmd: start moving the failed server in the online subsystem fails with failure rate α

t3: Re: end moving to the repair subsystem

t4: Re: end moving to the repair subsystem

t5: Enabled when a failed server is fixed well with repair rate μ

t6: Cmd: start moving a good server to the standby subsystem when the number of servers in the online subsystem is more than M

t7: Cmd: start moving a good server to the online subsystem when the number of servers in the online subsystem is less than M (i.e., the system is in short.)

t8: Re: end moving to the standby subsystem

t9: Re: end moving to the online subsystem

t10: Cmd: start moving a spare to the online subsystem when the number of servers in the online subsystem is less than M

t11: Re: end moving to the online subsystem.

4 Theoretical Model on Logistics

4.1 Mathematical background

The profit-oriented model on the LSA is considered to have N = M + S identical servers and one repairman. As many as M of these servers are regulated to provide service requirements for cloud users on the scene, the rest of the S servers are regarded as standby servers. Let the states n, n = 0, 1, 2,…, N (N = M + S), represent the number of failed servers in the system. The mean failure rate on the servers for M/M/1 MRP with spares is as follows:

$$\lambda_n = \begin{cases} M\lambda + (S-n)\,\alpha, & \text{if } n = 0,1,\ldots,S; \\ (M+S-n)\,\lambda, & \text{if } n = S+1, S+2,\ldots, N=S+M; \\ 0, & \text{Otherwise.} \end{cases} \quad (1)$$

Where the parameter vector: [λ, α] = [the failure rate of an operating server, the failure rate of a spare]. The mean repair rate by the repairman is given by $\mu_n = \mu$. To approach analytic steady-state results for the proposed model, we first construct the state-transition-rate diagram depicted in Fig. 5. The number inside the circle represents the number of failed servers in the system. Each circle in Fig. 5 shows the steady-state probability scenario that may happen during service in the system. For each circle except the first one (n = 0) and the last one (n = N), there are four arrows marked with the corresponding values of state-transition rate. The quantity marked along each arrow implies either flow-in probability into that state or flow-out probability off that state.

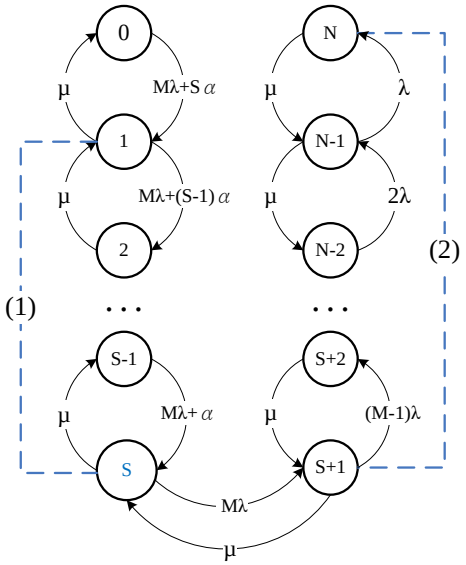

Fig. 5. State-transition-rate diagram for the proposed model

Let the notation: P(n) = the probability that there are n failed servers in the system where n=0,1, 2,.., N. Hence P(0) implies the probability that there are no server failed in the system. For steady-state case, the state probability functions P(n) can be obtained from the birth-and-death formula [7] in association with the state-transition-rate shown in Fig. 5. We define notations $\theta_\lambda = \lambda/\mu$ and $\theta\alpha = \alpha/\mu$ for the derivation based on the expression (1). According to the value n (number of failed servers) may happen, two segments are defined by the vector: [Segment 1, Segment 2] = [1 ≤ n ≤ S, (S+1) ≤ n ≤ N]. Based on the derivation process described in the Appendix, the state probability functions P(n) can be derived in terms of two segments as follows:

Segment 1: $1 \le n \le S$

$$P(n) = \frac{\lambda_0 \cdot \lambda_1 \cdot \lambda_2 \cdots \lambda_{n-1}}{\mu_1 \cdot \mu_2 \cdot \mu_3 \cdots \mu_n} P(0)$$

$$= \prod_{j=0}^{n-1} [M\theta_\lambda + (S-j)\theta_\alpha] P(0) \quad (2)$$

Segment 2: $(S+1) \le n \le N$, (N=M+S)

$$P(n) = \frac{\lambda_0 \cdot \lambda_1 \cdot \lambda_2 \cdots \lambda_{n-1}}{\mu_1 \cdot \mu_2 \cdot \mu_3 \cdots \mu_n} P(0)$$

$$= \frac{M! \cdot \prod_{j=0}^{S-1} [M\theta_\lambda + (S-j)\theta_\alpha] \cdot \theta_\lambda^{n-S}}{(N-n)!} P(0) \quad (3)$$

Equations (2) and (3) are the closed-forms for the state probability functions P(n) in terms of two segments in which the number of failed servers may happen. To obtain P(0), we substitute (2) and (3) in normalizing equation: $\sum_{n=0}^{N} P(n) = 1$, and general solutions for P(n) can be obtained for all $n \in N$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2016.2514506, IEEE Transactions on Services Computing

4.2 System Characteristics

Mathematical expectations are crucial for the long-run theoretical average values of relevant parameters in the system. To formulate the expressions regarding system performance metrics, it is necessary to construct mean-related functions such as expected number of failed/operating servers, expected number of spares in the system. Machine availability (MA) refers to the probability that a system is found to be in the running state at any point in time [20]. In cloud environment, this availability metric can be also regarded as a measure of performance committed to the cloud customers, and of a yardstick for quantitatively comparing the effectiveness of the fault-tolerance methods. We define the following system characteristics for the proposed model. Let

E[O] = the expected number of online servers in the system,
E[S] = the expected number of standby servers in the system,
E[B] = the expected fraction time of busy repairmen in the system,
E[I] = the expected fraction time of idle repairmen in the system,
Mn = the number of operating servers with n failed servers,
Sn = the number of spare servers with n failed servers,
MA = the machine availability of the system

Let n be the number of failed servers, and the range of n for two segments the values of [Mn, Sn] can be given for two cases as follows:
Segment 1: $1 \le n < S$, [Mn, Sn] = [M, (S-n)]
Segment 2: $S \le n < N$, [Mn, Sn] = [(N-n), 0]

With the data above, relevant expected values of decisive parameters including E[O], E[S], E[B] and E[I] can derived mathematically and expressed as equations (4), (5), (6) and (7) respectively.

$$\begin{aligned} E(O) &= \sum_{n=0}^{N} M_n P(n) \\ &= \sum_{n=0}^{S} M_n P(n) + \sum_{n=S+1}^{N} M_n P(n) \\ &= M \times \sum_{n=0}^{S} P(n) + \sum_{n=S+1}^{N} (N-n) P(n) \\ &= M \times \sum_{n=0}^{S} P(n) + M \times \sum_{n=S+1}^{N} P(n) + \sum_{n=S+1}^{N} (S-n) P(n) \\ &= M - \sum_{n=S+1}^{N} (n-S) P(n) \quad (4) \end{aligned}$$

$$\begin{aligned} E(S) &= \sum_{n=0}^{N} S_n P(n) = \sum_{n=0}^{S} S_n P(n) + \sum_{n=S+1}^{N} S_n P(n) \\ &= \sum_{n=0}^{S} S_n P(n) \\ &= \sum_{n=0}^{S} (S-n) P(n) \quad (5) \end{aligned}$$

$$E[I] = P(0)] \quad (6)$$

$$E[B] = 1 - E[I] = 1 - P(0) \quad (7)$$

$$MA = \frac{N - E[N]}{N} = 1 - \sum_{n=1}^{N} \frac{n \times P(n)}{N} \quad (8)$$

Based on the proposed model and equations, performance metrics are developed. The metrics include multifarious expected length of system parameters, and their relationships with the system. These performance metrics are needed for built up the evaluation function like the expected profit function for the proposed system.

4.3 An Illustrative Example on Mathematical Details

To perceive state probabilities and system metrics, a simple example is given to show the research issue of this article. For illustrative purpose, it is assumed that there are 5 online servers in online subsystem and 2 standby servers in standby subsystem respectively. There is a repair engineer with repair equipments for fixing the failed server in the repair subsystem. The repair time for the repair engineer is exponentially distributed with a mean of 2 days. When an online server is fixed, the time until the next breakdown is exponentially distributed with an average of 10 days. The failure rate of spares (α) in the standby subsystem is assumed to be $\lambda/2$ for warm-standby need.

In the mathematical language of M/M/1 MRP with spares, relevant parameters are given as follow:

M = 5, S = 2, λ = 1/10 (failure rate), μ = 1/2 (repair rate), α = 1/20, we have $\theta_\lambda = \lambda/\mu = 1/5$ and $\theta\alpha = \alpha/\mu = 1/10$. Based on expression (1), it yields

(i) $\lambda_n = 5\lambda + (2 - n)\alpha$, when n = 0, 1, 2
(ii) $\lambda_n = (7 - n)\lambda$, when n = 3, 4, 5, 6, 7

The state probability functions P(n) can be computed according to expressions (2) and (3) separately as follows:

Expression (2): $1 \le n \le 2$,

$$P(n) = \prod_{j=0}^{n-1} [Mq_l + (S-j)q_a] P(0) = P(0) \times \prod_{j=0}^{n-1} \left(\frac{12-j}{10}\right)$$

$$P(1) = \left(\frac{12}{10}\right) \times P(0) = 1.2\, P(0)\ ;$$

$$P(2) = \prod_{j=0}^{n-1} \left(\frac{12-j}{10}\right) \times P(0) = \left(\frac{132}{100}\right) \times P(0) = 1.32\, P(0)$$

Expression (3): $3 \le n \le 7$,

$$P(n) = \frac{M! \times \prod_{j=0}^{S-1} [Mq_l + (S-j)q_a] \times q_l^{n-S}}{(N-n)!} P(0)$$

$$= \frac{5! \times \left(\frac{1}{5}\right)^{n-2} \times \prod_{j=0}^{1} \left(\frac{12-j}{10}\right)}{(7-n)!} \times P(0)$$

$$= \frac{5! \times \left(\frac{1}{5}\right)^{n-2} \times (1.32)}{(7-n)!} \times P(0)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2016.2514506, IEEE Transactions on Services Computing

$$P(3) = \frac{5!\times(\frac{1}{5})\times(1.32)}{4!}\times P(0) = 1.32\ P(0)\ ;$$

$$P(4) = \frac{5!\times(\frac{1}{5})^2\times(1.32)}{3!}\times P(0) = 1.06\ P(0)$$

The values for five state probability functions in Segment 2 can be computed in a similar way. The whole seven state probability values in Segments 1 and 2 are collected as follows:

[P(1), P(2), P(3), P(4), P(5), P(6), P(7)] =
[1.2, 1.32, 1.32, 1.06, 0.63, 0.25, 0.05] ×P(0).

To obtain P(0), by the normalizing equation: $\sum_{n=0}^{N} P(n) = 1 \Rightarrow$

$$\sum_{n=0}^{7} P(n) = (7.56)\ P(0) = 1 \ \therefore P(0) = 0.146\ \#$$

From equations (4) ~ (8), system characteristics are calculated as follows:

$$E[O] = M - \sum_{n=S+1}^{N} (n - S)P(n)$$

$$= 5 - \sum_{n=3}^{7} (n - 2)P(n)$$

$$=5 - [P(3)+2P(4)+3P(5)+4P(6)+5P(7)]$$

$$= 5 - (1.32+2.12+1.89+1.0+0.25)$$

$$P(0) = 5 - (6.58\times 0.146) = 4.04$$

$$E[S] = \sum_{n=0}^{S} (S - n)P(n)$$

$$= \sum_{n=0}^{2} (2 - n)P(n)$$

$$= 2P(0) + P(1) = (3.2)\times 0.146 = 0.467$$

$$E[I] = P(0) = 0.146;$$

$$E[B] = 1 - E[I] = 0.854$$

$$\text{Machine Availability (MA)} = \frac{N - E[N]}{N} = 1 - \frac{\sum_{n=1}^{N} nP(n)}{N} =$$

$$1 - \frac{17.04}{7}\times P(0) = 0.64$$

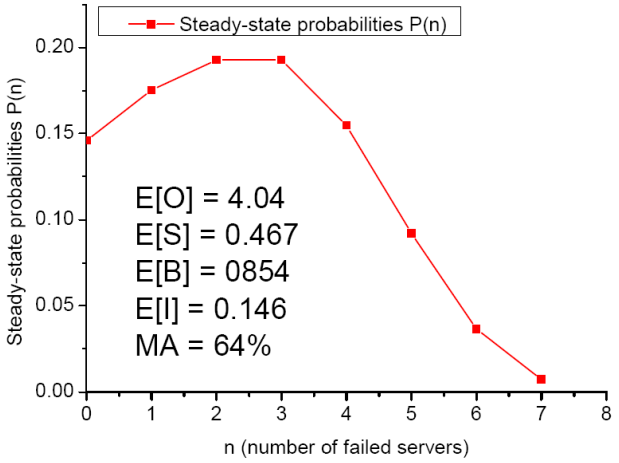

Fig. 6. Steady-state probabilities patterns with parameters [N, M, S] = [7,5,2].

The distribution of steady-state probabilities is depicted in Fig. 6. The relevant system performance measures, like E[O], E[S], and E[B], etc., are shown on the left-bottom side in Fig. 6 for gaining the whole picture concisely as well. On the average, the numbers of operating servers in online subsystem, the number of spares in standby subsystem and the MA metric are 4.04 (rating: 5), 0.467 (rating: 2) and 64% respectively.

5 Performance Evaluation and Design Support

5.1 Evaluation Formulation for Profit

Like all business, the profit pattern of a cloud platform is based on two components, namely, the income and the cost. The profit (also called the net business gain) is the income minus the cost. To optimize the profit, we develop a steady-state expected profit function per unit time, in which S is the decision parameter. It is reasonably assumed that part of repair facilities, like air conditioning, equipment and factory power, would be shutdown or closed if there is no failed server in it. In other words, the operating cost for repair facility in idle state would be much lower than that in busy state. Hence, let the parameters C_B and C_I be the cost per unit times of the repairman in busy state and idle state respectively. Also, let the parameter C_R be the revenue per unit time of one server in operation. To formulate the profit function, some cost/revenue parameters are defined in the following vector form as follows:

C_R = revenue per unit time of one online server in operation state.

$[C_O, C_S]$ = cost per unit time of one server in [online, standby] subsystem.

$[C_B, C_I]$ = cost per unit time when the repairman is [busy, idle] in repair subsystem,

Using the definitions of each cost/revenue element with its corresponding feature, the net profit function $P_F(S)$ can be developed in association with system metrics: E[O], E[S], E[I] and E[B] of which are given in equations (4), (5), (6) and (7) respectively. It is noted that the steady-state probabilities for two segments are given in expressions (2) and (3).

$$P_F(S) = (C_R - C_O)\ E[O] - C_S\ E[S] - C_I\ E[I] - C_B\ E[B]$$

$$= (C_R - C_O)\ [\ M - \sum_{n=S+1}^{N} (n - S)P(n)\ ] -$$

$$C_S \sum_{n=0}^{S} (S - n)P(n) - C_I\ P(0) - C_B\ [1 - P(0)] \quad (9)$$

The state probability functions P(n) for two segments are given in expressions (2) and (3), which are quite complex for the control parameter S. Examining equations (2) and (3), it is found that the parameter S occurs not only at the location of in-line but also of superscript of production symbol Π. Also the decision variable S occurs in the subscript and superscript of equation (9), which makes $P_F(S)$ a highly nonlinear and complex function. Instead, some numerical examples arepresented and intensively studied by applying the proposedmodels.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2016.2514506, IEEE Transactions on Services Computing

5.2 Profit Optimization with Various Warm-standbys

There are three types of standby servers configured in terms of failure rates and energy budget states. A standby server is called "hot-standby" if its failure rate and the energy budget are the same as those of an online server respectively. A standby server is called "warm-standby" when the failure rate is non-zero and less than that of an online server. A "cold-standby" is named if the spare is powered off and its failure rate is zero. On the failure rate and power cost, the spare in hot-standby state and in cold-standby incur the highest and the lowest metrics respectively while the running cost of warm-standby server lies between them.

Three types of standby subsystem are taken into consideration in the evaluation process for the proposed model. This is designed for administrator to gain the balance between management economics and response service quality. For example, keeping the spares in the state of hot-standby would undoubtedly respond the sudden breakdowns of any server in operation with minimum latency for their users, but however the management cost (including electric power cost and failure rate) would be much higher than other types of spares.

For providing decision support on making provision for cloud logistics, we study the effect of varying the kernel parameter (S) while keeping others constant. Three levels of warm-standby configuration are explored in terms of failure rate of spares (α) = $\lambda/2$, $\lambda/3$ and $\lambda/4$, which the parameter (λ) is the failure rate of online server. All simulations are performed with MATLAB tool package with customized MATLAB codes. System parameters for the amount of online servers and spares are assumed to be [M, S] = [100, 20]. Let the system's parameters be configured in terms of vector forms as follows:

(a) Averagefailure rate of an online server (λ) = 0.1,
(b) Average repair rate of a repairman (μ) = 10,
(c) Cost/revenue elements: [C_R, C_O, C_S, C_B, C_I] = [1.5, 0.3, 0.1, 100, 50] of which is charged by a specified time unit.

Since contour plots may provide the graphical representation of the optimization problem, and also may possess a powerful visualization that permits the solutions of the optimization problem by inspection. To validate the analytical materials, we have the graphical results composing three contours as shown in Fig. 7. Each data point in each contour is the profit value for its corresponding number of standby servers. The curve marked as $\alpha = \lambda/2$ (black box line) represents the profit patterns for the lowest warm-standby level among three configurations, whose the optimal value of profit approaches 16.51while if the amount of spares is set to be at S* = 15. For other two contours marked as $\alpha = \lambda/3$ (red circle line)and $\alpha = \lambda/2$ (blue triangle line),their corresponding optimal profit values reach 16.35 and 16.104 at S* = 13 an S* = 11, respectively.

As the level of warm-standby is decreasedlittle by little, the exemplified profit value goes higher gradually as revealed in Fig. 7. Logically, the failure rate and power cost would be incurred worse as the level of warm-standby servers in the standby subsystem has been tuned higher, and then it would deteriorate the value of optimal profit accordingly. In Fig. 7, such an improvement profile on profit metric by an amount of about twenty percentage of profit without logistic policy reveals the feasibility and effectiveness of the proposed LSA model.

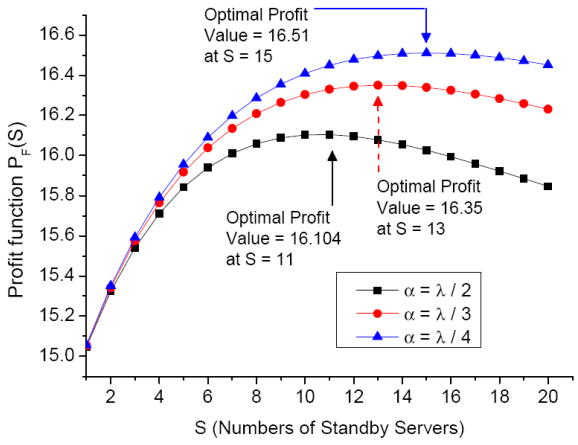

Fig. 7. Optimal profit patterns in three levels of warm-standby configuration

5.3 Sensitivity Analysis on Failure Rate

Cost-down approach in business would inevitably incur the risk on the quality-down on the horns of a dilemma more or less. The adoption of budget-saving servers with declining failure rate (λ) would drives the cost-down apparently. To show the varying tendency on profit on kernel parameter (λ), only the failure rate of server is varied each time while keeping others constant. Other system parameters are the same as those in Subsection 5.2 with warm-standby level $\alpha = \lambda/2$.

Observing on variation profile for curves in Fig. 8, it appears that the smaller the failure rate (λ) is tuned, the larger the profit $P_F(S)$ would go. Logically, tuning the failure rate (λ) smaller would elongate the lifetime of servers, and make parameter E[O] larger, which is a positive term in profit function $P_F(S)$ of expression (9) with $P_F(S) = (C_R - C_O)\,E[O] - C_S\,E[S] - C_I\,E[I] - C_B\,E[B]$. Meanwhile, the repairman loading in repair subsystem would be alleviated due to higher quality servers and then the parameter E[B] becomes smaller, which a negative term in (9), which can have positive contribution on $P_F(S)$. Although other two parameters E[S] and E[I] become larger with having negative impact on $P_F(S)$, their cost rates C_S and C_I, are much smaller than those of their counterparts E[O] and E[B] with $(C_R - C_O)$ and C_B. Taken together, the profit function $P_F(S)$ goes larger as the failure rate (λ) is tuned to be smaller, which embodies the variation profile in Fig. 8. The detailed simulated data constructing Fig. 8 are listed in Table 1 with data values of even-numbered spares (S).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2016.2514506, IEEE Transactions on Services Computing

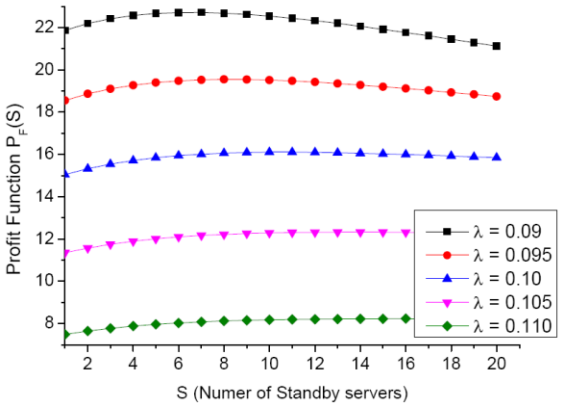

Fig. 8. Impact of tuning failure rate on profit profile

Table 1. Data values showing impact of tuning failue rate on profit

S	λ = 0.090	λ = 0.095	λ = 0.10	λ = 0.105	λ = 0.110
2	22.18967	18.86465	15.32343	11.56854	7.647206
4	22.57267	19.27376	15.71133	11.89111	7.879463
6	22.70744	19.47851	15.9399	12.09504	8.02809
8	22.67785	19.54409	16.0589	12.21765	8.120393
10	22.53772	19.51409	16.10258	12.28495	8.17526
12	22.32291	19.4188	16.09529	12.31541	8.205803
14	22.05811	19.27993	16.05469	12.32221	8.221062
16	21.76085	19.11344	15.99375	12.31473	8.227163
18	21.44391	18.93132	15.92206	12.29957	8.228151
20	21.11688	18.7427	15.84662	12.28118	8.22659

5.4 Issues on Profit with Constraints on Availability

In the cloud business, the profits gained by cloud enterprises and the performance given for cloud customers are the both ends of a seesaw. In cloud environment, the machine availability (MA) can be also regarded as a measure of performance committed to the cloud customers, and of a yardstick for comparing the effectiveness of the fault-tolerance methods in a quantitative manner. Practically, it is reasonable for cloud providers to guarantee target level on the system availability when they want attract potential cloud customers. How to approach the balance point between the profits and the performance metrics on the cloud logistics? The proposed system would like to explore the issue on decision support for optimal profit under the constraint of availability metric with some target level.

In Fig. 9 with the double-Y axis, the left Y-axis and the right Y-axis are set to be the profit values and the machine availability (MA) respectively. On the right Y-axis, the dash-curve marked with blue star represents availability patterns in variation with the numbers of spares (S). For the target level (for example, taking MA=90%) as a commitment to cloud customers, the simulation results in Fig. 9 provide an exemplified decision reference on deploying the amount of standby servers in standby subsystem in the proposed framework. Observing the solid-line contour marked with black boxes (i. e., the leftY-axis), the optimal profit value P_F = 16.104 occurs at S*= 11. However, the corresponding availability metric (MA =88.34 %) is smaller than the target level (MA = 90%).

How many standby servers are needed to meet the constraint on availability metric MA = 90%? Observing the blue-star contour with the right Y-axis in Fig. 9, it is found that the availability metric can be achieved by MA = 90.34% at S = 6 (pointed by the left-side upward arrow) at the negative impact of profit values $P_F(S=6)$ = 15.94 lower than $P_F(S=11)$ = 16.104. The detailed simulated data constructing Fig. 9 are listed in Table 2. From this exemplified data analysis, the proposed LSA can provide a quantitative decision support on the tradeoff study between the availability metric and the amount of spares in the standby subsystem.

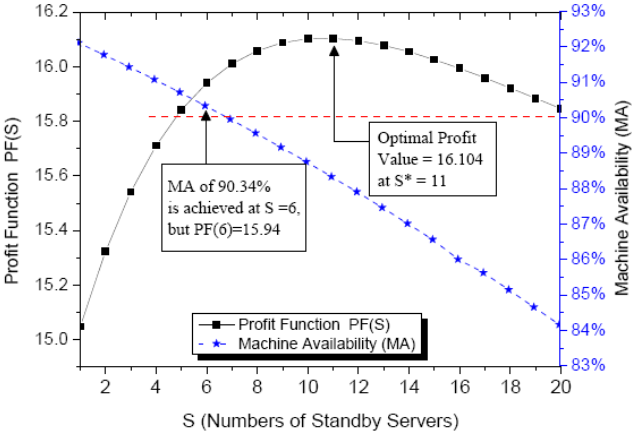

Fig. 9. Issues on Profit with Constraints on Machine Availability

Table 2 Detailed data on profit values and the corresponding MAs

S	Profit Values (P_F)	Machine Availability (MA)	S	Profit Values (P_F)	Machine Availability (MA)
1	15.04764	0.9211	11	16.10408	0.8834
2	15.32343	0.9178	12	16.09529	0.8791
3	15.54124	0.9143	13	16.07825	0.8747
4	15.71133	0.9108	14	16.05469	0.8702
5	15.84197	0.9072	15	16.02609	0.8657
6	15.9399	0.9034	16	15.99375	0.8601
7	16.01067	0.8996	17	15.95876	0.8563
8	16.0589	0.8957	18	15.92206	0.8515
9	16.08845	0.8917	19	15.88446	0.8466
10	16.10258	0.8876	20	15.84662	0.8417

6 Conclusions

The competition among enterprises may drive prices downward, but at what cost and profits? Logistic economy of cloud server farm emerges to be an important aspect of cloud computing which is of crucial interest for cloud providers to gain the profit of running the business. Since the inevitable breakdown of servers in operation would definitely incur the reduction of profit for cloud providers, proper backup policy should be adopted for alleviating the penalty. On the proposed logistic system, the decision on the amount of standby servers has been studied comprehensively using the queuing model. The goal of this research is to provide an effective decision

support for cloud logistics to get around the haphazard selection on spares quantities. To this aim, an optimization framework for decision support has been proposed using finite-source queuing theory.

From the qualitative aspect, an operational framework has been designed and visualized with Petri Net design. An analytical model in closed-form for the steady-state probability patterns has also been established in terms of both system measures and the expression for the profit evaluation of the cloud platform. Quantitatively, the proposed LSA approach are used for independent parameters for exploring the optimal profit under kernel system parameters like average numbers of online/standby servers, failure rate of servers, and repair rate, etc. We have further conducted simulation experiments to validate our model on the issues regarding the levels of standbys, failure rate of servers and availability metric. Numerical and simulated results showed that the optimal profit value can be approached for the decision support on deploying the amount of standby servers in standby subsystem. Such an optimizing profile reveals that the proposed LSA would be a feasible and cost-effective approach in cloud logistics.

ACKNOWLEDGMENT

The work of this paper was partially supported by the Ministry of Science and Technology, Taiwan, R.O.C. under Grant no. MOST 103-2218-E-007-021 and MOST 102-2221-E-216 -008 -MY3.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D.Patterson, A. Rabkin, I. Stoica, M. Zaharia, A view of cloud computing, Communications of ACM. 53 (4) (2010) 50–58.

[2] L. Moser, B. Thuraisingham and J. Zhang, "Services in the Cloud," *IEEE Trans. Services Computing*, vol. 8, no. 2, pp. 172–174, March/April 2015.

[3] R. Buyya R, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility," Future Generation Computer Systems, vol. 25, no. 6, pp.599–616, 2009.

[4] K.-W. Park, J. Han, J. Chung and K. H. Park, "THEMIS: A Mutually Verifiable Billing Systemfor theCloud Computing Environment," *IEEE Trans. Services Computing*, vol. 6, no. 3, pp. 300–313, July-September 2013.

[5] S.Caton, C. Haas, K. Chard, K.Bubendorfer and O. F. Rana, "A Social Compute Cloud: Allocating and Sharing Infrastructure Resources via Social Networks," *IEEE Trans. Services Computing*, Vol. 7, no. 3, pp. 359–372, July-September 2014.

[6] R.B. Cooper, Introduction to Queuing Theory, 2nd edition. 1981. Elsevier Science Publishing Co., Inc. 52 Vanderbilt Avenue, New York 10017.

[7] D. Gross, J.F. Shortle, J.M. Thompson, and C. Harris, Fundamentals of Queuing Theory, 4th edition, 2008, A John Wiley &Sons, Inc., New York.

[8] P. Patel, A. Ranabahu, and A. Sheth, Service level agreement in cloud computing. In Proceeding of the 24th ACM SIGPLAN Conference Companion on Object Oriented Programming System Languages and Applications (OOPSLA'09).

[9] K. Xiong K andH. Perros, Service performance and analysis in cloud computing. In Proceedings of IEEE World Conference Services, 2009, pp 693–700

[10] Y.-K. Lin and P.-C. Chang, "Maintenance reliability estimation for a cloud computing networkwith nodes failure," Expert Systems with Applications, 38 (2011), 14185–14189.

[11] B. Hu, Y. Sudo, K. Hato, Y. Murata and J. Murayama, "Cost reduction evaluation of sharing backup servers in intercloud," 19th Asia-Pacific Conference on Communications (APCC), Bali, Indonesia, 2013, pp.256–261.

[12] L. Chen, X. Zheng and G. Chen, "A system architecture for intelligent logistics system," 2013 *International Conference on Cloud Computing and Big Data*, 2013, pp. 426–431.

[13] S. Li, Y. Zhou, L. Jiao, X. Yan, X. Wang and M. R.-T.Lyu, "Towards Operational Cost Minimization in Hybrid Clouds for Dynamic Resource Provisioning with Delay-Aware Optimization," *IEEE Trans. Services Computing*, vol. 8, no. 3, pp. 398-409, May/June 2015.

[14] Y.C. Lee, C.Wang, A.Y. Zomaya, and B.B. Zhou, "Profit-driven scheduling for cloud services with data access awareness," *Journal of Parallel and Distributed Computing*, vol. 72, pp. 591–601, 2012.

[15] H. Khazaei, J. Misic, and V.B. Misic, "Performance analysis of cloud computing centers using M/G/m/m+r queuing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol.23, no. 5, pp. 936–943, 2012.

[16] T. Murata, Petri Nets: Properties, Analysis and Applications. *Proceedings of The IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

[17] J.-S. Lee, and P.-L. Hsu, Implementation of a remote hierarchical supervision system using Petri Nets and agent technology, *IEEE Transactions of Systems, MAN, and Cybernetics* – Part C: Applications and Reviews. vol. 37, no. 1, pp. 77–85, 2007.

[18] J.-S. Lee, M.-C. Zhou and P.-L. Hsu, "An Application of Petri Nets to Supervisory Control for Human-Computer Interactive Systems," *IEEE Transactions on Industrial Electronics*, vol. 52, no. 5, pp.1220-1226, 2005.

[19] J.-S. Lee, "A Petri Net of Command Filters for Semiautonomous Mobile Sensor Networks," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 4, pp. 1835–1841, 2008.

[20] C.-W. Ang, and C.-K. Tham, "Analysis and optimization of service availability in an HA cluster with load-dependent machine availability," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 9, pp. 1307–1319, 2007.

[21] Saif U. R. Malik, Samee U. Khan, and Sudarshan K. Srinivasan, "Modeling and analysis of state-of-the-art VM-based cloud management platforms", *IEEE Transactions on Cloud Computing*, Vol.1, No.1, pp. 1, 2013.

[22] Rahul Ghosh, Francesco Longo, Flavio Frattini, Stefano Russo, and Kishor S. Trivedi, "Scalable analytics for IaaS cloud availability", *IEEE Transactions on Cloud Computing*, Vol. 2, No. 1, pp. 57-70, 2014.

[23] Liuhua Chen, Haiying Shen, K. Sapra, "RIAL: Resource intensity aware load balancing in clouds", Proceedings of *IEEE INFOCOM*, pp. 1294–1302, 2014.

[24] Ching-Hsien Hsu, Kenn Slagter, Shih-Chang Chen and Yeh-Ching Chung, "Optimizing Energy Consumption with Task

Consolidation in Cloud", *Information Science*, Vol. 258, pp. 452-462, Feb. 2014.

[25] Yi-Ju Chiang, Yen-Chieh Ouyang and Ching-Hsien Hsu, "An Efficient Green Control Algorithm in Cloud Computing for Cost Optimization", *IEEE Transactions on Cloud Computing*, Vol. 3, No. 2, pp. 145-155, 2015.

[26] Yi-Ju Chiang, Yen-Chieh Ouyang and Ching-Hsien Hsu, "An Optimal Control Policy to Realize Green Cloud Systems Based on Petri Nets", *Journal of Supercomputing*, Vol. 69, No. 3, 1284-1310, 2014.

Fuu-Cheng Jiang Dr. Fuu-Cheng Jiang worked as a design engineer with the Aeronautical Research Lab., Chung Shan Institute of Science and Technology (CSIST) when he was assigned to a partnership project at General Dynamic, Fort Worth, Texas. Currently, he is a member of faculty in the department of computer science at Tunghai University in Taiwan. Dr. Jiang was the recipient of the Best Paper Award at the 5th International Conference on Future Information Technology 2010 (FutureTech 2010), which ranked his paper first among the 201 submittals. He has served dozens of the TPC for worldwide international conferences like BWCCA 2010, ICCCT 2011-2012, IEEE CloudCom 2012, IEEE BIOCAS-2013, NPC 2014, SC2 2014, IEEE SPICES 2015, SC2 2015, IoT 2015, CCBD 2015and also the Session Chair of CSE2011 and IEEE ICCE-Taiwan 2014, publication Chair of NPC 2014. Moreover, he served as journal reviewer of the Computer Journal, Ad Hoc Networks, Journal of Network and Computer Applications (JNCA), Journal of Supercomputing (JOS), Journal of Internet Technology (JIT), International Journal of Communication Systems (IJCS), and IEEE Transactions on Cloud Computing. Dr. Jiang has served as a member of an Editorial Board Member on CIP-JWCMCN Journal and an assistant editor on the Cloud-Link Editorial Team of IEEE society. His research interests include network modeling,cloud computing, wireless sensor networks and simulation. Dr. Jiang is a member of IEEE society.

Ching-Hsien (Robert) Hsu is a professor in department of computer science and information engineering at Chung Hua University, Taiwan; and distinguished chair professor in school of computer and communication engineering at Tianjin University of Technology, China. His research includes high performance computing, cloud computing, parallel and distributed systems, big data analytics, ubiquitous/pervasive computing and intelligence. He has published 200 papers in refereed journals, conference proceedings and book chapters in these areas. Dr. Hsu is the editor-in-chief of international journal of Grid and High Performance Computing, and international journal of Big Data Intelligence; and serving as editorial board for a number of prestigious journals, including *IEEETransactions on Service Computing*, *IEEE Transactions on Cloud Computing*, etc. He has been acting as an author/co-author or an editor/co-editor of 10 books from Springer, IGI Global, World Scientific and McGraw-Hill. He has also edited a number of special issues at top journals, such as *IEEE Transactions on Cloud Computing*, *IEEE Transactions on Services Computing*, *IEEE System Journal*, *Future Generation Computer Systems*, *Journal of Supercomputing*, etc. Prof. Hsu was awarded eight times distinguished award for excellence in research and annual outstanding research award through 2005 to 2015 from Chung Hua University. He has been serving as executive committee of Taiwan Association of Cloud Computing (TACC) from 2008-2012; executive committee of the IEEE Technical Committee of Scalable Computing (2008-2012); IEEE Cloud Computing (2012-present); Dr. Hsu is an IEEE senior member;

Shangguang Wang is associate professor at the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. He received his Ph.D. degree at BUPT in 2011. His PhD thesis was awarded as outstanding doctoral dissertation by BUPT in 2012. Dr. Wang is 2015-2016 President of the Service Society Young Scientist Forum in China, General Co-Chair of ICCSA 2016, Application Track Co-Chair of IEEE SCC 2015, and Program Chair of the 2014 International Conference on Internet of Vehicles (IOV), Program Chair of the 2014 International Symposium on Cloud and Service Computing (SC2), and Special Track Chair of APSCC 2014. His research interests include Service Computing, Cloud Computing, and QoS Management.