Received: January 28, 2017

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

An Approach to Evaluate the Availability of System in Cloud Computing Using Fault Tree Technique

Deepa Mani[1]* Anand Mahendran[2]

[1]*School of Information Technology and Engineering, VIT University, India*
[2]*School of Computer Science and Engineering, VIT University, India*
* Corresponding author's Email: mdeepa@vit.ac.in

Abstract: The Fault tree furnishes an engraving and analytical structure to dissect the dependability of systems. A fault tree gives an abstractly essential demonstrating schema to embody the level of system communications among all the component reliabilities in cloud computing environment. Fault tree Investigation is that fundamental technique which concentrates on failure modes for whatever system and the probabilities for occurrence/ likelihood risk dissection connected with it. Fault Tree Analysis signifies to significant deficiency or critical failures connected to the system and reasons for the faults graphically. Here we will realise that likelihood from claiming the failure of the primary event with that assistance about which we get that probability of failure of the top event. Dynamic fault trees include a progressive thought of the conventional issue tree tactic: system failures might rely on component failure. Exceptional motivation dynamic fault tree gates can model gradual supplanting of neglected segments. Starting with pools from claiming spares, failures that happen just if others happen for certain orders, dependencies that propagate failure, also circumstances the place failures could happen best to a predefined request. With the help of the dynamic gates, the fault trees need to be tackled by automatic change with proportional Markov models regularly in a cloud environment. The proposed system uses fault tree analysis and Markov chain process to efficiently identify the repair and failures of each virtual machine and reduces the energy consumption rate and cost accordingly. The result obtained in our proposed model definitely bounds optimum solutions and deceptively specifies the triumph of an efficient fault tree analysis of failures on minimal cost and energy consumption.

Keywords: Cloud Computing, Fault Tree Analysis (FTA), Markov chain process, Top Event (TE), Dynamic gates.

1. Introduction

Well established requisitions on cloud nature's domain display a significant amount of intricacy since the issues are not best identified. With the well- adequating of the requisition as well as of the cloud platform that provides for those figments from claiming unlimited registering resources accessible ahead interest [1]. So, we target to define and recognise the undesired practices from claiming such system in place on assessing how particular failure occasions or a different combination of fault events which is relative to the system parts can reason the failure from claiming subsystems alternately of the entire scheme [25]. Those primary objectives of the paper comprise in assessing the availability level of cloud requisitions keeping tabs on the fault tree use [2].

In the Infrastructure-as-a-Service (IaaS) model, capitals need to virtualized and conveyed as "a service" utilising Service Level Agreements (SLA) which ensures convinced levels about Quality of Service (QoS) to limit clients [24]. Inside IaaS, amenity dependability and accessibility are of an extraordinary vitality for an excellent administration conveyance [3]. Furthermore, these targets might be attained through mind overwhelming distributed fault tolerant configuration. Adaptability for asset provisioning of the cloud is gained by virtualization with that significance that however end customers request additional enrolling resources, they can provision on-request, straightforward to the client [4].

DOI: 10.22266/ijies2017.0831.26

Fault Tree Analysis may be a top down deductive analysis on which those reason for an occasion would deduce. It provides a graphic show from claiming how component let-down, human mistake and exterior variables are donated to a mischance or occasion [5]. It employments legitimate logical gates and also little occasions will display the way for a mischance through distinctive steps and henceforth a fault line tree may construct for the particular off chance. The functional failures can signify as an essential event, and same time human errors can be meant to escalate causes of technical failure [6]. Signify as an essential event, and same time human errors can be meant to escalate causes of technical side.

Fault tree analysis is a mainstream and diffused method for the reliability demonstrating and assessment about comprehensive, wellbeing basic frameworks, similar to those Programmable electronic frameworks (PES) [7]. This technology is based on the recognisable proof of a specific undesired off chance with the occasion to be broke down (e.g., system failure, virtual machine failure and power failure), called the Top Event (TE) [8]. The development of the Fault Tree (FT) returns in a top/ down manner, from those occasions with their reasons, till let-downs from claiming essential segments arrive [22]. The technique is based on the accompanying suspicions: (i) occasions are paired events (working/not-working); (ii) events need to be statistically autonomous, and (iii) associations the middle of events and reasons need to be spoken to by method for logical AND and OR gates [9].

However, some fault tree instruments unwind the most original supposition and also permit the consideration of the NOT gate and besides related (e. g. XOR) gates [10]. In fault tree analysis, the examination may be conveyed out in two steps: A subjective venture in which the legitimate expression of the top event will be inferred as far as major implicants (the negligible cut-sets) [23]. A quantifiable experiment in which on the support of the likelihoods allocated of the failures occasions of the fundamental components, the likelihood of an event of the TE (and of any inner occasion comparing should a logical sub-system) may ascertain [11].

In [29] the proposed a dynamic replication factors and some placement algorithm in order to improve the performance of the software system. The placement algorithm used to identify the replicas in the nodes of the system. The placement algorithm attempts to raise the data availability, improvement in execution rate and system bandwidth. [30] Proposed algorithm is QSC_MCD which combines the quality of service parameters with the number of clouds. PROMETHEE method and MCDM solution in order to select the best service from other cloud combination set. Attempted to show the best services with minimum number of clouds. In [30] discussed the fault tolerant system by using Markov chain. The proposed work shows the origin of failure, average time to repair. The main motivation of this paper is identifying the reason of the failures of virtual machine like single system failure, multiple system failure, and power failure. The queuing models like Markovian and non-Markovian models are inspected using the phase type expansion and renewal theory.

This paper is organised as follows: we discuss the fault tree analysis and the extended version of fault tree in section 2. System modelling can be presented with some parameters like the construction of fault tree and the performance evaluation using Markov process can be explained in section 3. At last, we conclude the paper in section 4.

2. Background

This section manages with the ideas of fault tree analysis, Dynamic fault tree analysis.

2.1. Fault Tree Analysis

Fault tree was designed to simplify the unreliability analysis. They give a compact, graphical, natural strategy with dissect system consistent quality and assessing those routes that make a system failure, also proving the instruments to hazard identification [12]. When FTA might have created, scientific techniques and computational codes need to be formed to fault trees; the same time that primary objective for FTA may be deficiency tree outline [21]. The action of a framework can view as from two focuses of view: the approaches towards the system let-down and the routes that lead it towards triumph [13].

It may be fascinating to point out that certain recognisable proof focuses looking into success space correspond for certain practically equivalent to focuses in failure area [19]. Furthermore, it is not difficult to accomplish event around the thing that constitutes failure over it may be to consent ahead what constitutes success [18]. Success will be related to the effectiveness of a system, the measure of output, the level of convenience and also creation and showcase offers, aspects that need describable eventually by continuous variables, not effortlessly demonstrated regarding single discrete events [14].

2.2. Dynamic fault tree analysis

Dynamic fault tree analysis is playing a significant role which classifies and knobs the components that might lead the framework to a failure. It is a logical dissection concentrated to a particular unsought occasion and also gives a system for deciding the reason for such occasion. That major undesirable event is seen as a TOP occasion in the system fault tree [16]. It heads those systems for an aggregate failure. The top event needs on be decided carefully, in view it will be key to the accomplishment of the investigation. Assuming that this event is to be universal, that investigation can be uncontrollable, and that event is excessively particular, at that point, the investigation doesn't provide a review of the system. So, it is required to be preserved till the legitimate level [19]. In this paper, they created and assessed an adjustment to non-basic failure show for accomplishing the multifaceted nature and the heterogeneity of hardware structures [20]. So the cloud-based frameworks inclined to noteworthy measures of failures and working incidents [26]. They utilise machine replication and live movement as fundamental instruments and display a fault tolerant design to genius successfully pick when reiteration and live relocation process ought to be started. The health state of the virtual machine is anticipated with the guide of the fault tree examination, to keep up the high accessibility and unwavering quality [28].

3. Related Work

Elinda et al., have analysed and evaluated the consistency of Dynamic Fault Tree using web applications in Cloud Computing. They defined the model in which the dynamic fault tree ran on a cloud platform and demonstrated the possibility of this method used in online payment system which is facilitated on a cloud platform. In exact operation conditions with nominal useful resources on the cloud stage, on system undependability is close twofold the impact of utilisation sort. They reviewed all the failure parts of the framework and to recognise the purpose behind each of the failures by utilising Dynamic Fault Tree (DFT) strategy. Their approach fails to focus on energy efficiency and cost factor of the datacenter. Alexandru et al., [2] in this paper, they created and assessed an adaptation to non-critical failure display for accomplishing the multifaceted nature and the heterogeneity of hardware frameworks [26]. Their proposed work fails to calculate response time and network traffic information of the virtual machine. They utilise machine duplication and animate movement as vital instruments and present a fault-tolerant outline to pro- effectively and choose when replication and live relocation process ought to be started. The health state of the virtual machine is predicted with the help of the FTA, to maintain the high availability and reliability [28].

Ragavan et al., [6] they developed a reliability modelling with the help of Dynamic Innovative Fault trees (DIFtree). This model is fit for showing different transferences of time to failure, no time module, steady hazard rate, time- differing danger rate (Weibull) and lognormal. Their approaches incorporate both Binary Decision Diagrams (BDD). The BDD algorithm will evaluate the functions of failed components and their probabilities [27]. But their approach fails to focus on energy efficiency and cost factor. Wang et al., [7] they propose architecture driven outline technique has been utilised as a part of escalated framework plan of the computer. This method is used to identify the risk, i.e., detect the areas of potential hazards within the intensive system design. The primary functionality of this approach is to determine the component-level vulnerabilities. The fault tree based structural analysis is more robust in determining the vulnerability analysis of e-business system architecture [8]. Their work fails to focus on calculating failure rate, tie to repair factors. A. Bobbio et al., [10], elegantly proposed algorithms like conversion algorithm, mapping algorithm which have been applied in various real world tasks. But they have acknowledged a small attention in the space of reliability. They compared the Bayesian Networks (BN) with one of the traditional techniques for security analysis, safety critical systems called Fault Trees (FT). They show that by implementing these algorithms, some additional power has been gained [25]. Their proposed work fails to focus on cost factor of data center.

Ravi et al., [26] deliberated about the failure qualities of cloud-based administrations and inspected the impact of every failure type. They introduced a fault tolerance model will helpful in detecting the robustness of failure protocols, state synchronisation methods and strength of the fail-over granularity. But their proposed work fails to focus on energy efficiency of the virtual machine. Yu Nan et al., [21] used Stochastic Petri Net (SPN) modelling which is helpful in improving the analytic reliability of the microsatellite onboard computer. The dynamic reliability of the systems can be obtained by SPN modelling, and also investigate the fault events with the help of the error indicator.

Table 1. Input parameter values

Notation	Meaning
Pr(X_{m+n})	Transition probability of the components
p_{ij} (n)	Probability of state changes
i,j	States
X	Number of components in working condition
Re	Reliability
F	Distribution function of the component lifetime
N_0	Number of identical components
N_f(t)	Number of failed components
N_s(t)	Number of survived components
La_f	Laplace transforms density function
La_w	Laplace transforms density variable
La_g	Probability density function
La_m	Average number of repairs or replacements
G_t(x)	Survival rate of the component
Re(t)	Reliability of the component
h(t)	Instantaneous failure rate

They also demonstrated the optimisation capabilities of the Dynamic Fault Tree algorithm [19]. Their proposed work fails to focus on cost factor of the virtual machine [20].

From the above analysis, the availability evaluation is carried out using Markov chain process in cloud computing. There are only a limited works, which addressed the availability issues using a mathematical model in the cloud computing. Among these works, there is no work which addresses the failure detection issues based on the Fault Tree Analysis model with QOS metrics like energy efficiency, cost and repair rate and failure rate. The proposed work moderates power and cost and failure detection by using Dynamic Fault Tree (DFT).

4. Problem Formation

This section deals with the fault tree model and the evaluation techniques using Markov chain process. In this paper, we have referred some formulas [15]. Based on our proposed work, the variables/ parameters in the formulas are slightly modified.

4.1. Fault tree model

This model is a visual model about parallel and progressive issue blends that will achieve those occasions of a predefined undesired occasion. Failures are events connected with virtual machine failure, power outage, hardware part failures, and human mistakes alternately different events, which can prompt that undesired occasion. The fault tree delineates those real interrelationships of these fundamental occasions that motivate that undesired occasion, which might be that TOP event of the tree. The TOP event relates should cause an entire system failure. So, the tree is created from claiming every last one of segments that produce this event. Fault trees are a qualitative model that can assess quantitatively. They consolidate an arrangement of units, known as "entryways" that allow or prevent basic stream in the tree. Entryways represent to the connection between the objects, which result in thus occasioning higher in the tree to happen. This event is perceived to be the gate yield, same time different activities require help entryway inputs.

A portion of the images that create the fault tree incorporates essential events, outer events, undeveloped occasions, contingent occasions, delegate occasions which portray particular instances of failures. There are likewise two principle sorts of gates: OR gate, AND gate and transfer symbols. The fault tree needs some preceding which will be helpful to check the outcome. The steps involved in successful FTA construction are as follows,

1. Classify the aim of the FTA
2. Describe the TOP event of the FTA
3. Outline the space of the FTA
4. Express the firmness of the FTA
5. State the fundamental principle of the FTA
6. Build FT
7. Assess the FT
8. Elucidate and showcase the result

4.2. Assessment Methods of Fault Trees

After constructing a fault tree, it can be assessed to gain the subjective and quantitative impacts. The quality measures include i) minimal cut units in a fault tree, ii) essential qualitative factors, iii) most basic cut sets that can explanation behind the failure. Cut sets provide information about the causes of failures and how that affects the entire system to fail. These cut sets can also identify how the pool of components fails from a single failure.

Let us consider the scenario, for example; we have *‘n’* number of states like S_1, S_2...S_n. If any one of the States wants to transfer from the current state to another state, let us assume the current state may be the idle state and the change of state may be a repair state. Our system will identify the probability of transfer among the states by using the Markov chain concept. That is, Markov chain model depicts the states. Our model will be helpful in identifying the failure causes and repair rate and the performance metrics of cloud computing using the fault tree analysis which is depicted in figure 1.

4.3. Probability evaluation using Markov chain process

In this model, the basic events are considered as mutually independent, and the occurrence is based on the failure probability, the rate of failure and the cause of unavailability. The event occurrence is denoted by logic 1 at that node; otherwise, the logic value of a node is 0.

Case 1: Computation of n-step probabilities

Let the expression for evaluating the n-step probability is $p_{ij}\ (n)$ and if the transition from one state to another state like from the idle state to some repair state then the probability is $p_{ij}(1)= p_{ij}$.

So, $p_{ij}(n)= Pr(X_{m+n}= j|X_m= i)$ (1)

If the Markov chain is homogeneous then the transition probability for such chain is,

$p_{ij}(n)=Pr(X_{m+n}=k|\ X_m=j)$ (2)

$pij=Pr(X_n = k|\ X_n\text{-}1=j)$ (3)

Where $n \geq 1$

To define 0-step probabilities of the state of the fault tree by,

$p_{ij}\ (0) = \{1, 0$ $\textit{ if it is 1 then } i = j,$
$\textit{Otherwise } 0\}.$ (4)

To get the joint probability of the fault tree will consider all the values of n,

$Pr(X_0 = i_0, X_1 =i_1, \ldots, X_n =i_n)$

$= Pr(X_0 = i_0, X_1 =i_1 \ldots X_{n\text{-}1} =i_{n\text{-}1})$

$. Pr(X_n = i_n |, X_0 =i_0, \ldots, X_{n\text{-}1} =i_{n\text{-}1})$

$= Pr(X_0 = i_0, X_1 =i_1 \ldots X_{n\text{-}1} =i_{n\text{-}1})$

$. Pr(X_n = i_n |, X_{n\text{-}1} =i_0 =i_{n\text{-}1})$

$= Pr(X_0 = i_0, X_1 =i_1, \ldots, X_{n\text{-}1} =i_{n\text{-}1})\ pi_{\ n\text{-}1,}\ i_n$ (5)

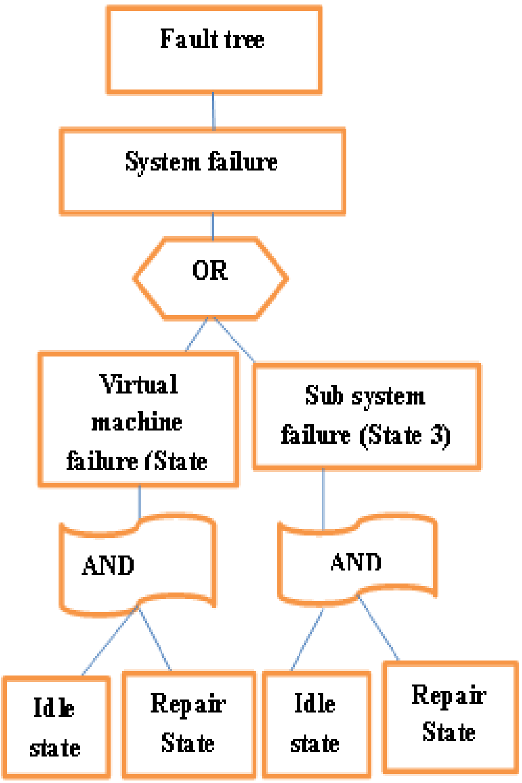

Figure1. A typical Fault tree based on the system failure

Case 2: Distribution of times between state changes

Assume that the state at the nth step is $X_n=i$ and the probability of the next state is j- that is, $X_{n+1}=j$- and it should depend on the current state I but it does not depend on the time the chain expended in the current state.

Let the random variable t_i denote the time that the Markov chain model spends in any one of the states, i.e., I during the single visit to the state i.

Thus the geometric distribution if t_i, so that,

$Pr(t_i =n) = (1\text{-}p_{ii})p_{ii}^{n\text{-}1}, i \in I$ (6)

By using the properties of the geometric distribution, the expected number of steps the chain of the process spends in the state I, per visit to state I is given by,

$E\left[ti = n\right] = \qquad , i \in I$ (7)

4.4. Reliability evaluation using Markov chain process

Let us assume that the cloud environment has *'N'* virtual machines and the time to failure of the component is X. The probability that the virtual machine survives until sometimes t is called the reliability *Re(t)* of the component. Figure 2 illustrates the causes of failure and classifies each failure as static and dynamic by using Markov chain concept. The outcome of this scenario is performance measurement.

Thus, $Re(t) = Pr(X>t) =1- F(t)$ (8)

Consider a fixed number of identical components N_0 under the test process. If *t*, $N_f(t)$ components have failed and $N_s(t)$ components has survived with,

$$N_f(t) + N_s(t) = N_0 \quad (9)$$

So, the estimated survival component probability

$$Pr' = N_s(t)/N_0 \quad (10)$$

So the reliability of the survival state of the component is calculated by,

$$Re(t) \simeq N_s(t)/N_0 = N0-N_f(t)/N_0 \quad (11)$$

$$Re(t)= 1- N_f(t)/N_0 \quad (12)$$

Let us assume that the total number of components N_0 is constant, while the number of failed components N_f increase with time. By taking the derivatives we get,

$$Re'(t) = -\frac{1}{N_0} N'_f(t) \quad (13)$$

In the above equation N'f(t) is the rate at which components fail. Therefore $N_0 \rightarrow \infty$, then interpret as the negative of the failure density function of the virtual machine, $f_x(t)$,

$$Re'(t)=-f_x(t) \quad (14)$$

Sometimes the conditional probability of the component does not survive for an additional interval of time x given that the component has survived until time t can be written as:

$$Gt(x) = \frac{\Pr(t < X < t+x)}{\Pr(X > t)} \quad (15)$$

From equation 15 we derive,

$$Gt(x) = \frac{F(t+x)-F(t)}{\mathrm{Re}(t)} \quad (16)$$

Now we have instantaneous failure rate of the component be h(t) at the time interval is t,

$$h(t) = \lim x \rightarrow 0 \frac{1}{x} \frac{F(t+x)-F(t)}{\mathrm{Re}(t)} \quad (17)$$

So that h(t)= f(t)/ Re(t) (18)

4.5. Availability evaluation using Markov chain process

The availability is factor which is helpful to identify the non- functioning, integrity and authentication of the customers.

Let us assume that we have *'n'* components of the virtual machine and μ_n be the repair rate for n components and λ_n is the failure rate for n components.

The average number of repairs and replacements are calculated by using the Laplace convolution formulas,

$$Lap_M(s) = \frac{Lap_w(s)\ Lap_g(s)}{S\left[1-Lap_w(s)\ Lap_g(s)\right]} \quad (19)$$

From that, we can derive the failure and the repair rate of the system,

1/λ is the mean time to failure (MTTF)

1/ μ is the average time to repair (MTTR)

So, the component availability is calculated by,

$$Ava = \frac{1-[1-s/\lambda]}{1-\left[[1-s/\lambda]\,[1-s/\lambda]\right]} \quad (20)$$

Inference of Fault tree analysis in Healthcare system:

Fault tree will be very helpful in identifying the peril in the medical field. The comprehensive study masks the identified risks and unmasks to determine the effects of it. [16]. On account of software failure, the normal anxiety is to decide the issues that will bring about the arrangement of standards to neglect to convey a system administration, for example, observing a system [17].

With the help of the fault tree, all the possible situations are linked., to classify the correlation among the failures and identify the segments.

Here is a case of the fault tree, as connected to the Insulin conveyance system. In figure 3 the potential software errors are shown. These failures including equipment, for example, low battery, blood screen or sensor let down, patient over-effort, or the nonattendance of medicinal staff failure [18].

The system risk assessment process done with the help of safety specification of the fault tree. Initially, the probability of the hazard event must be surveyed.

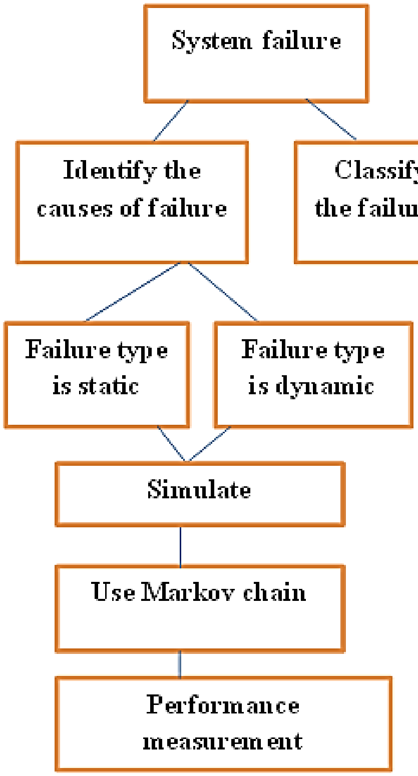

Figure 2. Analysis model for performance evaluation

The hazard occurrence is regularly quantifiable so numbers might be allocated in light of things like MTBF, latency impacts and other known entities. Finally, the failure assessment must include the severity of the risk and the associated cost. The software tool used to assess the following system is using SHARPE and ConceptDraw Pro diagramming.

5. Comparative analysis

The simulation of availability modelling is done using SHARPE tools. We have analysed the proposed model using SHARPE tools, and it is evident that the proposed availability model with Markov chain process predicts the steady state reliability of the virtual machines. The simulation of this model is examined to envisage the QOS metrics component. The suggested model is compared with traditionally proven techniques like BDD (Binary Decision Diagrams) [7], DIFtree (Dynamic Innovative Fault trees) [5], and SPN (Stochastic Petri Net) modelling [21].

Figure 4 and Figure 5 illustrates that Dynamic Fault Tree (DFT) efficiently increases the performance and availability when compared to the standard BDD, SPN and DIFtree [5]. Figure 4 shows the repair rate based on the error rate in the cloud data center [21]. The proposed method attains less repair cloud data center. The proposed method attains less repair rate for more number of mistakes compared to SPN and DIFtree [9].

Figure 5 shows the availability of the system based on the time to failure. The time to failure increases when the availability of the system increases [7]. Our proposed approach stabilises the failure rate to reduce the repair rate accordingly.

Thus the comparison result proves that DFT with Markov process is more efficient and performs well. Figure 6 depicts how the reliability of the cloud systems increases based the speed up gradation of the systems. In our proposed work we have essentially focussed on availability factors of quality of service. We have related our proposed technique with other existing techniques like Programmable electronic frameworks (PES), Dynamic Innovative Fault trees (DIFtree), Binary Decision Diagrams (BDD), Bayesian Networks (BN), Stochastic Petri Net (SPN) modelling, QoS-aware service composition in multiple cloud domain (QSC_MCD), PROMETHEE method and multi-criteria decision making (MCDM) solution, Continuous Time Markov Chain (CTMC) and our proposed method have inferred an distinctive outcome.

Our major goal is to achieve 100% steady state availability of the system. We have examined the QOS metrics such as cost, repair rate, failure rate and energy with the previously existing system but some the system failed to satisfy all the factors of QOS metrics.

Our proposed method balance out the accessibility to decrease the repair rate accordingly. Thus, Table 2. Describes the comparison result shows, that the fault tree based technique is well-organised and performs well when the failure rate of the entire virtual machine increases.

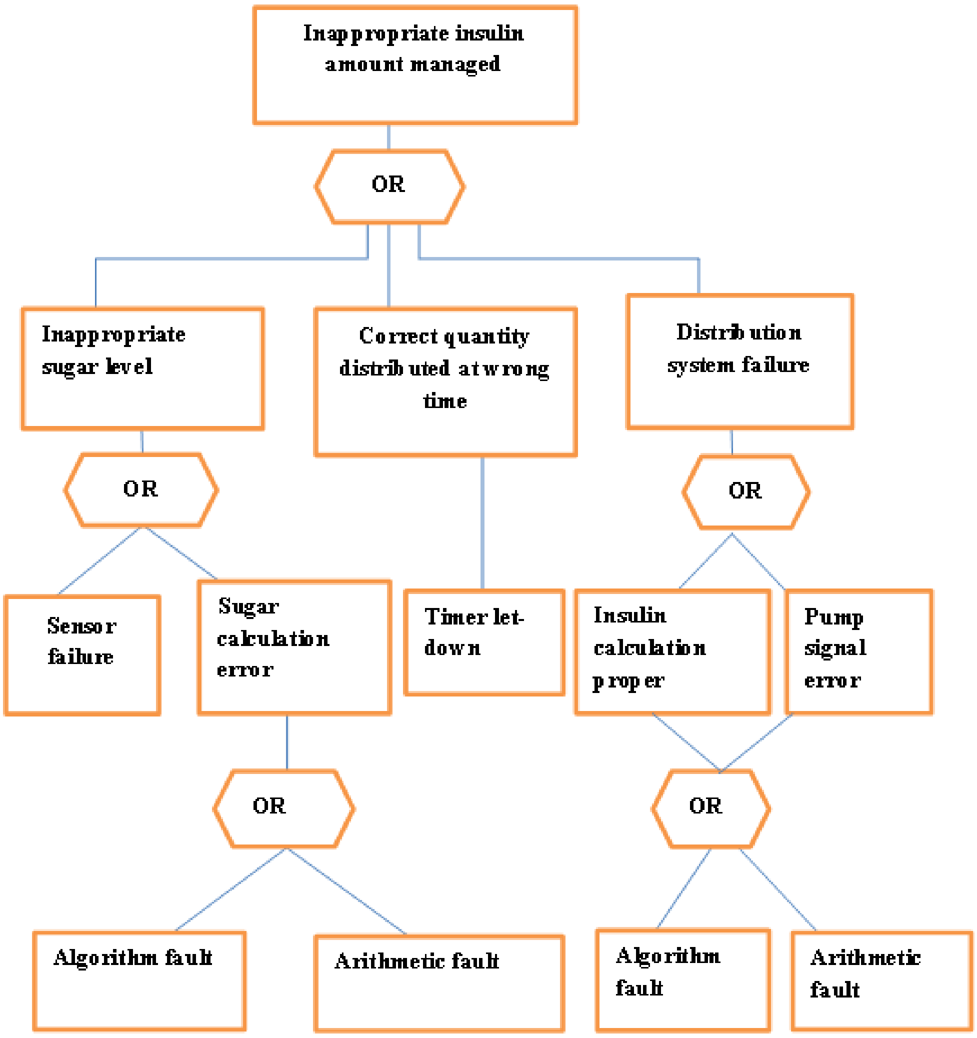
Inappropriate insulin amount managed
OR
Inappropriate sugar level
Correct quantity distributed at wrong time
Distribution system failure
OR
OR
Sensor failure
Sugar calculation error
Timer let-down
Insulin calculation proper
Pump signal error
OR
OR
Algorithm fault
Arithmetic fault
Algorithm fault
Arithmetic fault

Figure.3 Simulation of healthcare system

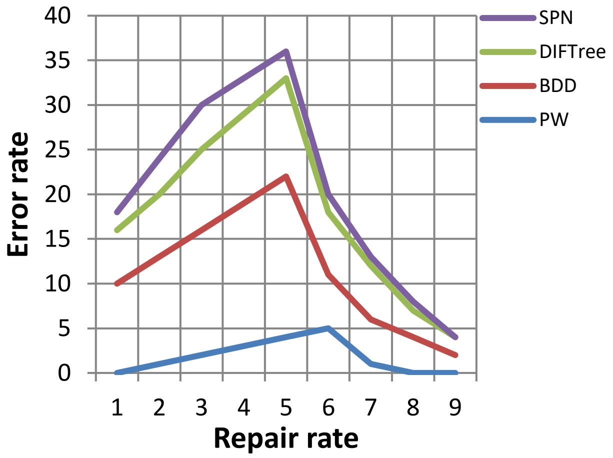

Figure. 4 Repair Rate Vs Error Rate

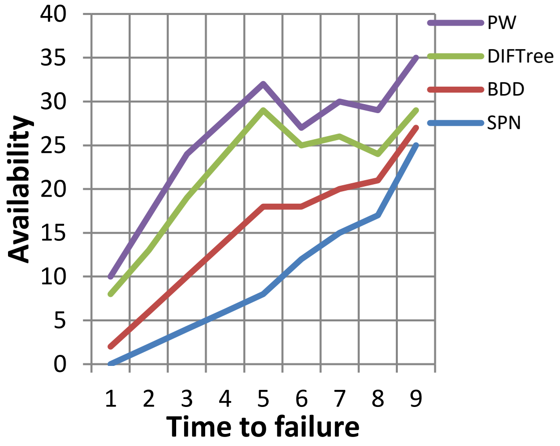

Figure. 5 Time to failure Vs Availability

Table 2. Comparison of proposed work with existing techniques

Work carried out	**Tools used**	**Analytical model/ Numerical model**	**Quality of service factors**	**Metrics of QOS achieved**
An Approach to Evaluate the Availability of System in Cloud Computing Using Fault Tree Technique (proposed work)	SHARPE	Analytical model	Steady state availability	Energy, failure rate, repair rate, Cost, response time, network traffic information.
Availability modelling of fault tolerant system using Markov chain model	SHARPE	Analytical model	Steady state availability	Cost, energy, failure rate, repair rate.
An Efficient Placement Algorithm for Data Replication and To Improve System Availability in Cloud Environment	Java (jdk 1.6)	Analytical model	Data Replication and Performance enhancement.	Down time and the acquisition cost
A QoS Aware Cloud Service Composition Algorithm for Geo-Distributed Multi Cloud Domain	PROMETHEE, multi-criteria decision making (MCDM)	Analytical model	Storage availability, system availability, and user-perceived availability	Response time, Throughput, Availability, Successability, Price.

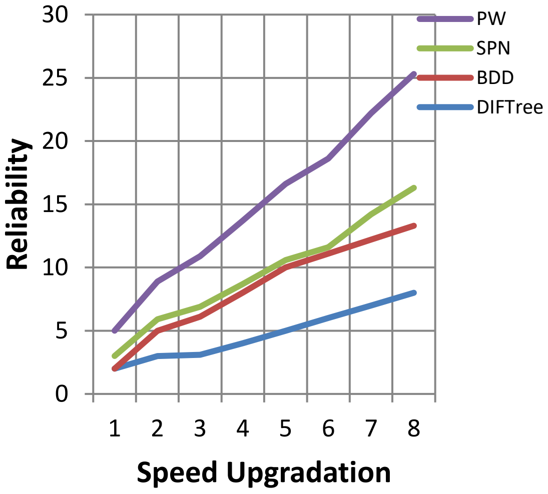

Figure. 6 Speed up gradation Vs Reliability

6. Conclusion

The primary objective of the paper is developing a dynamic model for the assessment of failure factors using the fault tree analysis. After presenting the success and failure factors, we concentrated on constructing a typical system fault tree by recognising the components that distress the system availability and reliability in the cloud environment. In this paper, we have computed the probability evaluation technique using Markov chain model, and the same method is used to identify the failures and repair rate of the components in the cloud environment. We simulated our proposed model, the availability of the cloud queuing system. The availability is measured by using cost, repair time, failure rate and response time. The simulation result discloses that our proposed approach attains better performance compared to the existing method such as BDD, QSC_MCD, MCDM and SPN techniques. The result also disclosures that the proposed method is efficient, rational and address enhance usage of the virtual machines with apt failure detection and energy saving factors from the datacentre view. In future, we aim to encompass the work with the QOS factors using availability modelling risk factors of cloud computing. Simulation result shows that fact that repair rate has decreased by 15% compared to the existing model, and increases the performance effectively. In future we intend to extend our work to identify the risk factors in availability modelling in cloud environment.

References

[1] E. Kajo , E.Driza, "An approach to evaluate the reliability of Web applications in cloud computing using Dynamic fault tree", In: *Proc. of International Conf. On Informatics*, Balgan, pp.151-157, 2013.

[2] A.Butoi, C. Alexandru, A.Stan and G. Cosmin, "Autonomous management of virtual machine failures in IaaS using fault tree analysis", In: *Proc. of International Conf. On grid economics and Business models*, pp.206-221, 2014.

[3] A. Ali Baig, R.Ruzli and B.Buang, "Reliability Analysis using fault tree analysis: A Review", *International Journal of Chemical Engineering and Applications*, Vol. 4, No.3, pp. 169-173, 2013.

[4] F. Chiacchi, L. Compagno, D. Urso, G. MAnno and N. Trapani, " An open-source application to model and solve dynamic fault tree of real industrial systems", In: *Proc. of International Conf. On SKIMA*, Vol.2, pp.1-8, 2011.

[5] X. Xu, L. Zhu, I. Weber,L. Bass, D. Sun, "POD-Diagnosis: Error Diagnosis of Sporadic Operations on Cloud Applications", In: *Proc. of International Conf. On DSN*, Vol.37, pp.1-12, 2014.

[6] R. Manian, J. Bechta, "Combining various solution techniques for dynamic fault tree analysis of computer systems", In: *Proc. Of International Symposium On High-Assurance Systems Engineering*, pp.1-7, 1998.

[7] W. Chu, and Y. Feng, "Fault Tree Based Architectural Analysis for E-Business Systems", In: *Proc. of International Symposium On Computer Science and Computational Technology*, pp. 52-57, 2009.

[8] J. Bechta, J. Sullivan David Coppit, "Developing a Low-Cost, High-Quality Software Tool for Dynamic Fault Tree Analysis", *IEEE Transactions on Reliability*, Vol.49, pp.49-59, 2002.

[9] G. Nieuwhof, "Introduction to Fault Tree Analysis with Emphasis on Failure Rate Evalauation", *Microelectronics and Reliability*, Vol. 14, No.2, pp. 105-119, 1970.

[10] D. Long, Thomas, A *fault tree Manual (pdf). Master's Thesis* (Texas A& M University), 1970.

[11] C. Eckberg, *Fault Tree Analysis Program Plan , Seattle*, WA: The Boeing Company, 1964.

[12] A. Hixenbaugh, *Fault Tree for Safety. Seattle*, WA: The Boeing Company, 1968.

[13] L. DQ, J. SH, Y. Chen, C. Zhou, "System reliability analysis of rock slope stability involving correlated failure modes", *Journal of Civil Engineering*, Vol.15, pp.1349-1359, 2011.

[14] B. Ford, Tim and R. Cooke, *Probabilistic Risk Analysis: Foundation and methods*, Cambridge University Press, NewYork, 2001.

[15] K.S. Trivedi, *Probability and Statistics with Reliability, Queuing and Computer Science Applications*, Vol.2, Wiley, 2001.

[16] M. Afshari, A. Ghaffaripour, "Modelling of Imperfect data in medical sciences by Markov chain with Numerical computation", *Advances in Bioscience and Biotechnology*, Vol.5, pp. 1003-1008, 2014.

[17] A. Saleh. M. Mosa hoi Yoo, L. sheets, "A systematic review of Healthcare application for smartphones", *Medical informatics and decision making*, Vol.12, pp.1-31, 2012.

[18] *C.* Rolim, C. Oberdan, "A cloud computing solution for patients data collection in health care institutions" , In: *Proc. of International Conf. On eHEalth, Telemedicine, and Social Medicine*, pp.95-99,2010.

[19] B. Dugan, J. Kevin, J. Sullivan, "Quality Software Tool for Dynamic Fault- Tree Analysis", In: *Proc. Of Symposium on Software Reliability Engineering*, Vol.49,pp.49-59, 2002.

[20] E. Vesely, *Fault Tree Handbook*, United States Nuclear Regulatory Commission Press, Washington, 1993.

[21] C. Raiter, *Extended Fault Trees Analysis supported by Stochastic Petri Nets*, Doctoral Thesis, Torino, Italy, 2005.

[22] J. Jhawar, R. Piuri, *Fault tolerance and resilience in cloud computing environment*, Vol.2, ACM Press, United States, 2013.

[23] D. Kim, D. Machida, K.S.Trivedi, "Availability modeling and analysis of a virtualized System", In: *Proc. of International Symposium On Dependable Computing*, pp. 365- 371, 2009.

[24] J. Dugan , K. S. Trivedi, "Coverage modeling for dependability analysis of fault- tolerant

systems.", *IEEE Transactions on Computers,* Vol. 38, No.6, pp. 775-787, 1989.

[25] R. Gulati, J. Dugan, "A modular approach for analyzing static and dynamic fault trees." *In: Proc. of International Conf. On Annual Reliability and Maintainability Symposium,* Vol. 45, pp.1-7, 1997.

[26] J. Dugan, B. Venkataraman and R. Gulati, "DIFtree: A software package for the analysis of dynamic fault tree models", *Annual Reliability and Maintainability Symposium,* Vol. 47, pp. 1-7, 1997.

[27] P. Gill, N. Jain, N. Nagappan, "Understanding Network Failures in Data Centers: Measurement, Analysis and Implications", *ACM Computer Communication Review,* Vol.41, pp.350-361, 2011.

[28] S. Karuppusamy, M. Muthaiyan, "An Efficient Placement Algorithm for Data Replication and To Improve System Availability in Cloud Environment", *International Journal of Intelligent Engineering and Systems,* Vol.9, No.4, pp. 88-97, 2016.

[29] S.Bhushan, Pradeep Reddy.CH, "A QoS Aware Cloud Service Composition Algorithm for Geo-Distributed Multi Cloud Domain", *International Journal of Intelligent Engineering and Systems,* Vol.9, No.4, pp.147-156, 2016.

[30] D. Mani, A. Mahendran, "Availability Modelling of Fault Tolerant Cloud Computing System", *International Journal of Intelligent Engineering and Systems,* Vol. 10, No.1, pp.154-165, 2017.