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/TCC.2017.2691323, IEEE Transactions on Cloud Computing

Correlation Modeling and Resource Optimization for Cloud Service With Fault Recovery

Xiwei Qiu, Yuanshun Dai, *Member, IEEE*, Yanping Xiang, and Liudong Xing, *Senior Member, IEEE*

Abstract—Energy-efficient cloud computing has recently attracted much attention, where not only performance but also energy consumption are important metrics to be considered for designing rational resource scheduling strategies. Most of existing approaches for achieving energy efficient computing focus on connecting these two metrics and balancing the tradeoff between them, which however is inadequate because another important factor reliability is not considered. In fact, both virtual machine (VM) failures and server failures inevitably interrupt execution of a cloud service, and eventually result in spending more time and consuming more energy on completing the cloud service. Therefore, reliability significantly affects service performance and energy consumption, and thus they should not be handled separately. Connecting these correlated metrics is essential for making more precise evaluation and further for developing rational cloud resource scheduling strategies. In this paper, we present a correlated modeling approach applying Semi-Markov models, the Laplace-Stieltjes transform (LST), a Bayesian approach to analyze reliability-performance (R-P) and reliability-energy (R-E) correlations for cloud services using a retrying fault recovery mechanism. A recursive method is also proposed for modeling the correlations for cloud services using a check-pointing fault recovery mechanism. The proposed correlation models can be used to calculate the expected service time and energy consumption for completing a cloud service. Moreover, the models can contribute to analyzing the expected performance-energy tradeoff. We formulate the expected performance-energy optimization problem by describing performance and energy consumption metrics as functions of assigned CPU frequencies. Finally, we use a derivation approach to determine Pareto optimal solutions for the formulated optimization problem. Illustrative examples are provided.

Index Terms—Cloud service, correlation modeling, fault recovery, virtual machine failure, server failure, resource optimization.

1 Introduction

CLOUDcomputing is a newly developed technology with numerous novel characteristics, such as large-scale resource sharing, on-demand resource provisioning and safe isolation of co-located workloads [1]. Virtualization, a core technology of cloud computing, enables flexible resource management for various cloud services. The use of virtualization supports cloud providers in developing rational resource scheduling strategies (e.g., flexible CPU resource assignment by using a frequency scaling technology) to reduce power consumption of a physical server. However, designing cloud resource scheduling strategies only from energy perspective is inadequate for real-world scenarios. In fact, a rational and efficient cloud resource scheduling strategy usually needs to optimize multiple correlated metrics, particularly reliability, performance and energy consumption [2]. Thus, a systemic model for evaluating these important metrics is fundamental and essential for reliable and energy-efficient design and operation of a cloud system.

In general, there exists an important tradeoff between performance and energy consumption metrics. For example, energy consumption for successfully completing a cloud service is decided by power consumption of the host server and completion time of the service. However, the CPU frequency assigned to the cloud service has inverse effects on power consumption and completion time indices, and thus forms a complicated tradeoff relationship. Many studies have proposed various approaches for quantifying and balancing this complicated performance-energy (P-E) tradeoff [3]-[5], but none of them captured random changes of performance and energy consumption caused by failures and recovery, i.e., the reliability factor.

In fact, performance, energy consumption and reliability are mutually correlated factors in realistic scenarios of cloud computing. For example, if the execution of a cloud service is interrupted by a random failure, the cloud service may be resumed, which inevitably results in spending more time and consuming more energy consumption for completing the cloud service. Meanwhile, subsequent repair actions for removing the occurred failure also lead to additional time and energy consumption. Therefore, reliability, performance and energy consumption metrics do affect one another and should not be considered separately.

To analyze the reliability-performance-energy (R-P-E) correlation in detail, theoretical modeling is a feasible and

- *X. Qiu, Y. Dai, and Y. Xiang are with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China. E-mail: qiu_uestc@hotmail.com;dai@cloudian.org yanping_xiang@163.com.*
- *L. Xing is with the Department of Electrical and Computer Engineering, University of Massachusetts, Dartmouth, MA 02747, USA. E-mail: lxing@umassd.edu.*

*Please note that all acknowledgments should be placed at the end of the paper, before the bibliography(**note that corresponding authorship is not noted in affiliation box, but in acknowledgment section**).*

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/TCC.2017.2691323, IEEE Transactions on Cloud Computing

efficient approach that can further contribute to developing comprehensive resource scheduling strategies. Many recent research has focused on correlation modeling and corresponding resource optimization considering only two factors, for example, performability analysis capturing R-P correlation [6], [7], and energy efficient scheduling considering P-E tradeoff [8], [9]. However, R-P-E correlation models have not attracted much attention, especially, R-E correlation. Performance and energy consumption metrics are indeed random variables affected by reliability. These critical connections cannot be ignored for achieving precise evaluation of the metrics. Although our prior research has studied a R-P-E correlation model for a cloud system [10], it does not consider specific fault recovery mechanisms, the work requirement of the cloud service, and the inverse effect on performance and energy metrics caused by assigning different CPU frequency to the cloud service.

To remedy this lack, we systemically study R-P-E correlation models and corresponding resource optimization technologies for cloud services using retrying and check-pointing fault recovery mechanisms. The primary innovation of the proposed model is that it establishes essential connection among system reliability, performance and energy consumption for a cloud service with a specific work requirement. We first model cloud service process with retrying and check-pointing recovery mechanisms by using Semi-Markov processes, which takes VM faults, server failures, and corresponding recovery actions into account. Then, the LST, a Bayesian approach, and a recursive method are used to evaluate the expected execution time and expected energy consumption of the cloud service. Finally, the calculation of these two important metrics is described as one-variable functions whose inputs are scaled CPU frequencies representing amount of computational resource assigned to cloud services. The optimal CPU frequencies minimizing expected service time and expected energy consumption are obtained using derivation approaches. Moreover, our models also contribute to finding Pareto optimal solutions for expected performance-energy combined optimization problems.

The remainder of the paper is organized as follows. Section 2 introduces the existing correlations among reliability, performance and energy consumption. Section 3 presents a correlation modeling approach which systemically applies Semi-Markov models, the LST and a Bayesian approach to build a R-P-E correlation model for a cloud service with the retrying recovery mechanism. Section 4 presents another R-P-E correlation model for a cloud service with the check-pointing recovery mechanism. Section 5 describes expected performance-energy combined optimization problems based on the proposed correlation models, and a derivation approach is used to analyze optimal solutions. Numerical examples and results are illustrated in Section 6. Section 7 describes some related researches followed by highlights of new contributions made by this paper. Section 8 concludes this paper.

2 Analysis of Correlations

2.1 Description of Correlations

In principle, a cloud system typically has a critical and centralized node to facilitate the uniform implementation of certification, authorization, and resource monitoring and assignment capabilities. After the centralized node receives a user's request of creating a VM, it first needs to find a physical serve that can host the VM and make a decision on how much resource capacity should be assigned to the VM. Then, the VM is created or instantiated at the selected serve, and the corresponding cloud service running in the created VM can be finally delivered to the user. For making a comprehensive and efficient decision that assigns optimal resource capacity to the created VM, complicated correlations among reliability, performance and energy must be taken into account using the model suggested in this work.

The execution of a cloud service mainly relates to three logical layers. The resource layer containing servers and VMs decides reliability of the cloud service. The application layer concerns performance of the cloud service, and the management layer usually focuses on controlling energy consumption. The interactions between different layers identify the existing correlations, which are described as follows.

1) R-P Correlation: For a cloud service, VM failures and server failures inevitable interrupt the execution of a cloud service. Thus, performance is a random variable affected by the reliability factor, which implies that performance is a resulting attribute of reliability.
2) R-E Correlation: Any failure also results in spending more time for completing the cloud service and addition time for recovering the failure, which also means more energy consumption. Thus, energy consumption is also a resulting attribute of reliability.
3) P-E Correlation: The amount of resource (i.e., the CPU frequency) assigned to the cloud service has inverse effects on completion time and server power. This implies that the P-E correlation is essentially a tradeoff relationship.

2.2 Critical Factors Affecting Correlations

Computing intensive tasks, such as bio-computing, image processing, 3D rendering, and complicated scientific computation, constitute an important category of cloud services, i.e., a cloud service with a specific work requirement. This kind of cloud services is usually time consuming and has a specific computation complexities which can be quantified by the number of instructions to be executed, for example, parameterized algorithms for some large-scale NP-complete problems [11]. For successfully executing a cloud service, different kinds of fault recovery mechanisms can be used, such as retrying (the failed task is retried on the same or another resource) and check-pointing (the failed task resumes processing from the last saved checkpoint instead of from the beginning) [12].

Besides fault recovery mechanisms, there also exists an

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/TCC.2017.2691323, IEEE Transactions on Cloud Computing

important factor affecting the R-P-E correlation, i.e., the assigned CPU frequency. If more resource capacity (i.e., higher CPU frequency) is assigned to the VM, its computational speed will increase, which has a positive effect on accelerating completion of a given work requirement. However, a higher CPU frequency also implies increased dynamic power consumption of the host server, which may lead to more energy consumed on completing the work requirement.

In this paper, with the consideration of retrying and check-pointing recovery mechanisms, our evaluation models aim to derive potential functions that take assigned CPU frequencies as inputs and return expected service time and expected energy consumption as outputs. The correlated models for retrying and check-pointing are presented in Section 3 and Section 4, respectively.

3 Correlation Model For Retrying

We define cloud service time as the total time spent in completing a specific work requirement. This random time also includes the additional time spent in failure recovery processes. To take the randomness into consideration, we use the expected service time of a cloud service as a performance metric. The computing capability of a VM is directly decided by an assigned CPU frequency, which identifies the processing speed of commands or instructions. Failures in created VMs and hosting servers are treated as different kinds of failures in our reliability model as they have different repair actions.

3.1 R-P Correlation Modeling

For completing a cloud service with a specific work requirement w, the cloud system created a VM for efficiently and flexibly utilize the CPU resource of the host server. In pure performance analysis without taking failures into account, the fault-free completion time of the cloud service (i.e., the time for successfully completing the given work requirement) is calculated by

$$\tau = \frac{w}{c} \tag{1}$$

where $c = u \cdot f_{max}$ is the computing speed, i.e., logical CPU frequency of the VM, and u $(0 < u \leq 1)$ is the CPU utilization of the server hosting the VM. Note that the completion time of the cloud service may be affected by other components, such memory, disk and network. However, capacity of those components can be easily improved by deploying better hardware devices. For more complicated situations where those components may become a bottleneck of τ, a bin-packing approach can be used to determine the value of τ [13].

In real-world scenarios, the execution of a cloud service may be interrupted by various failures, particularly VM failures and server failures in cloud computing environments. This implies that the completion time of the cloud service has a substantial connection with reliability factors, which also means that it is not a constant but a random value. Thus, the expected completion time is a more precise metric for describing service performance.

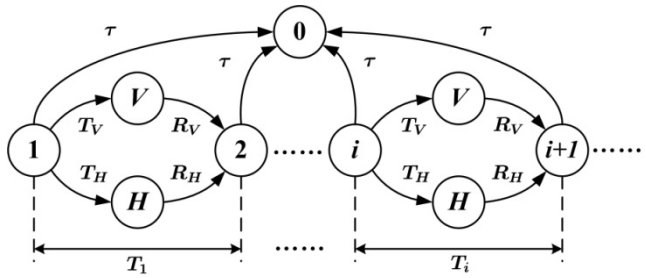


Fig. 1.Markov model for completing a cloud service with retrying.

In this work, we investigate VM failures and server failures to build our R-P correlation model. VM failures are considered as software failures, since a VM is essentially a kind of middleware software. We first build the model for the retrying recovery mechanism. This kind of recovery has no pre-requisites on the system level or the infrastructure level, and thus it is arguably accessible to a cloud environment with a virtualization technology. Moreover, it can also be applied for any kind of cloud services hosted on a VM. In this work, we make the following assumptions for reliability modeling:

A1) A server may be down because of a certain hardware failure. The hardware failure occurrence on a physical server follows a Poisson process with a failure rate θ. Once a hardware failure occurs, the server cannot operate and the cloud services running on it are lost. The new VM of the cloud service will run again as soon as the server is repaired.

A2) A software failure of a running VM is an obvious failure that can be immediately detected by the cloud system, which follows a Poisson process with a failure rate λ.

A3) Any failure initiates a repair process immediately. The server recovery time and the VM recovery time follow exponential distributions with repair rates η and μ, respectively.

A4) The resource capacity assigned to finish a service request remains constant, which means that each run of the VM for serving the request has the same computing speed c.

For the first and second assumptions, the server and VM failures are assumed to follow Poisson processes, which can be explained as being either within the operational phase or in a steady state after a long-time run [14]. The third assumption on repair time following exponential distributions is widely accepted in literature [15].

Let the stochastic process $\{X(t), t \geq 0\}$ represent the system state for completing the cloud service with the work requirement w and the CPU computing speed c, as shown in Fig. 1. State i $(i = 1, 2, \ldots)$ represents the i-th run of the cloud service and state 0 means the cloud service is complete. As for the i-th run, if any failure occurs over the time period τ (i.e., a server failure or a VM failure occurs before the completion of the work requirement), the execution of the service is terminated immediately. States H and V represent a server hardware failure and a VM failure occurs, respectively. Random variables

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/TCC.2017.2691323, IEEE Transactions on Cloud Computing

T_H, T_V, R_H and R_V describe a server failure time, a VM failure time, a server recovery time and a VM recovery time, respectively.

If the cloud service is interrupted by a failure, the execution before the failure and the subsequent recovery phase form an *unsuccessful run*. The execution time can be derived as $T_e = \min(T_V, T_H)$, and the recovery time (denoted by T_r) clearly depends on the kind of the failure that has occurred. The probability that the i-th run becomes unsuccessful can be obtained by

$$\begin{aligned} q &= \Pr(T_e < \tau) = 1 - \Pr(T_V \geq \tau, T_H \geq \tau) \\ &= 1 - e^{-(\lambda+\theta)\tau} \end{aligned} \tag{2}$$

Note that T_e is a continuous random variable ranging from 0 to τ. This implies that the upper bound τ is imposed on T_e under the condition that the failure has occurred over the time period τ. Meanwhile, the i-th run has the probability $p = 1 - q$ to complete the cloud service, and the equation $T_i = T_e = \tau$ is satisfied.

As shown in Fig. 1, the i-th unsuccessful run has the state subset $\phi_i = (i, H, V, i+1)$. Let $F_{A,B}(t)$ represent the one-step transition probability from state A to B during time interval t, where $A \in \phi_i$ and $B \in \phi_i$. Subject to the condition that the unsuccessful execution time cannot exceed τ, the expressions for all existing transition probabilities are given by

$$\begin{aligned} F_{i,H}(t) &= \Pr(T_H < t, T_H < T_V | X(0) = i, T_e < \tau, t < \tau) \\ &= \int_0^t \frac{\theta e^{-(\lambda+\theta)x}}{1 - e^{-(\lambda+\theta)\tau}} dx = \frac{\theta}{\lambda+\theta}\left(\frac{1 - e^{-(\lambda+\theta)t}}{1 - e^{-(\lambda+\theta)\tau}}\right), \ t < \tau \end{aligned} \tag{3}$$

$$\begin{aligned} F_{i,V}(t) &= \Pr(T_V < t, T_V < T_H | X(0) = i, T_e < \tau, t < \tau) \\ &= \int_0^t \frac{\lambda e^{-(\lambda+\theta)x}}{1 - e^{-(\lambda+\theta)\tau}} dx = \frac{\lambda}{\lambda+\theta}\left(\frac{1 - e^{-(\lambda+\theta)t}}{1 - e^{-(\lambda+\theta)\tau}}\right), \ t < \tau \end{aligned} \tag{4}$$

$$F_{H,i+1} = \Pr(R_H < t | X(0) = H) = 1 - e^{-\eta t}, \ 0 < t < \infty \tag{5}$$

$$F_{S,i+1} = \Pr(R_V < t | X(0) = S) = 1 - e^{-\mu t}, \ 0 < t < \infty \tag{6}$$

The cumulative distribution function(CDF) of T_i can thus be derived by

$$F_i(t) = \Pr(T_i < t) = F_{i,V} * F_{V,i+1} + F_{i,H} * F_{H,i+1} \tag{7}$$

where '$*$' denotes the Stieltjes convolution of two functions. Apply the LST (i.e., $\widetilde{F}(s) \equiv \int_0^\infty e^{-st} dF(t)$) to (3)-(6) and substitute the transformed expressions into (7), we can obtain

$$\begin{aligned} \widetilde{F}_i(s) &= \widetilde{F}_{i,V}(s) \cdot \widetilde{F}_{V,i+1}(s) + \widetilde{F}_{i,H}(s) \cdot \widetilde{F}_{H,i+1}(s) \\ &= \frac{1 - e^{-(s+\lambda+\theta)\tau}}{1 - e^{-(\lambda+\theta)\tau}} \cdot \frac{1}{s+\lambda+\theta}\left(\frac{\theta\eta}{s+\eta} + \frac{\lambda\mu}{s+\mu}\right) \end{aligned} \tag{8}$$

From the beginning of any run, if no failure occurs before the completion of the work requirement, the model finally transits into state 0, i.e., the *successful run*. The probability that any run becomes the successful run can be obtained from (2) as

$$p = 1 - q = e^{-(\lambda+\theta)\tau} \tag{9}$$

Since any failure leads to restarting the cloud service under the retrying fault recovery mechanism and each run has the same beginning state without failures, it is rational to treat runs of the service as independent events. Suppose the entire process for completing the service includes N $(N = 1, 2, \ldots)$ runs (i.e., $N-1$ unsuccessful runs and the last successful run), where N is a discrete random variable in accordance with a geometric distribution with parameter q. The probability mass function (pmf) of N is given by

$$p(n) = \Pr(N = n) = pq^{n-1} \tag{10}$$

Let W be the random total time spent in completing the cloud service. It satisfies the following equation

$$W = \begin{cases} \sum_{i=1}^{N-1} T_i + \tau & N \geq 2 \\ \tau & N = 1 \end{cases} \tag{11}$$

Let $F_W(t)$ represent the CDF of W. The corresponding conditional CDF $F_W(t|n) = \Pr(W < t | N = n)$ can be in principle found by taking the convolution of $F_i(t)$ with itself $n-1$ times, and then with $\delta(t-\tau)$. The LST expression of $F_W(t|n)$ is obtained by

$$\widetilde{F}_W(s|n) = e^{-sT_0}[\widetilde{F}_i(s)]^{n-1}, \tag{12}$$

where the equation $\widetilde{F}_1(s) = \widetilde{F}_2(s) = \ldots = \widetilde{F}_{n-1}(s)$ is satisfied, and $e^{-s\tau}$ is the LST expression of $\delta(t-\tau)$. Then, from (10) and (12), using the Bayesian approach to remove the condition of n, the LST of $F_W(t)$ is obtained as

$$\widetilde{F}_W(s) = \sum_{n=1}^{\infty} \widetilde{F}_W(s|n) \Pr(N = n). \tag{13}$$

In this work, we use the inverse of the expected total time spent in completing the cloud service as the performance metric, which is defined as

$$I_P = 1/\mathrm{E}(W). \tag{14}$$

From(13), apply the property of LST that becomes a Moment Generator, $\mathrm{E}(W)$ can be evaluated as

$$\mathrm{E}(W) = -\left.\frac{d\widetilde{F}_W(s)}{ds}\right|_{s=0} = \left(1 + \frac{\theta}{\eta} + \frac{\lambda}{\mu}\right)\left[\frac{e^{(\lambda+\theta)\tau} - 1}{\lambda+\theta}\right] \tag{15}$$

As seen from (15), the performance metric substantially depends on reliability factors, i.e., the parameters λ, μ, θ, and η. Based on (1), it is essentially a function of the assigned CPU utilization u. Although we assume that the failure rate remains a constant during the execution of the cloud service, they could be changed for different runs of the service (see [6]). Our model can be extended with updating failure rates in each run of the cloud service for fitting such a complicated situation.

3.2 R-E Correlation Modeling

The energy consumption of a server within a time interval Δt can be calculated by

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/TCC.2017.2691323, IEEE Transactions on Cloud Computing

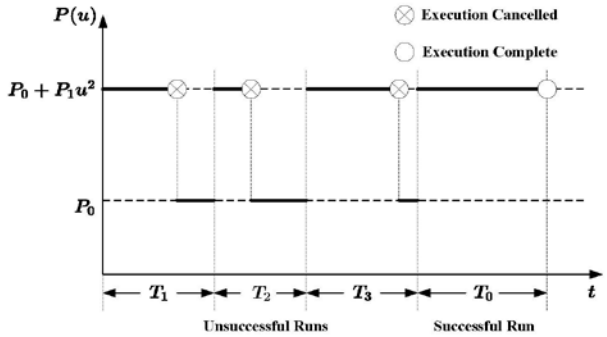


Fig. 2. Change of power consumption caused by random failures and recovery.

$$\Delta E = \int_{t_0}^{t_0+\Delta t} P(x)dx, \tag{16}$$

where $P(t)$ is the power consumption of the server. Theoretically, the power consumption of the server is the sum of power consumption of all components, mainly including CPU, memory, and disk, i.e.,

$$P = P_{cpu} + P_{memory} + P_{disk}. \tag{17}$$

However, in cloud computing environments, memory and disk power usually dominates the power consumption of user devices in the client side but not that of physical servers in the server side [16]. Thus, power consumption of physical servers are usually treated as being CPU-dominated and the power consumption of the other components are assumed to be constant independent of the system activity [17]. Besides, even though the simplest linear power models only taking CPU utilization into account have been proved to meet an accuracy of being less than 10% error [16]. In this paper, we adopt a power consumption model of physical servers as in (18), which is more precise than a linear function and has been widely accepted in the literature[17]-[19].

$$P(u) = P_0 + P_1 \cdot u^3, \tag{18}$$

where P_0 is a constant that describes the basic power consumption of the server including memory and disk power; u $(0 < u \leq 1)$ represents the average CPU utilization that has a significant influence on the dynamic power consumption of the server. For convenience, u is normalized as $u = f/f_{max}$, i.e., the ratio of an assigned CPU frequency to the maximum CPU frequency. In general, dynamic CPU power is CV^2f, where C is capacitance, V is voltage, and $f = uf_{max}$ is operating frequency of the CPU. Since V typically has a linear function in the frequency, i.e., $V = \alpha f$ [18], [19], the server power model can be simplified as (18), where $P_1 = \alpha^2 C f_{max}^3$ is a constant. The peak power of the server is reached when $u = 1$, i.e., $P(1) = P_0 + P_1$. For applying the power model to a specific physical server, parameters P_0 and P_1 usually need to be re-calibrated by using some statistical methods.

Note that the power consumption of the server is indeed a random variable due to random failures and recovery. Fig. 2 shows a sample of random power consumption for completing the cloud service. In this paper, the relatively small CPU frequencies caused by repair actions are negligible, as these actions, e.g. initiating a new VM and rebooting the server, do not occupy CPU resources in general. Thus, the power consumption of the server remains P_0 during recovery phases.

To capture the random change of power consumption, we use the similar modeling approach presented in Fig. 1. First, we build a semi-Markov process $\{X(\epsilon), \epsilon \geq 0\}$ having the same states as $\{X(t), t \geq 0\}$. The difference between $\{X(t)\}$ and $\{X(\epsilon)\}$ is that the latter stays in a given state for a random amount of energy consumption ϵ. Suppose A and B are two states of $\{X(\epsilon)\}$, and denote the random energy consumed in the transition from A to B as $Z_{A,B}$. If the CDF of $Z_{A,B}$, i.e., $G_{A,B}(\epsilon) = \Pr(Z_{A,B} < \epsilon)$ can be obtained, the total energy consumption for completing the cloud service is easily analyzed by associating it to the transitions of the Markov chain.

Note that the power consumption of the server remains as a constant over each one-step state transition of $\{X(\epsilon)\}$, which means the corresponding energy consumption satisfies $Z_{A,B} = P(u) \cdot T_{A,B}$. Thus, the CDF of $Z_{A,B}$ is derived as

$$G_{A,B}(\epsilon) = \Pr\left(T_{A,B} < \frac{\epsilon}{P(u)}\right) = F_{A,B}\left(\frac{\epsilon}{P(u)}\right) \tag{19}$$

Since the CDF of $T_{A,B}$ has been studied in (3)-(6), applying the property of LST, the LST expression of $G_{A,B}(\epsilon)$ can be obtained as

$$\widetilde{G}_{A,B}(s) = \widetilde{F}_{A,B}(P(u)s) \tag{20}$$

For the i-th unsuccessful run of the process $\{X(\epsilon)\}$, the LST expressions of all existing transition probabilities (i.e., $\widetilde{G}_{i,V}(s)$, $\widetilde{G}_{V,i+1}(s)$, $\widetilde{G}_{i,H}(s)$ and $\widetilde{G}_{H,i+1}(s)$) can be obtained by substituting (20) into (3)-(6). Then, denote the energy consumption of the i-th unsuccessful run as Z_i, the LST expression of its CDF yields

$$\begin{aligned}\widetilde{G}_i(s) &= \widetilde{G}_{i,V}(s) \cdot \widetilde{G}_{V,i+1}(s) + \widetilde{G}_{i,H}(s) \cdot \widetilde{G}_{H,i+1}(s) \\ &= \widetilde{F}_{i,V}[P(u)s] \cdot \widetilde{F}_{V,i+1}[P_0 s] + \widetilde{F}_{i,H}[P(u)s] \cdot \widetilde{G}_{H,i+1}[P_0 s]\end{aligned} \tag{21}$$

Let E represent the random total energy consumption for completing the cloud service. Similar to (13), the LST of the conditional CDF of E can be calculated as

$$\widetilde{G}_E(s|n) = [\widetilde{G}_i(s)]^{n-1} e^{-sP(u)\tau} \tag{22}$$

Then, use the Bayesian approach to remove the condition on n, we can obtain

$$\widetilde{G}_E(s) = \sum_{n=1}^{\infty} \widetilde{G}_E(s|n)p(n) \tag{23}$$

The energy consumption metric for completing the cloud service is defined as $I_E = \mathrm{E}(E)$, which can be evaluated as

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/TCC.2017.2691323, IEEE Transactions on Cloud Computing

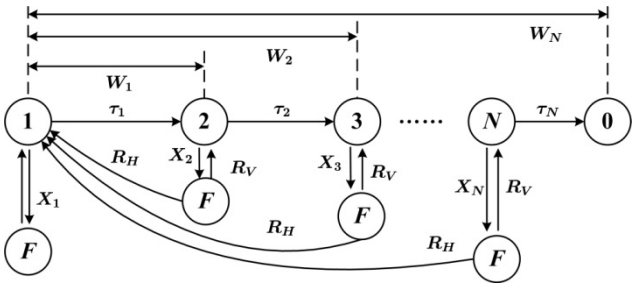


Fig. 3. Check-pointing recovery mechanism for a cloud service with multiple subtasks.

$$\begin{aligned} I_E &= -\frac{d\widetilde{G}_E(s)}{ds}\bigg|_{s=0} \\ &= \left(P(u) + P_0\frac{\theta}{\eta} + P_0\frac{\lambda}{\mu}\right)\left(\frac{e^{(\lambda+\theta)\tau}-1}{\lambda+\theta}\right) \end{aligned} \tag{24}$$

4. Correlation Model For Check-pointing

4.1 Check-pointing for Cloud Service with Multiple Subtasks

In many scenarios, a cloud service may have multiple subtasks. For example, a cloud service performing a data mining application may have multiple subtasks of data checking, data cleaning, data processing, and result verification. Those subtasks are executed in sequence, and the entire cloud service is complete as soon as the last subtask is finished. Obviously, it is inefficient to apply retrying mechanism when a failure occurs, since partial subtasks may be complete before the occurrence of the failure.

To efficiently execute such a cloud service, the cloud system can adopt a check-pointing recovery mechanism. Once a subtask has finished, a memory image (i.e., a checkpoint) of the VM can be created. Thus, if a VM failure occurs during the execution of the next subtask, the VM can be resumed from the memory image, that is, roll back to the prior checkpoint. On the other hand, a server hardware failure is treated as a catastrophic failure that leads to the lost of memory images. Thus, a server failure results in repeating the whole cloud service from the beginning.

Suppose the cloud service has $N(N \geq 1)$ subtasks. The execution of cloud service with the check-pointing recovery mechanism is described as a stochastic process shown in Fig. 3. State n $(n = 1, 2, \ldots, N)$ represents subtask n is finished, while a memory image of the VM is created. State 0 means the whole cloud service is complete. The relatively short time for creating the memory image (i.e., inserting a checkpoint) is neglected here. State F describes a failure occurs and X_n is the random failure time during the run of subtask n. For the other states and random variables, we use the same notations as in Section 3.1.

4.2 Evaluation of Expected Completion Time

Let w_n represent the work requirement of subtask n, and the cloud service has a work requirement of $w = \sum_{n=1}^{N} w_n$. The failure-free completion time of subtask n is $\tau_n = w_n/c$. The random completion time of the subtask n is denoted as T_n, and the random time from the beginning of the cloud service to the first completion of subtask n is denoted as W_n. Thus, the following equation is satisfied.

$$W_n = \begin{cases} T_1, & n = 1 \\ W_{n-1} + T_n, & n = 2, 3, \ldots, N \end{cases} \tag{25}$$

For the run of subtask n, the corresponding failure time X_n has the probability density function (pdf) of

$$f_n(t) = \frac{(\lambda+\theta)e^{-(\lambda+\theta)t}}{1-e^{-(\lambda+\theta)\tau_n}}, \quad 0 \leq t \leq \tau_n \tag{26}$$

Hence, the expected value of X_n (denoted by $\overline{X}_n$) can be calculated as

$$\overline{X}_n = \int_0^{\tau_n} t f_n(t)dt = \frac{1}{\lambda+\theta} - \frac{\tau_n e^{-(\lambda+\theta)\tau_n}}{1-e^{-(\lambda+\theta)\tau_n}} \tag{27}$$

From (3) and (4), an occurred failure has probability $\pi_H = \theta/(\lambda+\theta)$ of being a server failure, and has probability $\pi_V = \lambda/(\lambda+\theta)$ of being a VM failure. Different kinds of failures leads to different kinds of repair actions. The expected values of R_V and R_H are $1/\mu$ and $1/\eta$, respectively. Thus, for the execution of subtask 1, the expression of W_1 is written as

$$W_1 = \begin{cases} \tau_1, & \text{with probability } p_1 \\ X_1 + R_H + W_1, & \text{with probability } \pi_H q_1 \\ X_1 + R_V + W_1, & \text{with probability } \pi_V q_1 \end{cases} \tag{28}$$

where $p_i = \Pr(X_i \geq \tau_i) = \exp\big(-(\lambda+\theta)\tau_i\big)$, $i = 1, 2, \ldots, N$, and $q_i = 1 - p_i$. For the other subtask n $(n = 2, \ldots, N)$, it satisfies

$$T_n = \begin{cases} \tau_1, & \text{with probability } p_n \\ X_n + R_H + W_n, & \text{with probability } \pi_H q_n \\ X_n + R_V + T_n, & \text{with probability } \pi_V q_n \end{cases} \tag{29}$$

Substitute (27) into (28), the expected value of W_1 can be obtained as

$$\overline{W}_1 = \tau_1 + \overline{X}_1\frac{q_1}{p_1} + (\overline{R}_H\pi_H + \overline{R}_V\pi_V)\frac{q_1}{p_1} \tag{30}$$

From (29) and (30), the following recursive expressions are derived for $2 \leq n \leq N$.

$$\begin{aligned} \overline{W}_n &= \tau_n + \overline{X}_n\frac{q_n}{p_n} + (\overline{R}_H\pi_H + \overline{R}_V\pi_V)\frac{q_n}{p_n} \\ &\quad + \overline{W}_{n-1}\left(\frac{1-\pi_V q_n}{p_n}\right) \end{aligned} \tag{31}$$

Applying (31) recursively $N-1$ times, we can get

$$\overline{W}_N = \sum_{j=1}^{N-1}\left(k_j \cdot \prod_{i=j+1}^{N} h_i\right) + k_N \tag{32}$$

where

$$k_j = \left(1 + \frac{\theta}{\eta} + \frac{\lambda}{\mu}\right)\left(\frac{e^{(\lambda+\theta)\tau_j}-1}{\lambda+\theta}\right) \tag{33}$$

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/TCC.2017.2691323, IEEE Transactions on Cloud Computing

$$h_i = \frac{\lambda + \theta e^{(\lambda+\theta)\tau_i}}{\lambda + \theta} \tag{34}$$

The corresponding performance metric of the cloud service is written as

$$I_P = \frac{1}{\overline{W}_N} \tag{35}$$

4.3 Evaluation of Expected Energy Consumption

We use the similar modeling approach presented in section 3.2 to evaluate the expected execution time of the cloud service from the stochastic model shown in Fig. 3. For the run of subtask n, let Z_n, Z_H, and Z_V represent random energy consumed in execution, server hardware recovery, and VM recovery, respectively. According to the change of power consumption shown in Fig. 2, we can obtain

$$\begin{aligned} \overline{Z}_n &= P(u) \cdot \overline{X}_n \\ \overline{Z}_H &= P_0 \cdot \overline{R}_H \\ \overline{Z}_V &= P_0 \cdot \overline{R}_V \end{aligned} \tag{36}$$

Let Y_n $(n = 1, 2, \ldots, N)$ represent the energy consumed in completing subtask n, and E_n is the total energy consumed in the procedure from the beginning of the cloud service to the first completion of subtask n. From (28) and (29), we can obtain

$$Z_n = \begin{cases} P(u)\tau_n, & \text{with probability } p_n \\ Z_n + Z_H + E_n, & \text{with probability } \pi_H q_n \\ Z_n + Z_V + Z_n, & \text{with probability } \pi_V q_n \end{cases} \tag{37}$$

where $E_1 = Z_1$ and $E_n = E_{n-1} + Z_n$ is satisfied. Using a recursive calculation that is similar to (30) and (31), we can obtain the expected energy consumption for completing the entire cloud service as

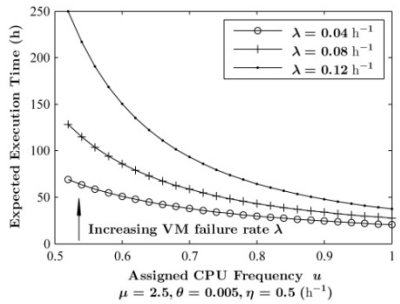
$$I_E = \overline{E}_N = \sum_{j=1}^{N-1} \left(g_j \cdot \prod_{i=j+1}^{N} h_i \right) + g_N \tag{38}$$

where h_i can be obtained from (34), and g_j is

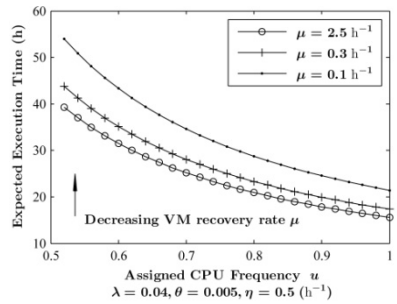
$$g_j = \left(P(u) + P_0 \frac{\theta}{\eta} + P_0 \frac{\lambda}{\mu} \right) \left(\frac{e^{(\lambda+\theta)\tau_j} - 1}{\lambda + \theta} \right) \tag{39}$$

5 Optimization Technique

For executing the cloud service, the assigned CPU frequency of the VM is an important decision variable. The performance and energy metrics are respectively expressed as functions $I_P(u)$ and $I_E(u)$, which take CPU utilization u as an important input. The traditional CPU assignment strategy only emphasizing the performance metric may not meet the optimal objective of saving energy consumption. Thus, it is important to find the Pareto set of decision variable u from both performance and energy perspectives. The combined performance-energy optimization model is given as



(a) Different VM failure rates



(b) Different VM recovery rates

Fig. 4. Expected execution time vs. assigned CPU frequency at different VM failure/recovery rates.

$$\begin{aligned} &\text{Decision Variable:} && u \\ &\text{Objective:} && \max I_P(u) \\ & && \min I_E(u) \\ &\text{Subject to:} && 0 < u \leq 1 \end{aligned} \tag{40}$$

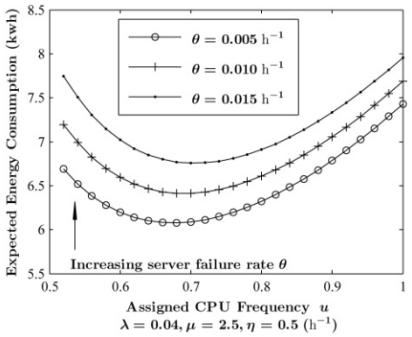
Note that we simplify I_P and I_E without identifying retrying or check-pointing. The optimization models for the retrying and check-pointing mechanisms can be specified by substituting (14) and (24), and (35) and (38) into (40), respectively. From (14) and (35), expressing the performance metrics of retrying and check-pointing, it can be found that $I'_P(u) > 0$, and thus $u^*_P = 1$ is the optimal solution for maximizing the expected performance metric for the two kinds of recovery mechanisms. From (24) and (38), it can be proved that $I''_E(u) > 0$ for all $u \in (0, 1]$, which means (24) and (38) are convex functions. Since $\lim_{u \to 0} I'_E(u) = -\infty < 0$, there must exist an optimal solution $u^*_E \in (0, 1]$ for minimizing the energy consumption metric. Solution u^*_E directly depends on the server power and reliability parameters, which can be derived as

$$u^*_E = \begin{cases} u_0, & \exists u_0 \in (0, 1), I'_E(u_0) = 0 \\ 1, & \text{otherwise} \end{cases} \tag{41}$$

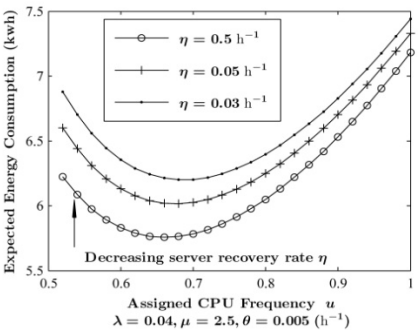
Now, the Pareto set of u for the combined optimization model (43) is obtained as

$$u^* = \begin{cases} \forall u \in [u_0, 1], & u^*_E = u_0, \quad u^*_P = 1 \\ 1, & u^*_E = 1, \quad u^*_P = 1 \end{cases} \tag{42}$$

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/TCC.2017.2691323, IEEE Transactions on Cloud Computing



(a) Different server failure rates



(b) Different server recovery rates

Fig. 5. Expected energy consumption vs. assigned CPU frequency at different server failure/recovery rates.

As seen from (42), solution u^* gives an important lower bound of u_0, which means that from both performance and energy perspectives, it is seriously inefficient to make the cloud service be executed in a low CPU frequency that $u < u_0$.

6 EXAMPLES

6.1 Examples for Retrying

First, we use numerical examples to show important R-P and R-E correlations. The server executing the cloud service is assumed to be Tecal X6000 deployed with the CPU of Intel Xeon X3480. The maximal computing ability of the server is 16×3.0 GHz. The basic power and the peak power are $P_0 = 109$ w and $P(1) = 470$ w, respectively (refer to the test results in [20]). From (18), the maximal dynamic power is $P_1 = P(1) - P_0 = 361$ w, and the corresponding power model of the server is $P(u) = 361 \times u^3 + 109$. The cloud service is supposed to have the work requirement of 2×10^3 T instructions.

Fig. 4 shows significant effects on the expected execution time of the cloud service caused by different reliability parameters. It can be found that the expected execution time of the cloud service varies inversely with assigned CPU frequency, which means that assigning more resource capacity has a more significant effect on improving performance. Fig. 4(a) and Fig. 4(b) demonstrate that the execution time can be shortened by reducing the VM failure rate and increasing the VM recovery rate, respectively. They essentially illustrate the R-P correlation that higher reliability leads to better performance. Note that

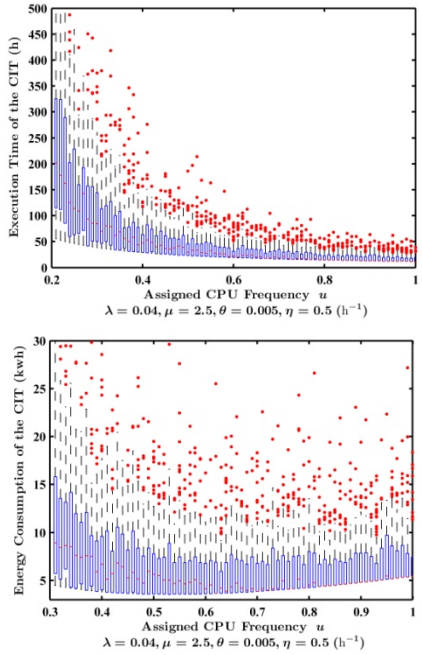


Fig. 6. Experimental results for retrying.

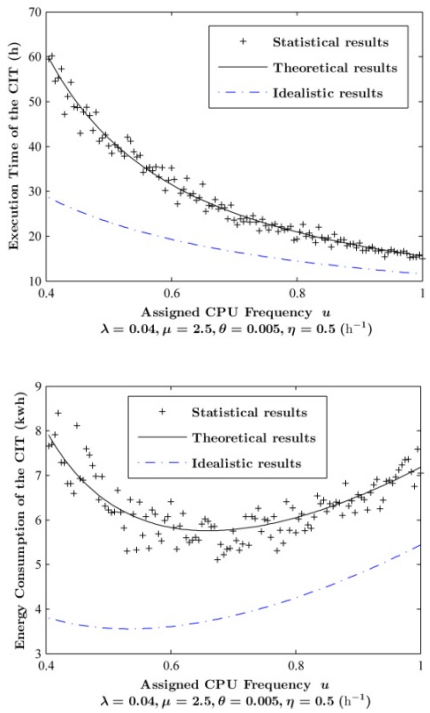


Fig. 7. Statistical results vs. theoretical results for retrying.

the changing trend of the curves in Fig. 4(a) is slightly different from that in Fig. 4(b). In Fig. 4(a), it can be observed that a higher VM failure rate results in that the expected execution time increases more dramatically when the CPU frequency decreases, which does not exist for the VM recovery rate shown in Fig. 4(b). This phenomenon can be explained by the fact that a lower CPU frequency induces a longer fault-free completion time τ, which potentially incurs a higher occurrence probability of failures, especially when the VM failure rate increases. This phenomenon is different from recovery processes

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/TCC.2017.2691323, IEEE Transactions on Cloud Computing

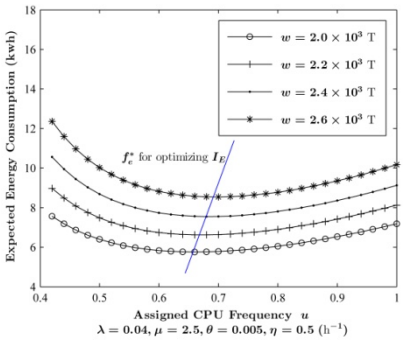


Fig. 8. Optimal CPU frequency for minimizing expected energy.

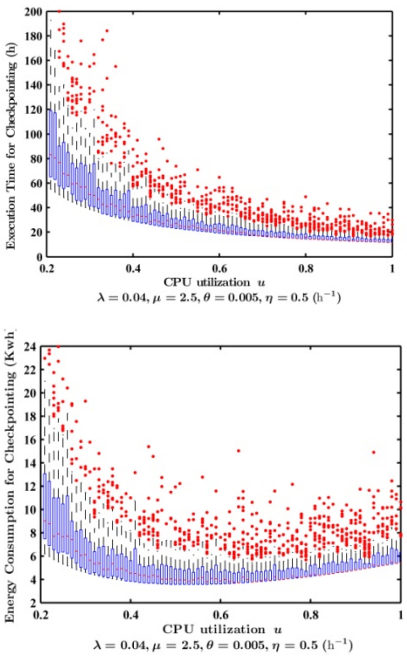


Fig. 9. Experimental results for check-pointing.

that are independent of the execution of the cloud service. Another important R-E correlation is also illustrated in Fig. 5. In general, the values of failure rates and recovery rates need to be assessed by using some statistical methods for different physical devices [14], [15].

To validate the proposed R-P-E correlation model for the cloud service, we design an experimental program based on the Monte Carlo method for simulating the run of the cloud service. The reliability parameters are given as $\lambda = 0.04$, $\mu = 2.5$, $\theta = 0.005$, and $\eta = 0.5$ (h^{-1}). For a special value of u, the experimental program runs 100 times for evaluating the execution time and energy consumption of the cloud service. Fig. 6 shows the experimental results obtained by running the experimental program, which are depicted by box-plots. Fig. 7 illustrates comparisons between statistical results and theoretical results, and also shows the corresponding fault-free completion time (i.e., idealistic results). It can be found that the fluctuation range of experimental results increase gradually with a decrease in the assigned CPU frequency, as shown in Fig. 6. This phenomenon implies performance and energy consumption have more chance to be

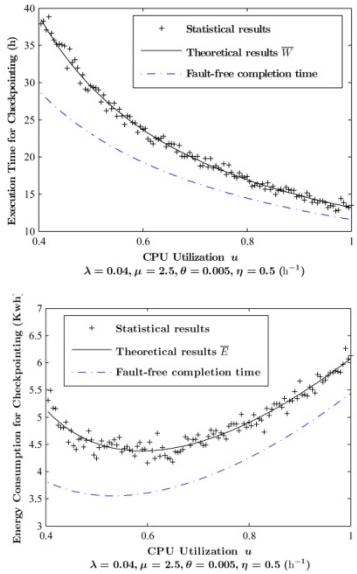
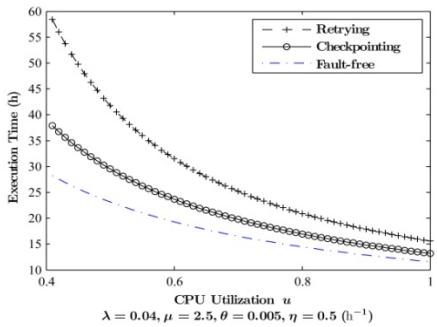
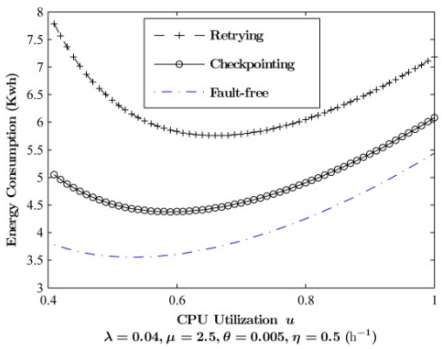


Fig. 10. Statistical results vs. theoretical results for check-pointing.



(a) Performance



(b) Energy consumption

Fig. 11. Comparison between retrying and check-pointing.

affected by the random failures when the assigned CPU frequency becomes lower. Thus, making the cloud service work at an excessively low CPU frequency is not efficient from both performance and energy consumption perspectives. In Fig. 7, it also can be observed that the statistical results are very close to the theoretical results, which verifies the proposed R-P-E correlation model.

From (44), the optimal CPU frequency for minimizing expected energy consumption is $u_E^* = 0.6442$, and the Pareto set for optimization model (43) is $u^* = [0.6442, 1]$. Fig. 8 depicts optimal CPU frequencies of different work

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/TCC.2017.2691323, IEEE Transactions on Cloud Computing

requirements. It can be observed that the energy saving caused by CPU scaling approach (i.e., $u \geq u_E^*$) is obvious and feasible. Meanwhile, the most important meaning of the Pareto set is that it gives an important lower bound u_E^*, and the assigned CPU utilization should not be less than u_E^* at least.

6.2 Examples for Check-Pointing

We take a data mining application as an example for illustrating the presented modeling, analysis and evaluation of cloud service using a check-pointing mechanism. The cloud service is supposed to have four subtasks, i.e., data checking, data cleaning, data processing, and result verification, executed in sequence. For the purpose of comparison with retrying, the work requirement of the entire service and other parameters take the same values as in Section 6.1. The subtasks of the cloud service have percentage work requirements of 20%, 25%, 45% and 10%. To verify the proposed theoretical model for check-pointing, we also design a corresponding experimental program. The experiment results obtained by running the program is shown in Fig. 9. The comparison between statistical results and theoretical results, shown in Fig. 10, verifies the correctness of the proposed model.

The comparison between check-pointing and retrying in performance and energy metrics is illustrated in Fig. 11. In general, given a specific work requirement of the cloud service, check-pointing can achieve better performance and energy metrics. Note that the optimal CPU utilization for minimizing energy consumption of check-pointing is lower than that for retrying, as shown in Fig. 11 (b). This is because that check-pointing can achieve a shorter expected completion time as compared with retrying, and thus reducing power consumption can have a more obvious effect on saving energy consumption.

7 Literature Review and Discussion

In this paper, we proposed a theoretical correlation model for evaluating multiple correlated indices for a cloud service. Theoretical modeling is always a critical research field that can further contribute to designing a rational resource scheduling strategy. There are many studies focusing on building theoretical models for analyzing a single metric in detail. For example, Fan et al. investigated various power models for full systems [17]. Baliga et al. proposed a power model including power of various components (CPU, memory, disk, network, etc.) [21]. The service time determining energy consumption is also a key performance metric. Khazaei et al. presented an iterative modeling approach combining multiple fine-grained sub-models for evaluating accurate service time of cloud services [22]. Bruneo applied stochastic reward nets to construct analytical performance model for analyzing several performance metrics: utilization, waiting time and responsiveness [23]. For reliability modeling, Dai et al. systemically studied multiple types of failures like network failures, time-out failures, matchmaking failures and resource failures and proposed a hierarchical correlated model in terms of virtualized service reliability [24],[25]. However, these models did not consider complicated correlations among reliability, performance and energy factors, and thus may not meet a comprehensive optimization demand.

Considerable recent studies have gradually concerned on joint modeling and optimization for correlated metrics. For example, Tudoran et al. studied an environment-aware data transmission optimization strategy considering the R-P correlation, which can deliver fast response for improving performance, and can alleviate adverse effects caused by low reliability and reusability [26]. Ghosh et al. quantified the R-P correlation as performability metrics, including expected response time, expected execution time, and expected delay time, for a cloud service [27]. As for the P-E tradeoff, Green et al. investigated the tradeoff between performance and power consumption for multi-core servers [28]. Xia et al. also studied correlation between energy consumption and network performance, and applied dynamic voltage scaling to reduce energy consumption [5]. Lee and Zomaya presented a composite evaluation function covering energy consumption and performance based on CPU frequency scaling and task consolidation [8]. Chen et al. considered data storage and processing in mobile cloud, i.e., dynamic topology of network, and then addressed problems about reliability and energy efficiency in an integrated manner (i.e., correlating reliability and energy factors) [29]. However, these existing joint models are hard to be further extended as R-P-E joint models. Although our prior work has studied a R-P-E joint modeling approach for a cloud service [10], which however did not address the retrying and check-pointing recovery mechanisms.

Another important research filed related to this paper is design of resource optimization strategies. In principle, comprehensive resource optimization strategies need to simultaneously take multiple correlation metrics into account. Many researches have studied various optimization strategies, such as network resource optimization [4], online optimization of data center [30], job scheduling optimization by using forecast technologies [31]. But these optimization technologies were proposed from a cloud service perspective without considering a specific work requirement. Although there are many studies that presented various optimization approaches to reducing average completion time [32], decreasing resource usage cost [33], and minimizing service delay [34], they cannot meet diverse requirements because the important R-P-E correlation is not addressed. The optimization models described in this paper are based on the proposed R-P-E correlation model, which can effectively solve the issue.

The optimization models studied in this work can significantly contribute to resource scheduling for a single cloud service. There exists another important kind of scheduling problems, i.e., job scheduling. The R-P-E models presented in this paper can be extended to evaluate the performance and energy consumption of a job allocation by taking $\Delta u = u_{job} - u_0$ as the input parameter of the R-P-E models, where u_0 is the CPU utilization of the server before the job allocation, and u_{job} is the utilization of the server after the job allocation. Thus, according

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/TCC.2017.2691323, IEEE Transactions on Cloud Computing

to the optimization technique presented in Section 5, an optimization solution Δu^* can also be derived for job scheduling. However, the job scheduling for multiple cloud services is a complicated NP-hard problem, which usually needs some heuristic algorithms to be solved. This will be studied in our future work.

8 Conclusion

Over the last few years, cloud computing has been widely deployed to provide flexible resource provisioning for various services. Cloud services with a specific work requirement are a typical kind of these services, whose performance is significantly affected by assigned resource capability. The cloud system needs to develop appropriate resource scheduling strategies for these services. This is a difficult issue because multiple correlated metrics (not only performance but also reliability and energy consumption) must be modeled and evaluated at the same time. Most of the existing research treats these metrics separately with no effective models to correlate these parameters, which makes it difficult for further developing a multi-objective optimization solution for satisfying increasingly diverse application requirements.

This paper makes original contributions by systematically studying correlations amongst reliability, performance and energy consumption metrics. The proposed model encompasses Markov models, Laplace-Stieltjes transform, a Bayesian approach and a recursive method. The important R-P-E correlations are investigated in our models for evaluating expected service time and energy consumption of a cloud service under two typical recovery mechanisms, retrying and check-pointing. The presented model makes evaluation of the correlated metrics clear by identifying assigned resource capacity (i.e., CPU frequency) as the critical correlation factor. Moreover, we build an expected performance-energy combined optimization problem based on the proposed models, which describe the trade-off between expected service performance and energy consumption of a cloud service. The Pareto optimal solutions are also determined.

The numerical examples in this work illustrate that server reliability has a significant influence on both service performance and energy consumption. Hence, it is not appropriate to treat these indices separately. Moreover, we demonstrate the different effects of adopting retrying and check-pointing recovery mechanisms on performance and energy consumption metrics. The examples also show that evaluation metrics provided by the proposed models can be used to assess an optimal CPU frequency for executing a cloud service.

In realistic environments, there exist more complicated application scenarios, such as migrating VM due to resource preemption, which may lead to dynamic changes in reliability parameters of the host server. Such a complicated situation will be considered in our future work. The power model presented in this paper is simplified by assuming server power is dominated by CPU. In the future we will also investigate a more comprehensive power model considering dynamic power consumptions caused by various components. We are also interested in extending our model to various cloud services whose completion time not only depends on the CPU frequency, but also are significantly affected by memory, disk and network. Another directions of our future work include modeling cloud services executing parallel subtasks under different fault recovery mechanisms, and designing some heuristic algorithms (e.g., genetic algorithms) for solving complicated job scheduling problems.

Acknowledgment

The authors would like to thank the anonymous referees for their constructive comments that have helped improving the quality of this work. This work was supported in part by the Natural Science Foundation of China under Grant 61170042, and by the Innovational Team Project of Sichuan Province (No. 2015TD0002).

References

[1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg and I. Brandic, "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as 5th Utility," *Future Gener. Comput. Syst.*, vol. 25, no. 3, pp. 599-616, 2009.

[2] R. Moreno-Vozmediano, R. S. Montero and I. M. Llorente, "Key Challenges in Cloud Computing Enabling the Future Internet of Services," *IEEE Internet Comput.*, vol. 17, no. 4, pp. 18-25, 2013.

[3] E. Gelenbe and R. Lent, "Energy-Qos Trade-offs in Mobile Service Selection," *Future Internet*, vol. 5, no. 2, pp. 128-139, Sep. 2013.

[4] D. Kliazovich, P. Bouvry, and S. U. Khan, "DENS: Data Center Energy-efficient Network-aware Scheduling," *Cluster Comput.*, vol. 16, no. 1, pp. 65-75, Mar. 2013.

[5] Y. Xia, M. C. Zhou, X. Luo, S. C. Pang, and Q. Zhu, "A Stochastic Approach to Analysis of Energy-aware DVS-enabled Cloud Datacenters," *IEEE Trans. Syst. Man, Cybern., Syst.*, vol. 45, no. 1, pp. 73-83, Jan. 2015.

[6] K. Tokuno, and S. Yamada, "Codesign-oriented Performability Modeling for Hardware-software Systems," *IEEE Trans. Reliab.*, vol. 60, no. 1, pp. 171-179, 2011.

[7] R. Ghosh, K. S. Trivedi, V. K. Naik, and D. S. Kim, "End-to-end Performability Analysis for Infrastructure as a Service Cloud: An Interacting Stochastic Models Approach," *Proc. 16th IEEE Pac. Rim Int. Symp. Depend. Comput.*, pp. 125-132, 2010.

[8] Y. C. Lee, A. Y. Zomaya, "Energy Efficient Utilization of Resources in Cloud Computing Systems," *J. Supercomput.*, vol. 60, no. 2, pp. 268-280, 2012.

[9] J. Subirats and J. Cuitart, "Assessing and Forecasting Energy Efficiency on Cloud Computing Platforms," *Future Gener. Comput. Syst.*, vol. 45, pp. 70-94, 2015.

[10] X. Qiu, Y. Dai, Y. Xiang, and L. Xing, "A Hierarchical Correlation Model for Evaluating Reliability, Performance, and Power Consumption of a Cloud Service," *IEEE Trans. Syst. Man, Cybern., Syst.*, vol. 46, no. 3, pp. 401-412, 2016.

[11] M. Xiao, T. Kloks, and S. H. Poon, "New Parameterized algorithms for the edge dominating set problem," *Math. Found. Comput. Sci.*, Berlin: Springer, pp. 604-615, 2011.

[12] M. Hamill, and K. Goseva-Popstojanova, "Common Trends in Software Fault and Failure Data," *IEEE Trans. on Software Eng.*, vol. 35, no. 4, pp. 484-496, 2009.

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/TCC.2017.2691323, IEEE Transactions on Cloud Computing

[13] R. Grandl, G. Ananthanarayanan, S. Kandula, S. Rao, and A. Akella, "Mlti-Resource Packing for Cluster Schedulers," *ACM SIGCOMM Comput. Commun. Review*, vol. 44, no. 4, pp. 455-466, 2014.

[14] M. Xie, Y. S. Dai, and K. L. Poh, *Computing Systems Reliability: Models and Analysis*. New York: Kluwer, 2004.

[15] K. S. Trivedi, *Probability and Statistics with Reliablity, Queuing, and Computer Science Applications*. New York: Wiley, 2001.

[16] R. Rivoire, P. Ranganathan, and C. Kozyrakis, "A Comparsion of High-level Full-system Power Models," *Proc. 2008 Conf. Power Aware Comput. Syst.*, pp. 3-8, 2008.

[17] X. Fan, W. D. Weber, and L. A. Barroso, "Power Provisioning for a Warehouse-sized Computer," *Proc. 34th Annu. Int. Symp. Comput. Archit. (ISCA '07)*, pp. 13-23, 2007.

[18] E. N. M. Elnozahy, M. Kistler, and R. Rajamony, "Energy-efficient Server Clusters," *Power Aware Comput. Syst.*, pp. 179-197, 2002.

[19] T. Mudge, "Power: A First-class Architectural Design Constraint," *Comput.*, vol. 4, pp. 52-58, 2001.

[20] SPEC, "All Published SPECpower_ssj2008 Results," http://www.spec.org/power_ssj2008/results/power_ssj2008.html, 2016.

[21] J. Baliga, R. W. Ayre, K. Hinton, and R. Tuchker, "Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport," *Proc. IEEE*, vol. 99, no. 1, pp. 149-167, Jan. 2011.

[22] H. Khazaei, J. Misic, V. B. Misic, "A Fine-Grained Performance Model of Cloud Computing Centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 11, pp. 2138-2147, 2013.

[23] D. Brueno, "A Stochastic Model to Investigate Data Center Performance and QoS in IaaS Cloud Computing Systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 3, pp. 560-569, 2014.

[24] Y. S. Dai, Y. Pan, and X. K. Zhou, "A Hierarchical Modeling and Analysis for Grid Service Reliability," *IEEE Trans. Comput.*, vol. 56, no. 5, pp. 681-691, 2007.

[25] Y. S. Dai, Y. Xiang, Y. Li, L. Xing and G. Zhang, "Consequence Oriented Self-Healing and Autonomous Diagnosis for Highly Reliable Systems and Software," *IEEE Trans. Reliab.*, vol. 60, no. 2, pp. 369-380, 2011.

[26] R. Tudoran, A. Costan, and G. Antoniu, "Overflow: Mult-Site Aware Big Data Management for Scientific Workflows on Clouds," *IEEE Trans. on Cloud Comput.*, vol. 4, no. 1, pp. 76-89, 2015.

[27] R. Ghosh, K. S. Trivedi, V. K. Naik, and D. S. Kim, "End-to-end Performability Analysis for Infrastructure as a Service Cloud: An Interacting Stochastic Models Approach," *Proc. 16th IEEE Pac. Rim Int. Symp. Depend. Comput.*, pp. 125-132, 2010.

[28] R. C. Green, L. Wang, and M. Alam, "Applications and Trends of High Performance Computing for Electric Power Systems: Focusing on Smart Grid," *IEEE Trans. Smart Grid*, vol.4, no.2, pp. 922-931, 2013.

[29] C. Chen, M. Won, R. Stoleru, G. G. Xie, "Energy-Efficient Fault-Tolerant Data Storage and Processing in Mobile Cloud," *IEEE Trans. on Cloud Comput.*, vol. 3, no. 1, pp. 28-41, 2014.

[30] R. Urgaonkar, U. C. Kozat, K. Igarashi and M. J. Neely, "Dynamic Resource Allocation and Power Management in Virtualized Data Centers," *2010 IEEE-IFIP Netw. Oper. Manage. Symp.*, pp. 479-486, 2010.

[31] J. Subirats and J. Guitar, "Assessing and Forecasting Energy Efficiency on Cloud Computing Platforms," *Future Gener. Comput. Syst.*, vol. 45, pp. 70-94, 2015.

[32] Y. Zhao, K. Chen, W. Bai, M. Yu, C. Tian, Y. Geng, Y. Zhang, D. Li, and S. Wang, "Rapier: Integrating Routing and Scheduling for Coflow-Aware Data Center Networks,"*2015 IEEE Conf. Comput. Commun. (INFOCOM)*, pp. 424-432, 2015.

[33] B. Palansisamy, A. Singh, and L. Liu, "Cost-Effective Resource Provisioning for MapReduce in a Cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 5, pp. 1265-1279, 2014.

[34] C. Q. Wu, X. Lin, D. Yu, W. Xu, and L. Li, "End-to-End Delay Minimization for Scientific Workflows in Clouds under Budget Constraint," *IEEE Trans. Cloud Comput.*, vol. 3, no. 2, pp. 169-181, 2014.

Xiwei Qiu received his B.S. degree in electronic and information engineering from Jilin University in 2004, and M.S. degree in software engineering and Ph.D. degree in computer science from the University of Electronic Science and Technology of China (UESTC) in 2009 and 2016, respectively.

He is currently a postdoctoral research fellow at the Collaborative Autonomic Computing (CAC) laboratory at UESTC. His research interests include cloud computing, reliability, modeling and optimization, and energy-efficient computing.

Yuanshun Dai (M'03) received his B.S. degree from Tsinghua University, Beijing, China, in 2000, and the Ph.D. degree from the National University of Singapore, Singapore, in 2003.

He is currently the Dean of School of Computer Science and Engineering, University of Electronic Science and Technology of China. He is also Chaired Professor and Director of the Collaborative Autonomic Computing (CAC) Laboratory. He serves as Chairman of the Professor Committee in the School since 2012; and as the Associate Director at the Youth Committee of the "National 1000er Plan" in China. He has published more than 100 papers and 5 books, where there are 62 papers indexed by SCI including 31 IEEE/ACM Transactions papers. His current research interests include cloud computing, dependability, security, big data, and autonomic computing.

Dr. Dai was the Program Chair of the 12th IEEE Pacific Rim Symposium on Dependable Computing. He was also the Founder and the General Chair of the IEEE Symposium on Dependable Autonomic and Secure Computing. He served as a Guest Editor of the IEEE Transactions on Reliability. He is also on the editorial boards of several journals.

Yanping Xiang received her B.S., and M.S. degrees from Shanghai Jiao Tong University, Shanghai, China, in 1993, and 1996 respectively; and Ph.D. degree from National University of Singapore in 2003.

In 2008, Dr. Xiang joined the School of Computer Science, University of Electronic Science and Technology of China (UESTC). Now she is a Professor at the Collaborative Autonomic Computing (CAC) laboratory at UESTC. Before joining UESTC, she was a Research Fellow of the Industrial and System Engineering Department of National University of Singapore. Her current research interests include cloud computing, decision making, reliability, and autonomic computing.

Liudong Xing(S'00-M'02-SM'07) received the B.E. degree in computer science from Zhengzhou University, China in 1996, and the M.S. and Ph.D. degrees in electrical engineering from the University of Virginia in 2000 and 2002, respectively.

She is a Professor with the Department of Electrical and Computer Engineering, University of Massachusetts (UMass) Dartmouth,

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/TCC.2017.2691323, IEEE Transactions on Cloud Computing

USA. Her research focuses on reliability modeling and analysis of complex systems and networks.

Prof. Xing was the recipient of the Leo M. Sullivan Teacher of the Year Award in 2014, the Scholar of the Year Award in 2010, and the Outstanding Women Award in 2011 of UMass Dartmouth. She was the recipient of the 2015 ChangJiang Scholar award by the Ministry of Education of China and the 2007 IEEE Region 1 Technological Innovation (Academic) Award. She was also co-recipient of the Best (Student) Paper Award at several conferences. She is an Associate Editor or Editorial Board member of multiple journals including Reliability Engineering & System Safety, and International Journal of Systems Science.