

Cloud infrastructure planning considering different redundancy mechanisms

Erica Sousa¹ · Fernando Lins¹ ·
Eduardo Tavares² · Paulo Maciel²

Received: 28 March 2016 / Accepted: 19 December 2016 / Published online: 5 January 2017
© Springer-Verlag Wien 2017

Abstract Cloud computing has been established as a remarkable business model that allows the outsourcing of processing, memory, storage, and networking provided by cloud infrastructures. Cloud applications require computing resources being promptly available to serve the on-demand load. If resources are not available to meet user requests, service dependability and performance may be considerably impacted. Thus redundancy mechanisms are a suitable solution for rapidly providing resources to recover the failure in service delivery. However, one of the main difficulties in cloud infrastructures are related to the selection of redundancy mechanisms that can provide a high availability with acceptable costs. This paper provides a methodology, stochastic models and an optimization approach for assisting the planning of private cloud infrastructures, which are selected according to the availability, downtime and cost constraints. Two case studies based on cloud platform are adopted to demonstrate the feasibility of the proposed work.

✉ Erica Sousa
erica.sousa@ufrpe.br

Fernando Lins
fernando.aires@ufrpe.br

Eduardo Tavares
eagt@cin.ufpe.br

Paulo Maciel
prmm@cin.ufpe.br

¹ Department of Statistics and Informatics, Federal Rural University of Pernambuco, Recife, PE 52171-900, Brazil

² Center of Informatics, Federal University of Pernambuco, Recife, PE 50740-560, Brazil

Keywords Cloud computing · Dependability evaluation · Cost evaluation · Stochastic petri net · Reliability block diagram · Greedy randomized adaptive search procedure

Mathematics Subject Classification 60Gxx · 60Hxx · 68Wxx

1 Introduction

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [25].

High availability is an important challenge in cloud infrastructure planning, especially considering that failure events may degrade the quality of several hosted services [2]. In this context, redundancy is a suitable alternative for assuring strict quality of service levels, such as redundant equipment. Redundant equipment allows service continuation, minimizing the effects of the occurrence of failure events. However, the adoption of redundancy in cloud computing infrastructures increases design space, as issues related to quantity, type and cost of redundancy must be considered. A challenge in cloud infrastructure planning is the selection of the redundancy types that should be adopted to these infrastructures considering the required availability and financial cost [4, 32].

Dependability modeling has been widely adopted as an essential activity for improving infrastructure planning and reducing services cost [32]. There are two major categories of dependability models: combinatorial models (e.g.: RBD—reliability block diagram) [34] and state-space models (e.g.: SPN—stochastic petri net) [12, 22]. RBDs capture conditions that make a system fail (or to be working) in terms of structural relationships between the system components. SPNs represent the system behaviour (failures and repair activities) by its states and event occurrence expressed as state transitions [32]. A hierarchical and heterogeneous modeling allows the composition of models based on state-space and combinatorial models in order to mitigate problems related to representing larger systems as cloud infrastructures [32].

The proposed paper presents a methodology for cloud infrastructure planning. This methodology adopts optimization mechanism, stochastic models and cost equations for representation, evaluation and selection of private cloud infrastructures.

The optimization mechanism is based on greedy randomized adaptive search procedure (GRASP) [10], which generate prominent private cloud infrastructures meeting availability, downtime and cost constraints using the proposed dependability and cost models for evaluation.

The proposed stochastic models are based on SPN [12, 22] and RBD [34], which are combined using a hierarchical modeling approach [31] in order to mitigate issues to represent large private clouds. These models are able to represent cloud infrastructures with different redundancy mechanisms, such as cold standby, hot standby, warm standby and active-active redundancy mechanism, as well as allowing the assessment of the respective impact on availability and downtime. The cost models represent the acquisition cost of the equipment and software redundant, cost of the maintenance team, cost of the equipment replacement during the corrective maintenance.

For experimental results, this work adopts Eucalyptus platform [14] as the cloud computing framework, but the conceived models can be adopted to other platforms. A virtual learning environment, a webmail and a website configured on the private cloud are adopted as case studies, but this work can represent other applications configured on the private cloud.

This paper extends a previous work [31] in which a modeling strategy based on a hierarchical and heterogeneous modeling for cloud infrastructure planning is proposed. The proposed paper presents a methodology for cloud infrastructure planning and the detailing of the dependability evaluation activity. The detailing of this activity provides a better understanding of the representation of cloud infrastructures with redundant equipment through a hierarchical and heterogeneous modelling.

Another improvements of this proposed work is the presentation of models to represent the physical machine, virtual machine and management modules of the cloud infrastructure through the physical node model, virtual machine model and resource manager model, respectively. Some extensions of this proposed paper are the presentation of the active-active redundancy model to represent the active-active redundancy in virtual machines and the presentation of the maintenance model to represent the corrective maintenance in cloud infrastructures.

A differential of the proposed work is the representation of the maintenance team cost and equipment replacement cost during the corrective maintenance. Moreover, this paper presents an optimization approach for generating cloud infrastructures through the assignment of redundancy mechanisms to its components.

This paper is organized as follows. Section 2 presents the related works and Sect. 3 introduces an overview of basic concepts. Section 4 shows the proposed methodology. Section 5 presents the dependability and cost models and Sect. 6 presents the optimization model. Section 7 describes two case studies and some experimental results. Finally, Sect. 8 presents concluding remarks and future works.

2 Related work

Over the last years, some works have been proposed to evaluate dependability of cloud infrastructures. Wei et al. [35] present a hierarchical method based on heterogeneous models, combining RBD and SPN for dependability evaluation of virtual data center (VDC). In this method, a top-level model based on RBD represents the virtual data center infrastructure and a low-level model based on SPN contemplates the components of the VDC in failure and repair state. Dantas et al. [8,9] present a hierarchical and heterogeneous modeling to represent redundant architectures and compare their availability taking in account physical machines acquisition costs. In this method, a high-level model based on RBD represents the Eucalyptus platform subsystems and a low-level model based on Markov chains represents the respective subsystems employing warm-standby replication. Silva et al. [30] present a hierarchical and heterogeneous modeling strategy to dependability evaluation of services offered in cloud computing located in geographically distributed data centers, considering the occurrence of disasters. The strategy combines models based on SPN and

RBD. This strategy evaluates the impact of virtual machines migration in distributed systems in different data centers through dependability metrics.

Other works propose the cost evaluation of cloud infrastructures. Martens et al. [21] shows that the analysis of relevant cost types and factors of cloud computing services is an important pillar of decision-making in cloud computing management. In this way, such paper presents a total cost of ownership (TCO) approach for cloud computing services. Li et al. [18] provides metrics and equations for calculating the cloud total cost of ownership (TCO) and utilization cost, considering the elastic feature of cloud infrastructure and the adopted virtualization technology. This paper [18] provides a foundation for evaluating economic efficiency of cloud computing and it provides indications for cost optimization of cloud infrastructures.

Some works present an optimization model for data center planning that compose the cloud infrastructure. Callou et al. [3] proposes an integrated approach to evaluate and optimize dependability, cost and sustainability issues of data center infrastructures. Callou's approach utilizes a methodology that takes into account reliability block diagram, stochastic Petri nets and energy-flow models, as well as an optimization method based on GRASP. Ferreira et al. [11] proposes a power load distribution algorithm (PLDA) based on the Ford–Fulkerson technique to optimize energy distribution of data center power infrastructures.

Differently from previous studies, this paper proposes a methodology and models for private cloud infrastructure planning. An optimization model provides cloud infrastructures with different redundant components. Dependability and cost models evaluate the availability, downtime and costs of these cloud infrastructures. These results are adopted to select the private cloud infrastructure that meet the dependability and cost constraints. The proposed approach provides optimal private cloud infrastructure according to the established requirements.

3 Preliminaries

This section presents an overview of prominent concepts for a better understanding of this work.

The dependability of a system can be understood as the ability to deliver a set of services that can be justifiably trusted. Indeed, dependability is related to disciplines such as reliability, availability and maintainability [17,34].

Reliability is the probability of a system making its predefined functions without failures for a specified period of time: $R(t) = P\{T > t\}$, which T is the random variable representing the time to failure of the system (or a single component) [19].

Availability is the probability of a system being in a functioning condition. It considers the alternation of operational and nonoperating states. Steady-state availability (A) is commonly adopted, and the following equations are also taken into account: $A = \text{uptime}/(\text{uptime} + \text{downtime})$, or

$A = \text{MTTF}/(\text{MTTF} + \text{MTTR})$. MTTF is the mean time to failure and MTTR is the mean time to repair, such that [19].

Maintainability is the probability that a failed system can be made operable in a specified period of time: $M(t) = 1 - \exp^{-(\mu)t}$, where t is the random variable repre-

senting the time to repair of the system (or a single component) and μ is the repair rate [19].

Dependability metrics (e.g., availability, reliability and downtime) might be calculated either by using combinatorial models (e.g., RBD) and state-space based models (e.g., SPN). In general, Petri nets are a bipartite directed graph, in which places (represented by circles) denote local states and transitions (depicted as rectangles) represent actions. Arcs (directed edges) connect places to transitions and vice-versa. This work are adopting a particular extension, namely, stochastic petri nets (SPN) [20,24], which allows the association of probabilistic delays to transitions using the exponential distribution or zero delays to immediate transitions (depicted as thin black rectangles). Besides, reliability block diagram (RBD) [34] are graphically represented by blocks, in which are arranged using the following composition mechanisms: series, parallel, bridge, k-out-of-n blocks, or a combination of previous compositions.

Reliability block diagrams allow the representation of component networks and provide closed-form equations, so the results are usually obtained faster than using simulation or numerical analysis performed in state-based models [19]. Nevertheless, when facing the representation of dynamic redundancy mechanisms, combinatorial models experience drawbacks concerning the thorough handling of failures, activations, and repairing dependencies [34].

On the other hand, state-space based models can consider those dependencies, so allowing representing complex redundant mechanisms. However, such methods are more complex and suffer from the state-space explosion. Some of those formalisms allow both numerical analysis and stochastic simulation and SPN is one of the most prominent models of such class [19].

4 A methodology for cloud infrastructure planning

This work presents a methodology (see Fig. 1) for cloud infrastructure planning by assigning redundancy mechanisms to the cloud infrastructure components in order to meet availability, downtime and cost constraints. This proposed methodology is divided into five activities: dependability and cost planning; dependability evaluation; cost evaluation; analysis of dependability and cost scenarios and selection of dependability and cost scenarios.

Initially, dependability and cost planning activity provides the type and number of redundancy mechanisms attributed to cloud platform components. This activity concerns the attribution of redundancy mechanism types (e.g.: active-active, cold standby, hot standby, warm standby and none) to components of the Eucalyptus platform and network equipment (e.g.: CLC, CC, NC, VM, RT and SW) thought optimization approach. It allows the creation of cloud infrastructures with different redundancy mechanism types. The number of each component of the cloud infrastructure can be defined according to architecture adopted. This activity also provides the MTTR and MTTF of the Eucalyptus platform components and network equipment, activation time of the redundant components, maintenance parameters and dependability and cost requirements.

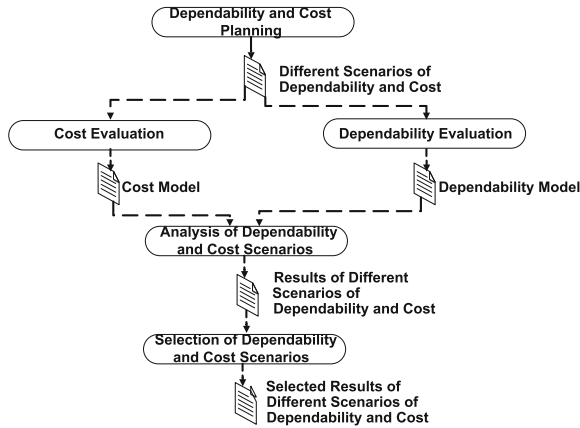


Fig. 1 Methodology for cloud infrastructure planning

Dependability evaluation activity adopts a heterogeneous and hierarchical modeling to represent cloud infrastructures with different redundancy mechanism types. This modeling considers the advantages of both stochastic Petri nets and reliability block diagrams to mitigate the complexity for representing cloud infrastructures. More specifically the most suitable model is selected for representing a cloud infrastructure subsystem and the results are combined to obtain the cloud infrastructure model. For each cloud infrastructure subsystem model, the mean time to failure (MTTF) and mean time to repair (MTTR) are computed. After obtaining the MTTF and MTTR, the cloud infrastructure model is generated.

Cost evaluation activity represents the costs of the redundant component, maintenance team and equipment replacement through equations.

Analysis of dependability and cost scenarios activity estimates the availability, downtime and costs of the cloud infrastructures with different redundancy mechanism types. These metrics are calculated through the dependability and cost models.

Selection of dependability and cost scenarios activity provides cloud infrastructures with availability, downtime and costs which are in accordance with the user requirements.

4.1 Dependability evaluation

The Dependability evaluation activity (see Fig. 2) is composed of seven activities: system understanding, parameter and metric definition, subsystem modeling, RBD modeling, SPN modeling, metric mapping and final model evaluation.

System understanding identifies the characteristics of the cloud platform, its components and hosted service on cloud computing. This activity also defines cloud platform dependability requirements that can be estimated in the final model evaluation.

Parameter and metric definition contemplates the dependability parameter (i.e., mean time to failure) of each cloud platform component and network equipment, redundancy parameter, maintenance parameter and the metrics for the cloud infras-

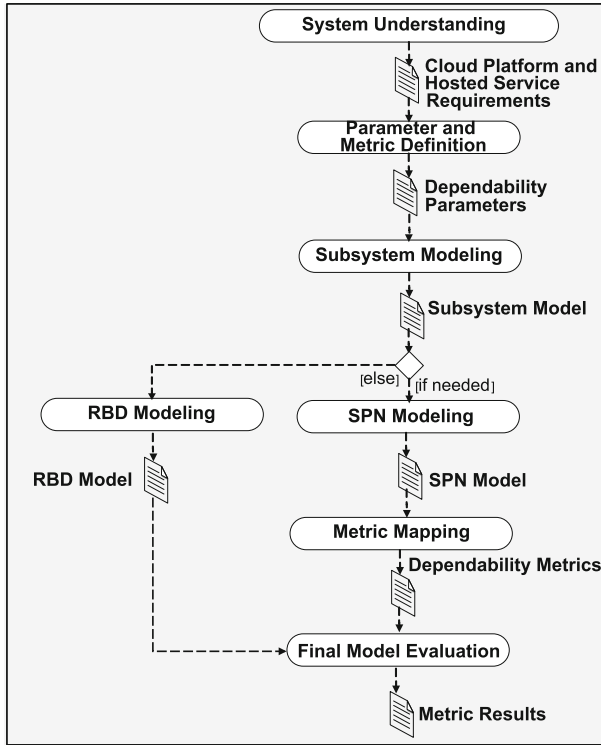


Fig. 2 Dependability evaluation

structure. The dependability parameters may be obtained using component datasheets or historical data. In this work, the metrics of interest are availability and downtime.

Subsystem modeling provides the generation of low-level models to represent the components of the cloud infrastructure and hosted service based on reliability block diagrams.

Reliability block diagram modeling contemplates the generation of high-level models to represent the Eucalyptus infrastructure based on reliability block diagrams. These models are adopted to estimate the availability and downtime of the cloud infrastructure whenever there is no dependence of failure and repair. For instance, RBD is adopted to model hot standby redundancy.

Stochastic petri net modeling contemplates the generation of high-level models to represent the systems of the cloud infrastructure based on stochastic Petri nets. In this case, SPN deals with components with dependencies regarding failure and repair. Redundancy mechanisms based on active-active, cold standby and warm standby are modeled using SPN models.

Metric mapping corresponds to the representation process of the dependability metrics through reference to the elements of the SPN. One example of these metrics is the availability.

Final model evaluation estimates the metrics of interest using the generated models.

5 Dependability and cost models

This section presents dependability and cost models for representing the cloud infrastructures and quantifying availability, downtime and redundancy cost. The dependability models are based on stochastic Petri nets or reliability block diagrams and the cost models are mathematical equations.

5.1 Dependability models

This section presents the models conceived for dependability evaluation of the Eucalyptus platform, Eucalyptus platform subsystems and redundancy mechanisms. Eucalyptus model represents the Eucalyptus platform. Physical node model, virtual machine model and resource machine model represent the Eucalyptus platform subsystems. The active-active redundancy model, hot standby model, cold standby model and warm standby model represent redundancy mechanisms assigned to Eucalyptus infrastructure components. The maintenance model represents the corrective maintenance of the Eucalyptus platform.

5.1.1 Eucalyptus model

The Eucalyptus model represents the cloud infrastructure configured with the Eucalyptus platform [14] and network equipment. Eucalyptus platform is composed of cloud controller (CLC), cluster controller (CC), node controller (NC) and virtual machine (VM). This model describes the overall cloud infrastructure state given the state of its components via reliability block diagrams [31]. The Eucalyptus model represents the cloud infrastructure in operational state when all its components are operational. Figure 3 shows the RBD adopted for estimating the availability and downtime of cloud infrastructure using the mean time to failure (MTTF) and mean time to repair (MTTR) of cloud infrastructure components (cloud controller (CLC), cluster controller (CC), node controller (NC), virtual machine (VM), switch (SW) and router (RT)).

Particularly, availability is estimated using expression $A_{cp} = \prod_{i=1}^6 A_i$ [19], in which A_i is steady state availability of an Eucalyptus infrastructure component, indicating that the system is operational when all components are operational [34]. Downtime is calculated using expression $D_{cp} = (100 - A_{cp}) \times 8760$ in hours for year [19].

5.1.2 Physical node model

The RBD model of the physical node represents the processing infrastructure (PI), primary memory (PM) and secondary memory (SM) of the physical machine (see

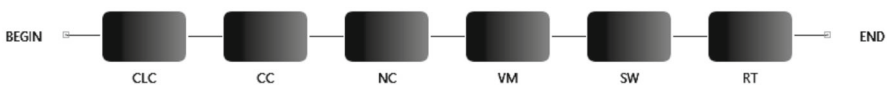


Fig. 3 Eucalyptus model

Fig. 4 Physical node model

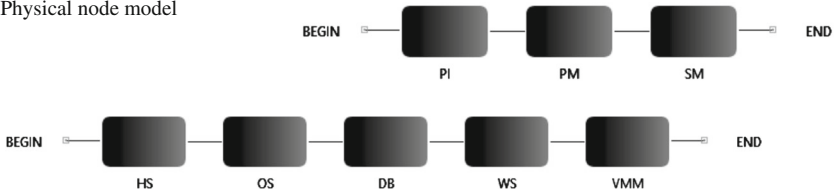


Fig. 5 Virtual machine model

Fig. 4). This RBD model is adopted to estimate the MTTF and MTTR of the physical machine.

5.1.3 Virtual machine model

The RBD model of the virtual machine represents the software configured on virtual machine (see Fig. 5). As an example, hosted service (HS), operating system (OS), database (DB), web server (WS) and virtual machine monitor (VMM). This RBD model is adopted to estimate the MTTF and MTTR of the virtual machine.

5.1.4 Resource manager model

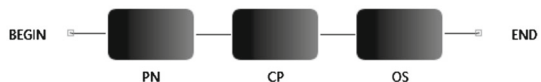
The RBD model of the resource manager is adopted to estimate the MTTF and MTTR of the resource managers of the cloud platforms through the reliability block diagrams of the physical node (PN), cloud platform (CP) and operating system (OS) (see Fig. 6).

5.1.5 Active-active redundancy model

The active-active redundancy mechanisms are employed when both primary and secondary virtual machine of the Eucalyptus infrastructure meet the requests of the user. This SPN model represents the virtual machine infrastructure with the active-active redundancy mechanism. The Fig. 7 shows the SPN model adopted to estimate the availability and downtime of the virtual machine with active-active redundancy.

The markings of the places VM1_ON and VM2_ON denote the operational states of the main virtual machine and redundant virtual machine, respectively, and the markings of the places VM1_OFF and VM2_OFF denote the failure states of these components. The main virtual machine and the redundant virtual machine share the system workload. When the main virtual machine fails, the redundant virtual machine becomes responsible for the system workload. The timed transitions MTTF_VM1, MTTF_VM2, MTTR_VM1 and MTTR_VM2 represent the occurrence of failure events and repair activities on the main virtual machine and redundant virtual machine,

Fig. 6 Resource manager model



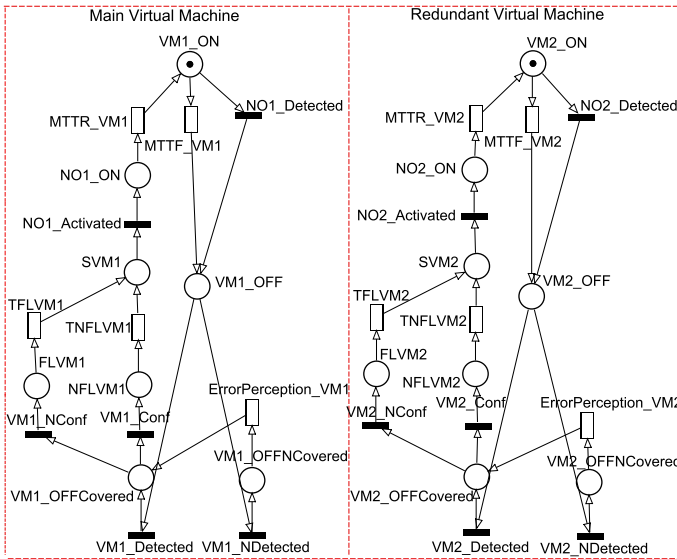


Fig. 7 Active-active redundancy model

and the times associated with these timed transitions represent the MTTF and MTTR of these components.

The virtual machines are instantiated on the physical machine where the NC service is provided. Each virtual machine fails when the NC fails or the virtual machine fails. The enabling function ($\#NC_ON=0$) allocated to the immediate transitions $NO1_Detected$ and $NO2_Detected$ represents the failure of each virtual machine due to occurrence of failure events in the NC. The virtual machines are repaired when the NC is operational. This condition is represented by the enable function ($\#NC_ON=1$) assigned to the immediate transitions $NO1_Activated$ and $NO2_Activated$.

After the occurrence of failure event on the virtual machine with the firing of the immediate transitions $MTTF_VM1$ and $MTTF_VM2$, this failure event can be detected with the firing of the immediate transitions $VM1_Detected$ and $VM2_Detected$ and can not be detected with the firing of the immediate transitions $VM1_NDetected$ and $VM2_NDetected$. The probability of the detection and non-detection of failure events is represented by the weights associated with the immediate transitions $VM1_Detected$, $VM2_Detected$, $VM1_NDetected$ and $VM2_NDetected$. The places $VM1_OFFCovered$ and $VM2_OFFCovered$ denote the detection of the failure event and the places $VM1_OFFFNcovered$ and $VM2_OFFFNcovered$ represent the non-detection of the failure event. When the failure is not detected, there is a perception error. The timed transitions $ErrorPerception_VM1$ and $ErrorPerception_VM2$ represent this perception error and the times associated to these transitions represent the duration of this perception error. After this time, the failure is perceived.

The main virtual machine and redundant virtual machine share the system workload. When the main virtual machine fails, the system can be configured to send the workload only for the redundant virtual machine. The system configuration for sending all

workload for the redundant virtual machine occur after the firing of the transitions VM1_Conf and TNFLVM1 or of the transitions VM2_Conf and TNFLVM2. The places NFLVM1 and NFLVM2 represent the start of the system configuration and the places SVM1 and SVM2 represent the completion of the system configuration.

When the system configuration fails, the workload is sent to the main virtual machine and redundant virtual machine. However, the requests sent to the main virtual machine are not met. The places FLVM1 and FLVM2 represent the not performing of the system configuration. While the system cannot be configured to send all workload for the redundant virtual machine, the requests are not met. In this case, the system configuration occurs after the firing of the immediate transitions VM1_NConf and TFLVM1 or VM2_NConf and TFLVM2. The probability of the system configuration and non-system configuration is modeled by weights associated to the immediate transitions VM1_Conf, VM2_Conf and VM1_NConf, VM2_NConf, respectively.

The detection of the failure and the system configuration for sending the workload for the redundant virtual machine allows the repair of the system after the firing of the timed transitions MTTR_VM1 and MTTR_VM2.

The following statements are adopted for estimating availability:

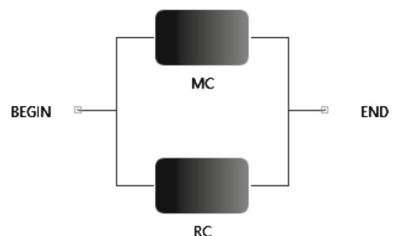
$A_{aa} = P\{(\#VM1_{ON} = 1 \text{ OR } \#VM2_{ON} = 1)\}$, which indicates the probability of the inner expression [36] and Downtime is calculated using expression $D_{aa} = (100 - A_{aa}) \times 8760$ in hours for 1 year [19].

5.1.6 Hot standby model

The active-passive redundancy mechanisms are adopted when the main components deal with requests of the client system and the redundant components are in standby. The hot standby model represents the hot standby redundancy mechanism through reliability block diagrams. Cold standby model and warm standby model represent the cold standby and warm standby redundancy mechanisms through stochastic Petri nets.

Hot standby model represents a component with the hot standby redundancy. In the hot standby redundancy, the failed component is replaced without significant delay since the spare modules are also powered [29, 32, 34]. The hot standby model depicts the model in which an active component (MC) has a hot standby spare (RC). Figure 8 shows the adopted RBD for estimating the availability and downtime of a component with hot standby redundancy through the MTTF and MTTR of the component and hot standby redundancy.

Fig. 8 Hot standby model



Availability is estimated using the expression $A_{hs} = 1 - \prod_{i=1}^2 (1 - A_i)$ [19], in which A_i is steady state availability of the component or hot standby redundancy, meaning that the system is operational whenever when at least one component is operational [29,34]. Downtime is calculated using expression $D_{hs} = (100 - A_{hs}) \times 8760$ in hours for 1 year [19].

5.1.7 Cold standby model

A component with cold standby redundancy is based on a non-active spare module that waits to be activated when the (main) active module fails. Hence, when the main module fails, the spare module’s activation takes a certain amount of time to be activated. This time period is named mean time to activate (MTA). As the spare component is switched off, it is considered that it does not fail until becoming operational [29,34].

The cold standby model is depicted in Fig. 9. Places Component_ON, Spare_ON, Component_OFF and Spare_OFF denote the operational and failure states for the main and spare modules, respectively. The spare module is initially deactivated, since no tokens are initially stored in places Spare_ON and Spare_OFF. As the main module fails, the transition ActiveSpare is fired. This transition ActiveSpare delay represents the Mean Time to Activate (MTA) and a marking in place WaitSpare denotes the spare module is not operational. Transitions MTTF_Component, MTTF_Spare, MTTR_Component and MTTR_Spare represent a failure as well as a repair and its delay represents MTTF and MTTR, respectively. The MTTF and MTTR associated to the main module may be different from the spare one.

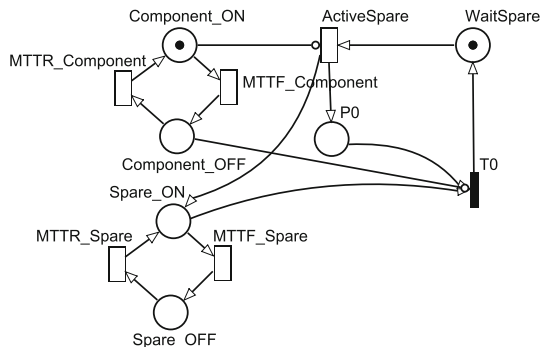
The following statements are adopted for estimating availability:

$A_{cs} = (P\{\#Component_ON = 1 \text{ OR } \#Spare_ON = 1\})$ [36] and Downtime is calculated using expression $D_{cs} = (100 - A_{cs}) \times 8760$ in hours for 1 year [19].

5.1.8 Warm standby model

A component with warm standby redundancy is based on a non-active spare module that waits to be activated when the active module fails. The difference with the cold standby is that the active and spare modules have failure rates λ and spare module has a failure rate ϕ when it is de-energized, considering $0 \leq \phi \leq \lambda$.

Fig. 9 Cold standby model



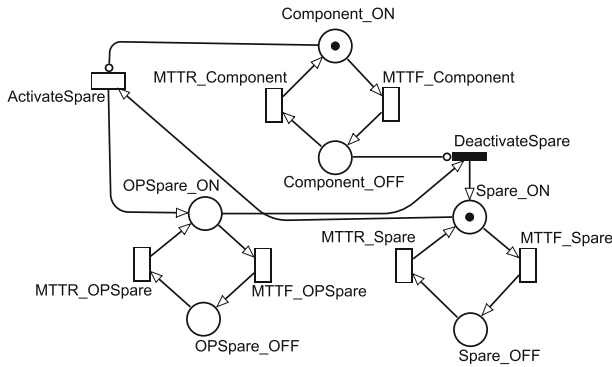


Fig. 10 Warm standby model

The SPN model [13] (see Fig. 10) includes six places, Component_ON, Component_OFF and their similar pairs. Places Spare_ON and Spare_OFF represent the spare of Component in non-operational state. Places OPSpare_ON and OPSpare_OFF represent the spare in operational state. At the moment that the main module fails, the transition ActivateSpare is enabled. Its firing represents the start of the spare in operational state. This period is named Mean Time to Activate (MTA). The immediate transition DeactivateSpare represents the return to normal operation after a failure. Transitions MTTT_Component, MTTT_Spare, MTTT_OPSPare, MTTR_Component, MTTR_Spare and MTTR_OPSPare represent a failure as well as a repair and its delay represents MTTT and MTTR, respectively.

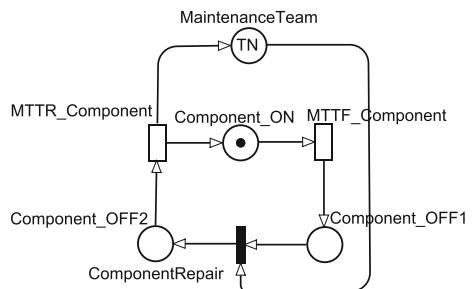
The following statements are adopted for estimating availability:

$A_{ws} = (P\{\#Component_ON = 1 \text{ OR } \#Spare_ON = 1 \text{ OR } \#OPSPare_ON = 1\})$ [36] and Downtime is calculated using expression $D_{ws} = (100 - A_{ws}) \times 8760$ in hours for 1 year [19].

5.1.9 Maintenance model

The maintenance model is based on stochastic Petri nets and represents the allocation of maintenance teams for corrective maintenance of a cloud infrastructure component. Figure 11 shows the SPN model adopted to estimate the availability and downtime of the cloud infrastructure.

Fig. 11 Maintenance model



The following statements are adopted for estimating availability:

$A_m = (P\{\#Component_ON = 1\})$, which indicates the probability of the inner expression [36] and Downtime is calculated using expression $D_m = (100 - A_m) \times 8760$ in hours for 1 year [19].

5.2 Cost model

This section presents the models conceived for cost evaluation. This paper deals with the costs for redundant components, maintenance team and component replacement.

5.2.1 Redundant component cost model

The cost for redundant components is represented by $RCC = \sum_{i=1}^N RCN_i \times RC_i$, which contemplates all hardware and software components adopted by the redundancy mechanisms in the cloud infrastructure. N denotes the number of distinct redundancy types (e.g.: active-active redundancy, hot, cold and warm standby module), RCN_i is the amount of a specific type of redundancy, and RC_i indicates the unit cost of a specific type of redundancy.

5.2.2 Maintenance team cost model

This model consists of the expenditures for the maintenance team, and it is represented by $MTC = \sum_{i=1}^N MTN_i \times MTC_i \times MTT_i$. N denotes distinct expertise types of the maintenance team. MTN_i is the number of maintenance team members with specific expertise. MTC_i represents the unit cost of maintenance team member with specific expertise. MTT_i indicates the work time of the maintenance team member with particular expertise.

5.2.3 Equipment replacement cost model

This model consists of the expenditures with the equipment replacement which are obtained through the $ERC = \sum_{i=1}^N REN_i \times EMT_i \times REC_i$. N denotes the replaced equipment types (e.g., server, router and switch). REN_i indicates the number of a replaced equipment type. EMT_i represents the equipment maintenance time. REC_i consists of the unit cost of a replaced equipment type.

6 Optimization approach

The most problems found in industry, government and science are computationally intractable by their nature or sufficiently large, so as to preclude the use of exact algorithms. In such cases, heuristic methods are usually employed to find good solutions [27].

GRASP (greedy randomized adaptive search procedures) is an heuristic search technique that provides good solutions to difficult combinatorial optimization problems [27]. The GRASP metaheuristic is an iterative technique in which each iteration

contemplates a construction phase and local search phase. Construction phase generates a random solution, and the local search phase investigates the neighborhood of the constructed solution to obtain an improved solution [10].

This section presents an optimization approach based on the GRASP meta-heuristic for generating cloud infrastructures through the assignment of redundancy mechanisms (active-active, cold standby, hot standby and warm standby) to cloud infrastructure components (cloud controller, cluster controller, node controller, virtual machine, switch and router) as shown in Algorithm 1.

Algorithm 1 GRASP(CT, RT, MaxIter, MaxCLS)

```

1:  $f(s) := \emptyset$ ;
2: for  $i = 1$  to  $i = \text{MaxIter}$  do
3:    $s := \text{GreedRandomized}()$ ;
4:   if elite set  $f(s)$  has at least  $\rho$  elements then
5:     if  $s'$  then is not feasible
6:       Randomly select a new solution  $s' \in f(s)$ ;
7:     end if
8:      $s := \text{LocalSearch}(s')$ ;
9:     Randomly select a solution  $s' \in f(s)$ ;
10:    if elite set  $f(s)$  is full then
11:      if  $c(s') \leq \max c(s) | s \in f(s)$  and  $s' \neq f(s)$  then
12:        Replace the element most similar to  $s'$  among all;
13:        elements with cost worst than  $s'$ ;
14:      end if
15:      else if  $s' \neq f(s)$  then
16:         $f(s) := f(s) \cup s'$ ;
17:      end if
18:      else if  $s'$  is feasible and  $s' \neq f(s)$  then
19:         $f(s) := f(s) \cup s^*$ ;
20:      end if
21:    end for
22: return  $s^* = \text{minc}(s) | s \in f(s)$ ;

```

The cloud infrastructures generated are represented through the dependability and cost models. These models provide the estimation of the availability, downtime and cost of the cloud infrastructures. The metric results are used by the optimization approach to select the cloud infrastructures that meet the dependability and cost requirements.

The input data are the component type (CT), component number (CN), redundancy mechanism type (RT) and the output data is an assignment vector (s^*) specifying the redundancy mechanisms types assigned to each component type of the cloud infrastructure. The set of elite solutions $f(s)$ for the cloud infrastructure (availability, downtime and cost result) is initialized with 0 in Line 1. The maximum number of iterations (MaxIter) is computed from Line 2–21. The maximum number of iterations (MaxIter) is defined by the user. During each iteration, a random and greedy solution s' is generated in Line 3. If the set of elite solutions $f(s)$ does not have at least ρ elements and s' is viable and sufficiently different from all other solutions in the set of elite solutions $f(s)$, s' is added to the set of elite solutions in Line 19. If the set of elite solutions $f(s)$ has at least ρ elements, the steps in Lines 5–17 are computed.

The construction phase does not guarantee the generation of a feasible solution. If this phase returns a non-feasible solution, the feasible solution s' is selected randomly from the set of elite solutions $f(s)$ in Line 6. The local search phase uses the solution s' as a start point in Line 8, resulting in a local minimum s' . If the set of elite solutions $f(s)$ meets the requirements, s' is a better solution than the worst solution and $s' \neq f(s)$, then this solution is added to the set of elite solutions $f(s)$ in Line 12. Among all elite solutions with a less cost than s' , the solution s most similar to s' is selected to be removed from the set of elite solutions $f(s)$. A solution s has a lower cost than s' in which its availability is higher and redundancy cost is lower. However, if the set of elite solutions is not complete, s' is added to the set of elite solutions in Line 16.

The construction phase generates cloud infrastructures through the assignment of redundancy mechanisms to cloud infrastructure components. The Algorithm 2 presents the construction phase.

Algorithm 2 *GreedyRandomized(CT, RT, MNCT, MNRT)*

```

1:  $RT := 0; CT := 0; MNRT := 5; MNCT := 7;$ 
2: for  $RT = 0$  to  $RT = MNRT$  do
3:   Randomly generate a redundancy  $RTI \in RS;$ 
4:    $RS := RS \cup RTI;$ 
5: end for
6: for  $CT = 0$  to  $CT = MNCT$  do
7:   Randomly generate a component type  $CTI \in CS;$ 
8:    $CS := CS \cup CTI;$ 
9:   Randomly select a redundancy type  $RTI \in RS;$ 
10:  Assign redundancy type  $RTI$  to component type  $CTI;$ 
11: end for
12: return assignment  $RTI \in RS;$ 

```

The redundancy mechanism type (RT) and component type (CT) are initialized with 0. The maximum number of redundancy type (MNRT) and maximum number of component type (MNCT) are initialized with 5 (e.g., active-active redundancy, could standby, hot standby, warm standby and none) and 6 (e.g., CLC, CC, NC, virtual machine, switch and router), respectively, in Line 1. In Line 3, the redundancy mechanism type (RTI) is randomly generated until a maximum number of redundancy mechanism type MNRT. Each redundancy mechanism type (RTI) is added to redundancy mechanism set (RS) in Line 4. In Line 7, the component type (CT) is randomly generated until a maximum number of component type MNCT. Each component type (CTI) is added to component set (CS) in Line 8. In Lines 9–10, the redundancy mechanism type (RTI) is randomly selected and assigned to each component type (CTI) generated. Each redundancy mechanism type is assigned to a Eucalyptus platform component.

The local search phase investigates the neighborhood of the constructed solution s . If an improvement is found regarding the constructed solution s , this new solution is updated and the neighborhood around the new solution is investigated. The process repeats until no improvement is found [10]. The Algorithm 3 presents the local search phase which utilizes the neighborhood structure known as 1-move. The solution s is obtained in which there is a change in the assignment of the redundancy mechanism type to a component. The search is repeated until an optimized solution occur in the

neighborhood. In this work, instead of evaluating all solutions in the neighborhood, a candidate list CLS is created with the best solutions. One of the best solutions is randomly selected and a movement is performed.

The input data are the solution s , parameters MaxCLS and MaxInter. Lines 1–13 are repeated until obtaining the local minimum. In Line 2, the counter and candidate list (CLS) are initialized with 0. In each interaction in Lines 3–9, a movement in the neighborhood of s is performed through function Move(s) without replacement of the previous solution in Line 4. If this neighborhood is a better solution, it is inserted into CLS in Line 6. This procedure occur until the candidate list (CLS) becomes full or a maximum number of iterations. The candidate list (CLS) size is defined by the user. In Lines 10–12, the candidate list is not empty, a solution $s \in CLS$ is randomly chosen. If the candidate list is empty, the procedure terminates returning the solution s .

Algorithm 3 Procedure LocalSearch(s ,MaxInter,MaxCLS)

```

1: repeat
2:  $count := 0$ ;  $CLS := \emptyset$ ;
3: repeat
4:  $s' := Move(s)$ ;
5: if  $s'$  is feasible and  $cost(s') < cost(s)$  then
6:    $CLS := CLS \cup s'$ ;
7: end if
8:  $count := count + 1$ ;
9: until  $|count| \leq MaxCLS$  or  $count \geq MaxInter$ ;
10: if  $CLS \neq 0$  then
11:   Randomly Select a solution  $s \in CLS$ ;
12: end if
13: until  $CLS = 0$ ;
14: return  $s$ ;
```

7 Case study

This section presents two case studies to illustrate the feasibility of the proposed methodology and models for assessing redundancy mechanisms in cloud infrastructures. Particularly, this work adopts a cloud environment configured with the Eucalyptus platform [14], which provides a virtual learning environment set up with Moodle [23], a webmail configured with Roundcube [28] and a website to an educational institution. The first case study presents a cloud environment composed of a cloud controller, a cluster controller, four node controllers, four virtual machines, a switch and a router (see Fig. 12). The second case study provides a cloud environment composed of a cloud controller, two cluster controllers, four node controllers, four virtual machines. In the case study 2, the virtual learning environment was configured in cluster 1 and the webmail and website were configured in cluster 2.

The proposed methodology provided the planning of the cloud environment adopted. This methodology is composed of the dependability and cost planning activity; dependability evaluation activity; cost evaluation activity; analysis of dependability and cost scenarios activity and selection of dependability and cost scenarios activity.

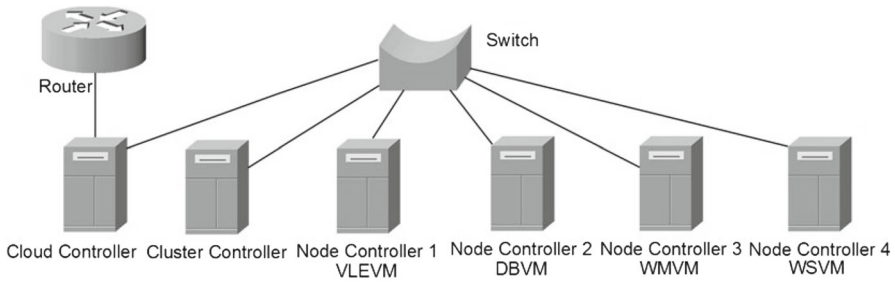


Fig. 12 Cloud infrastructures

In the dependability and cost planning activity, the Eucalyptus infrastructures are generated according to the proposed optimization approach. Each cloud infrastructure is conceived through the assigning the active-active, cold standby, hot standby, warm standby and none redundancy mechanism to components CLC, CC, NC, VM of the virtual learning environment, VM of the database, VM of the webmail, VM of the website, router and switch of the cloud infrastructure. These Eucalyptus infrastructures are selected according to the results of the availability, downtime and costs. In the case study 1, the criteria of the educational institution are a maximum number of 3 cloud infrastructures with the availability greater than 99.99%, downtime smaller than 0.8760 hours/year and cost smaller than US\$ 40,000.00. The case study 2 provides 3 cloud infrastructures with the availability greater than 99.99%, downtime smaller than 8.76 hours/year and cost smaller than US\$ 20,000.00.

Taking into account the dependability evaluation activity, the proposed dependability models are adopted to the representation and evaluation of the availability and downtime of the conceived Eucalyptus infrastructures. RBD models represent the Eucalyptus infrastructure subsystems. The MTTFs of the physical node model (see Fig. 4), virtual machine model (see Fig. 5) and resource manager model (see Fig. 6) are estimated to represent the Eucalyptus infrastructure.

The Eucalyptus model (see Fig. 3) represents the Eucalyptus infrastructure. This RBD model is adopted when the redundancy mechanism hot standby (see Fig. 8) is assigned to the Eucalyptus platform components, because there is not a dependence between the main component and the redundant component. But when the redundancy mechanisms active-active (see Fig. 7), cold standby (see Fig. 9) and warm standby (see Fig. 10) are assigned to the Eucalyptus platform components, SPN models are adopted to represent the dependence between the main component and the redundant component.

The computational resources of the Eucalyptus platform servers are represented by the physical node model (see Fig. 4). The computational resources are the processing infrastructure (PI), primary memory (PM) and secondary memory (SM). The MTTFs of the computational resources of the servers that run the CLC, CC or NC services are 2,500,000.00 h for processing infrastructure, 480,000.00 h for primary memory and 1,800,000.00 h for Secondary Memory. The MTTRs of the computational resources are 8 h. The MTTF and MTTR estimated to physical node are 329,067.64 and 8.00 h, respectively.

The virtual machine model (see Fig. 5) represents the virtual machines of the Eucalyptus platform. The MTTFs and MTTRs [16] of the software configured on the virtual machines are 4320.00 and 8.00, respectively. These software are the Moodle (HS) [23], Ubuntu (OS) [33], MySQL (DB) [26], Apache (WS) [1], KVM (VMM) [15] and Roundcube [28].

The services hosted on Eucalyptus platform are composed of four virtual machine types that are the VLE virtual machine, database virtual machine, webmail virtual machine and website virtual machine. The VLE virtual machine (VLEVM) is composed of the Moodle VLE (HS), Ubuntu (OS), Apache (WS) and KVM (VMM). The MTTF and MTTR calculated to VLEVM are 1080.00 and 8.00 h, respectively. The database virtual machine (DBVM) is composed of Ubuntu (OS), MySQL (DB) and KVM (VMM). The MTTF and MTTR calculated to DBVM are 1440.00 and 8.00 h, respectively. The webmail virtual machine (WMVM) is composed of RoundCube (HS), Apache (WS), Ubuntu (OS), MySQL (DB) and KVM (VMM). The MTTF and MTTR calculated to WMVM are 864.00 and 8.00 h, respectively. The website virtual machine (WSVM) is composed of Apache (WS), Ubuntu (OS), MySQL (DB) and KVM (VMM). The MTTF and MTTR calculated to WSVM are 1080.00 and 8.00 h, respectively.

The resource manager model (See Fig. 6) represents the management modules (CLC, CC and NC) of the Eucalyptus platform. The MTTFs of the management modules components of the Eucalyptus platform are 4320.00 h for Cloud Platform, 329,067.64 h for Physical Node and 4320.00 h for Operating System. The MTTRs of the management modules components of the Eucalyptus platform are 8 h. The MTTF and MTTR obtained to management modules are 2145.91 and 8.00 h, respectively.

The physical node model (Fig. 4), virtual machine model (Fig. 5) and resource manager model (Fig. 6) are used to calculate the dependability parameters of the Eucalyptus model. This model represents the cloud controller (CLC), cluster controller (CC), node controller (NC), VLE virtual machine (VLEVM), database virtual machine (DBVM), webmail virtual machine (WMVM), website virtual machine (WSVM), router (RT) and switch (SW). Figure 3 shows the RBD model used to estimate the availability and downtime of the Eucalyptus platform. The MTTFs of the Eucalyptus infrastructure components, router and switch [5,6] are 2145.91 h for CC, CLC and NC; 400,000.00 h for RT; 56,000.00 h for SW; 1440.00 h for DBVM; 1080.00 h for VLEVM and WSVM; and 864.00 h for WMVM. The MTTRs of the Eucalyptus infrastructure components, router and switch are 8 h.

The MTTFs of the redundancy mechanisms assigned to the Eucalyptus infrastructure components are shown in Table 1. Active-active, hot standby and warm standby redundancy mechanisms are operational when the main component is operational. Thus, this work considers that the main component and redundant component are similar. The MTTFs of these redundancy mechanisms are equal to the main components. In contrast, the cold standby and warm standby redundancy mechanisms are not operational when the main component is operational. In these case studies, this work considers that the main component and redundant component are different. Thus, these studies adopted a 0.3 reduction factor for the MTTFs of these redundancy mechanisms in relation to the MTTFs of the main components. MTAs are 0.16 s for cold standby redundancy and 0.08 s for warm standby redundancy.

Table 1 MTTFs of the redundant components

Component	Active-active (h)	Cold standby (h)	Hot standby (h)	Warm standby OP (h)	Warm standby NOP (h)
CC/CLC/NC	2145.91	1502.14	2145.91	2145.91	1502.14
DBVM	1440.00	1008.00	1440.00	1440.00	1008.00
VLEVW/WSVM	1080.00	705.60	1080.00	1080.00	705.60
WMVM	864.00	604.80	864.00	864.00	604.80
RT	105,120.00	73,584.00	105,120.00	105,120.00	73,584.00
SW	56,000.00	39,200.00	56,000.00	56,000.00	39,200.00

Considering the cost evaluation activity, the redundant component cost model is adopted to support the obtaining of the redundancy cost of the equipment and software of the cloud infrastructure. These costs are related to active-active, cold standby, hot standby and warm standby redundancy mechanisms. As previously explained, these mechanisms utilize the same component, and, thus, component and spare component are equal. However, the cold standby component is different, and this work assumes the unit cost of this redundancy mechanism is reduced by a 0.3 factor. The redundant components costs are US\$ 500.00 [7] for redundancies active-active, hot standby and warm standby of the components CLC, CC, NC; US\$ 350.00 for redundancy cold standby of the components CLC, CC, NC; US\$ 3291.46 [7] for redundancies active-active, hot standby and warm standby of the component router; US\$ 2304.02 for redundancy cold standby of the component router; US\$ 4000.00 [7] for redundancies active-active, hot standby and warm standby of the component switch; US\$ 2799.30 for redundancy cold standby of the component switch.

The maintenance team cost model provides the cost with maintenance team, which consists of a technician. The unit cost of the work hour of the technical is US\$ 20.00. The equipment replacement cost model provides the cost of each Eucalyptus infrastructure component. The equipment replacement cost are US\$ 250.00 for CLC, CC and NC, US\$ 1645.00 for router and US\$ 2000.00 for switch. These values are the unit costs of the Eucalyptus infrastructure components with a 0.5 reduction factor.

The proposed optimization approach generated the Eucalyptus infrastructures, which were represented through the dependability and cost models. These infrastructures were evaluated considering the analysis of dependability and cost scenarios activity.

In the selection of dependability and cost scenarios activity, 3 cloud infrastructures were selected according to the optimization approach, with the availability greater than 99.99%, downtime smaller than 0.8760 hours/year and costs smaller than US\$ 40,000.00 for case study 1 (see Table 2) and with the availability greater than 99.99%, downtime smaller than 0.8760 hours/year and costs smaller than US\$ 20,000.00 for case study 2 (see Table 3). The first column of these tables list the chosen Eucalyptus infrastructures (CI) and the second column shows the redundancy mechanisms assigned to the Eucalyptus infrastructure components. The redundancy mechanism types are active-active—AA, cold standby—CS, hot standby—HS, warm standby—

Table 2 Case study 1—chosen eucalyptus infrastructures

CI	Redundancy mechanism types assigned to components CLC, CC, NC, RT, SW, DBVM, LBVM, WMVM, and WSVM	Availability (%)	Downtime (hour/year)	Cost (US\$/year)
1	HS, CS, CS, CS, HS, WS, CS, None, CS	99.9905	0.8322	30,921.85
2	HS, CS, CS, None, HS, WS, CS, None, CS	99.9903	0.8497	28,617.83
3	HS, CS, None, WS, HS, WS, CS, None, CS	99.9903	0.8497	26,309.29

Table 3 Case study 2—chosen eucalyptus infrastructures

CI	Redundancy mechanism types assigned to components CLC, CC, NC, RT, SW, DBVM, LBVM, WMVM, and WSVM	Availability (%)	Downtime (hour/year)	Cost (US\$/year)
1	CS, WS, None, None, None, CS, CS, CS, CS	99.9945	0.4818	19,017.83
2	WS, WS, None, None, None, CS, CS, HS, CS	99.9944	0.4906	19,167.83
3	CS, WS, None, None, None, None, CS, HS, CS	99.9944	0.4906	19,017.83

WS and no redundancy mechanism—None. The other columns show the availability (%), downtime (hour/year) and cost (US\$/year) results of the chosen Eucalyptus infrastructures.

In the case study 1, Eucalyptus infrastructures with different redundancy mechanisms attributed to its components were chosen because met the availability, downtime and cost requirements. Although the availability of the selected Eucalyptus infrastructures are very similar, the third infrastructure presents the lowest cost, compared to others infrastructures.

In the case study 2, the availability of the selected Eucalyptus infrastructures also are very similar, but the first infrastructure presents the highest availability, lowest downtime and cost, compared to others infrastructures.

7.1 Modeling strategy validation

The scenario adopted to validate the conceived modeling strategy is composed of a cloud controller (CLC), a cluster controller (CC), a node controller (NC), a router and a switch. The dependability parameters (MTTFs and MTTRs) previously presented are adopted in this study.

In the proposed modeling strategy (MS1), the dependability and cost planning activity allows the generation of cloud infrastructures with different redundancy mechanisms. This study provided cloud infrastructures with only one redundant module in hot standby assigned to its components. The dependability evaluation activity provides SPN and RBD models in relation to these cloud infrastructures. The analysis of dependability and cost scenarios activity provides the evaluation of these models. The modeling strategy (MS2) [35] presents a hierarchical method based on heterogeneous

Table 4 Availability results calculated through the modeling strategy (MS1) and modeling strategy (MS2)

CI	Hot standby redundancy assigned to components	Availability (%)—MS1	Availability (%)—MS2	Percentage relative error (%)
1	CLC, CC, NC	99.5641	99.5633	0.0008
2	RT	99.1980	99.1969	0.0011
3	SW	99.2102	99.2090	0.0012

models, combining RBD and SPN models for dependability evaluation of virtual data centers (VDC).

The availability results of the suggested modeling strategy (MS1) are compared with the availability results of the modeling strategy (MS2) [35]. Table 4 shows the availability results calculated using both approaches.

The percentage relative error was applied to the availability results calculated through both approaches and low values were found for all cloud infrastructures provided. The validation of the proposed modeling strategy allows that it is used in other scenarios.

8 Conclusions

This work proposed a methodology, stochastic models and an optimization approach for cloud infrastructure planning. The proposed methodology generated Eucalyptus infrastructures with different redundancy mechanisms through an optimization approach based on GRASP. This methodology takes into account the advantages of both RBD and SPN formalism to compute availability and downtime of Eucalyptus infrastructures, in the sense that RBD and SPN models are combined to represent the Eucalyptus infrastructure, redundancies and corrective maintenance. Equations estimate the cost of redundant components, maintenance team and equipment replacement. The optimization mechanism selected the Eucalyptus infrastructures based on results obtained by the cost equations, RBD and SPN models. Two case studies based on virtual learning environment, webmail and website were presented in order to illustrate the feasibility of the proposed methodology and models. Eucalyptus infrastructures were generated, and its availability, downtime and cost were assessed, resulting in 3 infrastructures that met the requirements. As future work, we intend to consider other cloud platforms, such as CloudStack, Open Nebula and Open Stack. We also intend to consider performance requirements.

References

1. Apache (2015) <http://www.apache.org/>. Accessed 28 July 2015
2. Bauer E, Adams R (2013) Service quality of cloud-based applications. Wiley, New York
3. Callou G, Ferreira J, Maciel P, Tutsch D, Souza R (2014) An integrated modeling approach to evaluate and optimize data center sustainability, dependability and cost. *Energies* 7(1):238–277 (**Multidisciplinary Digital Publishing Institute**)

4. Chee BJS, Franklin C Jr (2009) Cloud computing: technologies and strategies of the ubiquitous data center. CRC
5. Cisco, Cisco 7606-S Router Data Sheet (2015) http://www.cisco.com/c/en/us/products/collateral/routers/7606-router/product_data_sheet0900aecd8057f3c8.html. Accessed 16 July 2015
6. Cisco, Nexus 9000 Switching Series (2016). <http://www.cisco.com/c/dam/en/us/products/collateral/switches/nexus-9000-series-switches/cisco-n9k-lippisreport.pdf>. Accessed 15 July 2016
7. Cost Unit (2015) <http://www.bestbuy.com/>. Accessed 1 Oct 2015
8. Dantas J, Matos R, Araújo J, Maciel P (2012) An availability model for eucalyptus platform: an analysis of warm-standby replication mechanism. In: Systems, man, and cybernetics (SMC), 2012 IEEE international conference, pp 1664–1669
9. Dantas J, Matos R, Araújo J, Maciel P (2012) Models for dependability analysis of cloud computing architectures for eucalyptus platform. *Int Trans Syst Sci Appl* 8:13–25
10. Feo TA, Resende MGC (1995) Greedy randomized adaptive search procedures. *J Glob Optim* 6(2):109–133 (Springer)
11. Ferreira J, Callou G, Maciel P (2013) A power load distribution algorithm to optimize data center electrical flow. *Energies* 6(7):3422–3443 (Multidisciplinary Digital Publishing Institute)
12. German R (2000) Performance analysis of communication systems with non-Markovian stochastic petri nets. Wiley, New York
13. Guimarães A, Maciel P, Matias R Jr (2013) An analytical modeling framework to evaluate converged networks through business-oriented metrics, reliability engineering & system safety. Elsevier, Amsterdam
14. HPE Helion Eucalyptus (2015) <http://www8.hp.com/br/pt/cloud/helion-eucalyptus-overview.html>. Accessed 3 July 2015
15. Kernel Based Virtual Machine (2015) http://www.linux-kvm.org/page/Main_Page. Accessed 30 Aug 2015
16. Kim, D. S., Machida, F. and Trivedi, K. S. (2009), Availability modeling and analysis of a virtualized system. In: International symposium on dependable computing, 2009. PRDC'09. 15th IEEE pacific rim international, IEEE Computer Society, pp 365–371
17. Laprie JCC, Avizienis A, Kopetz H (1992) Dependability: basic concepts and terminology. Springer, New York
18. Li X, Li Y, Liu T, Qiu J, Wang F (2009), The method and tool of cost analysis for cloud computing. In: Cloud Computing, 2009. CLOUD'09. IEEE international conference, IEEE, pp 93–100
19. Maciel P, Trivedi K, Matias R, Kim D (2012) Dependability modeling. In: Performance and dependability in service computing: concepts, techniques and research directions. IGI Global, pp 53–97
20. Marsan MA, Balbo G, Conte G, Donatelli S, Franceschinis G (1998) Modelling with generalized stochastic petri nets. In: ACM SIGMETRICS performance evaluation review, vol 26, number 2. ACM Press, New York
21. Martens B, Walterbusch M, Teuteberg F (2012), Costing of cloud computing services: a total cost of ownership approach. In: System Science (HICSS), 2012 45th Hawaii international conference, IEEE, pp 1563–1572
22. Menascé D, Dowdy L, Almeida V (2004) Performance by design: computer capacity planning by example. Prentice Hall PTR
23. Moodle (2015) <http://www.moodle.org.br/>. Accessed 3 Aug 2015
24. Murata T (1989) Petri nets: properties, analysis and applications. In: Proc. IEEE, vol 77, number 4, pp 541–580
25. NIST (2015) <http://www.nist.gov/itl/cloud/>. Accessed 28 Aug
26. Oracle (2015) <http://www.oracle.com/us/corporate/pricing/price-lists/mysql-pricelist-183985.pdf>. Accessed 1 July 2015
27. Resende M (1998) Greedy randomized adaptive search procedures (GRASP). AT&T Labs Research Technical Report, Citeseer, vol 98, number 1, pp 1–11
28. Roundcube webmail project (2016) <http://roundcube.net/>. Accessed 1 Nov 2016
29. Shooman ML (2001) Reliability of computer systems and networks: fault tolerance, analysis and design. Wiley-Interscience, New York
30. Silva B, Maciel P, Tavares E, Zimmermann A (2013) Dependability models for designing disaster tolerant cloud computing systems. In: Dependable systems and networks (DSN), 2013 43rd annual IEEE/IFIP international conference, pp 1–6

31. Sousa E, Lins F, Tavares E, Cunha P, Maciel P (2015) A modeling approach for cloud infrastructure planning considering dependability and cost requirements. *IEEE Trans Syst Man Cybern Syst* 45(4):549–558
32. Trivedi K (2002) *Probability and statistics with reliability, queueing and computer science applications*, 2nd edn. Wiley Interscience Publication, New York 2
33. Ubuntu (2015) <http://www.ubuntu.com/>. Accessed 8 July 2015
34. Way K, Zuo MJ (2003) *Optimal reliability modeling—principles and applications*. Wiley, New York
35. Wei B, Lin C, Kong X (2011) Dependability modeling and analysis for the virtual data center of cloud computing. In: *High performance computing and communications (HPCC)*, 2011 IEEE 13th international conference, IEEE, pp 784–789
36. Zimmermann A, Knoke M (2007) *A software tool the performability evaluation with stochastic and colored petri nets*. Technische Universitt Berlin. Real-Time Systems and Robotics Group