4-2014

Reliability of Heterogeneous Distributed Computing Systems in the Presence of Correlated Failures

Jorge E. Pezoa

Majeed M. Hayat

IEEE Transactions on Parallel and Distributed Systems, Vol. 25, No. 4 (April 2014): 1034-1043. DOI.

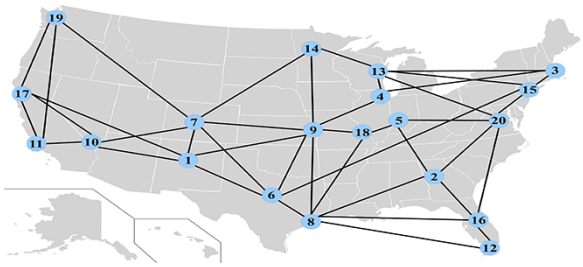
Reliability of Heterogeneous Distributed Computing Systems in the Presence of Correlated Failures

Jorge E. Pezoa
Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque
Majeed M. Hayat
Department of Electrical and Computer Engineering and also with the Center for High Technology Materials, University of New Mexico, Albuquerque

Abstract:

While the reliability of distributed-computing systems (DCSs) has been widely studied under the assumption that computing elements (CEs) fail independently, the impact of correlated failures of CEs on the reliability remains an open question. Here, the problem of modeling and assessing the impact of stochastic, correlated failures on the service reliability of applications running on DCSs is tackled. The service reliability is modeled using an integrated analytical and Monte-Carlo (MC) approach. The analytical component of the model comprises a generalization of a previously developed model for reliability of non-Markovian DCSs to a setting where specific patterns of simultaneous failures in CEs are allowed. The analytical model is complemented by a MC-based

procedure to draw correlated-failure patterns using the recently reported concept of probabilistic shared risk groups (PSRGs). The reliability model is further utilized to develop and optimize a novel class of dynamic task reallocation (DTR) policies that maximize the reliability of DCSs in the presence of correlated failures. Theoretical predictions, MC simulations, and results from an emulation testbed show that the reliability can be improved when DTR policies correctly account for correlated failures. The impact of correlated failures of CEs on the reliability and the key dependence of DTR policies on the type of correlated failures are also investigated.



description of the attached tpds-gagraphic-78.gif linked by @xlink:href

SECTION 1. Introduction

Traditionally, the problem of modeling and analyzing reliability in DCSs in the presence of CE failures has been tackled under the assumption that failures among CEs occur in a mutually-independent fashion. Under the assumption of independent failures, the reliability of a DCS depends on both the number of CEs composing the system and their individual likelihoods of failure. A vast amount of work has been developed under this assumption, and several approaches to improve the reliability of applications executed on such systems have been developed [1]–[2][3].

The assumption of independent failures of CEs greatly simplifies the analysis; however, such assumption may not be realistic for the type of failures occurring in modern DCSs, which may include heterogeneous CEs, non-negligible communication delays, unreliable CEs, unreliable communication links, and a dynamic topology that changes in a random fashion. For instance, Schroeder et al. [4], Kondo et al. [5], Gallet et al. [6], and Joshi et al. [7], analyzed different failure traces from large-scale high-performance computing (HPC) systems, Internet distributed systems as well as DCSs, and all of them concluded independently that all those systems are affected by frequent, correlated machine crashes and network failures that reduce the reliability of the entire system. Further, it was stated that to improve, at no extra cost, the reliability of large-scale DCSs, the software managing applications must provide means to compensate for correlated failures [5] [7].

DCSs that extend over large geographical areas, as in the case of donation grids and peer-to-peer (P2P) networks for instance, can be vulnerable to large-scale failures resulting from massive communication network malfunctions, wide-area power outages, wide-area natural disasters, or deliberate wide-area attacks to the system infrastructure as those inflicted by weapons of mass destruction (WMD) and high-power electromagnetic pulses. These events of stress occur at specific geographical locations and may induce correlated failures that disrupt specific parts of the DCS. Correlated failures may not only inflict disturbance to the system's availability but they may also induce further failures in other servers as a result of the lack of reliable communication between the DCS components, especially in situations where any data exchange takes large communication times [8]. In this work, we are interested in assessing the service reliability of DCSs in scenarios where servers fail, without recovery, in a correlated manner.

This paper has two contributions: 1) modeling the service reliability of applications executed on DCSs in the presence of correlated component failures by means of a hybrid analytical and MC-based approach, and 2) optimizing the service reliability by means of DTR policies. The service reliability is modeled by extending our

analytical non-Markovian model in [9] to include specific group failures of CEs at each failure event. This extension enables us to calculate the reliability conditional on the occurrence of a specific realization of correlated CE failures. By averaging the conditional reliability over a large number of correlated-failure realizations, the average service reliability of an application in the presence of correlated failures can be estimated. To develop a statistical model for correlated failures we have adopted the concept of probabilistic shared risk link group (SRLG), which was developed by the network-routing community and has been used it to introduce correlation in a meaningful and practical manner by defining sets of CEs that may suffer from a common stress event. To maximize the reliability of DCSs in the presence of correlated failures, a novel class of DTR policies is also developed. The DTR policies exploit statistical knowledge on correlated failures to preemptively redistribute tasks among the CEs with the goal of maximizing the service reliability of the application. Results show that the benefit of DTR in improving reliability can be elevated when policies account for the effects of correlated failures.

The rest of this paper is organized as follows. Section 2 presents a brief survey of related work on modeling correlated failures and assessing reliability in DCSs. In Section 3 we build a model for correlated failures and introduce the hybrid analytical and MC-based approach for predicting the service reliability of a DCS. The concept of correlated-failure-aware DTRs is introduced in Section 4. Section 5 presents our simulation results. Finally, our conclusions are given in Section 6.

SECTION 2. Related Work

Correlated failures have been extensively studied in other areas outside the context of DCSs. A simple taxonomy, based on the type of correlation exhibited by the failures, classifies correlated failures in temporal, spatial, logical or any combination of them.

Temporal failure correlations have been analyzed mostly in an empirical manner. In [10] the effects of failure patterns on the availability of a DCS's monitoring service were studied and simple techniques for improving the robustness of the monitoring services were developed. In [6] the time-varying behavior of failures in large-scale DCSs was empirically modeled from failure traces obtained from production systems. Zhang and Fu analyzed node, cluster, and system-wide failure behaviors to predict and capture temporal failure correlations (at different time scales) in a coalition cluster environment [11]. Spatially correlated failures have been modeled in large-scale systems as well. In [6] spatially correlated failures were identified from real data using the following intuitive approach: groups of failures occurring within a short time interval across the CEs were assumed to be spatially correlated. Other modeling approaches have assumed that spatially correlated failures are induced by massive events where a region containing several CEs are physically damaged [12]. Spatially correlated failures have also been modeled in wireless sensor networks, where spatial-failure patterns were modeled assuming the simultaneous failure of all the sensor nodes in a specific region [13].

Logical failure correlations have also been studied in DCSs, and have been obtained either from the logical data dependencies of the applications or from the logical interconnection between hardware components. Weatherspoon et al. analyzed logically correlated failures in P2P networks and developed a framework for discovering groups of CEs that are maximally independent in their failure characteristics and clustered them to compensate for correlated failures [14]. In [15] and [16], a software reliability modeling framework capable of incorporating the dependencies among successive software runs was reported. In [17], Dai et al. evaluated the reliability of a grid computing system considering the failure correlation of different subtasks executed by the grid; however, component failures were assumed to be independent. Recently, approximate analytical expressions for reliability in on-demand systems exhibiting correlated failures were presented [18].

Traces of real failures from several parallel, high- performance (HP), and distributed computing (DC) environments have become available to researchers in the last years [4][19]. In [4], Schroeder et al. statistically analyzed, and made public, traces of nine years of failures from a large HPC center. They noticed that the failure time of CEs follows a Weibull distribution, while their recovery time follows a lognormal distribution. In the context of correlated failures, Schroeder and Gibson noticed a certain degree of correlation between the failure rate of a CE and the type and intensity of the workload running on it. Iosup et al. analyzed traces of the long-term availability in a large-scale experimental grid environment [20]. The authors studied the effect of correlated failures in time, and built a failure model for the grid with no spatial correlation between the occurrence of failures at the different sites of it. Kondo et al. characterized the time availability in an Internet-based DCS focusing on identifying patterns of correlated availability [5]. They also modeled the availability and failure times in diverse DCSs; however, their analysis did not considered the effect of correlated failures [19].

SECTION 3. Modeling Reliability of DCSs in the Presence of Correlated Failures

1. Problem Definition

This paper tackles the problem of improving the reliability of non-Markovian DCS, by means of task reallocation, when failures in the CEs exhibit spatial correlation. We are particularly interested in improving the service reliability, which is defined as the probability that a given application can be entirely executed by a DCS. The DCS is assumed to be composed of n heterogeneous CEs, whose processing capabilities are of the processor-consistent type [21]; that is, the random time taken by any server to process any task follows a general distribution and depends only upon the random service time of the server executing such task. Parallel applications served by the DCS are assumed to belong to the class of applications with no data-dependence constraints between operations. Moreover, applications are supposed to be partitioned, at time $t = 0$, into M atomic tasks by an off-line application scheduler that allocates m_j tasks at the queue of the jth server. Further, all the CEs perform a synchronous DTR action at the prescribed time $t = t_b$.

Finally, we have also assumed that the exchange of any task, or any group of tasks, among any pair of servers experiences a stochastic communication delay. Such stochastic delays follow general distributions and depend upon both the number of tasks exchanged among the servers as well as network-related parameters such as heterogeneous end-to-end propagation times. We shall also assume that computing servers may fail permanently in a correlated fashion (to be described later) at any random instant following the so-called crash-stop failure model where tasks cannot be recovered from a failed server [22]. As a consequence, the application being executed on the DCS cannot be completed if at least one task remains unprocessed at a failed CE. Additionally, we further assume that small fixed-sized failure-notice (FN) messages are exchanged over the network in order to detect and isolate failed servers. These FN messages too experience stochastic end-to-end transfer delays that depend only on the end-to-end propagation time of each communication link. Finally, we adopt the simplifying assumption that servers employ a reliable message-passing protocol to guarantee that tasks are not discarded in situations such as a server failing while exchanges tasks with other servers.

1. Reliability in the Presence of Correlated Failures

2. Shared Risk Groups (SRGs) and Correlated Failures

We focus on modeling the service reliability in scenarios where servers fail without recovery. Specifically, we are interested in correlated failures triggered by large-scale geographical attacks to the DCS infrastructure, which diminish the ability of the DCS to reallocate tasks among CEs. Thus, we model the class of correlated failures resulting from real-world massive disruptions and/or physical attacks to the DCS infrastructure. To do so, we

have taken from the network routing community the concept of SRLGs and adjusted it here to introduce correlation in a meaningful and practical manner.

The concept of SRLG has successfully been used to address, in a systematic manner, the survivability of network topologies in the presence of multiple correlated communication link failures [23]. The key idea in SRLGs is that multiple yet different telecommunication services may be affected by a common network failure under the proviso that they share a common failure risk, such as a fiber-optic link, a routing device, a routing domain, etc. The consequence of a common risk failure is that all the services sharing the same risk would be affected or even totally interrupted. In [24], the concept of probabilistic SRLG was introduced as a generalization of the traditional SRLG to model stochastic failure events affecting the network topology, and upon the occurrence of a SRLG failure event, the communication links associated to the group fail with some probability. Here we take both concepts and redefine them to a DC environment.

Definition 1. A shared risk group in a DCS is a set of servers that may be affected by a common failure to the infrastructure of the DCS under the condition that they share a common failure risk.

In DCSs examples of common failure risks are: (1) infrastructure anomalies, such as power outages or spikes; (2) hardware failures, such as failures in memory modules, CPUs or even fans; (3) input/output errors, such as failures at disk drives or drive controllers; (4) network failures, such as failures at FastEthernet o GigaEthernet switches; and (5) software failures, such as failures at schedulers or distributed file systems. We note all these types of failures are logged by DCSs following the trace format of The Failure Trace Archive [25]. In fact, by examining traces in [25] from the HPC system at Los Alamos National Laboratory we have observed that the first failure triggered by a power outage produced a correlated failure at the nodes identified as 655 and 782. Other examples of real-world correlated failures found in the traces were triggered, for instance, by failures at UPSs and fiber drives. Examples of common failure risks of interest to this paper are groups of CEs sharing a close geographical area, groups of CEs within the same facility, groups of CEs facing a cyber attack to either the DCS, their distributed operating system, their communication network, or their Internet service provider, etc. [26].

Suppose now that there exists a set A of SRG events that may induce correlated failures to the DCS. Suppose also that each event $A \in \mathcal{A}$ has a probability of occurrence of $\pi(A)$. Further, assume that the underlying infrastructure of an n-server DCS is abstracted by a connected, undirected graph $G = (V, E)$, where $V = \{1,2, \dots, n\}$ is the set of CEs and $E \subset V \times V$ is the set of communication links. Consequently, when the SRG event A happens, the set of servers V can be partitioned into two sets: V_A and V_A^c, where the former set denotes the collection of all servers sharing the common risk associated to the SRG event A and the latter set denotes all those servers unaffected by the event A.

Definition 2. A PSRG in a DCS is a set of servers that do fail with a positive failure probability, in the event of a SRG failure. More precisely, the failure probability of the ith server, conditional on the SRG failure event A, is denoted as p_i^A and satisfies: $p_i^A > 0$ for all $i \in V_A$ and $p_i^A = 0$ otherwise.

Following [24], we assume that only one PSRG event may occur at a time, meaning that the PSRG failure events are mutually exclusive, and consequently the following relationship holds: $\sum_{A \in \mathcal{A}} \pi(A) = 1$. This otherwise arbitrary definition has been effectively used in the routing community and makes sense in the context of the class of failures regarded here [23]–[24][27].

Definition 3. We say that the servers i and j belonging to a DCS are correlated if p_i^A and p_j^A are both positive for the A PSRG. Moreover, upon the occurrence of the A SRG event, the probabilities p_i^A and p_j^A are mutually independent for all the pairs of servers in V_A.

Suppose now that X_i is a binary random variable representing if the ith server has failed ("1") or not ("0"). By arranging the n binary random variables in vector form, we introduce the failure vector $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n)$, which takes values in $\{0,1\}^n$, as the random vector defining the failure state of the DCS. Also, a realization of the failure vector $\mathbf{X}^A$ is denoted by the binary vector $\mathbf{x}$ and is termed as a failure. Finally as a more practical matter, to generate samples of correlated failures given a specific probabilistic SRG event, it only suffices to specify the probabilities p_i^A, independently generate realizations for the X_i^A random variables, and form the vector $\mathbf{x}^A$.

3. Service Reliability and Correlated Failures

In order to calculate the service reliability in the presence of spatially correlated failures, a hybrid analytical and MC approach is presented. By drawing samples of correlated failures from the PSRG model, a large number of realizations of correlated failure patterns can be generated. Thus, conditional on a particular failure pattern, an analytical model for the conditional service reliability of a DCS can be derived as shown in Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2013.78. More precisely, conditional on a sample, say, $\mathbf{X}^A = \mathbf{x}^A$, of correlated failures induced by the occurrence of the APSRG event, the system of recursive integral equations (9), with initial conditions (10), presented in Appendix A, available in the online supplemental material, can be used to compute the conditional service reliability.

In brief, we can estimate the service reliability of a DCS in the presence of correlated failures induced by an A PSRG event as follows. Let k be the kth sample of correlated failures induced by the occurrence of the A PSRG event, with $k = 1,2, \ldots, \kappa$. Let also $R_{\ell_0,k}(t_b|\mathbf{X}^A = \mathbf{x}^A)$ be the service reliability of the DCS when servers perform a synchronous DTR action at time $t = t_b$, the initial system configuration is as specified by ℓ_0, and the kth sample of correlated failures induced by the occurrence of the A PSRG event is as specified by $\mathbf{X}^A = \mathbf{x}^A$. The service reliability of the DCS in the presence of correlated failures induced by the occurrence of the A PSRG event, $R_{\ell_0}^A(t_b)$, can be estimated by simply averaging over the κ samples of correlated failures:

$$R_{\ell_0}^A(t_b) \approx \frac{1}{\kappa}\sum_{k=1}^{\kappa} \quad R_{\ell_0,k}(t_b|\mathbf{X}^A = \mathbf{x}^A). \text{ (1)}$$

SECTION 4. Correlated-Failure—Aware DistributedTask Reallocation Policy

In [9] [28] we developed a flexible class of DTR policies. Each policy in the class estimates, at $t = t_b$, the amount of load imbalance, $L_j^{ex}(t_b)$, that each server has with respect to the estimated total system load, $\hat{M}_i(t_b)$. The imbalance estimation criterion considers a general parameter, denoted as Λ_j, which represents different choices for the imbalance criterion, such as the relative computing power and the individual reliability of the CEs. Once the imbalance criterion is defined, each unbalanced 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:

$$l_{ij}^{(0)} \equiv l_{ij}^{(0)}(t_b) = \left\lfloor m_i(t_b) - \Lambda_j \hat{M}_i(t_b) / \sum_{\ell \in \mathcal{W}_j} \Lambda_\ell \right\rfloor, \text{ (2)}$$

where $\lfloor \cdot \rfloor$ is the floor function and $m_j(t_b)$ is the load at the jth server at time $t = t_b$. The values $l_{ij}^{(0)}$ are initial values for the partition of tasks at an unbalanced server.

To develop a DTR policy accounting for correlated failures, the ideas of PSRGs and correlation introduced in Definitions 2 and 3 must be considered. Here we have modified the general DTR policy and created a correlated-failure-aware policy by proposing the following reallocation criterion:

$$\Lambda_j = \lambda_{d_j}\left(1-\lambda_j^f / \sum_{k\in V}\lambda_k^f\right)(1-\pi(A))\left(1-p_j^A p_i^A\right), \text{(3)}$$

where λ_{d_j} and λ_j^f are, respectively, the processing speed and the failure rate of the jth CEs. The idea behind this definition for the reallocation policy is to favor the migration of tasks from overloaded servers to those CEs that are less vulnerable to fail in a correlated manner given the PSRG event A, while simultaneously penalize the migration of tasks to those CEs that are correlated, in the sense of Definition 3. Note that the processing speed of the servers as well as the failure rates are still considered in the definition of the reallocation criterion. When failures are uncorrelated the term $p_j^A p_i^A$ is zero as a consequence of Definition 2, and the proposed policy becomes proportional to: (1) the likelihood of not occurrence of the PSRG event A; and (2) $\lambda_{d_j}\left(1-\lambda_j^f / \sum_{k\in V}\lambda_k^f\right)$, which is exactly the reallocation policy for the independent failure case defined in [28], Section 2.2.

Note that in the development of the hybrid analytical and MC model for the service reliability, the DTR policy executed by the servers at time t_b has been considered as a parameter. We include such parameterization in the notation of the service reliability as $R_{\ell_0}^A(t_b;\mathbf{L}) \equiv R_{\ell_0}^A(t_b)$, where $\mathbf{L}$is an n-by-n matrix whose l_{ij}th element denotes the number of tasks to be reallocated from the ith to the jth server at time t_b. More importantly, we can exploit such parameterization to pose an optimization problem that allow us to optimally migrate tasks among the CEs such that the service reliability of the DCS, in the presence of correlated failures, can be maximized. Mathematically, the following mixed-integer optimization problem can be stated:

$$(t_b^*, \mathbf{L}^*) = \arg\min_{(t_b,\mathbf{L})} \; R_{\ell_0}^A(t_b;\mathbf{L}), \text{(4)}$$

subject to

$$\sum_{j=1, j\neq i}^{n} l_{ij} \leq m_i, i = 1, \ldots, n, \text{(5)}$$

$$l_{ij} \in \{0,1,\ldots,m_i\}, i, j = 1, \ldots, n, i \neq j, \text{(6)}$$

$$t_b \geq 0. \text{(7)}$$

The problem has $n(n-1)$ non-negative integer-valued variables, one non-negative 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; the efficient search algorithm based on a pairwise

decomposition of the DCS presented in [28] has been employed here to find feasible DTR policies maximizing the service reliability of a DCS in the presence of correlated failures.

We note that the proposed algorithm scales linearly in both the number of servers in the DCS and the number of tasks queued at the overloaded server. This claim is justified as follows. Suppose that the jth server is unbalanced and must reallocate tasks to η servers, where the relationship $0 < \eta < n$ holds. Since the DTR policy executed by the servers is distributed, each one must solve (9) and (10) independently. For $n = 2$ servers, the complexity in solving such equations is a function of the number of tasks queued at the jth server, that is $\mathcal{O}(f(m_j))$. Since the jth server decomposes the DCS into η pairs of DCSs, the overloaded server must solve at most η times the optimization problem (4) for $n = 2$. Further, by construction the algorithm must solve, at the jth server, no more than N times such optimization problem. From this, we observe that the complexity of the algorithm is $\mathcal{O}\left(N(n-1)f\left(m_j\right)\right)$. In addition, if an exhaustive search in the number of tasks to reallocate is conducted, then $f(m_j)$ is bounded by m_j because no more than m_j tasks must be reallocated.

SECTION 5. Results

1. Small-Scale Experiments

In this section the service reliability of a small-scale DCSs with a representative network topology has been analyzed. The DCS corresponds to a 20-node, nationwide distributed system where servers are located at several cities in the US as shown in Fig. 1. In our calculations we have considered two classes of CEs: HP servers and standard servers. Since in a DCS HP servers are expected to serve more tasks than standard CEs, they are supplied with a larger number of communication links. In particular, all those servers with five or more communication links in the topologies depicted in Fig. 1 are regarded as HP servers. Due to geographical proximity, in the DCS with 20-node servers the following PSRGs have been defined: PSRG-1 composed of servers 10, 11, 17, and 19; PSRG-2 composed of servers 1 and 7; PSRG-3 composed of servers 6 and 8; and PSRG-4 composed of servers 4, 5, 9, and 18. For simulation purposes, we suppose that each DCS may be affected only by four different PSRGs events, each one of them associated to PSRGs 1 to 4 and each one of them having a likelihood of occurrence of $\pi(A) = 0.25$.

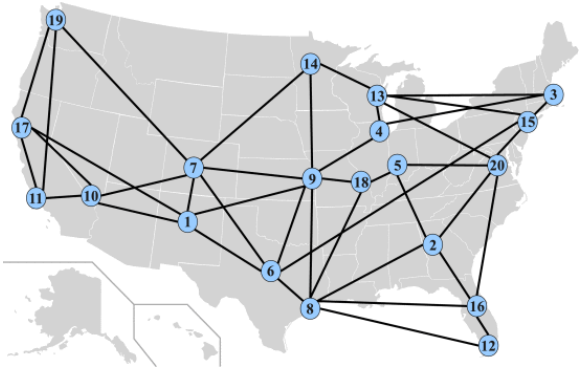


Fig. 1. Topology of a sample small-scale DCS.

The non-Markovian stochastic dynamics of DCSs have been simulated assuming that both, the service and the task transfer times, follow Pareto distributions. Pareto distributions have been selected because, after experimentally characterizing the dynamics of the testbed DCS described in Appendix B, available in the online supplemental material, the empirical probability distribution functions (pdfs) of service and task transfer times are best fitted by Pareto distributions. The average task-processing time of the HP servers was set to 1 s, while the standard deviation of the task-processing time was set to 0.25 s. The average task-processing time of the standard servers was set to ten times the average task-processing time of the HP

servers, and their standard deviation was set to 4 seconds. For the task transfer times, we follow [28]to introduce meaningful communication delays and define the average task-transfer time to be five times the average service time of the standard (slowest) servers. Further, the mean transfer time of l_{ij} tasks from the ith server to the jth, $\overline{T}_{ij}$, is calculated as $\overline{T}_{ij} = a_{ij} l_{ij} + b_{ij}$, where a_{ij} and b_{ij} are positive constants that depend on the link connecting the ith and the jth servers. The parameters a_{ij} and b_{ij} were set to 1 second per task and 2 seconds, respectively, as in [28].

Regarding failure times, we follow the results in [19] [20] and assume that the failure times in both cases, correlated and independent failures, follow Weibull distributions with the same average failure times, that is, the average failure time of a server that crashes independently is 600 s, while the average failure time of all those servers in a PSRG crashing simultaneously on the occurrence of a PSRG event is also 600 s. In addition, we have considered a scenario of relatively small and uniform failure probability for the servers conditional on a PSRG, namely, $p_i^A = 0.35$ for all i. To make fair comparisons, in our simulations we have adjusted the average number of failed servers to be the same for the cases of correlated and independent failures.

Regarding the workload processed by the DCSs, we assume in our calculations that an application composed of $M = 5{,}000$ tasks is allocated onto the CEs at time $t = 0$. We have considered four different initial allocations. The initial task allocations labeled as Uniform-1, Uniform-2, and Uniform-3 correspond, respectively, to an initial uniform allocation of tasks onto all the CEs, onto the HP servers only, and onto the standard CEs only, while the allocation labeled as Computing-Power corresponds to an initial allocation of tasks proportional to the relative processing speed of the CEs. In addition, a DTR policy with a reallocation criterion based solely on the relative processing speed of the CEs has been considered. All the estimated values reported here correspond to centers of intervals with 95 percent confidence, over which the estimates will not differ from the true value more than 5 percent.

Table 1 lists the optimal service reliability, for the four different initial task allocation considered, and for both cases independent and correlated failures. In the case of independent failures, the optimal service reliability was calculated by means of the pairwise decomposition of a DCS presented in [28]. In the case of correlated failures, the optimal service reliability was approximated by first generating a sample of PSRG correlated failures and, next, the conditional service reliability was calculated analytically by solving (9) and (10), and finally (1) was used to compute the estimated service reliability for a fixed DTR policy. It can be observed from Table 1 that, in spite of the DTR and the average number of failures are the same, PSRG correlated failures diminish the service reliability as compared to the case of independent failures. For the cases presented here, as a result of correlation in the failures, the service reliability has been reduced up to 21 percent for the 20-node DCS. The reduction in the service reliability is explained by the fact that the likelihood of failure of an HP server increases when correlated failures induced by a PSRG affect a DCS, as compared to the case of independent failures. For independent failures, any server may fail in the system; however, when a PSRG affects a DCS only a specific subset of servers is prone to fail. Recall that the average number of failures is the same for both independent and correlated failures, and recalling also that each PSRG, with the exception of PSRG-1 in the 20-node DCS, contains one or two HP servers. Thus, it can be easily observed that the likelihood of failure of an HP server increases in the presence of correlated failures induced by a PSRG. The same ideas explain also why independent and correlated failures yield approximately the same service reliability for the 20-node DCS in the case of the PSRG-1 event. In Appendix C, available in the online supplemental material, the negative effects of correlated failures induced by PSRG events on the average fraction of tasks served by the DCS are presented in detail as an additional result.

TABLE 1 The Service Reliability of Small- and Large-Scale Dcss in the Presence of Both Independent and Spatially Correlated Failures

20-node DCS

Initial Allocation	PSRG-1		PSRG-2		PSRG-3		PSRG-4	
	Indep.	Corr.	Indep.	Corr.	Indep.	Corr.	Indep.	Corr.
Uniform 1	0.835	0.830	0.869	0.681	0.866	0.694	0.822	0.775
Uniform 2	0.787	0.785	0.812	0.653	0.829	0.649	0.794	0.701
Uniform 3	0.820	0.818	0.850	0.673	0.852	0.682	0.813	0.746
Comp-Pwr.	0.877	0.874	0.897	0.681	0.889	0.691	0.874	0.757

Gnd5000 DCS

Initial Allocation	PSRG-1.		PSRG-2		PSRG-3		PSRG-4	
	Indep	Corr	Indep	Corr	Indep	Corr	Indep	Corr
Uniform	0.791I	0.527	0.762	0.646	0.781	**0.629**	**0.794**	**0.582**
Comp-Pwr	0.913	0.602	0.905	0.692	0.927	**0.688**	**0.916**	**0.657**

20-node DCS

Initial Allocation	PSRG-1		PSRG-2		PSRG-3		PSRG-4	
	Indep.	Corr.	Indep.	Corr.	Indep.	Corr.	Indep.	Corr.
Uniform 1	0.835	0.830	0.869	0.681	0.866	0.694	0.822	0.775
Uniform 2	0.787	0.785	0.812	0.653	0.829	0.649	0.794	0.701
Uniform 3	0.820	0.818	0.850	0.673	0.852	0.682	0.813	0.746
Comp-Pwr.	0.877	0.874	0.897	0.681	0.889	0.691	0.874	0.757

Grid5000 DCS

Initial Allocation	PSRG-1		PSRG-2		PSRG-3		PSRG-4	
	Indep.	Corr.	Indep.	Corr.	Indep.	Corr.	Indep.	Corr.
Uniform	0.791	0.527	0.762	0.646	0.781	0.629	0.794	0.584
Comp-Pwr.	0.913	0.602	0.905	0.692	0.927	0.688	0.916	0.657

We show in Fig. 2a the service reliability of the DCS as a function of the DTR policy, when the initial task allocation is Uniform-2 and the PSRG event 2 induces correlated failures in the system. The DTR policy is represented in the figure as the ratio of tasks exchanged among all the CEs. Since in the Uniform-2 allocation tasks are initially sitting at the standard CEs, the DTR policy showed in the figure corresponds to the case of transferring tasks from standard servers to HP servers. Results suggest that the service reliability, in the presence of correlated failures generated by a PSRG event, may drop between 5 and 25 percent as compared to the case of independent-failures. Once again, this result is attributed to the fact that, when correlated failures occur, it is more likely that an HP server fails as compared to the likelihood of failure of HP servers when independent failures affect the DCS. Moreover, Fig. 2a also illustrates the effect of properly selecting the number of tasks to migrate among the CEs: when the task transfer delays are negligible compared to the average service times of tasks, the optimal DTR policy corresponds to the initial partition specified by (2). However, when the task-transfer delays are not negligible, as in the example shown here, such selection is no longer optimal and by transferring only 95 percent (90 percent) of the tasks as specified by (2), a maximal service reliability of 0.812 (0.653) is achieved when independent (correlated) failures affect DCS's dynamics. In Fig. 2 b the service reliability of the DCS in the presence of both independent and correlated failures is depicted as a function of the instant when the DTR policy is executed by the CEs. Results suggest that an excessive delay in executing the DTR policy has the effect of considerably reducing the service reliability regardless of the type of failure affecting the DCS.

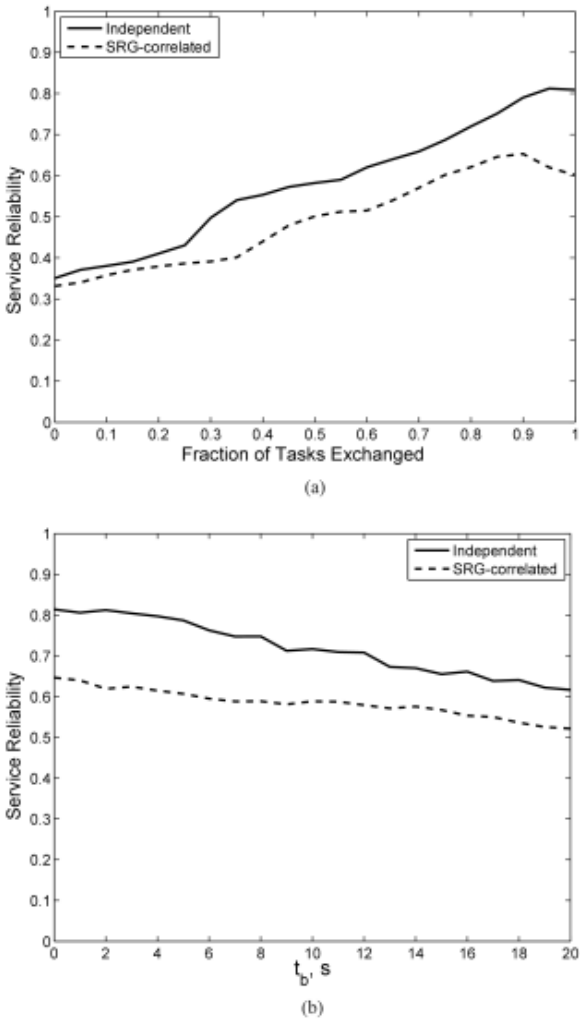


Fig. 2. The service reliability of the small-scale DCS as a function of the: (a) fraction of tasks reallocated; and (b) instant when the DTR policy is executed.

In order to experimentally validate our theory we coded the DTR policy for maximizing the service reliability on the testbed DCS described in Appendix B, available in the online supplemental material, and emulated the 20-node DCS shown in Fig. 1. In order to yield predictions for the service reliability, the random times driving the dynamics of the testbed must be first experimentally characterized. To do so, Pareto distributions were fitted for the service times and transfer times of the testbed. The average service times, the average transfer times, and the a_{ij} and b_{ij} parameters were estimated using maximum likelihood estimators as in [28]. From the experimental characterization, we have adjusted the DC application so that the average service times at the CEs are between 5 and 10 seconds for the standard servers and between 1 and 1.5 seconds for the HP servers. Also, traffic shapers have been set so that the estimated a_{ij} and b_{ij} parameters are between 5 and 10 seconds per task, and 1 to 3 seconds, respectively. The initial workload in the DCS was set to 2,000 tasks, the failure times of the CEs failing in a correlated manner was assumed to follow Weibull distributions with average failure times of 400 s. As in simulations, we considered a scenario of relatively small and uniform failure probability for the CEs, conditional on a PSRG, i.e., $p_i^A = 0.35$ for all i. Also, to make fair comparisons, we have adjusted the average number of failed CEs to be the same for the cases of correlated and independent failures. In the experimental setup, the reliability was calculated by averaging a total of 500 independent realizations of failure patterns for each policy, while in the case of the theoretical predictions 10,000 realizations of independent failures patterns were employed for each policy.

Fig. 3 shows the theoretical predictions and the experimental results for the service reliability in the presence of PSRG-2 correlated failures, as a function of the fraction of tasks reallocated among the servers, when the Uniform-1 initial task allocation was employed to partition the application at time $t = 0$. As in the case of simulations, in Fig. 3 two types of DTR policies have been considered: one that disregards the fact that failures occur in a correlated manner (labeled as "CF-Unaware") and another policy accounting for correlated failures (labeled as "CF-Aware"). First, observe that Fig. 3visually suggests a fairly good agreement between the theoretical predictions and the experimental results. This subjective assessment is confirmed by the fact that maximum absolute errors smaller than 4 percent have been obtained between the theoretical and curves obtained from the emulated DCS. As before, we see that if a DTR policy does not take into account the effects of correlated component failures then the service reliability is reduced by up to 22 percent. The larger reduction in the service reliability is observed in an operational regime where more tasks are exchanged among servers. Last, in Table 2 we list the maximal service reliability for both the correlated-failure-unaware and the correlated-failure-aware DTR policies for different initial allocation of tasks of the 2,000 tasks, when PSRG-2 events induce correlated failures. The optimal values for the service reliability have been obtained by solving the optimization problem (4). Table 2 shows that regardless of how the tasks have been initially allocated onto the servers, the service reliability is indeed greater when a DTR policy exploits the knowledge about the correlated component failures occurring in the system.

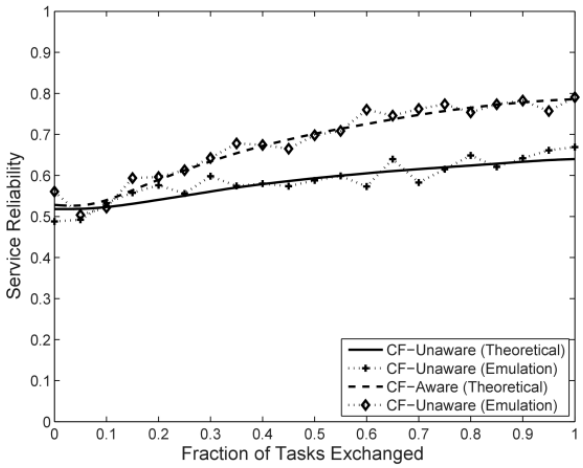


Fig. 3. The service reliability as a function of fraction of tasks reallocated among the CEs, when independent and correlated failures affect an emulated DCS.

TABLE 2 Maximal Service Reliability for Two Types of DTR Policies and Different Initial Allocation of Tasks

Initial load	Maximal Service Reliability			
	Theoretical		Emulated	
	CF-Aware	CF-Unaware	CF-Aware	CF-Unaware
Uniform 1	0.795	0.653	0.772	0.667
Uniform 2	0.786	0.640	0.791	0.669
Uniform 3	0.751	0.622	0.772	0.614
Comp-Pwr.	0.702	0.608	0.726	0.615

Initial load	Maximal Service Reliability			
	Theoretical		Emulated	
	CF-Aware	CF-Unaware	CF-Aware	CF-Unaware
Uniform 1	0.795	0.653	0.772	0.667
Uniform 2	0.786	0.640	0.791	0.669
Uniform 3	0.751	0.622	0.772	0.614
Comp-Pwr.	0.702	0.608	0.726	0.615

Finally, we present in Appendix C, available in the online supplemental material, experiments conducted over a 38-nodes DCS, which show the diminishing effects of correlated failures on the service reliability.

1. Large-Scale Experiments

In order to study the effect of correlated failures in a more realistic scenario, we have analyzed the large-scale grid computing system called Grid5000 [29]. Grid5000 is a computing grid geographically distributed across nine cities, termed as sites, and composed of 15 clusters with a total number of 1,288 nodes. For simplicity, we have studied here Grid5000's behavior in the presence of correlated failures at the cluster level, that is, we considered the Grid5000 as a 15 server DCS where each one of the 15 servers summarizes the behavior of all the nodes forming the corresponding cluster. Since Grid5000 is geographically distributed across nine different sites, which host one or more clusters per site, we have defined PSRGs for all those sites containing two or more clusters. Thus, the so-called PSRG-1 is composed of clusters 1 to 4, the PSRG-2 is composed of clusters 7 and 8, the PSRG-3 is composed of clusters 9 and 10, while the PSRG-4 is composed of clusters 11 and 12. The remaining clusters (5, 6, 13, 14, and 15) are not associated to any SRGs and we have assumed that they fail independently. Based on this definition of PSRGs, we suppose that Grid5000 may be affected by four different PSRG events, each one of them associated to PSRGs 1 to 4. The likelihood of occurrence of each PSRG event is $\pi(A) = 0.25$.

To model the dynamics of the 15 clusters in Grid5000, we downloaded traces from The Grid Workload Archive, [30], and The Failure Trace Archive [25]. We pooled samples of all the nodes forming a cluster and employed parametric as well as non-parametric pdf estimation methods to fit proper distributions for both, the task execution times and the failure times, at the cluster level. Heterogeneity was introduced in the task execution times by filtering out the data, using the “Application class” field provided in the traces, and considering only those 15 application classes with larger number of samples. Table 6 in Appendix C, available in the online supplemental material, lists statistics and the fitted distributions for servers’ task execution times in Grid5000. Note that, at the cluster level, average task service times are highly heterogeneous as observed in Table 6 and as indicated by a coefficient of variation of 1.985. The heterogeneity in the service times is also observed noting that such times seem to follow different parametric distributions, such as Gamma, Extreme Value, and Pareto, as well as non-parametric distributions, such as mixtures of two and four Gaussian kernels.

Regarding failure times, we also pooled samples of all the nodes in a cluster and filtered out the data to consider only the availability times of the 15 clusters. In Table 6 statistics as well as the fitted distributions for the failure times in Grid5000 are listed. We note that, at the cluster level, the average failure times in Grid5000 are more-or-less homogeneous with a coefficient of variation of 0.645. To establish fair comparisons and simplify our simulations, the failure time of all the clusters within a PSRG is assumed to follow the distribution with the largest average failure time. In addition, as in the previous examples, we have considered a scenario of relatively small yet uniform failure probability for the clusters, conditional on a PSRG. We assumed that the ith cluster in a PSRG may fail with a probability of $p_i^A = 0.35$ for all i. Regarding task transfer times, we safely assumed that communication delays are negligible due to sites in

Grid5000 are interconnected using high-speed VLANs at a speed of 1 Gbps. Further, we assumed also that the topology of the network is a full-mesh.

As in the previous simulations, and in order to establish fair comparisons, we have adjusted the average number of failed servers to be the same for the cases of correlated and independent failures. The workload to be processed by the Grid5000 DCSs is composed of $M = 100{,}000$ tasks, which are allocated onto the servers, at time $t = 0$, using an uniform allocation of tasks, labeled as "Uniform," and an allocation proportional to the relative processing speed of the clusters, labeled as "Computing-Power." Note that, on average, the simulated Grid5000 DCS is capable of serving a workload of approximately 190,000 tasks. Regarding the DTR policy, we have employed again a reallocation criterion based on the relative processing speed of the servers.

In Table 1 we have also listed the optimal service reliability for the two different initial task allocations considered, and when independent and correlated failures affect the Grid5000 DCS. As in the case of the 20-node DCSs, results in Table 1 show that, regardless of the DTR policy employed by the system, correlated failures heavily reduce the service reliability as compared to the case of independent failures. Results also show that the largest reduction in the service reliability occurs when the PSRG-1 is affected by correlated failures. This is attributed to the following issues. First, a correlated failure occurring at the PSRG-1 produces the simultaneous failure of four clusters, meaning that about 25 percent of the total number of clusters in the DCS becomes unavailable. Second, the four clusters in the PSRG-1 have a combined processing power of 0.886 tasks per second, which represents 60 percent of the total computing power of the DCS. Third, the DTR policy used by the DCS reallocates more workload onto the clusters 3, 4, and 12 since they have the largest processing speeds. Consequently, a correlated failure at the PSRG-1 reduces the likelihood of serving the workload because of the simultaneous reduction in the number of available clusters, the large reduction in the processing power of the Grid5000 DCS, and the incorrect reallocation of workload onto clusters that are prone to fail in a correlated manner. We comment that the second larger reduction in the service reliability occurs when correlated failures affect the PSRG-4. Such a reduction is explained because: 1) PSRG-4 has a combined processing power of 0.550 tasks per second, which represents about 37 percent of the total computing power of the DCS; and 2) PSRG-4 contains cluster 12, which has the second larger processing speed in the system, making such cluster an excellent candidate cluster to receive reallocated workload.

1. Discussion

The reduction in the service reliability in the presence of correlated failures is a consequence of using a non-suitable DTR policy, which is not aware of the type of correlation present in the failure patterns. In order to observe the effect on the service reliability of including the information about the correlation induced by PSRG in the failure patterns, we compare the DTR reallocation criterion based solely on the processing speed of servers (labeled as "CF-Unaware") and a DTR with a correlated-failure-aware policy (labeled as "CF-Aware") that we proposed in (3). In this example (as in the previous cases), the workload is distributed at time $t = 0$ using four different allocations. The comparison results are listed in Table 3. It can be clearly observed that by including the information about the correlation induced by the PSRGs into a DTR policy a considerable increase in the service reliability is obtained as compared to a policy that neglects such information. Moreover, by comparing Tables 1 and 3 it can be observed that for the 20-node DCS, the optimal values for reallocating tasks dictated by the correlated-failure-aware DTR policy increase the service reliability to a level which is almost the same as in the case of independent failures, thereby compensating the negative impact induced by correlated failures on the system's reliability. This compensation effect is achieved because the correlated-failure-aware policy does: 1) smartly move the

calculations (tasks) away from those servers that are prone to fail in a correlated manner, to all those servers that do not share the same likelihood of failure; and 2) simultaneously leave a small fraction of the load on those servers that may fail in a correlated manner, thereby exploiting their computing capabilities. These two features of the correlated-failure—aware DTR policy devised in this paper are key, since they represent the fundamental tradeoff appearing in DC in the presence of correlated failures. On one hand, migrating less tasks among the CEs is suitable in scenarios when CEs work independently; as such, the effect of correlated failures can be partially compensated. On the other hand, migrating more tasks among the CEs is suitable for cooperative work and it increases the coupling between the CEs, which, in turn, has a negative effect on the reliability if correlated failures are not accounted for in a DTR policy. As an additional result, in Appendix C, available in the online supplemental material, we compare the CF-Aware and the CF-Unaware DTR policies and their effect of the service reliability for the case of the Uniform-2 initial task allocation.

TABLE 3 Service Reliability of Small- and Large-Scale DCSs Achieved by Correlated-Failure Aware and Unaware DTR Policies

LU-node UL

	PSRG-1		PSRG-2	
Initial Allocation	DTR policy		DTR policy	
	CF-Aware	CF-Unaware	CF-Aware	CF-Unaware
Uniform 1	0.841	0.830	0.823	0.681
Uniform 2	0.775	0.785	0.801	0.653
Uniform 3	0.819	0.818	0.810	0.673
Comp-Pwr.	0.877	0.874	0.837	0.681
	PSRG-3		PSRG-4	
Initial Allocation	DTR policy		DTR policy	
	CF-Aware	CF-Unaware	CF-Aware	CF-Unaware
Uniform 1	0.831	0.694	0.846	0.775
Uniform 2	0.809	0.649	0.828	0.701
Uniform 3	0.817	0.682	0.814	0.746
Comp-Pwr.	0.825	0.691	0.829	0.757

Gnd5000 DCS

	P KG-1		PSRG-2	
Initial Allocation	DTR policy		DTR policy	
	CF-Aware	CF-Unaware	CF-Aware	CF-Unaware
Uniform	0.631	0.527	0.660	0.646
Comp-Pwr.	0.781	0.602	0.796	0.692
	PSRG-3		PSRG-4	
Initial Allocation	DIR policy		DIR policy	
	CF-Aware	CF-Unaware	CF-Aware	CF-Unaware
Unitorm	0.671	0.629	0.644	0.584
Comp-Pwr.	0.791	0.688	0.795	0.657

20-node DCS				
	PSRG-1		PSRG-2	
Initial Allocation	DTR policy		DTR policy	
	CF-Aware	CF-Unaware	CF-Aware	CF-Unaware
Uniform 1	0.841	0.830	0.823	0.681
Uniform 2	0.775	0.785	0.801	0.653
Uniform 3	0.819	0.818	0.810	0.673
Comp-Pwr.	0.877	0.874	0.837	0.681
	PSRG-3		PSRG-4	
Initial Allocation	DTR policy		DTR policy	
	CF-Aware	CF-Unaware	CF-Aware	CF-Unaware
Uniform 1	0.831	0.694	0.846	0.775
Uniform 2	0.809	0.649	0.828	0.701
Uniform 3	0.817	0.682	0.814	0.746
Comp-Pwr.	0.825	0.691	0.829	0.757
Grid5000 DCS				
	PSRG-1		PSRG-2	
Initial Allocation	DTR policy		DTR policy	
	CF-Aware	CF-Unaware	CF-Aware	CF-Unaware
Uniform	0.631	0.527	0.660	0.646
Comp-Pwr.	0.781	0.602	0.796	0.692
	PSRG-3		PSRG-4	
Initial Allocation	DTR policy		DTR policy	
	CF-Aware	CF-Unaware	CF-Aware	CF-Unaware
Uniform	0.671	0.629	0.644	0.584
Comp-Pwr.	0.791	0.688	0.795	0.657

For the Grid5000 DCS, we note that the use of the correlated-failure—aware DTR policy enhances the service reliability of the system up to 28 percent, as compared to the use of a policy that unwisely disregards the effect of correlated failures. Unlike the case of the 20-node DCS, the maximal service reliability yielded by the correlated-failure—aware DTR policy does not achieve the same level of reliability as in the case of independent failures. This is due to the fact that those clusters with the larger individual and combined processing capabilities are more likely to fail in a correlated manner because they belong to the PSRG-1 and to the PSRG-4. Thus, the correlated-failure—aware policy is severely constrained and cannot reallocate enough tasks to the fastest clusters in the system, and consequently, it is capable of only partially compensating for the negative impact of correlated failures on the system's reliability.

SECTION 6. Conclusions

This paper sheds light on the impact of spatially correlated failures on the reliability of DCSs and presents strategies for how to mitigate the adverse effects of correlated failures using simple preemptive DTR policies that are aware of the failure statistics. The concept of SRLG, which is used in the routing community as an effective mechanism to model and simulate correlated failures, has been adapted in this paper to introduce the idea of PSRGs. Under this concept, correlated failures resulting from real-world massive disruptions and/or physical attacks to the DCS infrastructure can be modeled, and correlation can be introduced by defining or identifying those CEs that may suffer from a common stress event.

The effects of correlated failures on the service reliability have been investigated thoroughly using the proposed reliability model. Results show that correlated failures reduce both the service reliability and the average number of tasks executed by a DCS as compared to scenarios of independent failures. Notably, a correlated-failure—aware DTR policy has been proposed in order to enhance the service reliability of the system. Moreover, it has been observed that the optimal selection of the number of tasks to be reallocated among the CEs depends upon the degree of correlation in the failures.

ACKNOWLEDGEMENT

This work was supported by Defense Threat Reduction Agency (Combating WMD Basic Research Program). J.E. Pezoa was also supported by CONICYT, FONDECYT Iniciación Folio 11110078. The authors wish to thank the help of the Center for Advanced Research Computing at the University of New Mexico.

References

1. M. Litzkow, "Condor - A Hunter of Idle Workstations", *Proc. Eighth Int'l Conf. Distributed Computing Systems*, pp. 104-111, 1988.

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

3. J. Palmer, "Empirical and Analytical Evaluation of Systems with Multiple Unreliable Servers", *Proc. Int'l Conf. Dependable Systems and Networks*, pp. 517-525, 2006.

4. B. Schroeder, "A Large-Scale Study of Failures in High- Performance Computing Systems", *Proc. Int'l Conf. Dependable Systems and Networks*, pp. 249-258, 2006.

5. D. Kondo, "On Correlated Availability in Internet- Distributed Systems", *Proc. Ninth IEEE/ACM Int'l Conf. Grid Computing (GRID '08)*, pp. 276-283, 2008.

6. M. Gallet, "A Model for Space-Correlated Failures in Large-Scale Distributed Systems", *Proc. 16th Euro-Par Conf. Parallel Processing*, pp. 88-100, 2010.

7. P. Joshi, *Prefail: A Programmable Failure-Injection Framework*, Apr. 2011.

8. Y.S. Dai, "Reliability Analysis of Grid Computing Systems", *Proc. Pacific Rim Int'l Symp. Dependable Computing*, pp. 97-103, 2002.

9. J.E. Pezoa, "Performance and Reliability of Non-Markovian Heterogeneous Distributed Computing Systems", *IEEE Trans. Parallel and Distributed Systems*, vol. 23, no. 7, pp. 1288-1301, July 2012.

10. S. Nath, *Tolerating Correlated Failures in Wide-Area Monitoring Services*, 2004.

11. Z. Zhang, "Proactive Failure Management for High Availability Computing in Computer Clusters", *Proc. Third Int'l Joint Conf. Computational Science and Optimization*, pp. 377-381, 2010.

12. K. Kim, "Assessing the Impact of Geographically Correlated Failures on Overlay-Based Data Dissemination", *Proc. IEEE GLOBECOM*, pp. 1-5, 2010.

13. N.H. Azimi, "Data Preservation Under Spatial Failures in Sensor Networks", *Proc. 11th ACM Int'l Symp. Mobile Ad Hoc Networking and Computing*, pp. 171-180, 2010.

14. H. Weatherspoo, "Introspective Failure Analysis: Avoiding Correlated Failures in Peer-to-Peer Systems", *Proc. IEEE Symp. Reliable Distributed Systems*, 2002.

15. K. Goseva-Popstojanova, "Failure Correlation in Software Reliability Model", *IEEE Trans. Reliability*, vol. 49, pp. 37-48, Mar. 2000.

16. Y. Dai, "Modeling and Analysis of Correlated Software Failures of Multiple Types", *IEEE Trans. Reliability*, vol. 54, pp. 100-106, Mar. 2005.

17. Y.-S. Dai, "Performance and Reliability of Tree-Structured Grid Services Considering Data Dependence and Failure Correlation", *IEEE Trans. Computers*, vol. 56, pp. 925-936, July 2007.

18. L. Fiondella, "Estimating System Reliability with Correlated Component Failures", *Int'l J. Reliability and Safety*, vol. 4, no. 2/3, pp. 188-205, 2010.

19. D. Kondo, "The Failure Trace Archive: Enabling Comparative Analysis of Failures in Diverse Distributed Systems", *Proc. 10th IEEE/ACM Int'l Conf. Cluster Cloud and Grid Computing*, pp. 398-407, 2010.

20. A. Iosup, "On the dynamic resource availability in grids", *Proc. Eighth IEEE/ACM Int'l Conf. Grid Computing (GRID '07)*, pp. 26-33, 2007.

21. V. Shestak, "Robust Sequential Resource Allocation in Heterogeneous Distributed Systems with Random Compute Node Failures", *Proc. IEEE Int'l Symp. Parallel and Distributed Processing*, 2009.

22. T. Ma, "Evaluation of the QoS of Crash-Recovery Failure Detection", *Proc. ACM Symp. Applied Computing*, pp. 538-542, 2007.
23. D. Papadimitriou, "Inference of Shared Risk Link Groups Internet Draft IETF Internet Draft", 2002.
24. K. Lee, "Cross-Layer Survivability in WDM-Based Networks", *IEEE/ACM Trans. Networking*, vol. 19, no. 4, pp. 1000-1013, Aug. 2011.
25. "The Failure Trace Archive", *INRIA*, 2012.
26. M.J. Ciaraldi, "Risks in Anonymous Distributed Computing Systems", *Proc. Int'l Network Conf. (INC '00)*, 2000.
27. S. Neumayer, "Assessing the Vulnerability of the Fiber Infrastructure to Disasters", *IEEE/ACM Trans. Networking*, vol. 19, no. 6, pp. 1610-1623, Dec. 2011.
28. J.E. Pezoa, "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.
29. "The Grid5000", *INRIA*, 2012.
30. "The Grid Workload Archive", *PDS Group TU Delft*, 2012.
31. M. Dodge, The Atlas of Cyberspace, Addison Wesley, 2008.