Blockchain provisioning over private cloud computing environments: Availability modeling and cost requirements

Carlos Melo*, Jamilson Dantas*, Ronierison Maciel*, Paulo Pereira*, Eder Quesado* and Paulo Maciel*
*Informatics Center, Federal University of Pernambuco, Recife, Brazil
{casm3, jrd, rsm4, prps, eraqs, prmm}@cin.ufpe.br*

978-1-7281-4832-8/19/$31.00 ©2019 IEEE

***Abstract*—This paper proposes and evaluates availability models for blockchain provisioning over cloud computing infrastructures as well as their respective deployment expenses in order to establish a cost $\times$ benefit relationship. To demonstrate these models' feasibility, we provide two case studies considering blockchain provisioning over a baseline architecture, and three other alternative redundant environments.**

***Index Terms*—Availability, Blockchain, cloud computing, Cost-Benefit.**

I. Introduction

The Blockchain's based technologies aim to solve trust and inefficient utilization problems [1]. Those solutions are mainly the results of the shared ledger immutability, which guarantees that after a transaction be processed and stored in the system, it can no longer be changed, thus guaranteeing its integrity [2].

Besides blockchain technologies, cloud computing has emerged as one of the main distributed computing paradigm, which is a result of the virtualization process [3]. Large data centers have in the cloud computing the main paradigm to test, refine and develop new technologies mainly due to the virtualization technology that enables the reduction of idle resources and increases the profit margin by reducing the physical space required to accomplish the service provisioning.

This paper evaluates a blockchain platform provided over a set of cloud computing architectures and provide to each one their acquisition cost as well as their availability values. We also point out the best cost$\times$benefit relationship for these environments.

II. Architectures

This section presents the evaluated environments and proposes three alternative architectures considering different redundancy mechanisms. For the evaluated scenario we consider a blockchain that connects two different organizations (Org. 1 and Org. 2), which is one of the mostly common scenarios that may represent the interaction between i.e two banks, two companies, two persons and so on, both organizations may be distributed around the globe and each one has its respective data center (S_1, S_2). Figure 1 shows a high-level view of the organizations interaction.

In this paper, we model, evaluate and consider only one data center, that could be the S_1 from organization 1, which is why it stand as a white box in Figure 1. The availability from organization 2 is a blackbox, we do not know how much resources they have, which is a common case when considering a peer-to-peer environment.

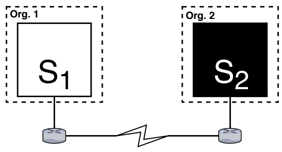

Fig. 1: Scenario

For the S_1 data center we consider three types of nodes: Active Nodes, Warm Nodes, and Cold Nodes. Based on the available nodes, we provide four different architectures. The first one, called baseline architecture is the one that contains the minimum requirements to accomplish the service provisioning. The second environment, which has double redundancy in an active-active mode contains two active nodes, while the third considers a warm-standby redundancy and has an active and a warm node. Finally, the fourth has an active and a cold node.

III. Proposed Models

A. Baseline Architecture

To represent this architecture and the fact that all components must be operational, we considered an RBD with 8 (eight) serial blocks. The correspondent RBD is expressed by the reliability Expression 1, which shows how the availability of S_1 was calculated, considering that we have only one machine performing the service provisioning.

$$R_{S_1} = \prod_{i=\{\text{Components}\}} R_i(t) = \prod_{i=\{\text{Components}\}} e^{-\lambda_i t} = e^{-\sum_{i=\{\text{Components}\}} \lambda_i t} \quad (1)$$

Where **Components** stands for the correspondent component (Hardware, Operating System, Compute, Storage, Network, Docker, Fabric and Composer). Meanwhile, the respective MTTF_{S_1} can be calculated through the Expression 2.

$$\lambda_{eq} = \sum_{i=\{\text{Components}\}} \lambda_i \Rightarrow \text{MTTF} = \frac{1}{\lambda_{eq}} = \frac{1}{\sum_{i=\{\text{Components}\}} \lambda_i} \quad (2)$$

B. Redundant Architectures

For the three redundant types we provided a different CTMC, it is important to mention that the transitions for a CTMC availability model are failure (λ) and repair (μ) rates, that were previously obtained (See Expression 2).

Active-Active Redundancy

This architecture is denoted by a CTMC with two states (S1 and S2), and starts at the **S2** state, in this state all the components in both nodes are operational. The λ and μ rates represent the failure and repair of each machine, a rate is inversely proportional to the time *t*, so a rate $\gamma = 1/t$. We change from S2 to S1 state if at least one node enter into a failure state. At state S1 the service still being provided, we may repair the node in failure with a μ rate or the operational node may fail in a λeq rate. We formally define this CTMC as a 4-tuple: $(Q, \Gamma, \delta, \text{UP})$; Where $Q = \{$S2, S1, S0$\}$, $\Gamma = \{\lambda eq, \mu\}$; δ is the transition function, while the UP represents the set of states where the system is operational that means UP = $\{$S2, S1$\}$.

This architecture availability corresponds to $A_{\text{S}} = \Pi_{\text{S2}} + \Pi_{\text{S1}}$ which can be described in Expression 3.

$$A_{\text{S}} = \frac{\mu^2}{2\lambda eq^2 + 2\lambda eq\mu + \mu^2} + \frac{2\lambda eq\mu}{2\lambda eq^2 + 2}\lambda eq\mu + \mu \tag{3}$$

Warm-Standby Redundancy

This CTMC has five states that consider all possibilities between UP, Waiting and Down for two nodes. An architecture is said to be in a warm-standby redundancy if the main component, responsible for providing the service, has a waiting component *waiting* for its failure before it can start up. The standby environment is "pre-ready", unlike components in active-active redundancy, which were fully configured.

This architecture availability corresponds to the probability of being in at least one of the operational states, which means that $\text{A}_{\text{WS}} = \Pi_{UW} + \Pi_{UD} + \Pi_{DU}$ and can be described by the Expression 4. We formally define this CTMC as a 4-tuple: $(Q, \Gamma, \delta, \text{UP})$; Where $Q = \{$UW, UD, DU, DW, DD$\}$, $\Gamma = \{\lambda eq, \gamma, \lambda eqw, \mu\}$; δ is the transition function, while the UP represents the set of operational states $\{$UW, DU, UD$\}$. The γ rate represents the activation period, which means the time required to configure the server with the service, for a warm-standby this value may reach up to 2 minutes [4].

$$\begin{aligned}\text{A}_{\text{WS}} = {} & \frac{\mu^2(\gamma w+\lambda eqw+\mu)}{\gamma w\big(\lambda eq^2+\lambda eq(\lambda eqw+\mu)+\mu(\lambda eqw+\mu)\big)+(\lambda eq+\mu)\big(\lambda eq\lambda eqw+(\lambda eqw+\mu)^2\big)} + \\ & \frac{\mu(\gamma w\lambda eq(\lambda eq+\lambda eqw)+\gamma w\lambda eqw\mu+\lambda eqw(\lambda eq+\mu)(\lambda eq+\lambda eqw+\mu))}{(\lambda eq+\mu)\big(\gamma w\big(\lambda eq^2+\lambda eq(\lambda eqw+\mu)+\mu(\lambda eqw+\mu)\big)+(\lambda eq+\mu)\big(\lambda eq\lambda eqw+(\lambda eqw+\mu)^2\big)\big)} + \\ & \frac{\gamma w\lambda eq\mu^2}{(\lambda eq+\mu)\big(\gamma w\big(\lambda eq^2+\lambda eq(\lambda eqw+\mu)+\mu(\lambda eqw+\mu)\big)+(\lambda eq+\mu)\big(\lambda eq\lambda eqw+(\lambda eqw+\mu)^2\big)\big)}\end{aligned} \tag{4}$$

Cold-Standby Redundancy

In a cold-standby architecture, the standby node is turned off and needs to be configured, so it can assume the main role in case of failure of the primary node. The switching period between the operant node and its counterpart is called *switch*, this time has a value considered negligible in redundancies of the active-active type, but in an environment that is suspended or even disconnected is determinant in the total availability of the service offered.

This architecture CTMC has four states (S2, S1T, S1, S0), and the availability corresponds to the probability of being in at least one of the operational states, which means that $\text{A}_{\text{S}_1} = \Pi_{\text{S2}} + \Pi_{\text{S1}}$ and is described in Expression 5. Already the respective MTTF for this architecture is the same as the presented in the warm-standby model and is given by $\frac{1}{\lambda\text{eq}}$. This CTMC starts at the **S2** state, in this state all the components in the active node are operational, the cold-standby node is turned off, so it can not enter into a failure state. The S1T state represents a temporary state that indicates that the machine is being configured to enter into state S1. This CTMC is formally defined as a 4-tuple: $(Q, \Gamma, \delta, \text{UP})$; Where $Q = \{$S2, S1T, S1, S0$\}$, $\Gamma = \{\lambda eq, \gamma, \mu\}$; δ is the transition function, while the UP represents the set of operational states $\{$S2, S1$\}$. The γ rate represents the activation period for a cold-standby node, this value may reach up to 10 minutes [4].

$$\begin{aligned}\text{A}_{\text{CS}} = {} & \frac{\mu^2(\gamma+\mu)}{\gamma\big(\lambda\text{eq}^2+\lambda\text{eq}\mu+\mu^2\big)+\mu^2(\lambda\text{eq}+\mu)} + \\ & \frac{\gamma\lambda\text{eq}\mu}{\gamma\big(\lambda\text{eq}^2+\lambda\text{eq}\mu+\mu^2\big)+\mu^2(\lambda\text{eq}+\mu)}\end{aligned} \tag{5}$$

IV. Case Studies

This section provides two case studies that demonstrate how feasible are the proposed models: (1) Availability, (2) Cost-Benefit evaluation of five different architectures.

The values used in this paper can be seen in Table I they were obtained from literature review [5]–[8].

TABLE I: Input Parameters for Baseline Architecture

Component	MTTF (h)	MTTR (h)
Hardware (HW)	8760	1
Operating System (OS)	2893	1
Computing, Network, Storage	788.4	1
Docker (DO)	2516	1
Fabric (HF), Composer (HC)	1258	1

A. Case Study I - Availability Evaluation

The first case study is the proposed models' availability evaluation. We started by evaluating the the baseline architecture and obtaining the λeq and μ that will be used in the subsequent architectures. The baseline architecture indicate an availability of 99.38% and an annual downtime of 54.60 hours, more than two days. With the respective MTTF and MTTR values of this RBD, we were able to evaluate the redundant architectures. The Table II presents the obtained availability for each of the four evaluated architectures.

TABLE II: General Availability Values

Model	MTTF (h)	Av.(%)	Annual Down.(h)
Baseline (A1)	159.95	99.38	54.60
Active-Active (A2)	13029	99.992	0.70
Warm-Standby (A3)	159.95	99.973	2.36
Cold-Standby (A4)	159.95	99.907	8.15

B. Case Study II - Cost-Benefit Evaluation

A three-step survey characterizes the cost evaluation process. The first step consists in establishing the number of components to service provisioning. In our case, the number of nodes. The second step is characterized by the establishment of the acquisition cost of each component.

The chosen server to service provisioning was the Dell PowerEdge T320. The third step is characterized by the analysis of each component technical specification, searching by their respective power values (W). The considered cost of the kilowatt-hour was US$0.16. We calculated the kWh for each device with the expression 6 and the obtained value multiplied by US$0.16.

$$\text{kWh} = \frac{\text{W} \times \text{Number of hours per day} \times \text{Number of days per year}}{1000} \quad (6)$$

The Table III presents the relationship between the components and their respective acquisition cost and energy consumption expenses for their first year of use.

TABLE III: Estimated Expenses for Each Architecture

Architecture	**Energy(USD/y)**	**Acq.(USD)**	**Total(USD)**
Baseline	140.16	1,209.82	1,349.98
Active-Active	280.32	2,419.64	2,699.96
Warm-Standby	208.41	2,419.64	2,628.05
Cold-Standby	145.41	2,419.64	2,565.05

The most expensive architecture was the active-active environment whose deployment and administration cost for an entire year was US$ 2,699.96, as both machines are active and running the deployment expenses are the double of the baseline architecture.

This paper tries to find an ideal value for annual downtime and the required expenses to deploy an architecture. However, both are distinct measures, making it necessary to normalize them, that is, to place them in the same interval $\{0,1\}$. The Expression 7 presents the normalization process conducted.

$$\text{NormalizedXY} = \frac{\text{NoX} - \text{MinNoX}}{\text{MaxNoX} - \text{MinNoX}} \quad (7)$$

Where X $=\{$Cost, Downtime$\}$, Y $= \{$Baseline Architecture, Active-Active, Warm-Standby, Cold-Standby$\}$; MinNoX $=$ $\{$minimum value for the architecture$\}$ and MaxNoX = $\{$maximum value for the architecture$\}$. By obtaining the normalized values for cost and downtime, we are able to relate them through the Euclidean distance; the architecture with the shortest distance to the origin is said to have the best cost-benefit relationship. The euclidean distance calculation may be done through $\text{DistanceZ} = \sqrt{\text{NC}^2 + \text{ND}^2}$, where Z is the respective architecture identifier, NC stands for the Normalized Cost, and ND is the acronym for Downtime Normalized. The ranking of the cost-benefit relationship is presented in Table IV.

TABLE IV: Architecture Rank

Rank	**Architecture**	**Norm. Cost**	**Norm. Down.**	**Eucli. Dist.**
1st	A4	0.900	0.138	0.678
2nd	A3	0.947	0.030	0.688
3rd	A2	1.000	0.000	0.707
4th	A1	0.000	1.000	0.707

The ranking shows that the cold-standby architecture has the best cost-benefit relationship. This architecture does not have the greatest availability, but it does not have the highest deployment price either, which positively impacts the normalization formula.

V. Conclusions

This paper presented two case studies aiming at planning cloud computing infrastructures for the blockchain provisioning. The first case study dealt with the evaluation of the availability of the proposed architectures, starting from the architecture with the minimum requirements to provide the service until a set of redundant architectures in active-active, warm and cold-standby modes. For the second case study, the cost-benefit evaluation of the proposed infrastructures was performed.

References

[1] M. Gupta, *Blockchain for DUMMIES*. John Wiley & Sons, Inc., 2017.

[2] G. Zyskind, O. Nathan, and A . Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in *2015 IEEE Security and Privacy Workshops*, May 2015, pp. 180–184.

[3] P. Mell, T. Grance *et al.*, "The nist definition of cloud computing," 2011.

[4] E. Sousa, P. R. M. Maciel, C. Arajo, G. Alves, and F. Chicout, "Performance modeling for evaluation and planning of electronic funds transfer systems," in *ISCC'09*, 2009, pp. 73–76.

[5] J. Dantas, "Modelos para analise de dependabilidade de arquiteturas de computao em nuvem," Master's thesis, Centro de Informtica - Universidade Federal de Pernambuco (Recife, Brasil), 2013.

[6] C. Melo, J. Dantas, J. Araujo, and P. Maciel, "Availability models for synchronization server infrastructure," in *Proceedings of the IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC'16)*, Budapest, Hungary, 2016.

[7] E. Sousa, F. Lins, E. Tavares, and P. Maciel, "A MODELING STRATEGY FOR CLOUD INFRASTRUCTURE PLANNING CONSIDERING PERFORMANCE AND COST REQUIREMENTS," vol. 4, no. 1, p. 14, 2016.

[8] S. Sebastio, R. Ghosh, and T. Mukherjee, "An availability analysis approach for deployment configurations of containers," *IEEE Transactions on Services Computing*, 2018.