Performance and Reliability of Non-Markovian Heterogeneous Distributed Computing Systems

Jorge E. Pezoa

Majeed M. Hayat

Performance and Reliability of Non-Markovian Heterogeneous Distributed Computing Systems

Jorge E. Pezoa
Universidad de Concepción, Concepción
Majeed M. Hayat
University of New Mexico, Albuquerque

Abstract:

Average service time, quality-of-service (QoS), and service reliability associated with heterogeneous parallel and distributed computing systems (DCSs) are analytically characterized in a realistic setting for which tangible, stochastic communication delays are present with nonexponential distributions. The departure from the traditionally assumed exponential distributions for event times, such as task-execution times, communication arrival times and load-transfer delays, gives rise to a non-Markovian dynamical problem for which a novel age dependent, renewal-based distributed queuing model is developed. Numerical examples offered by the model

shed light on the operational and system settings for which the Markovian setting, resulting from employing an exponential-distribution assumption on the event times, yields inaccurate predictions. A key benefit of the model is that it offers a rigorous framework for devising optimal dynamic task reallocation (DTR) policies systematically in heterogeneous DCSs by optimally selecting the fraction of the excess loads that need to be exchanged among the servers, thereby controlling the degree of cooperative processing in a DCSs. Key results on performance prediction and optimization of DCSs are validated using Monte-Carlo (MC) simulation as well as experiments on a distributed computing testbed. The scalability, in the number of servers, of the age-dependent model is studied and a linearly scalable analytical approximation is derived.

Keywords

SECTION 1 Introduction

In the analysis of performance and reliability of applications executed on heterogeneous DCSs, analysts, designers, and users alike need formal methods to assess and compare different designs and implementations before deciding on a specific solution. Modeling DCSs and assessing application's performance and reliability are complicated tasks due to several reasons. First, servers in a DCS offer heterogeneous processing capabilities to their users. Second, the communication network of a DCS connects geographically dispersed servers, as in the case of wide area networks, peer-to-peer (P2P) networks and donation grids, where significant stochastic, heterogeneous communication delays may be imposed to the exchange of information. Finally, a DCS comprises a large number of elements that interplay in a collective and stochastic manner creating a complex system dynamics [1], [2], [3].

Because of this unavoidable complexity, simulation, and in particular discrete event simulation (DES), has emerged as the most employed method for both modeling as well as performance and reliability analysis in heterogeneous DCSs. The most appealing aspect of DES is its simplicity and general application, while its main drawback is its high computational cost to achieve accurate results [1]. In addition to DES, modeling has been conducted by means of model description languages such as Petri Nets [4] and graphical modeling tools such as fault trees [5] and reliability graphs [6].

Analytical methods for performance and reliability have also been developed in the literature. To circumvent the inherent complexity of the models, researchers usually impose a set of simplifying assumptions and constraints so that system's behavior can be characterized in a tractable manner. The most common simplifying assumptions employed in the literature are heuristics related to applications' service time [7], [8], [9], homogeneous capabilities of servers and/or communication links [8], time-invariant DCS topology [10], and the deterministic behavior of the transfer time of tasks [9], [11], [12]. Researchers have argued that these assumptions are meaningful solely in parallel computing environments [4], [13], [14]. They further argue that performance and reliability analysis in DCSs must consider stochastic models with nonhomogeneous parameters due to the large amount of uncertainty introduced by both communication delays and the number of working servers.

The stochastic model most widely used to represent a DCS is the Markovian distributed queuing network. Such model is obtained under the assumption that all the timed-events that collectively govern the behavior of the DCS (e.g., task-execution times, communication arrival times, load-transfer times and server failure times) follow exponential distributions [12], [15], [16], [17], [18], [19]. The vast amount of research on Markovian models has yielded several modeling and analysis tools such as TimeNET and WebSPN [12], [17], [18], [19]. The main advantages provided by the Markovian assumption are that highly simplifies the calculations, yields tractable

solutions, avoids the time-dependent behavior of the system dynamics [20], and leads to closed-form [17], [18], [19] or recursive characterizations [16], [21] of complex performance metrics.

Modeling and analysis of queuing systems in non-Markovian settings has also been conducted. The classical approach is to abstract a DCS in terms of $G/G/n$ queues [3], [22]. Analysis can be conducted under this model using basic principles and standard methods from stochastic processes, such as state-space expansion based on phase-type distributions [1] or on the method of supplementary variables introduced by Cox in [23]. Specifically, Cox analyzed a completely reliable $G/G/n$ queuing system that follows distributions with rational Laplace transforms, and modeled the probability of finding a server available using the Laplace transform.

Recently, modeling and analysis of non-Markovian queuing systems has been conducted in terms of non-Markovian stochastic Petri Nets [4], [24], [25], [26], [27], [28]. In these works non-Markovian stochastic Petri Nets have been developed and their capabilities have been exploited to model the performance of web server networks [26], queues of embedded systems [26], [27] and the queues of two network terminals [28] among other applications. The common approach taken in the aforementioned works is that non-Markovian stochastic Petri Nets have been developed by considering both general distributions for the random times firing state transitions and continuous supplementary variables, in the form of age variables, to account for the memory of the general distributions.

While the Markovian assumption highly simplifies calculations it is not always appropriate to model practical systems. For instance, it is easy to observe that physical constraints of practical systems impose unavoidable minimum response times to both service and transfer times of applications. For the service time, this unavoidable time results from typical operations carried out by applications, such as loading the data into the memory, making system calls and initializing variables. For the transfer times, actual communication networks always introduce a nonzero end-to-end propagation delay to any exchange of information. In fact, in earlier works we have observed that empirical characterizations for both task service times and task transfer times that were obtained from actual DCSs follow power-law distributions like the Pareto distribution [16], [29]. Moreover, researchers have shown that Markovian models may introduce significant errors on the calculation of performance and reliability metrics [16], [25]. In particular, in [16], we showed by means of Monte-Carlo (MC) simulations that the service reliability of a DCS calculated under the Markovian assumption can be highly inaccurate in settings where the average task-transfer delays are large compared to the average task service times.

In this paper, we extend the stochastic regeneration-based models presented in [16], [21], and [29] in significant ways. (Here, we use the terms regeneration and renewal interchangeably.) First and foremost, we relax the assumption on the exponential distribution of the random event times that govern the dynamics of a DCS. To this end, we introduce in our analysis auxiliary age variables and constructed a continuous-time age matrix that keeps track of the memory of all the nonexponential random times. The age matrix augments the queuing state-space model presented in [16] and [21] yielding a hybrid continuous and discrete state-space. This hybrid setting enables the development of an age-dependent stochastic regeneration theory for the distributed queuing problem at hand, which, in turn, leads to the analytical, recursive characterization of the service time of an application being executed on a DCS with an arbitrary number of servers. Second, the scalability (in the number of servers) of the hybrid state-space model for the application service time is studied and a linearly scalable analytical approximation of the model is presented.

The characterization of the stochastic service time of an application has been employed to calculate three performance metrics of great interest to system analysts and designers, namely, the average service time of an application, quality-of-service (QoS) guarantees in executing an application, and the service reliability in executing an application. Moreover, in order to account for the efficient execution of parallel applications, our

characterizations have been parameterized by a dynamic task reallocation (DTR) action that is synchronously executed by the system servers. This parameterization, in conjunction with a set of DTR policies, yields a flexible mathematical framework for systematically devising DTR strategies in heterogeneous DCSs such as grid computing systems [3], distributed pattern-searches in DNA databases [30], and reliable distributed computing in P2P networks and donation grids [31]. In this paper, we will exploit the developed framework to optimize performance and reliability in a distributed computing testbed. Our results indicate that when communication delays are severe compared to the average service time of the servers, Markovian approximations not only result in prediction inaccuracies of the performance and reliability metrics, but also introduce significant errors in the prediction of the number of tasks that need to be reallocated among the servers as DTR is performed. Finally, the mathematical framework developed here can be used also to characterize metrics and devise allocation policies in parallel computing systems. This is illustrated here by assessing the performance of a parallel application executed on a 12-node production cluster.

This paper is organized as follows: in Section 2, we state formally the problem tackled in this paper. In Section 3, we build the novel age-dependent renewal theory for the random service time of applications in a DCS, describe the scalability issues and derive analytical approximations for the service time of an application. In Section 4, we compare our age-dependent approach with our previous Markovian renewal-based solution with emphasis on DTR policies for minimal average service time, maximal QoS and maximal reliability. Our conclusions are given in Section 5.

SECTION 2 Problem Description

Consider the class of parallel applications that has no data-dependence constraints between operations. Due to this data independence, such applications can be arbitrarily partitioned into an integer number, say, M, of indivisible and independent tasks. After this partitioning process, the M tasks can be allocated onto the heterogeneous computing servers of a DCS for their parallel execution. Suppose that the DCS is composed of n servers, and let us suppose that, at $t = 0$, an offline scheduler has allocated m_j tasks in the queue of the jth server, where $M = \sum_{j=1}^{n} m_j$ and m_j is a nonnegative integer. Here, the heterogeneous capabilities of the servers have been assumed as of the processor-consistent type [32], [33], that is, the random time taken by any server to process any task depends only upon the service time of the server executing the task.

In order to efficiently parallelize the execution of applications on the DCS, servers are allowed to exchange their tasks by synchronously executing, at a prescribed time $t = t_b$, a DTR policy. We have denoted by $l_{ij}(t_b)$ the number of tasks to be reallocated from the ith to the jth server at time tb. For simplicity of notation, we may drop the explicit dependence on tb whenever there is no confusion and write l_{ij}. By arranging the l_{ij} quantities in matrix form, the DTR policy is denoted as $\mathbf{L} = (l_{ij})_{n \times n}$. To model the limitations of practical communication networks utilized by DCSs to transfer data among the servers, we shall assume hereafter that the exchange of any task, or any group of tasks, among any pair of servers experiences a stochastic communication delay. Such stochastic delay depends upon both the number of tasks exchanged and network parameters such as the heterogeneous end-to-end propagation time.

Additionally, we will assume that computing servers can fail permanently at any random instant following the socalled crash-stop failure model [34], [35]. This failure model has the practical implication that avoids the use of complicated software and/or hardware replication systems for recovering tasks from failed server. Consequently, the application cannot be successfully serviced by the system if at least one task remains unprocessed at a failed server. Also, it has been further assumed that upon failure a server broadcasts to the network a small fixed-sized failure-notice (FN) message. Such type of messages are required by practical DCSs to detect and isolate failed servers. Of course, FN messages faces random end-to-end transfer delays, which

depends solely on the heterogeneous end-to-end propagation time of each communication link. Finally, we will also assume that servers employ any type of reliable message-passing protocol, such as MPI, P4, and PVM. These protocols guarantee that tasks are not discarded by the network in situations like the failure of a server while exchanging tasks with other nodes.

For mathematical tractability, Assumptions A1 and A2 (stated below) are imposed on the random times driving the events occurring in the DCS.

2 Assumption A1
For any $j \neq k$, the following times are random and their probability distribution functions (pdfs) are known.

1. W_{ki}: *the service time of the ith task at the kth server, with pdf $f_{W_{ki}}(x)$*
2. Y_k: *the failure time of the kth server, with pdf $f_{Y_k}(x)$.*
3. X_{jk}: *the transfer time of an FN message sent from the jth to the kth server, with pdf $f_{X_{jk}}(x)$*
4. Z_{ik}: *the transfer time of l_{ik} tasks from the ith to the kth server, with pdf. $f_{zk}(x)$.*

2 Assumption A2
. All the random times listed in Assumption A1 are mutually independent.

In this paper, we consider three key performance and reliability metrics: the average service time of an application, the QoS and the service reliability in executing an application [2], [3], [16], [22]. While Precise Mathematical Definitions of These Metrics Are Presented in Section 3.3, At This Point We Will Briefly Introduce Them and Describe Their Scope of Applicability. the Average Service Time Is Critical To Assess the Speedup in the Runtime of Applications When Executed in Parallel on a DCS. the Average Service Time Is a Reasonable Metric (i.e., It Takes a Finite value) Only in Settings Where Servers Are Completely Reliable or in Settings Where Servers Are Allowed To Recover After a Failure. the Qos, Which Is Defined as the Probability of Executing An Application By a Prescribed Time Deadline, Is a Reasonable Metric in Settings Where Server Nodes May or May Not Fail. the Qos Metric Is of Interest To System Users and Analysts, Specially in Real-Time or in Time Constrained Applications. Finally, the Service Reliability, Which Is Defined as the Probability of Successfully Executing An Entire Application, Is An Important Metric for Assessing the Dependability of Applications Executed on Dcss, and It Is a Reasonable Metric Only When Servers Can Fail Without Recovery And/or in Settings Where Applications Cannot Continue Their Execution After a failure.

The Upcoming Section Is Dedicated To Developing the Mathematics Required for Characterizing the Above-Referenced Three Metrics analytically.

SECTION 3 Age-Dependent Queuing Theory
In [16] and [21], we presented discrete state-space models for the random time taken by the DCS to execute an application in a setting where all the stochastic timed events comprising task-execution times, communication arrival times, load-transfer times, and server failure times all follow exponential distributions. The exponential-distribution causes the dynamical queue to be Markovian, which aids the mathematical analysis greatly. However, when the distributions of these random events follow alternative probability distributions, memory is introduced into the system dynamics and the Markovian property of the underlying stochastic process is destroyed. Unfortunately, the departure from the Markovian property makes the analytical tractability of the solution far more challenging than that for the Markovian case. The key theoretical contribution of this paper is the introduction of a mathematical trick that converts the non-Markovian stochastic distributed queuing process

to that which is Markovian, albeit in a generalized sense (and with some added overhead, of course), thereby making the analysis tractable as in the Markovian case.

In order to retain the Markovian property of the process in a generalized sense, the memory of all the nonexponential distributions will be tracked by means of real-valued age variables. These age variables augment the discrete state-space models presented in [16] and [21] yielding a hybrid discrete and continuous state-space representation for the underlying stochastic process governing dynamics of the DCS.

3.1 Supplementary Age Variables

We begin by illustrating how the concept of age variables is exploited in this paper. Suppose that T is random variable representing some random time. Loosely speaking, if it is known that a units of time have elapsed for the random time T, then the remaining random time $T_a = T - a$ can be introduced as the replacement of the random time T measured from a, and also, one can think of T_a as the *aged* version of T with age a Proceeding formally, the age parameter, a, associated with the random time T is defined as the nonnegative, real-valued quantity that defines the random variable, $T_a = T - a$, on the restriction of the sample space to the event $A = \{T \geq a\}$. The random time T_a has a pdf equal to the *conditional* pdf of T given that A has occurred; namely, $f_{T_a}(t; a) = f_{T|A}(t|a)$. It must be noted that if T is exponentially distributed, then the pdfs of T and T_a are identical due to the memoryless property of the exponential distribution. Here, we exploit the relationship between a random variable and its aged version as follows: as soon as a random time T is triggered by some event, its associated age variable is set to zero, and as time elapses, the age variable increases its value to track the elapsed time of T and adjusts accordingly the pdf of T to show the effect of the elapsed time on its likelihood. By convention, if a random time has not been triggered by an event, then the age variable associated with it is set to infinity.

3.2 Age-Dependent Configuration of a DCS

Following the ideas presented in [16] and [21] for a Markovian setting, the configuration of an n-server DCS can be partially described by means of one state vector and two state matrices. First, let m be an n-dimensional column vector describing the number of unserved tasks in the DCS, where the ith component of $\mathbf{m}$ specifies the number of tasks queued at the ith server. Second, let $\mathbf{F}$ be an n-by-n binary matrix describing the failed *("O")* or functioning *("1")* state of each server, where the ijth element describes the working or failed state of the jth server as perceived by the ith server. Third, let C be an $n- by - n$ matrix specifying the number of tasks in transit over the network; the ijth element of C is an nonnegative integer determining the number of tasks being transferred from the ith to the jth server.

Note that the vector $\mathbf{m}$ and the matrices $\mathbf{F}$ and C have finite dimensions and take values on the discrete sets $\Omega_1 = \{0,1,\dots,M\}^n, \Omega_2 = \{0,1\}^n$ and $\Omega_3 = \{0,1,\dots,M\}^n$ respectively. For $\Omega = \Omega_1 \times \Omega_2 \times \Omega_3$, we can define any one-to-one mapping $h: \Omega \rightarrow \mathcal{I}$ such that, for each possible value of the concatenated matrix $(\mathbf{m}, \mathbf{F}, \mathbf{C})in\Omega$, $h(\mathbf{m}, \mathbf{F}, \mathbf{C})$ assigns a positive integer in the index set $\mathcal{I} = \{1,2,\dots,\kappa\}$, where κ is the cardinality of Ω.

In a non-Markovian setting, the vector m and the matrices $\mathbf{F}$ and $\mathbf{C}$ must be supplemented with enough age variables in order to completely describe the configuration of the DCS. To do so, let a_{M_i} and a_{F_i} be age variables associated with the *first* stochastic service time of a task at the ith server and the *first* random failure time of the ith server, respectively, with $i = 1,\dots,n$. Also, let $x_{F_{ij}}$ be the age variable associated with the *first* random transfer time of an FN packet from the ith to the jth server, with $i, j = 1,\dots,n, i \neq j$. We can arrange all these age variables in the column vector $\mathbf{a}_M$ and the *n-by-n* matrix aF. The $\mathbf{a}_M$ vector contains the α_{M_i} age variables and the $\mathbf{a}_F$ matrix contains both the a_{F_i} variables (at the diagonal of the matrix) and the $a_{F_{ij}}$ variables (at the off-diagonal positions). Similarly, let $a_{C_{ik}}$ be the age variable associated with the random transfer of l_{ik} tasks

from the ith to the kth server. We can also arrange these age variables in matrix form to obtain $\mathbf{a}_C$, whose ikth component is $a_{C_{ik}}$.

We define the *system-age* as the concatenated matrix $\mathbf{a} \triangleq (\mathbf{a}_M, \mathbf{a}_F, \mathbf{a}_C)$ Further, for a given time t we define the *age-dependent system-state* matrix as the concatenated matrix $\mathbf{S}(t) \triangleq (\mathbf{m}(t), \mathbf{F}(t), \mathbf{C}(t), \mathbf{a}(t))$, which describes completely the state of an n-server DCS. (Note that in a Markovian setting the memoryless property of the exponential distribution makes the system-age matrix unnecessary; therefore, the system-state matrix reduces to $\mathbf{S}(t) = (\mathbf{m}(t), \mathbf{F}(t), \mathbf{C}(t))$ as in [16].) With this, we can introduce the stochastic process $\{\mathbf{S}(t), t \geq 0\}$ characterizing the stochastic dynamics of the DCS.

3.3 Mathematical Definition of the Metrics
With the above preliminaries, we can mathematically define the service time of an application as well as the performance metrics described earlier. The *application service time* is defined as the random time taken by the DCS to execute the entire application when servers perform a synchronous DTR action at time t=tb and the initial system configuration is as specified by $\mathbf{S}(0) \equiv (\mathbf{m}_0, \mathbf{F}_0, \mathbf{C}_0, \mathbf{a}_{M_0}, \mathbf{a}_{F_0}, \mathbf{a}_{C_0})$. More precisely,

$$T_{\ell_0}(t_b, \mathbf{a}_0) \triangleq inf\{t > 0: \mathbf{m}(t) = 0 \text{and} \mathbf{C}(t) = 0\}, (1)$$

where $\ell_0 = h(\mathbf{m}_0, \mathbf{F}_0, \mathbf{C}_0)$ and $\mathbf{a}_0 = (\mathbf{a}_{M_0}, \mathbf{a}_{F_0}, \mathbf{a}_{C_0})$Note that since servers can fail permanently with nonzero probability, the application service-time is infinite when at least one task remains queued at a server that has already failed. Note also that in the special case where servers are completely reliable, the application service time is finite with probability one.

The *average service time of an application*, denoted as $\overline{T}_{\ell_0}(t_b, \mathbf{a}_0)$ is defined as the expected value of the random application service time; $\overline{T}_{\ell_0}(t_b, \mathbf{a}_0) \triangleq E[T_{ell_0}(t_b, \mathbf{a}_0)]$. This metric is defined in the special case of a DCS with completely reliable servers; that is, when $Y_k = \infty$ almost surely for all k. The *service reliability* is defined as the probability that the application can be entirely executed by the system, that is $R_{\ell_0}(t_b, a_0) \triangleq P\{T_{\ell_0}(t_b, \mathbf{a}_0) < \infty\}$. Finally, the *QoS in executing an application* is the probability that the application can be entirely executed by the user-specified due time T_M; that is $Q_{\ell_0}(t_b, T_M, \mathbf{a}_0) \triangleq P\{T_{\ell_0}(t_b, \mathbf{a}_0) < T_M\}$Note that the service reliability is a special case of the QoS for which the due time of the application is infinite.

In Section 3.4, Lemma 1 and Theorem 1 characterize the average service time in the form of recursively (in the discrete variable ℓ) coupled integral equations, and Lemma 2and Theorem 2 do the same for QoS. In addition, corollaries to Lemma 2 and Theorem 2provide a characterization for the service reliability. Furthermore, models for the approximations of these performance and reliability metrics are also provided.

3.4 Regeneration-Based Characterization of the Performance Metrics
Consider the process $\{\mathbf{S}(t), t \geq 0\}$ and suppose that at time $t = t_0$ the system configuration is as specified by $\mathbf{S} = (\mathbf{m}, \mathbf{F}, \mathbf{C}, \mathbf{a})$We define the *age-dependent regeneration time*, denoted by $\tau_{\mathbf{a}}$, as the minimum of the following four random variables: the time to the *first* task service by any server, the time to the *first* occurrence of failure at any server, the time to the *first* arrival of an FN packet at any server, or the time to the *first* arrival of a group of tasks at any server. Given the system-age a, for $t \geq t_0$, we can replace the random times listed in Assumption A1 by their aged versions, thereby we can define the age-dependent regeneration time as

$$\tau_{\mathbf{a}} \triangleq min(\min_k W_{k1}, \min_k Y_k, \min_{j \neq k} X_{jk}, \min_{k,i} Z_{ik}), (2)$$

where the subscript a emphasizes the dependency of the regeneration time on all the age variables associated with the nonexponential random times. The upcoming example illustrates how the age-dependent regeneration time and system-state matrix yield a recursive characterization for the application service time.

Suppose that the first event occurring in the DCS happens to be the execution of a task at the ith server at $t = s$, for $t_0 < t$. The occurrence of this event implies that all the random times governing the DCS have aged by s units of time (in addition to the ages specified in a) and there is one less task queued at the ith server; all the other dynamics remain unchanged. Thus, the occurrence of the event $\{\tau_{\mathbf{a}} =, s, \tau_{\mathbf{a}} = W_{i1}\}$ gives birth to a new (regeneration) DCS at $t = s$, represented by $\{\mathbf{S}(t), t \geq s\}$ that is statistically identical to the original process while having a new initial configuration $\mathbf{S}' = (\mathbf{m}'\mathbf{F}', \mathbf{C}', \mathbf{a}')$ resulting from the regeneration event $\{\tau_{\mathbf{a}} = s, \tau_{\mathbf{a}} = W_{i1}\}$ More precisely, the new initial system configuration is as follows: $\mathbf{m}'$ is identical to $\mathbf{m}$ but with one unit less at its ith element, $\mathbf{F} = \mathbf{F}, \mathbf{C} = \mathbf{C}$, and the new system-age matrix is $\mathbf{a} = \mathbf{a} + s$ with the ith component of $\mathbf{a}'_M$ set to zero if at least one task remains queued at the ith server. Similar transformations on the initial configuration are observed when the regeneration event is any of the other possible events: the failure of the i th th server, the arrival of an FN packet from server j to server k, or the arrival of the ith group of tasks to the kth server.

Before stating our main results, we introduce some useful definitions. Let us define the term $G_X(\alpha) \triangleq P\{X = \tau_a | \tau_a = \alpha\} f_{\tau_a}(\alpha)$ where X i s any of the random times listed in Assumption 1, $f_{\tau_a}(\alpha)$ is the pdf of the age-dependent regeneration time τ_a and $P\{X = \tau_a | \tau_a = \alpha\}$ is the probability that the regeneration event is $\{\tau_{\mathbf{a}} = X\}$ conditional on the event $\{\tau_a = \alpha\}$. Both, the and the conditional probability $P\{T^{(j)} = \tau_a | \tau_a = \alpha\}$ can be computed, either analytically or numerically, using

$$
\begin{aligned}
f_{\tau_{\mathbf{a}}}(t) &= \sum_{j \in \mathcal{J}} f_{T^{(j)}|A_j}(t|a_j) \prod_{k \in \mathcal{J}, k \neq j} \left(1 - F_{T^{(k)}|A_k}(t|a_k)\right), \\
P\{T^{(j)} &= \tau_a | \tau_{\mathbf{a}} = \alpha\} = P\{T^{(j)} < T^{(1)}, \dots, T^{(j)} < T^{(j-1)}, \\
&\quad T^{(j)} < T^{(j+1)}, \dots, T^{(j)} < T^{(p)} | \tau_{\mathbf{a}} = \alpha\},
\end{aligned}
$$

where $A_j = \{T_j \geq a_j\}$ and $F_{T^{(j)}|A_j}(t|a_j) = F_{T^{(j)}}(t; a_j)$ [correspondingly, $f_{T^{(j)}|A_j}(t|a_j) = f_{T^{(j)}}(t; a_j)]]$ is the conditional distribution [correspondingly, the conditional density] that defines the cumulative distribution function [correspondingly, the pdf] of the aged version of the random time $T^{(j)}$ whose age parameter is a_j. (For the sake of notation, we have indexed all the random times listed in Assumption A1 as well as their associated age parameters using an index set $\mathcal{J}$ and denoted these random times as $T^{(j}$ with $j \in \mathcal{J}$.).) Finally, let us introduce the vector $\mathbf{v}^{(i)}$ (correspondingly, the matrix $\mathbf{A}^{(ij)}$) which is identical to the vector $\mathbf{v}$ (correspondingly, the matrix $\mathbf{A}$) but with its ith (correspondingly, ijth) component set to zero and the vector δ_i denotes an $n-$ dimensional vector with all its entries equal to zero except that its i th element is equal to one.

When a DTR action is performed by the n servers at time $t_b = 0$, and the initial system configuration is arbitrarily specified by the vectors m and a_M and the matrices $\mathbf{E}, \mathbf{C}, \mathbf{a}_F$, and $\mathbf{a}_C$, the three performance and reliability metrics analyzed here are characterized by the results in Lemmas 1, 2 and Corollary 1, which are termed as initial conditions.

Lemma 1

Lemma 1 Initial Condition for the Age-Dependent Characterization of the Average Execution Time

For any $\ell \in \mathcal{J}$, the average application execution time satisfies the system of recursive, coupled integral equations

$$
\begin{aligned}
\overline{T}_\ell(0, \mathbf{a}_M, \mathbf{a}_F, \mathbf{a}_C) &= E[\tau_{\mathbf{a}}] \\
&+ \int_0^\infty [\textstyle\sum_{i=1}^n G_{W_{i1}}(\alpha) \times \overline{T}_{\ell_i}(0, (\mathbf{a}_M + \alpha)^{(i)}, \mathbf{a}_F + \alpha, \mathbf{a}_C + \alpha) \qquad (3) \\
&+ \textstyle\sum_{i=1}^n \sum_{j=1, j \neq i}^n G_{Z_{ji}}(\alpha) \times \overline{T}_{\ell'_{ji}}(0, \mathbf{a}_M + \alpha, \mathbf{a}_F + \alpha, \mathbf{a}_C + \alpha)] d\alpha,
\end{aligned}
$$

where $\ell = h(\mathbf{m}, \mathbf{F}, \mathbf{C}), \ell_i = h(\mathbf{m} - \delta_i, \mathbf{F}, \mathbf{C})$ and $\ell'_{ji} = h(\mathbf{m} + c_{ji}\delta_i, \mathbf{F}, \mathbf{C}^{(ji)})$

Remark

The recursion in Lemma 1 is in ℓ, which depends upon the discrete variables $\mathbf{m}, \mathbf{F}$, and $\mathbf{C}$. Throughout the remaining lemmas and theorems, recursions are also carried out in these discrete variables.

Lemma 2

Lemma 2 Initial Condition for the Age-Dependent Characterization of the QoS

For any $\ell \in \mathcal{I}$, the QoS in executing an application by the time deadline T_M satisfies the system of recursive, coupled integral equations

$$Q_\ell(0, T_M, \mathbf{a}_M, \mathbf{a}_F, \mathbf{a}_C) =$$
$$\int_0^\infty [\sum_{i=1}^n G_{W_{i1}}(\alpha) Q_{\ell_i}(0, T_M - \alpha, (\mathbf{a}_M + \alpha)^{(i)}, \mathbf{a}_F + \alpha, \mathbf{a}_C + \alpha)$$
$$+ \sum_{i=1}^n \sum_{j=1,j\neq i}^n G_{Z_{ji}}(\alpha) Q_{\ell_i'}(0, T_M - \alpha, \mathbf{a}_M + \alpha, \mathbf{a}_F + \alpha, \mathbf{a}_C + \alpha) \quad (4)$$
$$+ \sum_{i=1}^n \sum_{j=1,j\neq i}^n G_{X_{ij}}(\alpha) Q_{\ell_{ij}}, (0, T_M - \alpha, \mathbf{a}_M + \alpha, \mathbf{a}_F + \alpha, \mathbf{a}_C + \alpha)$$
$$+ \sum_{i=1}^n G_{Y_i}(\alpha) Q_{\ell_i'}(0, T_M - \alpha, \mathbf{a}_M + \alpha, (\mathbf{a}_F + \alpha)^{(ii)}, \mathbf{a}_C + \alpha)] d\alpha,$$

where $\ell_i' = h(\mathbf{m}, \mathbf{F}^{(ii)}, \mathbf{C})$ and $\ell_{ij} = h(\mathbf{m}, \mathbf{F}^{(ji)}, \mathbf{C})$.

Proof of Lemmas 1 and 2 are in Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2011.285.

Corollary 1

Corollary 1 Initial Condition for the Age-Dependent Characterization of the Service Reliability

For any $l \in \mathcal{I}$, the service reliability in executing an application satisfies the system of recursive, coupled integral equations

$$R_\ell(0, \mathbf{a}_M, \mathbf{a}_F, \mathbf{a}_C) =$$
$$\int_0^\infty [\sum_{i=1}^n G_{W_{i1}}(\alpha) R_{\ell_i}(0, (\mathbf{a}_M + \alpha)^{(i)}, \mathbf{a}_F + \alpha, \mathbf{a}_C + \alpha)$$
$$+ \sum_{i=1}^n G_{Y_i}(\alpha) R_{\ell_i'}(0, \mathbf{a}_M + \alpha, (\mathbf{a}_F + \alpha)^{(ii)}, \mathbf{a}_C + \alpha) \quad (5)$$
$$+ \sum_{i=1}^n \sum_{j=1,j\neq i}^n G_{Z_{ji}}(\alpha) R_{\ell_{ji}}(0, \mathbf{a}_M + \alpha, \mathbf{a}_F + \alpha, \mathbf{a}_C + \alpha)$$
$$+ \sum_{i=1}^n \sum_{j=1,j\neq i}^n G_{X_{ij}}(\alpha) R_{\ell_{ij}}(0, \mathbf{a}_M + \alpha, \mathbf{a}_F + \alpha, \mathbf{a}_C + \alpha)] d\alpha.$$

When the DTR action is performed by the n servers at any time $\xi > 0$, and the initial system configuration is arbitrarily specified by the vectors m and $\mathbf{a}_M$ and the matrices $\mathbf{F}, 0, \mathbf{a}_F$, and $\mathbf{a}_C$, the dynamical characterization of the performance and reliability metrics is given by the results in Theorems 1, 2 and Corollary 2, which employ the results provided by the initial conditions.

Theorem 1

Theorem 1 Age-Dependent Characterization for the Average Execution Time

For any $\ell \in \mathcal{I}$, the average application execution time satisfies the system of recursive, coupled integral equations in ξ

$$\overline{T}_\ell(\xi, \mathbf{a}_M, \mathbf{a}_F, \mathbf{a}_C)$$
$$= \int_0^\xi \sum_{i=1}^n G_{W_{i1}}(\alpha) \overline{T}_{\ell_i}(\xi - \alpha, (\mathbf{a}_M + \alpha)^{(i)} \mathbf{a}_F + \alpha, \mathbf{a}_C + \alpha) d\alpha \quad (6)$$
$$+ \int_0^\xi \alpha f_{\tau_a}(\alpha) d\alpha + (1 - F_{\tau_a}(\xi)) \times (\xi + \overline{T}_\ell(0, \mathbf{a}_M, \mathbf{a}_F, \mathbf{a}_C)),$$

where $\ell = h(\mathbf{m}, \mathbf{F}, 0)$, $\ell_i = h(\mathbf{m} - \delta_i, \mathbf{F}, 0)$ and $\overline{T}_\ell(0, \mathbf{a}_M, aF, aC)$ is the initial condition related to the ℓth integral equation.

Theorem 2

Theorem 2 Age-Dependent Characterization for the QoS

For any $\ell \in \mathcal{I}$, the QoS in executing an application by a predefined time-deadline T_M satisfies the system of recursive, coupled integral equations in ξ

$$
\begin{aligned}
&Q_\ell(\xi, T_M, \mathbf{a}_M, \mathbf{a}_F, \mathbf{a}_C) = \\
&\int_0^\xi [\sum_{i=1}^n \ G_{W_{i1}}(\alpha) Q_{\ell_i}(\xi - \alpha, T_M - \alpha, (\mathbf{a}_M + \alpha)^{(i)}, \\
&\mathbf{a}_F + \alpha, \mathbf{a}_C + \alpha) + \sum_{i=1}^n \ G_{Y_i}(\alpha) \times Q_{\ell_i'}(\xi - \alpha, T_M - \alpha, \\
&\mathbf{a}_M + \alpha, (\mathbf{a}_F + \alpha)^{(ii)}, \mathbf{a}_C + \alpha) + \sum_{i=1}^n \ \sum_{j=1, j \neq i}^n \ Q_{\ell_{ij}}(\xi - \alpha, \\
&T_M - \alpha, \mathbf{a}_M + \alpha, \mathbf{a}_F + \alpha, \mathbf{a}_C + \alpha) \times G_{X_{ij}}(\alpha)] d\alpha \\
&+ Q_\ell(0, T_M - \xi, \mathbf{a}_M, \mathbf{a}_F, \mathbf{a}_C)(1 - F_{\tau_a}(\xi)),
\end{aligned}
\tag{7}
$$

where $\ell_i' = h(\mathbf{m}, \mathbf{F}^{(ii)}, 0)$, $\ell_{ij} = h(\mathbf{m}, \mathbf{F}^{(ji)}, 0)$, $T_M - \xi, \mathbf{a}_M, \mathbf{a}_F, \mathbf{a}_C)$ is the initial condition related to the lth integral equation.

Proof of Theorems 1 and 2 are deferred to Appendix B, available in the online supplemental material.

Corollary 2

Corollary 2 Age-Dependent Characterization of the Service Reliability

For any $\ell \in \mathcal{I}$, the service reliability in executing an application satisfies the system of recursive, coupled integral equations in ξ

$$
\begin{aligned}
&R_\ell(\xi, \mathbf{a}_M, \mathbf{a}_F, \mathbf{a}_C) = \\
&\int_0^\xi [\sum_{i=1}^n \ G_{W_{i1}}(\alpha) R_{\ell_i}(\xi - \alpha, (\mathbf{a}_M + \alpha)^{(i)}, \mathbf{a}_F + \alpha, \mathbf{a}_C + \alpha) \\
&+ \sum_{i=1}^n \ G_{Y_i}(\alpha) R_{\ell_i'}(\xi - \alpha, \mathbf{a}_M + \alpha, (\mathbf{a}_F + \alpha)^{(ii)}, \mathbf{a}_C + \alpha) \\
&+ \sum_{i=1}^n \ \sum_{j=1, j \neq i}^n \ G_{X_{ij}}(\alpha) R_{\ell_{ij}}(\xi - \alpha, \mathbf{a}_M + \alpha, \mathbf{a}_F + \alpha, \mathbf{a}_C + \alpha)] d\alpha \\
&+ (1 - F_{\tau_a}(\xi)) R_\ell(0, \mathbf{a}_M, \mathbf{a}_F, \mathbf{a}_C).
\end{aligned}
\tag{8}
$$

To calculate the performance and reliability metrics the initial task allocation, the t_b instant and a specific DTR policy must be defined. To compute the QoS the value of T_M must also be specified. Here it is assumed that, initially, all the random times have age zero; we will be interested in computing the values $\overline{T}_{\ell_0}(t_b, 0)$, $Q_{\ell_0}(t_b, T_M, 0)$, and $R_{\ell_0}(t_b, 0)$. It must be noted that, for any ξ and ℓ, the values taken by the metrics are known in the following particular cases: 1) if the application is composed of a single task queued at the ith server then $\overline{T}_\ell(\xi, a) = E[W_{i1}]$; 2) if there are no tasks to be served in the DCS then $\overline{T}_l(\xi, \mathbf{a}) = 0$ while $Q_\ell(\xi, T_M, \mathbf{a}) = 1$ and $R_\ell(\xi, \mathbf{a}) = 1$; 3) if a server fails and contains at least one task in its queue then $\overline{T}_\ell(\xi, \mathbf{a})$ is infinite and $Q_\ell(\xi, T_M, \mathbf{a}) = 0$ and $R_\ell(\xi, \mathbf{a}) = 0$.

The system of integral equations are solved numerically using simple iterations and by means of an algorithm that dynamically constructs the mapping h and creates a sorted list of all the equations and symbols to solve for. Next, values are computed by visiting the list in reverse order thereby exploiting the recursive nature of the system of equations. The integral equations have been solved using the built-in functions for adaptive numerical integration over infinite intervals provided by The GNU Scientific Library [36].

Now, we compare our age-dependent model for the performance and reliability of a DCS with the general models, based upon stochastic Petri Nets, proposed by German [26], and Telek and Horváth, [28]. The main difference between the non-Markovian models in [26] and [28] and our work is that the former models aim to develop general tools for evaluating complex stochastic models. Moreover, the authors are interested in extending the class of stochastic Petri Nets to a class of non-Markovian stochastic Petri Nets, while in our work we are interested in modeling performance and reliability, for the specific case, of a non-Markovian DCS whose communication links impose nonnegligible delays and whose computing servers are unreliable. A list of some of the theoretical similarities between the works is the following. First, for all the models the state-space representation of the queues is finite and must be specified by analyzing the dynamics of the DCS. Second, in [26] and [28] state transitions are tracked by the firing times of each random time modeled by the Petri Net, while here the state transitions are tracked by a single time: the regeneration time. Third, in all the models the memory of the nonexponential distributions has been accounted for using real-valued age variables. Fourth, the initial marking, the marking process and the reachability set in [26] and [28] are conceptually similar to our initial system-age matrix, our system-age matrix, and the set of valid events occurring after the stochastic regeneration takes place. Fifth, the execution policies of the general transitions in [26] and [28] are akin to the aging rule defined in Section 3.1.

3.5 Linearly Scalable Approximation

Computing the metrics according to the recipes described in Section 3.4 is computationally expensive for large systems. First, since the dimension of the age-dependent state vector S is $4n^2 + 2n$, the age-dependent state-space model for the execution time scales polynomially in the number of servers. Second, for a fixed number of servers in a DCS, since both FN packets and tasks exchanged among the servers may arrive at any random instant, every recursion generated from Theorems 1 and 2 must consider a combinatorial number of values for the age-dependent state-vector, and as a consequence, the number of recursive equations that must be constructed to calculate the metrics scales exponentially in the number of messages exchanged.

To circumvent the scalability problem of our exact model, we have considered two simplifying assumptions to construct an approximated characterization for the service time, which in turn, yields an approximation to the metrics characterized in Theorems 1and 2. The first assumption neglects the random transfer time of FN packets on the network. This approximation reduces the matrix $\mathbf{F}$ to a binary vector that we have denoted as $\mathbf{f}$. The second assumption considers that all the tasks in transit to a particular server arrive simultaneously to it. More precisely, we assume that the $l_k = \sum_j l_{jk}$ tasks being transferred from other servers to the kth server arrive as a single batch of tasks to server k at a certain random time denoted as $\widetilde{Z}_k$. This approximation reduces the matrix $\mathbf{C}$ to a vector that we have denoted as $\mathbf{C}$. Note that these approximations reduce the dimension of the age-dependent state vector to $6n$, yielding a model with linear scalability in the number of servers.

Empowered by these approximations, we introduce the *approximated age-dependent random execution time of an application*, $T_\ell(t_b, \widetilde{\mathbf{a}}_M, \widetilde{\mathbf{a}}_F, \widetilde{\mathbf{a}}_C)$, when the system configuration is as specified by the reduced age-dependent state vector $\mathbf{S} = (\mathbf{m}, \mathbf{f}, \mathbf{c}, \widetilde{\mathbf{a}}_M, \widetilde{\mathbf{a}}_F, \widetilde{\mathbf{a}}_C))$ where $\widetilde{\mathbf{a}}_M; \widetilde{\mathbf{a}}_F$, and $\widetilde{\mathbf{a}}_C$ are vectors containing the age variables associated with the vectors m, f, and c, respectively, $\ell = h(\mathbf{m}, \mathbf{f}, \mathbf{c})$ and h is a mapping akin to h.

We can apply the principles presented in Section 3.4 *mutatis mutandis* to obtain regenerative age-dependent equations for the approximated random service time. To this end, we suppose that $\widetilde{Z}_i$ satisfies Assumption A2, that the pdf of these random times are known and denoted as $f_{\widetilde{Z}_i}(t)$. For the sake of the space, in Theorems 3 and 4, we present characterizations solely for the average approximated service time and the QoS.

Theorem 3

Consider an n-server DCS whose servers perform a synchronous DTR action at time $\xi \geq 0$. For any $\ell \in \mathcal{I}$, the average approximated application execution time satisfies the system of recursive, coupled integral equations in ξ

$$\overline{T}_\ell(\xi, \tilde{\mathbf{a}}) = \int_0^\xi [\alpha f_{\tau_\mathbf{a}}(\alpha) + \textstyle\sum_{i=1}^n \; G_{W_{i1}}(\alpha)\overline{T}_{\ell_i}(\xi - \alpha, (\tilde{\mathbf{a}}_M + \alpha)^{(i)},$$
$$\tilde{\mathbf{a}}_F + \alpha, \tilde{\mathbf{a}}_C + \alpha)]d\alpha + (1 - F_{\tau_\mathbf{a}}(\xi))(\xi + \overline{T}_\ell(0, \tilde{\mathbf{a}})), \quad (9)$$

where the vectors $\mathbf{m}, \mathbf{f}, 0, \tilde{\mathbf{a}}_M, \tilde{\mathbf{a}}_F$, and $\tilde{\mathbf{a}}_C$ denote an arbitrarily specified initial system configuration, $\tilde{\mathbf{a}} = (\tilde{\mathbf{a}}_M, \tilde{\mathbf{a}}_F, \tilde{\mathbf{a}}_C), \ell = h(\mathbf{m}, \mathbf{f}, 0), \ell_i = h(\mathbf{m} - \delta_i, \mathbf{f}, 0)$, and $\overline{T}_\ell(0, \tilde{\mathbf{a}}_M, \tilde{\mathbf{a}}_F, \tilde{\mathbf{a}}_C)$ is the initial condition related to the lth integral equation. Moreover, these initial conditions satisfy the system of recursive, coupled integral equations

$$\overline{T}_{\ell'}(0, \tilde{\mathbf{a}}) = \int_0^\infty \; \textstyle\sum_{i=1}^n \; G_{W_{i1}}(\alpha)\overline{T}_{\ell_i}(0, (\tilde{\mathbf{a}}_M + \alpha)^{(i)}, \tilde{\mathbf{a}}_F + \alpha, \tilde{\mathbf{a}}_C + \alpha)$$
$$+ \textstyle\sum_{i=1}^n \; G_{\tilde{Z}_i}(\alpha)\overline{T}_{\ell'_i}(0, \tilde{\mathbf{a}}_M + \alpha, \tilde{\mathbf{a}}_F + \alpha, \tilde{\mathbf{a}}_C + \alpha)d\alpha + \mathsf{E}[\tau_\mathbf{a}], \quad (10)$$

where the vectors $\mathbf{m}, \mathbf{f}, \mathbf{c}, \tilde{\mathbf{a}}_M, \tilde{\mathbf{a}}_F$, and $\tilde{\mathbf{a}}_C$ denote an arbitrarily specified initial system configuration, $\ell' = \tilde{h}(\mathbf{m}, \mathbf{f}, \mathbf{c})$, and $\ell'_i = \tilde{h}(\mathbf{m} + l_i\delta_i, \mathbf{f}. \mathbf{c}^{(i)})$.

Theorem 4

Consider an n-server DCS whose servers perform a synchronous DTR action at the time $\xi \geq 0$ For any $\ell \in \mathcal{I}$, the approximated QoS in executing an application by the time T_M satisfies the system of recursive, coupled integral equations in ξ

$$Q_\ell(\xi, T_M, \tilde{\mathbf{a}}) = \int_0^\xi \; \textstyle\sum_{i=1}^n \; G_{W_{i1}}(\alpha)Q_{\ell_i}(\xi - \alpha, T_M - \alpha, (\tilde{\mathbf{a}}_M + \alpha)^{(i)},$$
$$\tilde{\mathbf{a}}_F + \alpha, \tilde{\mathbf{a}}_C + \alpha) + \textstyle\sum_{i=1}^n \; G_{Y_i}(\alpha)Q_{\ell'_i}(\xi - \alpha, T_M - \alpha, \tilde{\mathbf{a}}_M + \alpha, \quad 11)$$
$$(\tilde{\mathbf{a}}_F + \alpha)^{(ii)}, \tilde{\mathbf{a}}_C + \alpha)d\alpha + (1 - F_{\tau_\mathbf{a}}(\xi))Q_\ell(0, T_M - \xi, \tilde{\mathbf{a}}),$$

where the vectors $\mathbf{m}, \mathbf{f}, 0, \tilde{\mathbf{a}}_M, \tilde{\mathbf{a}}_F$ and $\tilde{\mathbf{a}}_C$ denote an arbitrarily specified initial system configuration, $\tilde{\mathbf{a}} = (\tilde{\mathbf{a}}_M, \tilde{\mathbf{a}}_F, \tilde{\mathbf{a}}_C), \ell = h(\mathbf{m}, \mathbf{f}, 0), P_i = h(\mathbf{m} - \delta_i, \mathbf{f}, 0), \ell'_i = h(\mathbf{m}, \mathbf{f}^{(i)}, 0)$ and $Q_\ell(0, T_M - \xi, \tilde{\mathbf{a}}_M, \tilde{\mathbf{a}}_F, \tilde{\mathbf{a}}_C)$ is the initial condition related to the ℓth integral equation. Moreover, these initial conditions satisfy the system of recursive, coupled integral equations

$$Q_{\ell'}(0, T'_M, \tilde{\mathbf{a}}) =$$
$$\int_0^\infty [\textstyle\sum_{i=1}^n \; G_{W_{i1}}(\alpha)Q_{\ell_i}(0, T'_M - \alpha, (\tilde{\mathbf{a}}_M + \alpha)^{(i)}, \tilde{\mathbf{a}}_F + \alpha, \tilde{\mathbf{a}}_C + \alpha)$$
$$+ \textstyle\sum_{i=1}^n \; G_{Y_i}(\alpha)Q_{\ell'_i}(0, T'_M - \alpha, \tilde{\mathbf{a}}_M + \alpha, (\tilde{\mathbf{a}}_F + \alpha)^{(ii)}, \tilde{\mathbf{a}}_C + \alpha) \quad (12)$$
$$+ \textstyle\sum_{i=1}^n \; G_{\tilde{Z}_i}(\alpha)Q_{\ell''_i}(0, T'_M - \alpha, \tilde{\mathbf{a}}_M + \alpha, \tilde{\mathbf{a}}_F + \alpha, \tilde{\mathbf{a}}_C + \alpha)]d\alpha,$$

where the vectors $\mathbf{m}, \mathbf{f}, \mathbf{c}, \tilde{\mathbf{a}}_M, \tilde{\mathbf{a}}_F$ and $\tilde{\mathbf{a}}_C$ denote an arbitrarily specified initial systemconfiguration, $T'_M = T_M - \xi_r \ell' =, h(\mathbf{m}, \mathbf{f}, \mathbf{c}), P_i = h(\mathbf{m} - \delta_i, \mathbf{f}, \mathbf{c}), \ell_i = h(\mathbf{m}, \mathbf{f}^{(i)}, c)$ and $\ell_i =, h(\mathbf{m} + l_i\delta_i, \mathbf{f}, \mathbf{c}^{(i)})$.

Proofs of Theorems 3 and 4 are omitted as they are simplified versions of proofs of Theorems 1 and 2.

We end this section by commenting on the memory requirements of the state-space representation of our models and on the computational complexity of the Markovian and non-Markovian models. In a Markovian setting [16], [21], the state-space representation for the application execution time is discrete and has a dimension $2n^2 + n$; the Markovian model needs to store $2n^2 + n$ integer numbers. In the non-Markovian case

such state-space has discrete and continuous components; thus, not only $2n^2 + n$ integer numbers must be stored for this model but also $2n^2 + n$ real values. The computational complexity of our models is evaluated by means of the number of equations to be solved in the model. It must be noted that the recursions presented in the lemmas and theorems are carried out only in the discrete variables $\mathbf{m}, \mathbf{F}, \mathbf{f}, \mathbf{C}$ and $\mathbf{c}$. In the case of our exact models, for a given initial workload of M tasks, the total number of equations to be solved depends up on all the possible values taken by the discrete variables $\mathbf{m}, \mathbf{f},$ and $\mathbf{C}$. Since these variables can take at most $M^n, 2^{n^2}$ and M^{n^2} different values, the computational complexity of our exact models can be bounded by $\mathcal{O}(2^{n^2} M^{n^2+n})$. A similar analysis shows that the computational complexity of our approximated models is bounded by $\mathcal{O}(2^{2n} M^n)$. Finally, the runtime complexity of our algorithms can be bounded as follows. Note that every integral in the system of equations is parameterized by system-age matrix, consequently, the calculation of every integral can be considered as an independent calculation. Hence, if we assume that the maximum time taken by the numerical integration of each term is t_I, and the maximum time taken to lookup and retrieve each symbol stored in the sorted list is t_L the runtime complexity of the exact and approximated models are $\mathcal{O}((t_I +, t_L) 2^{n^2} M^{n^2} + n)$ and $\mathcal{O}\left((t_I + t_L) 2^{2n} M^n\right)$ respectively. In summary, the computational and the runtime complexity of our models scales: 1) exponentially in the number of nodes in the system due to the combinatorial nature of the state representation; and 2) polynomially in the number of tasks queued in the system. In Fig. 1, we have illustrate the number of equations to be solved in the Markovian and the non-Markovian cases as a function of the number of servers. Cases labeled as "M fixed" evaluate a situation where an application (composed of $M = 400$) tasks is executed by the DCS using n servers. Cases labeled as "M variable" evaluate a situation when $m_i = 50$ tasks are assigned to each server in the system. This explains why curves merge for $n = 8$ in each case. Fig. 1 also shows that the number of equations generated by the non-Markovian model is about two times the number of states in the Markovian case. For comparison, Fig. 1 also shows the number of equations to be solved by the approximate method (Theorems 3 and 4) in the non-Markovian case as a function of the number of system servers. The figure confirms the large savings in computational complexity (of the model) that the approximation provides.
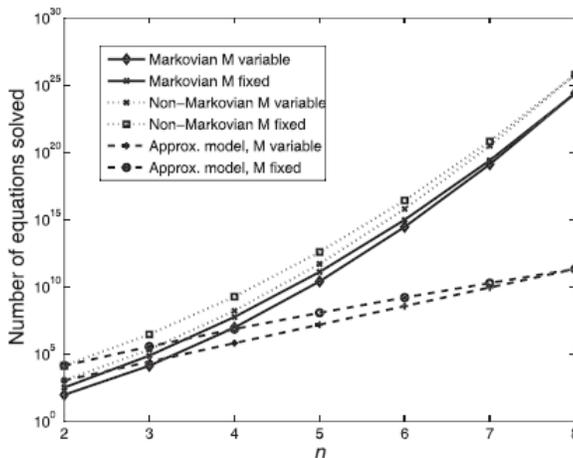

Fig. 1. Number of recursions to be solved as a function of the system servers for both markovian and non-markovian models. For comparison, the approximated models are also shown.

SECTION 4 Results

4.1 Predicting Performance and Reliability in Non-Markovian Heterogeneous DCSs

Now, we compare predictions for the characterized metrics using the non-Markovian models developed here and the Markovian models in [16] and [21]. In the examples considered, we assume that the communication network is homogeneous with three network-delay conditions to be considered: low, moderate, and severe.

Under low network-delay conditions, transferring a task to and processing such task at the fastest server takes, on average, the same time as processing the task at the slowest server. For the moderate and severe network-delay cases, the average transferring and processing times of a task to the fastest server is, respectively, two and five times the service time at the slowest server.

Different stochastic models for the task transfer times have been employed. The Markovian setting is represented by the exponential-distribution delay model. The Pareto 1 and 2 models represent the case where service and transfer times follow Pareto laws with unity and an infinite variance, respectively. The Pareto 3 model represents an experimental case were the delay-statistics in a two-server testbed DCS has been empirically characterized and its parameters have been adjusted, by means of traffic shapers, so that the variance of the transfer times are 5 and the average transfer times satisfy the network delay conditions in every case. For fair comparison, all distributions modeling the same random times have identical mean values. For simplicity, the Markovian and non-Markovian models are compared in a heterogeneous DCSs composed of two servers.

The accuracy of the predictions has been assessed by means of theoretical calculations, and in the case of the Pareto 3 model experimental results are also shown. Figs. 2, 3, and 4 show the service reliability as a function of the number of tasks reallocated from server 1 to 2 for the three network-delay cases under analysis, when an application composed of 150 tasks is executed (initially, $m_1 = 100$ tasks are allocated at server 1 and $m_2 = 50$ at server 2). The mean service time per task is 2 and 1 s for servers 1 and 2, respectively, while the mean failure times are $\lambda_{f_1}^{-1} = 1,000$ and $\lambda_{f_2}^{-1} = 500$. It can be noticed that the Markovian approximations for the service reliability are extremely accurate in the low network-delay condition in all the cases (the maximum relative approximation error is approximately 1 percent) but in the case of the Pareto model with infinite variance, which exhibits an approximation error of about 7 percent. Interestingly, the accuracy of the Markovian approximation is reduced in the case of moderate delays (approximation errors increase up to a 17 percent). In particular, when communication times are large the Markovian approximation becomes inaccurate, with approximation errors up to a 120 percent, as evidenced in Fig. 4.
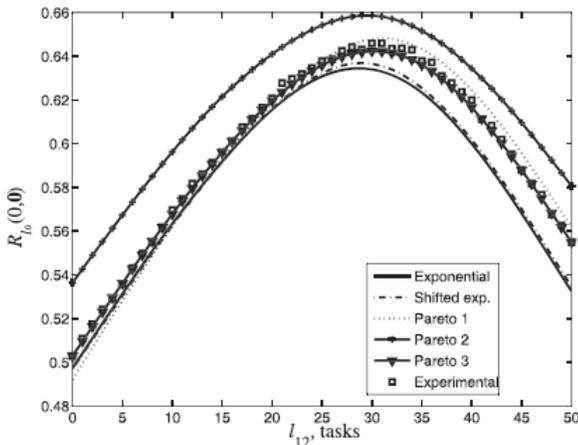

Fig. 2. Service reliability as a function of the number of tasks reallocated from server 1 to 2: low delay case.
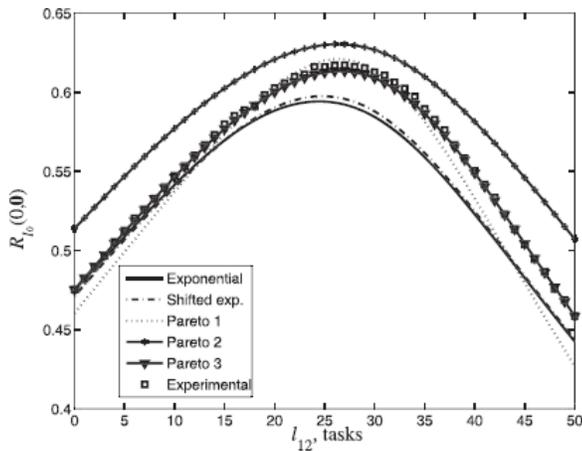
Fig. 3. Service reliability as a function of the number of tasks reallocated from server 1 to 2. Moderate delay case.
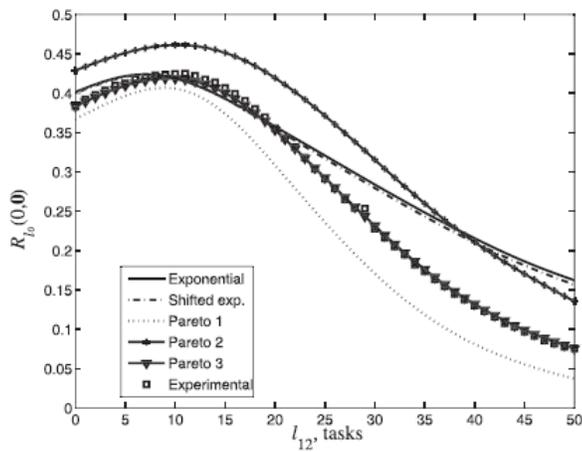

Fig. 4. Service reliability as a function of the number of tasks reallocated from server 1 to 2. Severe delay case.

Figs. 2, 3, and 4 also show the effect of the network-delays on the performance metric. As the mean transfer time increases the service reliability decreases considerably. In addition, to the reliability in Fig. 6, we show the theoretical predictions for the QoS metric when the due time is 140 s, as a function of the number of tasks exchanged. In this figure, the Pareto 1 model is employed to abstract moderate network delays. As expected, the QoS in executing the entire application increases as more tasks are transferred from server 1 to 2. The maximal QoS is only 0.471, which is attained when 33 to 35 tasks are transferred to the second server.
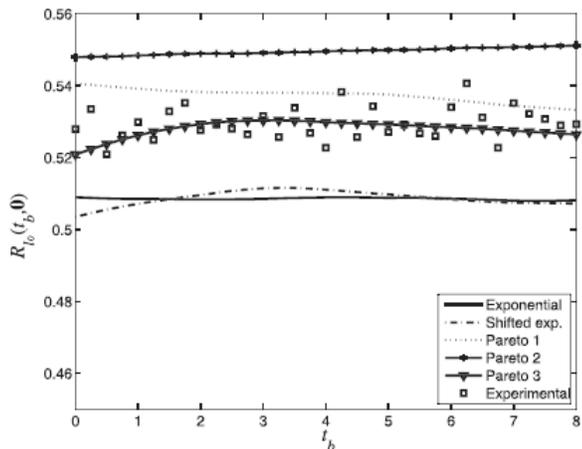
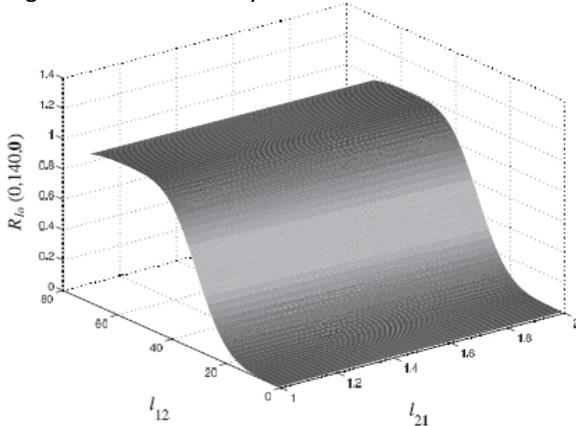Fig. 5. Service reliability as a function of the task reallocation instant in the moderate delay case.

Fig. 6. Qos in executing an application by the due time tm = 140 s as a function of the number of tasks exchanged.

Next, the service reliability of the two-server experimental DCS are calculated, in the moderate network-delay case, for the different event-time stochastic models as a function of the instant to trigger the DTR policy. Fig. 5 shows the theoretical predictions as well as the experimental results obtained for the Pareto 3 model. Note that in this dynamical situation, the Markovian approximations offer, once again, a fairly good approximation of the service reliability. It must be noted that the predictions yielded for the experimental case (Pareto 3 model) are remarkably accurate.

We now turn to evaluating the accuracy of the linear approximation for the random service time using theoretical calculations and Monte-Carlo simulations. Note that the simultaneous arrival of tasks to a server (which is the essence of the approximation) at a random time can be defined in several different ways. For instance, a conservative approximation can be obtained by defining $\tilde{Z}_i$ as the maximum of all the random task-arrival times, that is $\tilde{Z}_k = \max_i Z_{ik}$. A less conservative approximation can be obtained by defining $\tilde{Z}_k$ as any ℓth order statistics, where $\tilde{Z}_k = \min_i Z_{ik}$ corresponds to most aggressive approximation. In the evaluations presented here, the minimum random arrival time as well as the maximum random arrival time approximations are considered.

Results for different initial allocations of an application partitioned in $M = 250$ t asks have been calculated. The DCS considered is composed of five servers whose processing times following Pareto distributions, with average processing times of 1, 2, 3, 4, and 5 seconds for the first, second, third, fourth, and fifth node, respectively. The communication network has been considered to be homogeneous and the network delays are assumed to be moderate. Two different initial allocations for the tasks have been considered: a uniform allocation $m_1 = \cdots = m_5 = 50$ tasks for all i and nonuniform allocation $(m_1, \ldots m_5) = (150,0,0,0,100)$. Note that the execution times are expected to be longer in the imbalanced case of a nonuniform initial allocation because more tasks must be exchanged over the network in order to improve the QoS metric as compared to the uniform initial allocation. This behavior is clearly observed in Figs. 7 and 8, which show the results for the exact and approximated QoS metric. Note that the approximation based on the maximum random arrival time per server introduces less approximation errors than the aggressive approximation based on the minimum arrival time. This behavior is expected because as the task transfer time increases, either by transferring more tasks or by an increased average transfer delay, assuming that all the tasks arrive as a batch at the fastest time does not reflect the behavior of the network thereby impacting more severely on the service reliability as compared to the less aggressive approximation of the maximum transfer time.
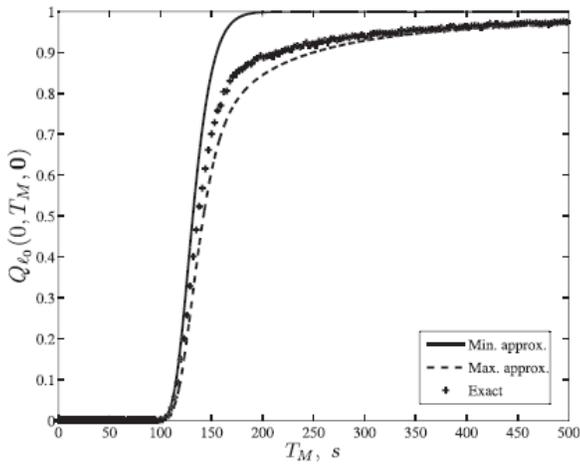
Fig. 7. Exact and approximated qos as a function of the time deadline. Initial task allocation is balanced.
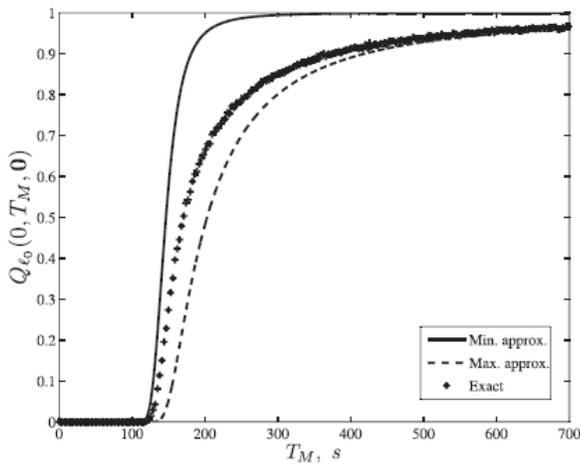

Fig. 8. Exact and approximated qos as a function of the time deadline. Initial task allocation is unbalanced.

Finally, we have employed our model to approximate the QoS in a parallel computing environment. To this end, we have considered a regime where the average transfer delays are 10 percent of the fastest average service time, and we have considered different stochastic models for the service times, thereby approximating a G/G/n queuing system. The application served by the DCS has been partitioned into $M = 250$ tasks, which are assigned uniformly to a DCS composed of five servers. The average service time of the servers are 1, 2, 3, 4, and 5 seconds. The stochastic models evaluated in this example are Pareto with finite ("Pareto 1") and infinite ("Pareto 2") variance and a uniform distribution with a coefficient of variation (standard deviation divided by the mean) of 2 percent, which aims to approximate a deterministic service times. As in the previous comparisons all the stochastic models have the same mean. Fig. 9 compares predictions using the exact stochastic models with the predictions generated by a Markovian approximation ("Exponential") for the QoS metric as a function of the due time. It can be observed from the figure that, once again, the Markovian approximation yields inaccurate results. Notably, larger approximation errors are obtained for those models that depart the most from an exponential distribution, that is, Pareto 2 and the uniform model with narrow variance. Note that the QoS metric is reduced in the case of the Pareto 2 model as compared to other models for the service times. This is due to the fact that the Pareto 2 model has an infinite variance; as such, large service times are more likely to occur and a reduction in the QoS, as compared to other models, is expected. Finally, the steep change for the QoS around $T_M = 120$s in the uniform model is attributed to the narrow variance specified for such distribution, which eliminates the likelihood of extremely short and long service times.
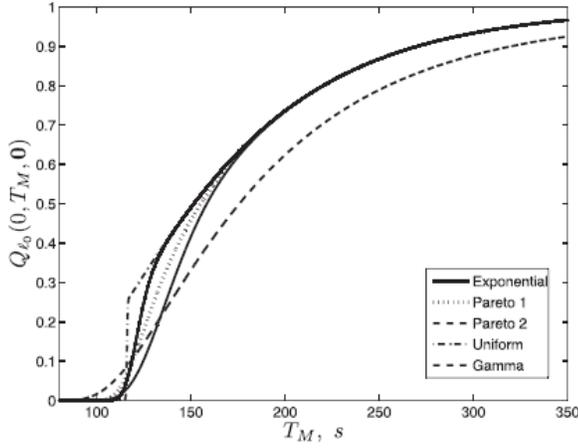
Fig. 9. The qos in executing an application in a parallel computing system as a function of the due time. Several stochastic models and the markovian approximation are considered for the service time.

4.2 Application of the Theory to Optimal Task Reallocation in DCSs

In order to account for the efficient execution of parallel applications in heterogeneous DCSs, the characterizations developed here have been parameterized by a DTR action that is synchronously executed by the system servers. This parameterization allows us to pose optimization problems, which, in turn, enable us to optimally devise a task migration scheme such that the metrics characterized in Theorems 1 and 2 are optimized. More precisely, the following mixed-integer optimization problems can be formulated

$$\min_{(t_b),\mathbf{L}} \overline{T}_{l_0}(t_b, 0),$$
$$\max_{(t_b,\mathbf{L})} Q_{\ell_0}(t_b, T_M, 0), \text{ (13)(14)(15)}$$
$$\max_{(t_b,\mathbf{L})} R_{\ell_0}(t_b, 0),$$

which are subject to: 1) $\sum_{j=1,j\neq i}^{n} l_{ij} \leq m_i, i = 1, \dots, n,$; 2) $l_{ij} \in \{0, 1, \dots, m_i\}, i, j = 1, \dots, n, i \neq j$; and 3) $t_b \geq 0$. Each problem has $n(n-1)$ nonnegative integer-valued variables, one nonnegative real-valued variable and $n^2 + 1$ restrictions. This type of optimization problem is known to be NP-hard due to the combinatorial explosion of the search space.

To provide a scalable, efficient, yet accurate solution to the problem of devising task migration policies for heterogeneous DCSs, we follow here the pairwise decomposition presented in [29] and [37] to find feasible optimal DTR policies. The idea consists in decomposing an n-server DCS into several two-server systems every time tasks must be exchanged between pairs of servers, and the optimal number of tasks to be migrated among the servers is computed in an iterative manner. To account for the the non-Markovian dynamics of the DCS, Theorems 1 to 4 are employed in the pairwise decomposition to compute the performance and reliability metrics. With this, the complexity of the solution remains bounded to the complexity of a two-server case, while the scalability turns out to be linear in the number of servers as shown in [29] and [37].

Distributed Task Reallocation Policy

We have employed a flexible class of distributed task reallocation policies developed in [29] and [37]. The policy estimates, at time $t = t_b$, the amount of load imbalance, which is denoted by $L_j^{ex}(t_b)$, that each server has with respect to the estimated total system load, $\hat{M}$. The imbalance estimator considers a general parameter, denoted as Λ_j, which represents different imbalance criteria. For example, if Λ_j is associated with the processing speed of the servers, the imbalance is determined by the relative computing power of the servers. Alternatively, if Λ_j is

associated with the reliability of the servers, then the relative resilience of the servers determines the amount of imbalance in the DCS. Next, each imbalanced server determines the initial amount of tasks to reallocate among the remaining servers in the system. This step is carried out by partitioning the excess load among all the candidate task-receiver servers listed in the set $\mathcal{W}_j$

$$l_{ij}^{(0)} \equiv l_{ij}^{(0)}(t_b) = \lfloor m_i(t_b) - \frac{\Lambda_j}{\sum_{\ell \in \mathcal{W}_j} \Lambda_\ell} \hat{M}_i(t_b) \rfloor, (16)$$

where $\lfloor \cdot \rfloor$ is the floor function and $m(t_b)$ is the load at the jth server at time $t = t_b$. The values $l_{ij}^{(0)}$ will be used as initial values in the search algorithm.

Table 1 Average service time and service reliability for different models under severe network-delay conditions

Initial load $(m_1, \ldots, m_5)$	Pareto 1	Pareto 2	$\overline{T}_{\ell_0}(0,0), s$ Shft. Exp.	Uniform	Exp.
(250,0,0,0,0)	171.13	191.21	171.38	186.05	218.28
(0,0,0,0,250)	179.91	199.79	172.11	185.71	215.75
(50,50,50,50,50)	170.12	199.70	175.11	184.02	212.92
(23,26,38,54,109)	165.10	171.39	157.92	171.28	195.39
Initial load $(m_1, \ldots, m_5)$	Pareto 1	Pareto 2	$R_{\ell_0}(0,0)$ Shft. Exp.	Uniform	Exp.
(250,0,0,0,0)	0.505	0.513	0.518	0.453	0.271
(0,0,0,0,250)	0.454	0.480	0.461	0.512	0.250
(50,50,50,50,50)	0.519	0.559	0.559	0.517	0.392
(78,60,45,37,30)	0.643	0.612	0.621	0.551	0.410

The average service time and the service reliability of a heterogeneous, five-server DCS have also been optimized by solving the optimization problems (13) to (15). It has been assumed that the parallel application is partitioned in $M = 250$ tasks. To assess the service reliability, it has been assumed that failure times follow Weibull distributions with means 1,000, 800, 600, 500, and 400 s, for servers 1 to 5, respectively. The average service times were set to be 5, 4, 3, 2, and 1 s for servers 1 to 5, respectively.

Table 1 lists both the average service time and the maximal service reliability obtained under severe network-delay conditions. Both performance metrics were obtained by solving the system of equations generated by the recursion presented in Theorem 3 and Corollary 2. For comparison, the column "Exponential" presents results yielded using the optimal policies devised under Markovian assumptions. Also for comparison, the last row on each part of the table represents a benchmark for each performance metric, since the initial allocation of tasks is actually the optimal allocation. These optimal task allocations were obtained by performing an MC-based exhaustive search over all the DTR policies. It can be noted from Table 1 that the exponential approximation produces relative errors between 15 and 85 percent. Remarkably, using incorrect models for the random times yield not only inaccurate results, but also specifies inappropriate reallocation policies that, in turn, reduce the performance metrics under study. For example, when all the initial load is allocated at the first server, a DTR devised using the non-Markovian model specifies that, in total, 232 and 182 tasks must be reallocated to the remaining servers in order to minimize the average service time and maximize the service reliability, respectively. However, when the Markovian models are employed to devise a DTR policy, only 201 and 157 tasks must be transferred to the other four servers. This behavior is also observed in the evaluations conducted for the two-server systems. Consider, for example, the Pareto 3 model in Figs. 3 and 4, the maximal service

reliability is achieved when, respectively, 27 and 10 tasks are transferred from server 1 to 2. However, the exponential approximation wrongly dictates that the maximal reliability is obtained when 25 and 7 tasks are transferred from server 1 to 2.

Table 2 Comparison between models in predicting the average service time and the service reliability under severe delays

		$\bar{T}_{\ell_0}(0,0), s$			Avg. Comp Time, s	
Initial load $(m_1, \ldots, m_5)$	Exact	Approx.	Exp.	Exact	Approx.	Exp.
(250,0,0,0,0)	171.13	196.01	218.28	1940	520	58.39
(0,0,0,0,250)	179.91	204.95	215.75	1896	535	59.49
(50,50,50,50,50)	170.12	181.64	212.92	1900	536	56.08
(23,26,38,54,109)	165.10	172.97	195.39	1873	600	54.21
		$R_{\ell_0}(0,0)$			Avg. Comp Time, s	
Initial load $(m_1, \ldots, m_5)$	Exact	Approx.	Exp.	Exact	Approx.	Exp.
(250,0,0,0,0)	0.505	0.429	0.271	2053	728	64.79
(0,0,0,0,250)	0.454	0.377	0.250	2263	729	76.13
(50,50,50,50,50)	0.519	0.478	0.392	2335	730	68.03
(78,60,45,37,30)	0.643	0.590	0.410	2178	757	66.31

In Table 2, we have listed results for the average service time and the service reliability of the five-server DCS, under severe network-delay conditions, when the exact and the approximated model have been employed to solve for the optimization problems stated in (13) and (15). The stochastic model evaluated in this example is Pareto with finite variance ("Pareto 1" model). For the approximate model, we have assumed that all the tasks in transit to the servers arrive, as a single batch, at the maximum of all the random task-arrival times. In Table 2, columns labeled as "Exact" list the results obtained when the exact model of the DCS is employed. The time measurements have been obtained using a standard nondedicated desktop computer, equipped with a 2.20 GHz dual core processor, 8 GBytes of RAM and running a 64 bits, Linux-based operating system. Equations and algorithms to solve the optimization problems (13) and (15) were coded in C, and the GNU scientific library was exploited in order to efficiently calculate the integrals of each equation. The average computing times reported in Table 2 correspond to averaging ten executions of the software that solves the optimization problems. It can be noticed that the values for the metrics obtained by means of the approximated metrics incur in a maximum error of approximately 14 percent in the case of the average service time and 17 percent in the case of the service reliability. Notably, it can be observed from comparing Tables 1 and 2 that the approximation errors obtained by the approximated models are lower than those obtained by the Markovian models, for the case of severe network delays. Regarding the computing time of the models, our evaluations show that when the approximated models are used in solving the optimization problems (13) and (15), computing times are reduced in approximately 3 to 3.5 times with respect to the computing time taken by the exact models. From Table 2, it can be observed that the accuracy provided by the age-dependent non-Markovian model comes at the expense of a notorious increment in the computing time. Our calculations show that, as compared to the Markovian approximations, a computing-time overhead of approximately 10 times is demanded by the approximated models while an overhead 30 times is required by the exact models.

Finally, we have validated our theory in a larger DCS environment. To do so, we have used ten out of the 12 nodes available in the ristra cluster of the Center for Advanced Research Computing at the University of New Mexico. From experimentally characterizing the DCS, we have found that the empirical pdfs of the service times of the servers follow shifted gamma distributions with average service times between 2 and 5 seconds. Since we have employed in our experimental set-up a cluster with a high-speed communication network, traffic shapers have been used to artificially introduce transfer delays. Pareto distributions were fitted for the empirical pdfs of

the task transfer times, whose average transfer times per task range between 1 to 3 seconds. As in the previous evaluations, it has been assumed that failure times follow Weibull distributions, whose average failure times are between 3,000 and 5,000 seconds. The initial workload in the DCS was set to 10,000 tasks. We have initially allocated the tasks in four different manners. The "Uniform" initial allocation means that all the servers have in their queues the same number of tasks at time t = 0. The "Unbalanced 1" and "Unbalanced 2" initial allocations represent, respectively, cases where all the tasks are allocated at the slowest and at the fastest server. The initial "Random" allocation assigns a random allocation to the servers at time t = 0. Also as in the previous evaluations, the service reliability in the experimental case was calculated by averaging a total of 500 independent samples.

Table 3 The average service time and the service reliability of a 10-server non-markovian DCS

	$\bar{T}_{\ell_0}(0,0), s$		$R_{\ell_0}(0,0)$	
Initial Load	Theoretical	Experimental	Theoretical	Experimental
Uniform	3272.1	3653.5	0.7743	0.7001
Unbalanced 1	2928.9	3551.4	0.7901	0.7467
Unbalanced 2	2743.1	3375.8	0.6259	0.6174
Random	3100.2	3417.0	0.7810	0.7502

Table 3 shows the theoretical predictions, and the experimental results, obtained for the average service time and the service reliability of a 10-server non-Markovian DCS. These results have been obtained when the optimal task reallocation policy is executed by the servers in the system. The reallocation policy has been calculated by solving the optimization problems (13) and (15) using the two-server age-dependent models and the pairwise decomposition algorithm. Results in Table 3 for the predicted and the experimental metrics exhibit a maximum error of approximately 18 percent in the case of the average service time and 10 percent in the case of the service reliability. Regarding the computing time of the solution, the optimization process took an average time between 10 and 16 hours to produce the results on the standard nondedicated desktop computer previously specified.

SECTION 5 Conclusions

We have presented a novel, age-dependent analytical characterization of the average service time of an application, the service reliability and the QoS in executing an application in heterogeneous DCSs in the presence of stochastic communication delays with general probability distributions. In addition, the characterization is parameterized by a DTR action in order to efficiently execute parallel applications. This work, therefore, offers a general and rigorous mathematical framework for both assessing performance and reliability and systematically devising DTR polices for maximal performance and reliability in heterogeneous DCSs with non-Markovian dynamics. The developed age-dependent framework constitutes a major generalization of our earlier Markovian models for the tandem distributed queuing system, reported in [16] and [21], to a non-Markovian and hence more realistic settings. Our results indicate that when the network delays are relatively large compared to service times, the error in estimating any of these metrics, as a result of falsely assuming exponentially distributed random delays becomes significant, thereby necessitating the use of our age-dependent model. Similarly, when distributions of the random service times depart from the theoretical exponential distribution, the performance and reliability metrics are inaccurately approximated by the Markovian models. For example, our calculations show relative errors as large as 120 percent in estimating the average service time and the service reliability, respectively. It must be mentioned that the accuracy provided by the age-dependent model comes at the expense of increased computational complexity as compared to its Markovian counterpart. To alleviate such computational burden, an analytical approximation of the model was

also developed. The approximation has the appealing feature that dimensionality of the space-state vector characterizing the system scale linearly in the number of system servers.

In future works, refinements to the approximation of the exact age-dependent model can be derived using a probabilistic dimension-reduction technique, which exploits the likelihood of the regeneration events. Methods that exploit the concept of bisimulation, which are known for its effectiveness in reducing the state space of large hybrid systems, can also be employed to refine the approximations of the exact age-dependent model.

ACKNOWLEDGEMENT

References

1. G. Bolch, S. Greiner, H. de Meer, K.S. Trivedi, Queueing Networks and Markov Chains, John Wiley and Sons, Inc., 2006.

2. Y.-S. Dai, G. Levitin, "Optimal Resource Allocation for Maximizing Performance and Reliability in Tree-Structured Grid Services", *IEEE Trans. Reliability*, vol. 56, no. 3, pp. 444-453, Sept. 2007.

3. Y.-S. Dai, G. Levitin, K. Trivedi, "Performance and Reliability of Tree-Structured Grid Services Considering Data Dependence and Failure Correlation", *IEEE Trans. Computers*, vol. 56, no. 7, pp. 925-936, July 2007.

4. A. Bobbio, A. Puliafito, M. Tekel, "A Modeling Framework to Implement Preemption Policies in Non-Markovian SPNs", *IEEE Trans. Software Eng.*, vol. 26, no. 1, pp. 36-54, Jan. 2000.

5. M. Bouissou, J.-L. Bonc, "A New Formalism that Combines Advantages of Fault-Trees and Markov Models: Boolean Logic Driven Markov Processes", *Reliability Eng. and System Safety*, vol. 82, no. 2, pp. 149-163, 2003.

6. M.C. Kim, P.H. Seong, "Reliability Graph with General Gates: An Intuitive and Practical Method for System Reliability Analysis", *Reliability Eng. and System Safety*, vol. 78, no. 3, pp. 239-246, 2002.

7. X. Tanga, K. Li, R. Li, B. Veeravalli, "Reliability-Aware Scheduling Strategy for Heterogeneous Distributed Computing Systems", *J. Parallel and Distributed Computing*, vol. 70, no. 11, pp. 941-952, 2010.

8. S. Ali, H.J. Siegel, M. Maheswaran, S. Ali, D. Hensgen, "Task Execution Time Modeling for Heterogeneous Computing Systems", *Proc. Ninth Heterogeneous Computing Workshop (HCW '00)*, pp. 185-199, 2000.

9. D. Vidyarthi, A. Tripathi, "Maximizing Reliability of a Distributed Computing System with Task Allocation Using Simple Genetic Algorithm", *J. Systems Architecture*, vol. 47, pp. 549-554, 2001.

10. Y. Hamam, K. Hindi, "Assignment of Program Tasks to Processors: A Simulated Annealing Approach", *J. Operational Research*, vol. 122, pp. 509-513, 2000.

11. G. Attiya, Y. Hamam, "Task Allocation for Maximizing Reliability of Distributed Systems: A Simulated Annealing Approach", *J. Parallel and Distributed Computing*, vol. 66, pp. 1259-1266, 2006.

12. L.E. Holloway, B.H. Krogh, A. Giua, "A Survey of Petri Net Methods for Controlled Discrete Event Systems", *Discrete Event Dynamic Systems*, vol. 7, no. 2, pp. 151-190, 1997.

13. S. Pllana, I. Brandic, S. Benkner, "Performance Modeling and Prediction of Parallel and Distributed Computing Systems: A Survey of the State of the Art", *Proc. First Int'l Conf. Complex Intelligent and Software Intensive Systems (CISIS '07)*, pp. 279-284, 2007.

14. M.M. Hayat, S. Dhakal, C.T. Abdallah, J.D. Birdwell, J. Chiasson, "Dynamic Time Delay Models for Load Balancing. Part II: Stochastic Analysis of the Effect of Delay Uncertainty" in Advances in Time Delay Systems, Springer-Verlag, pp. 355-368, 2004.

15. J. Al-Jaroodi, N. Mohamed, H. Jiang, D. Swanson, "Modeling Parallel Applications Performance on Heterogeneous Systems", *Proc. 17th Int'l Symp. Parallel and Distributed Processing (IPDPS)*, pp. 160.2, 2003.

16. J.E. Pezoa, S. Dhakal, M.M. Hayat, "Maximizing Service Reliability in Distributed Computing Systems with Random Failures: Theory and Implementation", *IEEE Trans. Parallel and Distributed Systems*, vol. 21, no. 10, pp. 1531-1544, Oct. 2010.

17. C. Kelling, "TimeNET-Sim-a Parallel Simulator for Stochastic Petri Nets", *Proc. 28th Ann. Simulation Symp.*, 1995.

18. A. Bobbio, A. Puliafito, M. Scarpa, M. Telek, "WebSPN: A Web-Accessible Petri Net Tool", *Proc. Conf. Web-Based Modeling and Simulation*, 1998.

19. S. Haddad, P. Moreaux, "Approximate Analysis of Non-Markovian Stochastic Systems with Multiple Time Scale Delays", *Proc. IEEE CS 12th Ann. Int'l Symp. Modeling Analysis and Simulation of Computer and Telecomm. Systems (MASCOTS '04)*, pp. 23-30, 2004.

20. D. Logothetis, V. Mainkar, K. Trivedi, "Transient Analysis of Non-Markovian Queues via Markov Regenerative Processes", *Probability and Statistics—A.J. Medhi Festschrift*, pp. 109-131, 1996.

21. S. Dhakal, M.M. Hayat, J.E. Pezoa, C. Yang, D.A. Bader, "Dynamic Load Balancing in Distributed Systems in the Presence of Delays: A Regeneration-Theory Approach", *IEEE Trans. Parallel and Distributed Systems*, vol. 18, no. 4, pp. 485-497, Apr. 2007.

22. G. Levitin, Y.-S. Dai, "Service Reliability and Performance in Grid System with Star Topology", *Reliability Eng. and System Safety*, vol. 92, pp. 40-46, 2007.

23. D.R. Cox, "The Analysis of Non-Markovian Stochastic Processes by the Inclusion of Supplementary Variables", *Proc. Cambridge Philosophical Soc.*, vol. 51, pp. 433-441, 1965.

24. K.S. Trivedi, A. Bobbio, G. Ciardo, R. German, A. Puliafito, M. Telek, "Non-Markovian Petri Nets", *Proc. ACM Int'l Conf. Measurement and Modeling of Computer Systems (SIGMETRICS)*, pp. 263-264, 1995.

25. R. German, "Non-Markovian Analysis", pp. 156-182, 2002.

26. R. German, "Markov Regenerative Stochastic Petri Nets with General Execution Policies: Supplementary Variable Analysis and a Prototype Tool", *Performance Evaluation*, vol. 39, no. 1–4, pp. 165-188, 2000.

27. P. Bazan, R. German, "An Iterative Approximate Analysis Method for Non-Markovian Models Based on Supplementary Variables", *Proc. 12th GI/ITG Conf. Measuring Modelling and Evaluation of Computer and Comm. Systems (MMB '04)*, pp. 255-264, 2004.

28. M. Telek, A. Horváth, "Transient Analysis of Age-MRSPNs by the Method of Supplementary Variables", *Performance Evaluation*, vol. 45, no. 4, pp. 205-221, 2001.

29. J.E. Pezoa, M.M. Hayat, Z. Wang, S. Dhakal, "Optimal Task Reallocation in Heterogeneous Distributed Computing Systems with Age-Dependent Delay Statistics", *Proc. 39th Int'l Conf. Parallel Processing (ICPP)*, 2010.

30. Z. Tang, J.D. Birdwell, J. Chiasson, C.T. Abdallah, M.M. Hayat, "Resource-Constrained Load Balancing Controller for a Parallel Database", *IEEE Trans. Control Systems Technology*, vol. 16, no. 4, pp. 834-840, July 2008.

31. J. Sonnek, A. Chandra, J.B. Weissman, "Adaptive Reputation-Based Scheduling on Unreliable Distributed Infrastructures", *IEEE Trans. Parallel and Distributed Systems*, vol. 18, no. 11, pp. 1551-1564, Nov. 2007.

32. V. Shestak, J. Smith, A. Maciejewski, H. Siegel, "Stochastic Robustness Metric and Its Use for Static Resource Allocations", *J. Parallel and Distributed Computing*, vol. 68, pp. 1157-1173, 2008.

33. V. Shestak, E.K.P. Chong, A.A. Maciejewski, H.J. Siegel, "Robust Sequential Resource Allocation in Heterogeneous Distributed Systems with Random Compute Node Failures", *Proc. IEEE Int'l Symp. Parallel and Distributed Processing (IPDPS)*, 2009.

34. C. Fetzer, "Perfect Failure Detection in Timed Asynchronous Systems", *IEEE Trans. Computers*, vol. 52, no. 2, pp. 99-112, Feb. 2003.

35. T. Ma, J. Hillston, S. Anderson, "Evaluation of the Qos of Crash-Recovery Failure Detection", *Proc. ACM Symp. Applied Comnuting*, pp. 538-542, 2007.

36. "The GNU Scientific Library", 2011, [online] Available: http://www.gnu.org/s/gsl.

37. J.E. Pezoa, S. Dhakal, M.M. Hayat, "Decentralized Load Balancing for Improving Reliability in Heterogeneous Distributed Systems", *Proc. Int'l Conf. Parallel Processing Workshops (ICPP '09)*, 2009.