Analysis of Availability of Virtualized Servers

L. H. S. Bomfim; L. B. Silva; R. J. P. B. Salgueiro; R. S. Jacaúna

Departamento de Computação, Universidade Federal de Sergipe, 49100-000, São Cristóvão-Se, Brasil

leonardohsb@gmail.com

lb.luanab@gmail.com

salgueiro@ufs.br

rjacauna@gmail.com

A análise de disponibilidade de servidores virtualizados é uma importante ferramenta para gestores de tecnologia de informação e comunicação no que tange, sobretudo, ao planejamento e dimensionamento de datacenters. Se, por um lado, o uso da virtualização possibilita uma redução de custos, por outro, pode tornar o sistema mais susceptível a indisponibilidade. Este trabalho avalia a disponibilidade de dois ambientes, um com servidor virtualizado e outro com servidores não virtualizados. Os serviços oferecidos são de E-mail, DNS, Servidor Web e Servidor de Arquivos, um cenário típico em diversas empresas. É construído um estudo de caso utilizando modelagem analítica com Árvore de Falhas e Cadeias de Markov. A Árvore de Falha é usada para modelar os servidores e as Cadeias de Markov para obter o comportamento de cada componente de hardware e software. O ambiente não virtualizado é composto por quatro servidores, cada um provendo os serviços específicos, enquanto o virtualizado é formado por um único servidor com quatro máquinas virtuais, cada uma fornecendo um serviço. Através da análise dos modelos desenvolvidos, os resultados obtidos mostram que, embora, o sistema não virtualizado apresente uma menor indisponibilidade, por ter menor dependência entre os serviços, a diferença, neste caso de 0,06% anual, torna-se irrelevante, quando comparada às vantagens trazidas pela virtualização

Palavras-chave: virtualização; análise de disponibilidade; cadeias de Markov

The analysis of availability of virtualized servers is an important tool for managers in information technology and communication especially when it comes to planning and design of datacenters. If the use of virtualization enables a cost reduction, it can also make the system more susceptible to downtime. This work analyzes the availability of two environments, one with a virtualized server and the other with non-virtualized servers. The services offered are e-mail, DNS, Web Server and File Server, a typical scenario in many companies. It is developed a case study using analytical modeling with Fault Tree and Markov Chains. The Fault Tree is used to model the servers and Markov Chains to model the behavior of each component of hardware and software. The non-virtualized environment is composed of four servers, each one providing specific services, while the virtualized consists of a single server with four virtual machines, each one providing a service. By analyzing the models developed, the results show that although the non-virtualized system has less downtime, because has less dependence between the services, the difference in this case is 0.06% annually, becomes irrelevant when compared to the benefits brought by virtualization.

Keywords: virtualization; analysis of availability; markov chains

1. INTRODUCTION

The use of virtualization in the datacenter of companies has gained prominence. With the best performance of the hardware and cheapening the same since 1990, corporate servers began to be underutilized, with a processing load between 5 and 15% of the total resources available [1].

Virtualization can be justified in an environment of datacenters through server consolidation. The management policy for a single application server is still widely used, even when the application is idle in most part of their time [14].

To solve the problem of under-utilization of servers in corporate datacenters, it uses the concept of virtualization, which gives to each user an environment independent of the others. The uses of virtualization allows the reduction of costs with purchases of servers, consumer spending power, physical space and also reduce spending with cooling devices to keep the place with the proper temperature.

In spite of the benefits of virtualization, we must examine the availability of a virtualized server. Since, several services are offered by a single physical server, unlike a traditional datacenter, with the concept of one service per server.

The availability is a property of dependability of systems, which is the ability of a system to provide reliable service that is fault tolerant. The fault tolerance is the ability of the system continues to provide services even in the presence of faults, with techniques such as hardware redundancy, software configurations and virtualization [2].

To analyze the use of a virtualization in a company, this paper employs an analytic modeling to assess the availability of services, thus enabling an organization to make the choice to virtualize the datacenter. The feasibility study on whether or not to deploy virtualization in a datacenter environment is based on analytical modeling, which will assess the availability of non-critical services to be migrated to the environment proposed in this work.

The analytical modeling is used to make the evaluation of system. An example is in [8], which formulates an analytical model to investigate how the energy consumption in virtual servers depends on properties of the workload, infrastructure virtualization and the average density of virtual machines per server physical. Another example is in [10] which presents a performance model for virtual environments through analytical modeling.

A highlight work is that through analytical modeling present a model of availability and analysis of virtualized systems [7]. It was built two systems, a non-virtualized and another one virtualized, and with the use of analytic modeling determined the availability considering the failures that can occur in hardware, software and the hypervisor that performs the virtualization.

Thus, this paper analyzes the availability of a datacenter that provides the services of File server, Web service, DNS and Email. These services can be found in many companies in the area of information technology and other areas.

The server availability is achieved with the use of analytical modeling using Markov chains to analyze the behavior of hardware and software components. With the use of a Fault Tree was modeled the virtualized environment and the non-virtualized one for comparison between them.

This paper is organized as follows. Section 2 describes the concept of virtualization. Section 3 presents the approach to analysis of availability of computer systems, including the use of analytical modeling using Markov chains. Section 4 presents the analyzed scenarios and Section 5 presents the models developed with Fault Trees and Markov Chains. Section 6 discusses the results. And in section 7 are the final considerations of this work.

2. VIRTUALIZATION

The concept of virtual machine emerged in the 1960s when IBM developed the operating system M44/44X, from it others were designed with virtualization support, such as OS/370 [12].

Since 1990 the development of hardware it's with better performance and quality, and is developed the Java programming language that uses the concept of virtual machine so that the programs developed are capable of running in any platform [9].

Nowadays, virtualization is being used not only to reduce costs in datacenters, there is also the use in education [5], software testing, server consolidation [11], among other areas.

The virtualization is a process that allows to run multiple operating systems on a single device [13]. As can be seen in Figure 1, in which a physical machine has some virtual machines, each one with hardware resources, operating system and applications.

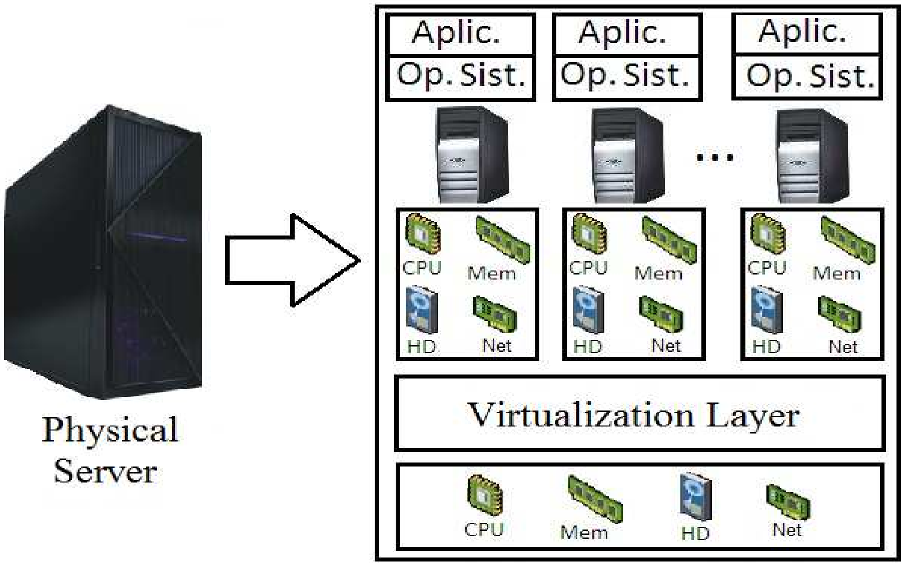


Figure 1 - Virtualization's Process

A virtual machine environment consists of three basic parts: the real system, which contains the actual hardware and software resources of the system; the virtual system running on the virtualized system; and the virtualization layer called hypervisor, which builds the virtual interfaces from the real one [16]. The hypervisor is a software that runs on the physical machine, examples are Xen, Hyper-V of Microsoft and VMWare Inc of VMWare [3].

For virtualization been adopted in companies, it is necessary a study that ensures the availability of the services provide. For this study we use Markov chains to investigate the behavior of each component involved in the process of virtualization.

3. AVAILABILITY ANALYSIS WITH MARKOV CHAINS

The availability of a system is defined as the fraction of time that the system is available to accept service requests from users. The length of time that the system is unavailable is called downtime, and the length of time that the system is available is called the uptime [6].

In this work, for the analysis of service availability in a virtualized server, is used Markov chains to describe the behavior of each piece of hardware and software.

Markov processes represent phenomena that can be classified into finite and discrete states, with a transition probability between states. The sequence of states following this process is called Markov Chain [4].

Markov chains can be represented by using state transition diagrams, as shown in Figure 2. The states are represented by circles named E_i and E_j, and transitions are p_{ij}, p_{ji}, p_{ii} and p_{jj}. The total rate of transitions into and out of a state is 1, representing a 100% of probability.

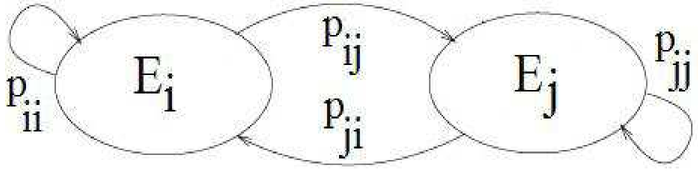


Figure 2 - Example of Markov Chain

The Markov chains have limitations in the model that are explained by [11]. One of these limitations is the fact that it Memoryless Assumption, thus it is assumed that all the necessary information system is described in the state, which also causes the time that it is irrelevant what happens in the same state. The only important thing to know is likely to go to a particular state through the current.

And, another limitation is to be Resulting Limitation, because all information must be contained in states, Markov chains are subject to being large, which causes an increased complexity and loss of accuracy.

4. SCENARIOS ANALYZED

Two scenarios are proposed for the development of the case study. The first proposed scenario is a non-virtualized datacenter, with the concept of one service per server, as shown in Figure 3a. With this configuration, the shutdown of a server leaves only the service it provides unavailable, without affecting others.

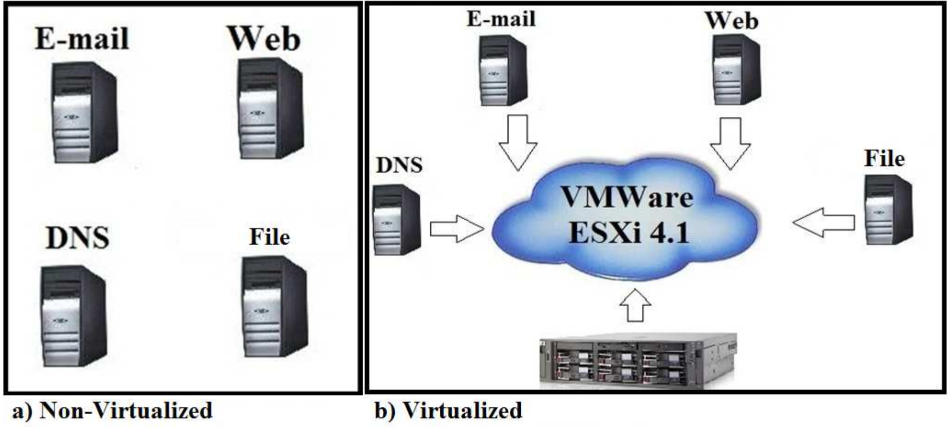


Figure 3 - Scenarios Analyzed

A virtualized datacenter with the same non-virtualized services is represented in Figure 3b. In this datacenter were created four virtual machines on a single physical server, in a way that do not exceed the limit of available computational resources.

The virtualized server has a processor Intel Xeon E5430 described in Table 1, with 18 GB RAM and a disk storage of 530 GB.

Specification	Description
Number of cores	4
Clock Speed	2.66 GHz
L2 Cache	12 MB
FSB Speed	1333 MHz
Instruction Set	64-Bit

Table 1 - Processor Settings

The server virtualization is performed with VMWare ESXi 4.1 hypervisor. The choice of this software is because its a tool with free license, documentation available at the manufacturer's web site and operating history in other virtualization processes. Table 2 shows the configuration of each virtual machine created.

Service	Processor	Mem	HD
Email	2x5.208 GHz	4 GB	100 GB
DNS	1x2.064 GHz	2 GB	8 GB
File	1x2.064 GHz	4 GB	100 GB
Web	1x2.064 GHz	512 MB	8 GB

Table 2 - Configuration of Virtual Machines

5. PROPOSED MODELS

The proposed model consists of a Fault Tree to calculate the probability of unavailability of hardware, hypervisor, virtual machines and applications, using Markov chains to capture the behavior of each system component.

The models used to represent the scenarios are the Fault Trees. A Fault Tree represents a system through nodes corresponding to the logical gates “OR” and “AND”.

A gate "OR" is faulty if any of the components depicted below the port has an unavailability. The gate type "AND" represents a failure if all components below it present unavailability.

The Fault Tree model shown in Figure 4 represents the system in a virtualized server. The first node is a port of type “OR” and corresponds to the physical server. Below this gate is the division of hardware, the virtualization layer (VMM) and server's virtual machines.

The gate "OR" described by "Hardware" shows the six components of server's hardware (CPU, memory, power, network, HD and cooling), if there is a failure in each of this components, and the gate is the type "OR", it corresponds to a system failure.

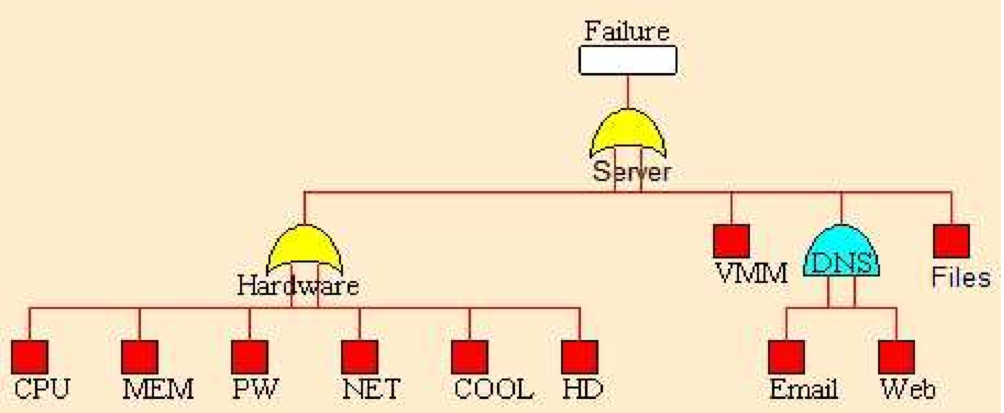


Figure 4 - Fault Tree System for Virtualized Datacenter

The DNS' server is represented by a gate of type "AND" with the virtual machines of Email and Web Server below it. This representation is because if the DNS' server stops, the services of E-mail and Web server continues running without address translation. However, if Web server and e-mail service become unavailable, the DNS service, even if available, it is not being used.

The file server is represented in a level below of the physical server, because a failure causes a downtime, because it has files used by other virtual machines.

For the non-virtualized datacenter, the Fault Tree is shown in Figure 5. The explanation is similar to the virtualized datacenter, however, instead of virtual machines there are physical servers with hardware and operating system.

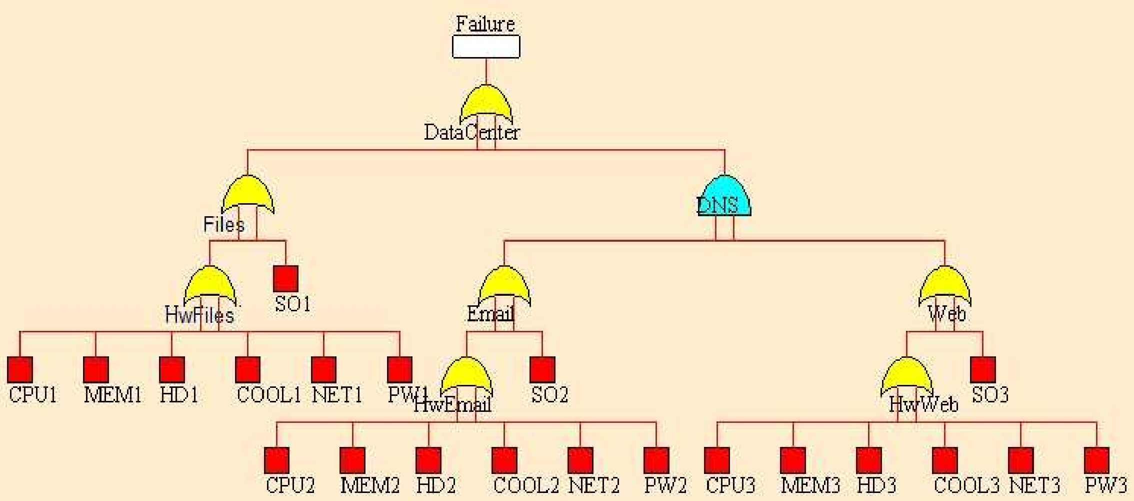
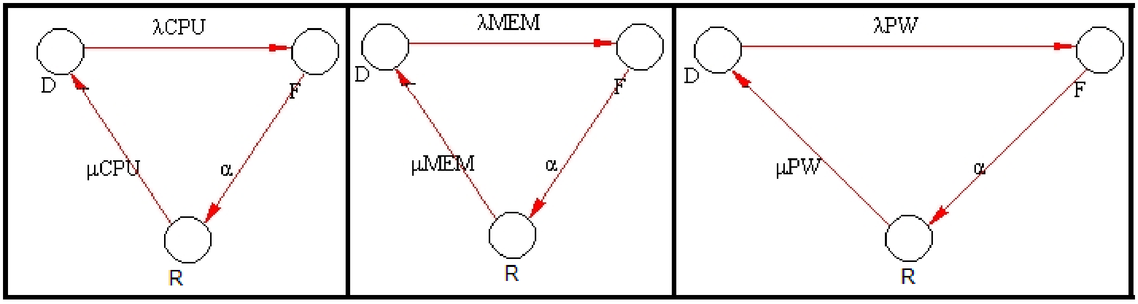


Figure 5 - Fault Tree System for Non-Virtualized Datacenter

The hardware's components (CPU, memory, hard drive, cooling system, network device and power supply), software (operating system and hypervisor) and virtual machines are represented by Markov chains to obtain the fault state of each one.

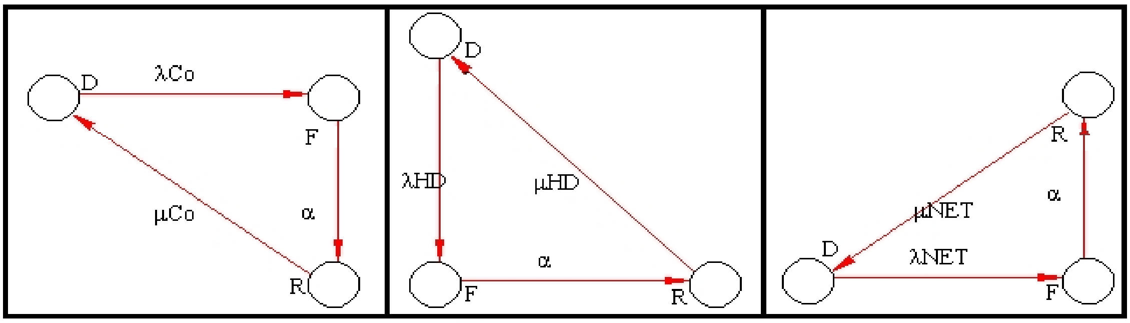
The Figure 7a presents the Markov chain to the behavior of the physical server's processor. In the state “D” the system is active, when the processor has a fault at a rate λ_{CPU}, the system enters the state "F", and a person is assumed to solve the problem with a rate α, going to the state “R”. With the repair completed in an average repair μ_{CPU}, the system returns to state “D”.



a) CPU **b) Memória** **c) Power Suply**

Figure 7 - Markov Chains for the subsystems: CPU, Memory, Power

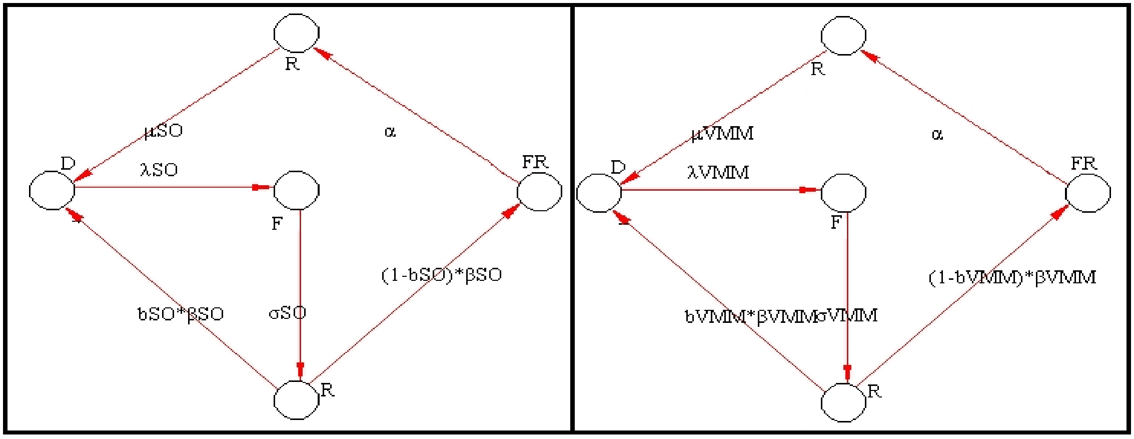
The subsystems of memory, power supply, cooling system, HD and network have the same chain shown in Figure 7a, only the values of the input parameters are different. Figures 7b, 7c, 8a and 8b, 8c, respectively, represent the Markov chain for components of memory, power supply, cooling system, HD and network.



a) Cooling **b) HD** **c) Network**

Figure 8 - Markov Chains for the subsystems: Cooling, HD, Network

The Markov chain for the behavior of the operating system has five states, shown in Figure 9a. The model starts in the “D”, if a fault occurrs with a rate λ_{SO}, the model goes to state “F”. After detecting a failure at a rate σ_{SO}, the model moves to the state “R” and the system is restarted. If this procedure return to work the system, the chain goes to state “D”, otherwise it moves to the state “FR”. In the “FR” a person is called to fix the problem by going to the state “R”. When the repair is completed, the system goes back to “D”. In this model, μ_{SO} represents the mean time to repair, b_{SO} the reset factor e β_{SO} average time of restart. he Markov chain for the hypervisor, shown in Figure 9b, is similar to the operating system.



a) Operating System **b) Hypervisor**

Figure 9 - Markov Chains for the subsystems: Operating System, Hypervisor

The Markov chain for the behavior of virtual machines is presented in Figure 10, having five states. In the “D” the virtual machine is working properly. If there is a service failure, the chain switches to “FS”. With the failure being detected switches to “FSD” in this state the fault is removed and the chain returns to state “D”.

For a unavailability in a virtual machine, the chain shown in Figure 10, follows the same behavior of the service failure.

In this model, λ_A is the mean time to failure of an application, σ_A is the average time to detect the fault, C_A is the factor to repair the application and μ_{1A} is the mean time to repair the application. The symbol λ_V is the average time to failure of the virtual machine, σ_V represents the average time to detection of failure in a virtual machine, C_V is the factor of repair to the virtual machine and μ_V is the mean time to repair for virtual machine.

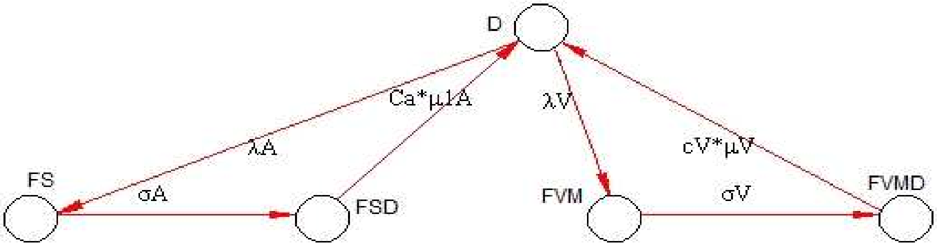


Figure 10 - Markov Chains for the subsystem:Virtual Machines

6. RESULTS AND DISCUSSION

The Markov Chains and Fault Trees are modeled using the tool SHARPE. This tool is used to measure the availability of components of Markov Chains. And with the Fault Trees are calculated failure probability of availability of each system and the annual mean time to failure.

SHARPE is a tool that provides a specification language and methods to solution most types of model used for the performance, reliability and performance modeling.

The tool introduces the concept of hierarchy, because it allows that measurements of a model can be use as input parameters for other models [15].

Obtaining the parameters of the models is given through the manuals of the components, observations and available works. The values of the input parameters in models for the mean time to failure of the components of this case study are obtained in [7].

The values for the mean time to repair components and time of an employee being assigned vary according to the working group for each company. In this work correspond to 1 hour the mean time to repair, and 30 minutes to an person be appointed.

To perform the experiment is important to put all the values in the same unit of reference. This exchange should consider the unit that best suit the study, producing fewer decimal places, thus reducing the error in arithmetic.

The values of the parameters in this case study are presented in Table 3.

Parameter	Description	Value
1/λ_{CPU}	Mean time to failure of CPU	2.500.000 hours
1/ λ_{MEM}	Mean time to failure of memory	480.000 hours
1/ λ_{PW}	Mean time to failure of Power suply	670.000 hours
1/ λ_{NET}	Mean time to failure of network	120.000 hours
1/ λ_{CO}	Mean time to failure of cooling	3.100.000 hours
1/ λ_{HD}	Mean time to failure of HD	20.000.000 hours
1/ λ_{VMM}	Mean time to failure of Hypervisor	2880 hours
1/ λ_{SO}	Mean time to failure of Operating System	1440 hours
1/ λ_{V}	Mean time to failure of Virtual Machine	2880 hours
1/ λ_{A}	Mean time to failure of Software	336 hours
1/μ_{CPU}	Mean time to repair of CPU	1 hour
1/ μ_{MEM}	Mean time to repair of memory	1 hour
1/ μ_{PW}	Mean time to repair of Power suply	1 hour
1/ μ_{NET}	Mean time to repair of network	1 hour
1/ μ_{CO}	Mean time to repair of cooling	1 hour
1/ μ_{HD}	Mean time to repair of HD	1 hour
1/ μ_{VMM}	Mean time to repair of Hypervisor	1 hour
1/ μ_{SO}	Mean time to repair of Operaing System	1 hour
1/ μ_{V}	Mean time to repair of Virtual Machine	1 hour
1/ μ_{1A}	Mean time to repair of Software	1 hour
1/σ_{VMM}	Mean time failure detection in hypervisor	30 seconds
1/ σ_{SO}	Mean time failure detection in Operating System	30 seconds
1/ σ_{V}	Mean time failure detection in Virtual Machine	30 seconds
1/ σ_{A}	Mean time failure detection in Software	30 seconds
1/β_{VMM}	Mean time to restart hypervisor	10 minutes
1/ β_{SO}	Mean time to restart Operating System	10 minutes
1/α	Mean time to a person be assumed	30 minutes
1/b_{VMM}	Factor of restart of Hypervisor	0.9
1/b_{SO}	Factor of restart of Operating System	0.9
1/C_{V}	Factor of repair of Virtual Machine	0.95
1/C_{A}	Factor of repair of software	0.9

Table 3 - Values for the parameters of the models

In this way, the analysis is conducted of the model by analyzing the behavior of the chains and fault trees with the proposed variation of parameters. It is also calculated the availability of systems and transient analysis. With the transient analysis we observe the behavior of systems with increasing operating time.

6.1. Model Analysis

Figure 11 shows the Fault Tree analysis of the virtualized system. When there is an increase in the rate of a CPU failure, the availability of the system decreases. As the Fault Tree of non-virtualized system is developed using the same structure, also exhibits the same behavior.

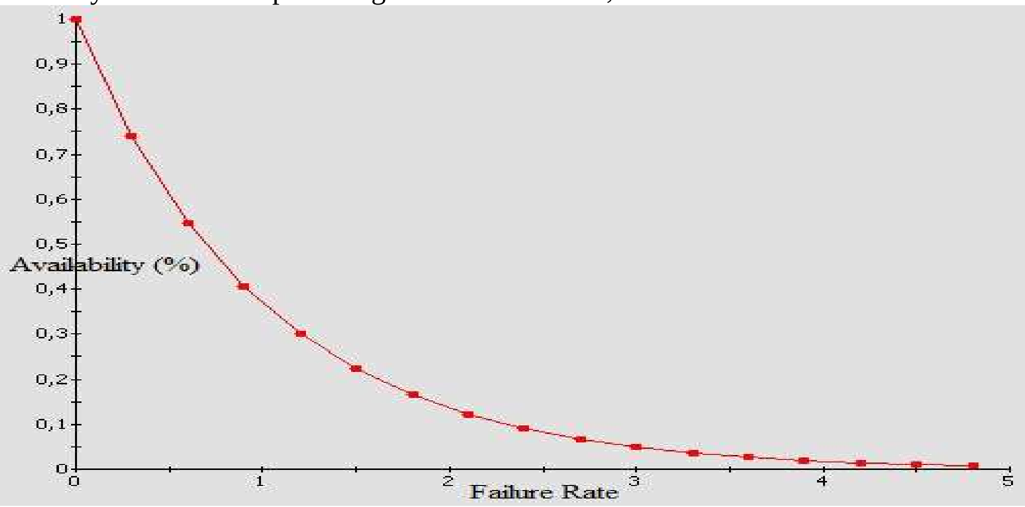


Figure 11 - Analysis of Availability Model Virtualized

The behavior of Markov chains for hardware and software devices are shown in Figure 12, with specific analysis of the chain for the device's physical memory. When the failure rate increases there is a decrease in the availability of the system. Similar behavior occurs with the Markov chain for the virtual machines, as shown in Figure 13.

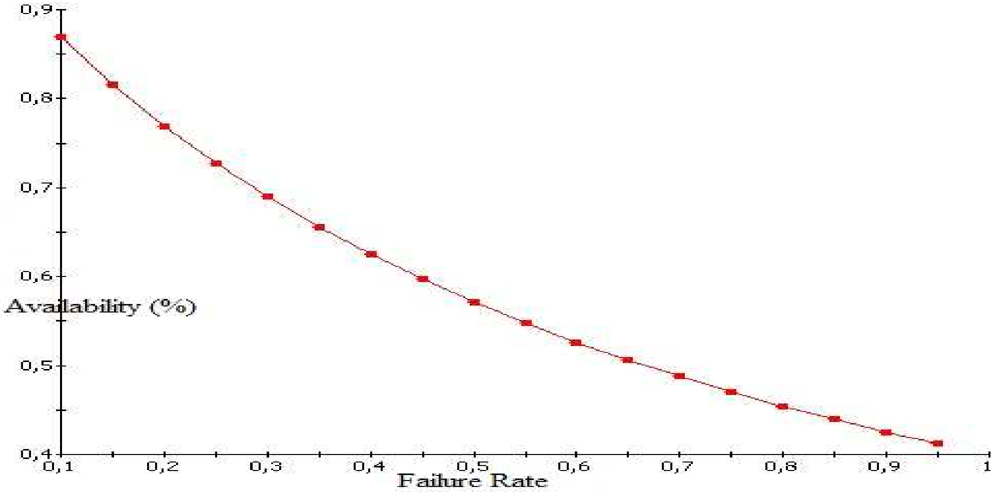


Figure 12 - Availability Analysis Using Markov Chain of Memory Subsystem

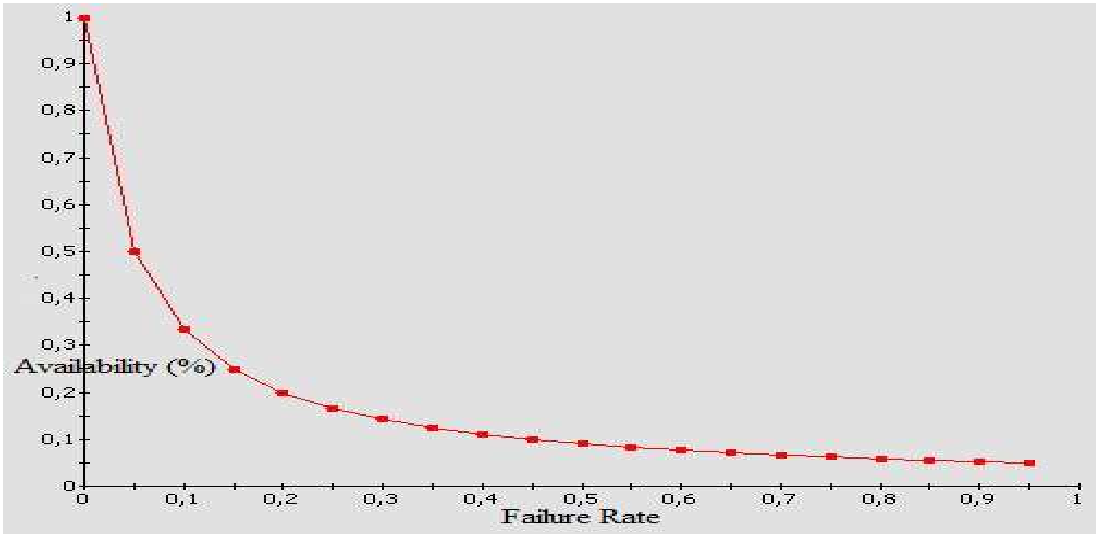


Figure 13 - Availability Analysis Using Markov Chain of Virtual Machines Subsystem

This analysis shows that the models are behaving in line with changes in given parameters, with increasing failure rate decreases the availability on all models.

6.2. Availability Systems

Table 4 presents the results for availability and average annual unavailability of systems.

	Measures	**Values**
Virtualized System	Probability of availability	99,92%
	Annual Downtime	468 minutes
Non-Virtualized System	Probability of availability	99,98%
	Annual Downtime	129 miuntes

Table 4 - Results of Availability Systems

The availability of virtualized server corresponds to approximately 99.92%, and downtime per year is approximately 468 minutes (7 hours and 48 minutes), which represents 0.09% of the year.

For the non-virtualized system, represented by four distinct physical servers, the availability of the system corresponds to approximately 99.98% and downtime per year is approximately 129 minutes (2 hours and 9 minutes), which represents 0.03 % of the year.

This difference in availability between virtualized and non virtualized, it can be viewed in the graph shown in Figure 14, which corresponds to transient analysis of systems. The X axis corresponds to the mean time to failure of the operating system, hypervisor and virtual machines in hours and Y axis to availability.

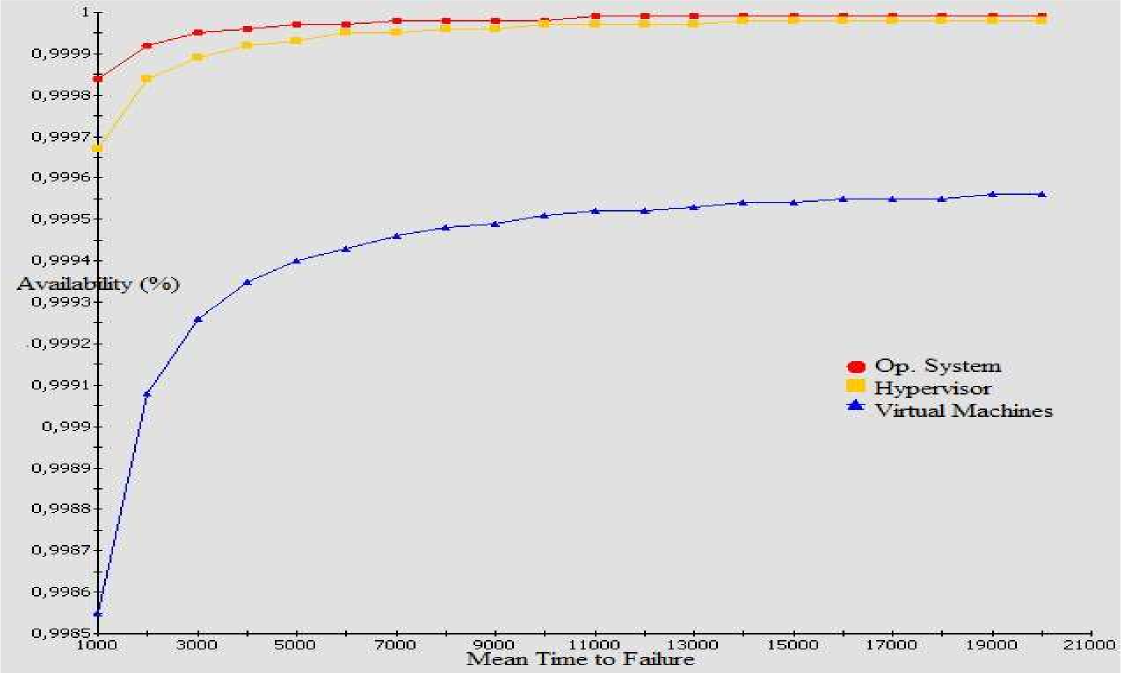


Figure 14 - Transient Analysis of the Proposed Models

In the virtualized environment there is only one server for all services provided. With the unavailability of the physical server, all four virtualized services stop operating. While four different servers, if only one has a unavailable, only one service is no longer provided.

The important thing is not just get results, but to analyze them. The difference in availability between the datacenters is approximately 0.06% annually, which for many companies that provide this service such difference becomes irrelevant when compared to the benefits brought by virtualization.

7. CONCLUSIONS

The process of virtualization is increasingly present in corporate datacenters. The profits obtained by the use of this concept range from economy to the non-acquire of new physical servers, to reduce of costs of electricity in the company.

To adopt a new technology is necessary to conduct a comparative study based on what services the company provides, and thus ensure the viability of a virtualized datacenter. This paper presents this comparison for a company that has services with Email, Web Server, File Server and DNS.

The results obtained for availability analysis shows that the non-virtualized system has a lower unavailability to have less dependency between services, because each service is on a different machine.

However, the difference is not high, considering the percentage of minutes of downtime per year below 0.1%.

The gains from implementing virtualization are bigger than the percentage of downtime for companies that do not have the services analyzed as being critical. The economy with the least expenditure on the purchase of servers, lower cost in energy companies, gains with centralized management of servers and an environmental policy for the lowest greenhouse gas emissions, should be priority issues in the management of any company.

1. ANDRADE, F.; OKANO, M. O impacto da virtualização nas empresas. Anais do IV Congresso Nacional de Excelência em Gestão, 1: 1–22 (2008).
2. A. AVIZIENIS, J. LAPRIE, B. RANDELL, AND C. LANDWEHR. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, 11–33 (2004).
3. BLAKE, E.; CLINCY, V. Virtualization, is it worth it? A technical, financial and economic approach. 5th International Conference on Future Information Technology, p. 01 06, 2010.
4. DESROCHERS, A.; AL'JAAR, R. Y. *Applications of Petri nets in Manufacturing Systems: Modelling, Control and Performance Analysis*. IEEE Press (1995).
5. DOBRILOVIC, D.; STOJANOV, Z. Virtualization software in operating systems course. *ITRE 06 - Information Technology: Research and Education*. 222 226 (2006).
6. JAIN, R. The Art of Computer Systems Performance Analysis. [S.l.]: John Wiley e Sons Inc, 1991.
7. KIM, D. S.; MACHIDA, F.; Trivedi, K. S. Availability modeling and analysis of a virtualized system. IEEE Proceedings of the IEEE International Symposium Pacific Rim Dependable Computing, 365–371 (2009).
8. KOCHUT, A. Power and performance modeling of virtualized desktop systems. IEEE International Symposium on Modeling, Analysis e Simulation of Computer and Telecommunication Systems. 1–10 (2009).
9. LAUREANO M. A. P.; MAZIERO, C. A. Virtualização: Conceitos e aplicações em segurança. [S.l.]: Anais do 26 Simpósio Brasileiro de Segurança da Informação (2008).
10. MENASCE, D. Virtualization: Concepts, applications, and performance modeling. CMG-Conference, p. 407–417, 2005.
11. MENASCE, D.; DOWDY, L.; ALMEIDA, V. Performance by design: computer capacity planning by example. [S.l.*]: Prentice Hall* (2004).
12. Meyer, R.; Seawright, L. A virtual machine time-sharing system. IBM Systems Journal. 9: 199–218 (1970).
13. SAHOO, J.; MOHAPATRA, S.; LATH, R. Virtualization: A survey on concepts, taxonomy and associated security issues. Second International Conference on Computer and Network Technology, 222–226. (2010).
14. SOUNDARARAJAN, V.; Govil, K. Challenges in building scalable virtualized datacenter management. ACM SIGOPS *Operating Systems Review*. 44 (2010).
15. TRIVEDI, K. SHARPE 2002: Symbolic Hierarchical Automated Reliability and Performance.
16. Uhlig, R. et al. Intel virtualization technology. Computer. 38: 48–56 (2005).