Analysis of Measures to Achieve Resilience during Virtual Machine Interruptions in IaaS Cloud Service

Priya Vedhanayagam[1], Subha S.[1], Balamurugan Balusamy[1], P. Vijayakumar[2] and Victor Chang[3]
[1]*School of Information Technology and Engineering, VIT University, Vellore, Tamilnadu, India*
[2]*University of College of Engineering Tindivanam, Melpakkam, Tamilnadu, India*
[3]*International Business School Suzhou, Xi'an Jiaotong Liverpool University, Suzhou, 215123, China*

Keywords: Cloud Computing, IaaS, Performance Evaluation, Queueing, Virtual machine, Resilience.

Abstract: In cloud computing era, the resilience issues faced by cloud computing services may be high. And therefore, the best alternative to reckon with the effects on the Quality-of-Service is to preserve resilience of Cloud computing service. To address this issue, an analytical model is proposed to study queueing system to handle various virtual machine interruptions. The proposed model recommends a secondary virtual machine to redeem the primary virtual machine during a probable halt. The work highlights the innovation employed for analysing the measures to achieve resilience during virtual machine interruptions in IaaS cloud service, the main objective of this research. The model is simulated using SHARPE and the results declare guaranteed performance for the IaaS clients to achieve high availability of service as the response time never deflate during VM interruptions.

1 INTRODUCTION

Cloud computing is the most advanced technology in the realms of computing and its services are attaining the stature of an intrinsic value that governs one's day to day life. Cloud services are supported by a framework namely Internet Data Center (IDC) (Armbrust, M.,et al., 2010). Nourished by the broad accessibility of rapid web access, and nurtured by the need to encourage clients to lessen IT operational costs, the utilization of Cloud computing has expanded widely in the past several years. Cloud Computing depends on a service-oriented architecture and renders three classifications of services namely Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). The IaaS is concerned with hardware, storage, servers, and networking components over the Internet. The PaaS concentrates on virtualized servers, operating systems, and other hardware and software computing platforms. And the SaaS delivers application software and other services to host the application (Rimal, B. P., et al., 2011). The entrepreneur's chief aim is to achieve more computing facilities and benefits with fewer resources in different environments. The cloud technology is a gift box loaded fully with benefits like adaptability, disaster recovery, automatic software upgrades, free capital-investment, expanded collaboration, work from any-place, document control, security, competitiveness and ecologically friendly.

Resilience is transforming into an essential service primitive for numerous cloud computing applications. Metrics for resilience are greatly associated with dependability metrics that are based on availability, performance and survivability. In our proposed model, resilience is defined as the capability of a system to recover from various virtual machine interruptions. To appraise resilience, we exploit dependability attributes of systems such as availability and performance (Javadi, B., et al., 2013). This model makes a deep study of resilience analysis of IaaS cloud (Ghosh, R.,et al., 2010) considering various interruption states of virtual machine. These interruptions result in downtime which degrades the overall performance of the system and violate the Quality-of-Service specified in the Service Level Agreement and significantly affect the availability of the system.

As per our knowledge, there is practically no existing work that asserts these issues, as will be seen in Section 2 below. To overcome these issues, the proposed system models the cloud data centre

Vedhanayagam, P., S., S., Balusamy, B., Vijayakumar, P. and Chang, V.
Analysis of Measures to Achieve Resilience during Virtual Machine Interruptions in IaaS Cloud Service .
DOI: 10.5220/0006419904490460
In *Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security (IoTBDS 2017)*, pages 449-460
ISBN: 978-989-758-245-5

having $M^{[x]}/G/1$ queueing system (Baba, Y, 1987) as an internal queue that takes the batch of task arrivals to a single virtual machine, on the assumption that each task is serviced by a single VM. During any of these interruptions, a secondary virtual machine is quickly substituted to attend to the scheduled task in the primary VM. This creates high availability of resources without substantial downtime.

The remainder of the paper is organized as follows: Section 2 presents the related work in performance analysis of cloud data centers; Section 3 depicts the evolution of our model and the details of the analysis. Section 4 introduces an analytical model considering different virtual machine interruptions. Section 5 discusses the different special conditions for analysing the parameters. Section 6 lists out the numerical results obtained from the analytical model and Section 7 summarizes the results.

2 RELATED WORKS

Over the years, cloud computing has pulled in extensive research attention, however just a small portion of the work has been carried out to address performance and resilience issues by analytical models.

The primary commitment of the researchers Khazaei, H., et al. (2011a) is to create performance models for cloud computing centers. They proposed a basic analytical model, using M/G/m queueing system for cloud computing centers to examine the most important aspects of present day cloud centers. Based on this work, the authors had published the book titled "Cloud Computing: Performance Analysis" (Wang, L., et al. 2011) by extending their work to consolidate the cloud's focuses with a hyper-exponential family. Later on, they included presumptions of finite capacity for cloud center, which made their model to be more like a genuine cloud center (Khazaei, H., et al. 2011b). Interestingly, Khazaei et al. (2012a) developed M/G/m/m+r queueing system, a conceptual model to manage the performance assessment of a cloud center with the two-stage approximation technique. Specifically, it permits precise modeling of cloud centers with countless Physical Machines. They extended their work (Khazaei et al. 2012b) to meet the demands of challenging circumstances where virtualization could be used to contribute a versatile characterized set of computing resources that would have high degree of virtualization. Khazaei, H., et al. (2013a) had improved the proposed analytical model to fasten critical perspectives such as pool administration, power utilization, resource allocation process and virtual machine organization of present day cloud centers. Khazaei et al. (2013b) introduced a performance model sensible for large IaaS clouds, utilizing interactive stochastic models. Khazaei et al. (2013c, 2013d) had furthermore presented an interactive stochastic model which was realistic for cloud computing centers with heterogeneous demands and resources. Moreover, in particular, a client's task may ask for various sorts of VMs.

Bruneo, D., et al. (2010) exhibited a novel strategy to interpret WS-BPEL forms into non-Markovian stochastic Petri nets which, when permitted, would systematically assess the importance of performance indices primarily, and evaluate the performance of web service at the initial design stage. The similar work was tended by Bruneo et al. (2011) with an objective of assessing various service parameters. Bruneo et al. (2013a) assessed QoS oriented performance analyses through the estimation of steady-state measures and by inspecting the delays presented in service endowment with an ultimate aim of decreasing energy costs. Bruneo et al. (2013b) offered an analytical model based on their previous work that could undoubtedly actualize resource allocation policies in a Green Infrastructure-as-a-Service cloud. Bruneo et al. (2013c) provided a dual solution for fluctuations in the workload. One way was through the conservation of reliability principle, and the other was to optimize the Virtual Machine Monitor software in case of an occurrence of workload changes. Bruneo et al. (2014) proposed a stochastic model to validate the performance of IaaS cloud service.

Ghosh et al. (2010b) adopts a rapid and reasonable technique for examining service excellence of huge sized IaaS cloud after quantifying the effects of variations in workload, fault load, system capacity and developed submodels for failure, repair, and migration of Physical Machines in Cloud using scalable and highly reliable stochastic models. Ghosh et al. (2010a) felt that resiliency quantification would be a critical task and he extends his previous work with an awareness to manage non-homogeneous interacting sub-models. The researcher's key inspiration for driving and creating adaptable stochastic models for the cloud is to help the service provider through what-if analyses utilizing the execution model created from his previous model (2010b). The Power-performance trade-off analysis for IaaS Cloud (2011)

demonstrates that natural gathering of Physical Machines taking into account their power utilization and response time conduct may not prompt craving results and portray the cost breakdown and capacity scheduling (Ghosh et al. ,2013; 2014a; 2014b).
Bacigalupo et al. (2011), Yang et al. (2013), Tian et al. (2013), Khomonenko and Gindin (2014), Vilaplana et al. (2014), Cao et al. (2014), Guo et al. (2014), Cheng et al. (2015), Liu et al. (2015), Sousa et al. (2015), Mei et al. (2015), Liu et al. (2015), Xia et al. (2015a, 2015b), Liu, X et al. (2014, 2015), Zhang et al. (2016), are the different research works that deal with various queueing models and provided diverse solutions for evaluating performance measures.

As per our studies on the works published previously, it is noted that this research work effectively contributes to the analysis of measures during the resilience of cloud services during different VM interruptions. Our proposed work applies Sumudu transform on M[x]/G/1 queueing system to overcome certain deficiencies found in other transforms like Fourier, Laplace, Mellin, etc., and also to make expressions simple, more intuitive and applicable in different cases (Khalaf & Belgacem, 2014).

3 SERVICE MODEL OF THE PROPOSED SYSTEM

In an IaaS cloud service model, the IaaS providers host client's applications and deal with responsibilities including system maintenance, backup, and resiliency planning. IaaS platforms strive to offer an on-demand scalable resource which makes IaaS appropriate for inconsistent workloads. Client's workloads may be severely affected, when an IaaS provider experiences downtime. Figure 1 shows the proposed IaaS cloud service model. An IaaS provider's responsibility is to provide service to more and maximum clients with high availability. In the IaaS cloud infrastructure, every physical machine in the physical layer serves as a virtualized environment through a Virtualization Server on top of which one or more VMs can be instantiated. A hypervisor decouples the VMs from the physical host and allocates resources dynamically to each VM as per the requirement to provide service for the client's task. Here, we assume that each task is served by a single virtual instance, i.e., primary VM. A Queueing system serves as an internal queue to study the resilience of the system during different VM interruptions.

Figure 2 shows the metrics Up Time and Down Time of the system during different interruptions like the vacation period of the primary VM, the extended vacation period of the primary VM, the repair period of primary VM and a delay time during the repair period. During these interruptions, a primary virtual machine can be supported by a secondary virtual machine to achieve resilience (VMware vLockstep, 2017). The secondary VM works along with primary VM in perfect synchrony and at the event of every primary VM interruption, the interrupted task goes to the head of the queue. An instance of secondary VM is created automatically, and it handles the task from the queue. We assume that once the primary VM recovers from the interruption, the workload is transferred back to the primary VM.

According to the problem design, from an operational point of view, the primary VM can be represented as a finite state machine categorized by different operating states for different primary VM interruptions as depicted in Figure 3. Once started, the primary virtual machine is in the active working state S_T with working event e_{wrk}, it might enter into any of the different interruption states. When the primary VM goes on vacation, it enters state S_{Ov} with an event e_{Ov} for the vacation period, and after completing the vacation, it enters state S_α with an event e_α and resumes the active state S_T. After returning, the primary VM can go for an extension of vacation, it enters state S_{Ev} with an event e_{Ev} for the extended vacation period, after completing the extended vacation enters state S_β with an event e_β and resumes the active state S_T. In case of any unexpected failure, the primary VM goes to repairing, and enters state S_{Ur} with an event e_{Ur} under repair period, and after completing the repairing process, enters state S_ω with an event e_ω and resumes active state S_T. The primary VM can go to repair after some time, and enters state S_{Ds} with an event e_{Ds} for a repair period after a delay, and after completing the repairs it enters state S_δ with an event e_δ and resumes active state S_T. In the event of any interruption, the primary VM gets automatically triggers the event e_{trig} to swap its state

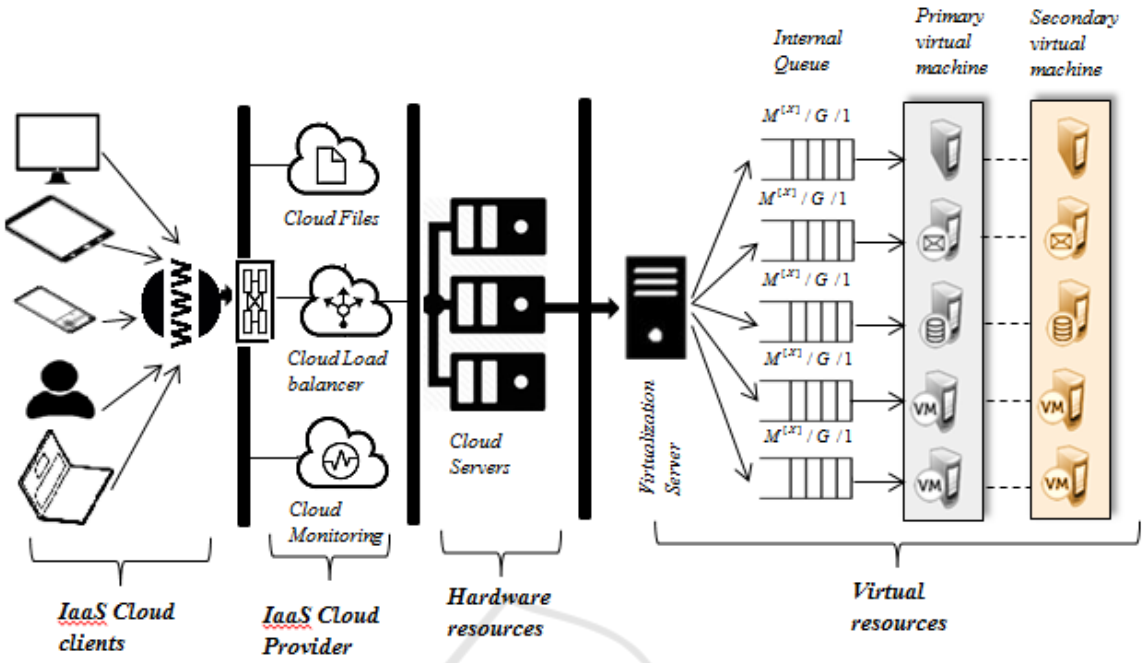

Figure 1: IaaS cloud model supported by secondary VM with $M^{[X]}/G/1$ internal queueing system.

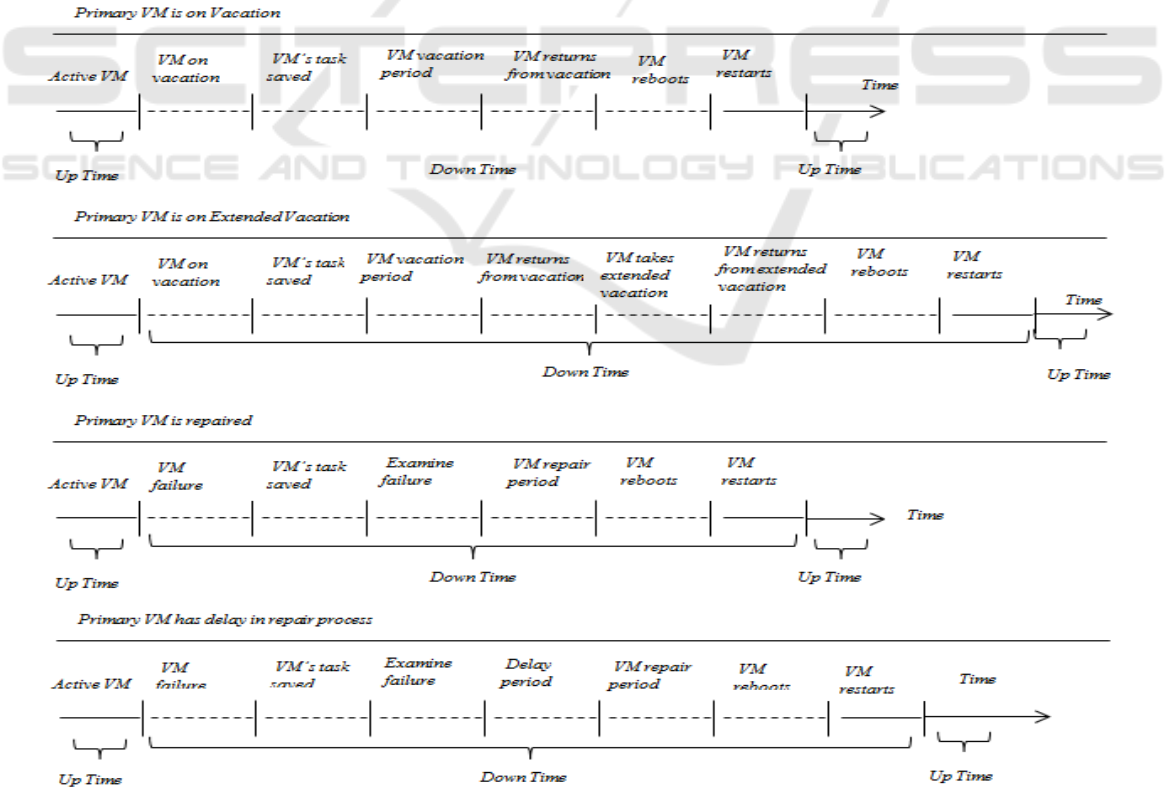

Figure 2: Timing metrics during different primary VM interruptions.

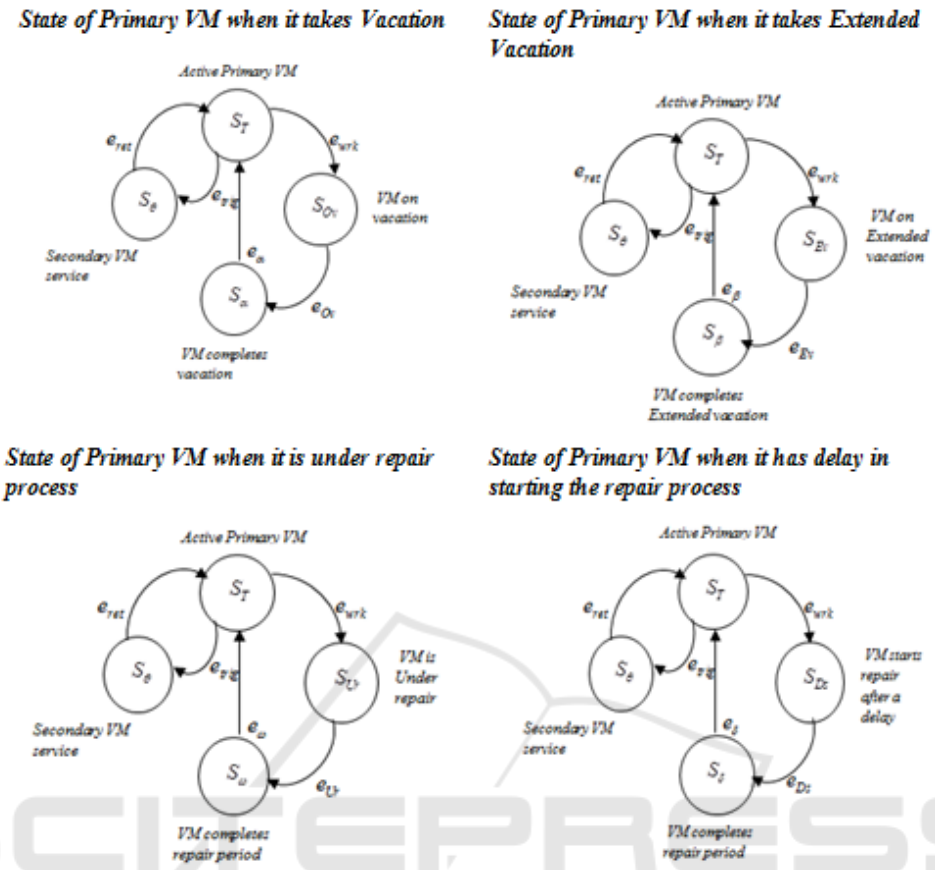

Figure 3: Operating states of primary VM during different interruptions.

S_T to secondary VM service state S_θ to handle the workload. Once the primary VM completely recovers, the secondary VM swaps its state S_θ to S_T with an event e_{ret}.

Tools of Queuing theory can be utilized analytically to examine the behaviour of the system described above and it resolves the most interesting performance factors such as response time, task blocking probability, probability of immediate service, mean number of tasks, mean number of customers, mean number of customers in the queue and the mean waiting time in the system. In reality, the service or server might face some accidental failure or breakdown. In such contents, the service provider can't provide reliability and availability, until the system recovers from the failure. The main motive of our proposed work is to consider the impact of resilience and great recovery options for various virtual machine interruptions in an IaaS cloud service. The IaaS clients should not experience any downtime, so as to have a high resilient system during different primary VM interruptions.

4 QUEUEING MODEL OF THE PROPOSED SYSTEM

In this work our focus was on $M^{[X]}/G/1$ queueing framework. Client's tasks arrive in batches of size k to the "Primary VM" and given service in an FCFS fashion $\lambda b_k \ (k = 1,2,3...)$. When the Primary VM becomes unavailable due to various interruptions, a "Secondary VM" is assigned automatically to provide continuous service to the clients in the same manner. Primary VM provides service with conditional probability $\mu(v)dv$ for a period of time $(v, v+dv)$. $\alpha(v)\Delta v$ is the probability of Primary VM completing its idle period and $\beta(v)dv$ is the probability of Primary VM completing its extended idle period. Breakdown

times of the primary virtual machine trail Poisson distribution using mean breakdown rate, $\psi > 0$. For the period of primary VM failure, the task whose service is hindered returns to the queue head, however, it is immediately taken up for service by the secondary VM and the repair procedure may begin at any time. $\omega(v)\Delta v$ is the probability of the Primary VM recovering after the repair and $\delta(v)\Delta v$ is the probability of the Primary VM returning to the active state after a delay time. The service rate of the Secondary VM follows an exponential distribution of $\theta > 0$. When the primary VM recovers from the interruptions, the task currently served by the secondary VM that swaps over to the primary virtual machine to begin the service. Every single stochastic procedure required in the framework is autonomous.

5 PROPOSED ANALYTICAL MODEL DEPICTING DIFFERENT INTERRUPTIONS OF THE PRIMARY VM

5.1 Service Time for a Virtual Machine

$$\frac{\partial}{\partial v}T_j(v) = -\left(\lambda + \mu(v) + \psi\right)T_j(v) + \lambda\sum_{k=1}^{j-1} b_k T_{j-k}(v) \tag{1}$$

$$\frac{\partial}{\partial v}T_0(v) = -\left(\lambda + \mu(v) + \psi\right)T_0(v) \tag{2}$$

5.1.1 Boundary Condition

$$T_j(0) = (1-m)\int_0^\infty T_{j+1}(v)\mu(v)dv + (1-n)\int_0^\infty Ov_{j+1}(v)\alpha(v)dv + \int_0^\infty Ev_{j+1}(v)\delta(v)dv + \int_0^\infty Ur_{j+1}(v)\omega(v)dv + {}_S\lambda b_{j+1} \tag{3}$$

5.1.2 Probability Generating Function

$$T_r(x) = \frac{-Su\left[1 - F^*(q)\right]}{q\left\{x - F^*(q)\left[1 - m + mA^*(\eta)e\right]\right\} - \psi xL} \tag{4}$$

5.2 Primary Virtual Machine on Idle Time

$$\frac{\partial}{\partial v}Ov_j(v) = -\left(\lambda + \alpha(v) + \theta\right)Ov_j(v) + \lambda\sum_{k=1}^{j} b_k Ov_{j-k}(v) + \theta Ov_{j+1}(v) \tag{5}$$

$$\frac{\partial}{\partial v}Ov_0(v) = -\left(\lambda + \alpha(v) + \theta\right)Ov_0(v) + \theta Ov_1(v) \tag{6}$$

5.2.1 Boundary Condition

$$Ov_j(0) = m\int_0^{\infty} T_j(v)\mu(v)dv \tag{7}$$

5.2.2 Probability Generating Function

$$Ov_r(x) = \frac{-qmSu\left[1-A^*(\eta)\right]}{\eta q\left\{x-F^*(q)\left[1-m+mA^*(\eta)e\right]\right\}-\psi x\eta L} \tag{8}$$

5.3 Primary Virtual Machine on an Extended Idle Time

$$\frac{\partial}{\partial v}Ev_j(v) = -(\lambda+\delta(v)+\theta)Ev_j(v) + \lambda\sum_{k=1}^{j} b_k Ev_{j-k}(v) + \theta Ev_{j+1}(v) \tag{9}$$

$$\frac{\partial}{\partial v}Ev_0(v) = -(\lambda+\delta(v)+\theta)Ev_0(v) + \theta Ev_1(v) \tag{10}$$

5.3.1 Boundary Condition

$$Ev_j(0) = n\int_0^{\infty} Ov_j(v)\alpha(v)dv \tag{11}$$

5.3.2 Probability Generating Function

$$Ev_r(x) = \frac{-qmnSuF^*(q)A^*(\eta)\left[1-C^*(\eta)\right]}{\eta q\left\{x-F^*(q)\left[1-m+mA^*(\eta)e\right]\right\}-\psi x\eta L} \tag{12}$$

5.4 Primary Virtual Machine is under Repair

$$\frac{\partial}{\partial v}Ur_j(v) = -(\lambda+\omega(v)+\theta)Ur_j(v) + \lambda\sum_{k=1}^{j} b_k Ur_{j-k}(v) + \theta Ur_{j+1}(v) \tag{13}$$

$$\frac{\partial}{\partial v}Ur_0(v) = -(\lambda+\omega(v)+\theta)Ur_0(v) + \theta Ur_1(v) \tag{14}$$

5.4.1 Boundary Condition

$$Ur_j(0) = \int_0^{\infty} Ds_j(v)\beta(v)dv \tag{15}$$

5.4.2 Probability Generating Function

$$Ur_r(x) = \frac{-\psi xSu\left[1-F^*(q)\right]D^*(\eta)\left[1-B^*(\eta)\right]}{\eta q\left\{x-F^*(q)\left[1-m+mA^*(\eta)e\right]\right\}-\psi x\eta L} \tag{16}$$

5.5 Primary Virtual has Delay in Recovery

$$\frac{\partial}{\partial v}Ds_j(v) = -(\lambda+\beta(v)+\theta)Ds_j(v) + \lambda\sum_{k=1}^{j} b_k Ds_{j-k}(v) + \theta Ds_{j+1}(v) \tag{17}$$

$$\frac{\partial}{\partial v}Ds_0(v) = -(\lambda+\beta(v)+\theta)Ds_0(v) + \theta Ds_1(v), \quad \frac{\partial}{\partial v}Ds_0(v) = 0 \tag{18}$$

5.5.1 Boundary Condition

$$Ds_j(0) = \psi\int_0^\infty T_{j-1}(v)\,dv = \psi T_{j-1}$$
$$Ds_0(0) = 0 \tag{19}$$

5.5.2 Probability Generating Function

$$Ds_r(x) = \frac{-\psi x Su\left[1-F^*(q)\right]\left[1-D^*(\eta)\right]}{\eta q\left\{x-F^*(q)\left[1-m+mA^*(\eta)e\right]\right\}-\psi x\eta L} \tag{20}$$

5.6 Overall System

$$\lambda S = (1-m)\int_0^\infty T_0(v)\mu(v)\,dv + (1-n)\int_0^\infty Ov_0(v)\alpha(v)\,dv + \int_0^\infty Ev_0(v)\delta(v)\,dv + \int_0^\infty Ur_0(v)\omega(v)\,dv + \int_0^\infty Ds_0(v)\beta(v)\,dv \tag{21}$$

$$Q_r(x) = T_r(x) + Ov_r(x) + Ev_r(x) + Ds_r(x) + Ur_r(x) \tag{22}$$

Probability Generating Function

$$Q_r(x) = \frac{-Su\left[1-F^*(q)\right]\{\eta+\psi x[\phi]\} + qmF^*(q)\left[1-A^*(\eta)e\right]}{\eta q\left\{x-F^*(q)\left[1-m+mA^*(\eta)e\right]\right\}-\psi x\eta L} \tag{23}$$

Substitute the subsequent values in the above equations $q = \lambda - \lambda Ba(x) + \psi$, $\eta = \lambda - \lambda Ba(x) + \theta - \theta/x$, $L = \left[1-F^*(q)\right]D^*(\eta)B^*(\eta)$, $u = \lambda - \lambda Ba(x)$, $e = 1-n+nC^*(\eta)$, $\phi = 1-D^*(\eta)B^*(\eta)$, A^*, B^*, C^*, D^* and F^* are the Sumudu transforms used in the equations and the Sumudu transform for any function is $S\left[f(t)\right] = \int_0^\infty f(ut)e^{-t}dt$. During the applying normalization condition, one finds S, $Q_r(1) + S = 1$.

$MQL_r = \frac{d}{dx}Q_r(x)\big|_{x=1}$, using this relation we can find the steady state of Mean number of tasks served in the virtual machine. Little's law is applied to find Mean waiting time of a task in the virtual machine, $MRT_r = MQL_r / \lambda$. Average number of task in the queue is calculated as $AQL = MQL_R + \rho$ where ρ is the traffic intensity. Average waiting time of the task in the queue can be calculated, using Little's Law $ART = AQL / \lambda$ (Little & Graves, 2008).

6 NUMERICAL VALIDATIONS

The analytical model is simulated, using SHARPE tool (Trivedi & Sahner, 2009). An extensive variety of qualities were established for our model parameters so that the model can state to an expansive assortment of cloud service provider. We assume that 1, 2, 4 or 8 Primary VMs are deployed on a single Physical machine supported by parallel Secondary VMs. The mean arrival rate of tasks to a primary VM is $\lambda > 0$ (we classify 500 to 1500 tasks per hour). Mean Service time $1/\mu$ (from examination 30 minutes to 1 hour). Mean delay to search a VM $1/\delta$ (for current study 1 to 5 seconds). Breakdown times of the virtual machine trail Poisson distribution using mean breakdown rate $\psi > 0$ (few hours). According to the research work to check the legitimacy of the results obtained, we have studied the service time, idle times, delay times, extended idle times and repair times and these timings appear to be exponentially distributed. To fulfil the stability conditions, all values have been chosen subjectively.

Table 1: Performance measures for the proposed system $\mu = 7, \alpha = 5, \omega = 4, \lambda = 2, \psi = 2, m = 0.5, n = 0.5.$

β / δ	ART			
	$\beta=5$	$\beta=6$	$\beta=7$	$\beta=8$
$\delta=3$	6.9422	5.7059	5.0579	4.6597
$\delta=5$	3.0695	2.7392	2.5423	2.4119
$\delta=7$	2.385	2.1595	2.0219	1.9294
$\delta=9$	2.1031	1.9155	1.7998	1.7215

Table 2: Performance measures for the proposed system $\mu = 7, \alpha = 5, \omega = 4, \lambda = 2, \psi = 2, m = 0.5, n = 0.5.$

β / δ	AQL			
	$\beta=5$	$\beta=6$	$\beta=7$	$\beta=8$
$\delta=3$	13.8843	11.4118	10.1157	9.3194
$\delta=5$	6.139	5.4784	5.0846	4.8238
$\delta=7$	4.777	4.3191	4.0438	3.8588
$\delta=9$	4.2063	3.8309	3.5996	3.4431

Table 3: Performance measures for the proposed system $\mu = 7, \alpha = 9, \lambda = 2, \psi = 2, m = 0.5, n = 0.5.$

θ / ω	MQL_r		
	$\theta=0$	$\theta=1$	$\theta=3$
$\omega=2$	13.9042	8.9989	5.4294
$\omega=3$	5.5188	4.3426	3.1283
$\omega=4$	3.8724	3.1876	2.4194
$\omega=7$	2.5682	2.1972	1.7525

Table 4: Performance measures for the proposed system $\mu = 7, \alpha = 9, \lambda = 2, \psi = 2, m = 0.5, n = 0.5.$

ω / θ	ART			
	$\omega=2$	$\omega=3$	$\omega=4$	$\omega=7$
$\theta=0$	7.4028	3.1625	2.3155	1.6328
$\theta=1$	4.9264	2.5531	1.9531	1.4289
$\theta=3$	3.1011	1.9097	1.5348	1.1751

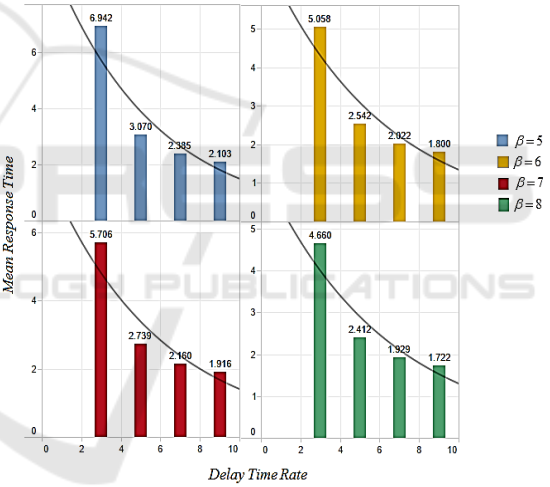

Figure 4: Delay time Vs. Mean response time of the VM.

The impact of the delay times and extended vacation times is depicted in Figure 4. Mean response time decreases, even when there is an increase in the delay rate, since secondary VM takes the responsibility of primary VM when it is idle for a long time.

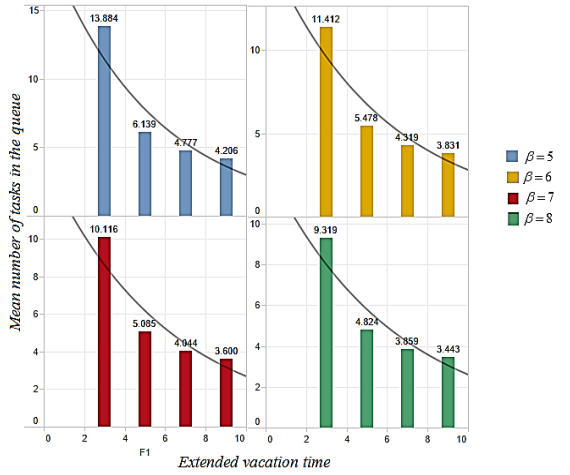

Figure 5: Extended vacation time Vs. Mean number of tasks in the queue.

The impact of the extended vacation times and Mean number of tasks in the queue is depicted in Figure 5. Mean number of tasks in the virtual machine decreases, even when there is an increase in the extended vacation times.

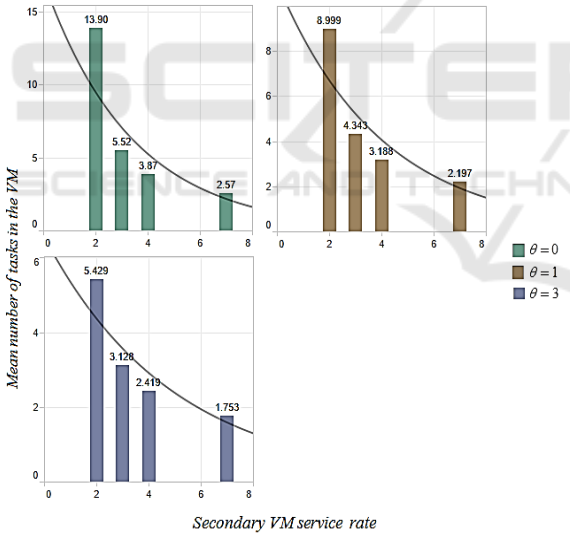

Figure 6: Secondary VM service Vs. Mean number of tasks in the VM.

The impact of the secondary VM service times and Mean number of tasks in the VM is depicted in Figure 6. Mean number of tasks in the queue decreases, even when there is an increase in the secondary service times and the repair rate times.

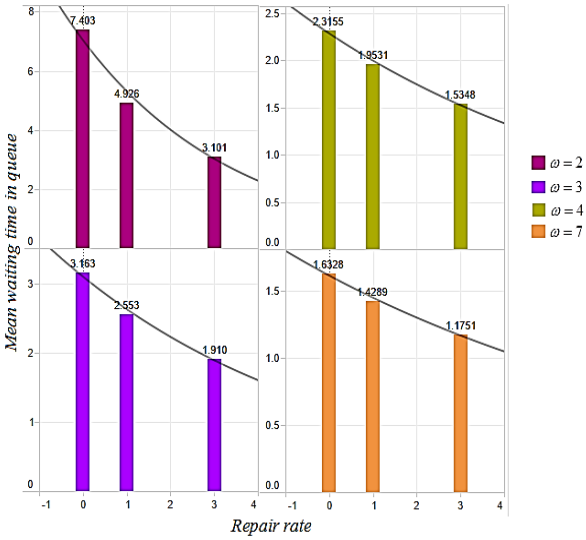

Figure 7: Repair time Vs. Mean waiting time in queue.

The impact of the repair times and Mean waiting time in the queue is depicted in Figure 7. Mean waiting time in the queue decreases, even when there is an increase in the repair times and the secondary service rate times.

7 CONCLUSION

In this paper, we studied the resilience for IaaS cloud and proposed an analytical model which probes deep into our modern data centre to bring new novelties. We are quite hopeful that this is an innovative and honest attempt in analysing the measures for the resiliency of IaaS cloud by considering $M^{[x]}/G/1$ queueing system and presents exceptional performance measures during distinct interruptions of primary virtual machine supported by a secondary virtual machine by utilizing the advantage of Sumudu transform. In future, we plan to extend our work in $M^{[x]}/G/1$ queueing system, as the task refuses to join the queue, i.e., balking; and the tasks leave the queue after entering, i.e., reneging during different virtual machine interruptions.

REFERENCES

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., & Zaharia, M. (2010). A view of cloud computing. Communications of the ACM, 53(4), 50-58.

Baba, Y. (1987). On the Mx/G/1 queue with and without vacation time under non-preemptive last-come first-served discipline. J. of Opns. Res. Society of Japan, 30, 150-159.

Bacigalupo, D. A., Van Hemert, J., Chen, X., Usmani, A., Chester, A. P., He, L., ... & Jarvis, S. A. (2011). Managing dynamic enterprise and urgent workloads on clouds using layered queuing and historical performance models. Simulation Modelling Practice and Theory, 19(6), 1479-1495.

Bruneo, D. (2014). A stochastic model to investigate data center performance and qos in iaas cloud computing systems. IEEE Transactions on Parallel and Distributed Systems, 25(3), 560-569.

Bruneo, D., Distefano, S., Longo, F., & Scarpa, M. (2010, April). Qos assessment of ws-bpel processes through non-markovian stochastic petri nets. In Parallel & Distributed Processing (IPDPS), 2010 IEEE International Symposium on (pp. 1-12). IEEE.

Bruneo, D., Distefano, S., Longo, F., & Scarpa, M. (2013c, October). Stochastic evaluation of QoS in service-based systems. IEEE Transactions on Parallel and Distributed Systems, 24(10), 2090-2099.

Bruneo, D., Distefano, S., Longo, F., Puliafito, A., & Scarpa, M. (2013a, June). Workload-based software rejuvenation in cloud systems. IEEE Transactions on Computers, 62(6), 1072-1085.

Bruneo, D., Lhoas, A., Longo, F., & Puliafito, A. (2013b, September). Analytical evaluation of resource allocation policies in green iaas clouds. InCloud and Green Computing (CGC), 2013 Third International Conference on(pp. 84-91). IEEE.

Bruneo, D., Longo, F., & Puliafito, A. (2011, June). Evaluating energy consumption in a cloud infrastructure. In World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a(pp. 1-6). IEEE.

Cao, J., Li, K., & Stojmenovic, I. (2014). Optimal power allocation and load distribution for multiple heterogeneous multicore server processors across clouds and data centers. IEEE Transactions on Computers, 63(1), 45-58.

Cheng, C., Li, J., & Wang, Y. (2015). An energy-saving task scheduling strategy based on vacation queuing theory in cloud computing. Tsinghua Science and Technology, 20(1), 28-39.

Ghosh, R., Longo, F., Frattini, F., Russo, S., & Trivedi, K. S. (2014a, March). Scalable analytics for iaas cloud availability. IEEE Transactions on Cloud Computing, 2(1), 57-70.

Ghosh, R., Longo, F., Naik, V. K., & Trivedi, K. S. (2010a, October). Quantifying resiliency of iaas cloud. In Reliable Distributed Systems, 2010 29th IEEE Symposium on (pp. 343-347). IEEE.

Ghosh, R., Longo, F., Naik, V. K., & Trivedi, K. S. (2013). Modeling and performance analysis of large scale iaas clouds. Future Generation Computer Systems, 29(5), 1216-1234.

Ghosh, R., Longo, F., Xia, R., Naik, V. K., & Trivedi, K. S. (2014b, December). Stochastic model driven capacity planning for an infrastructure-as-a-service cloud. IEEE Transactions on Services Computing, 7(4), 667-680.

Ghosh, R., Naik, V. K., & Trivedi, K. S. (2011, June). Power-performance trade-offs in IaaS cloud: A scalable analytic approach. In 2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W) (pp. 152-157). IEEE.

Ghosh, R., Trivedi, K. S., Naik, V. K., & Kim, D. S. (2010b, December). End-to-end performability analysis for infrastructure-as-a-service cloud: An interacting stochastic models approach. In Dependable Computing (PRDC), 2010 IEEE 16th Pacific Rim International Symposium on (pp. 125-132). IEEE.

Guo, L., Yan, T., Zhao, S., & Jiang, C. (2014). Dynamic performance optimization for cloud computing using m/m/m queueing system. Journal of Applied Mathematics, 2014.

Javadi, B., Thulasiraman, P., & Buyya, R. (2013). Enhancing performance of failure-prone clusters by adaptive provisioning of cloud resources. The Journal of Supercomputing, 63(2), 467-489.

Khalaf, R. F., & Belgacem, F. B. M. (2014, December). Extraction of the Laplace, Fourier, and Mellin transforms from the Sumudu transform. In AIP Conf. Proc (Vol. 1637, No. 142, pp. 6-1432).

Khazaei, H., Jelena, M., & Vojislav, B. M. (2013d). Performance evaluation of cloud data centers with batch task arrivals. Communication Infrastructures for Cloud Computing, 199-223.

Khazaei, H., Misic, J., & Misic, V. B. (2011a, June). Modelling of cloud computing centers using m/g/m queues. In 2011 31st International Conference on Distributed Computing Systems Workshops (pp. 87-92). IEEE.

Khazaei, H., Misic, J., & Misic, V. B. (2011b, December). Performance analysis of cloud centers under burst arrivals and total rejection policy. InGlobal Telecommunications Conference (GLOBECOM 2011), 2011 IEEE (pp. 1-6). IEEE.

Khazaei, H., Misic, J., & Misic, V. B. (2012a). Performance analysis of cloud computing centers using m/g/m/m+ r queuing systems. IEEE Transactions on parallel and distributed systems, 23(5), 936-943.

Khazaei, H., Misic, J., & Misic, V. B. (2013b, November). A fine-grained performance model of cloud computing centers. IEEE Transactions on parallel and distributed systems, 24(11), 2138-2147.

Khazaei, H., Misic, J., & Misic, V. B. (2013c, December). Performance of cloud centers with high degree of virtualization under batch task arrivals. IEEE Transactions on Parallel and Distributed Systems, 24(12), 2429-2438.

Khazaei, H., Mišić, J., Mišić, V. B., & Mohammadi, N. B. (2012b, December). Availability analysis of cloud computing centers. In Global Communications Conference (GLOBECOM), 2012 IEEE (pp. 1957-1962). IEEE.

Khazaei, H., Mišić, J., Mišić, V. B., & Rashwand, S. (2013a, May). Analysis of a pool management scheme for cloud computing centers. IEEE Transactions on parallel and distributed systems, 24(5), 849-861.

Khomonenko, A. D., & Gindin, S. I. (2014, October). Stochastic models for cloud computing performance evaluation. In Proceedings of the 10th Central and Eastern European Software Engineering Conference in Russia (p. 20). ACM.

Little, J. D., & Graves, S. C. (2008). Little's law. In Building intuition (pp. 81-100). Springer US.

Liu, M., Dou, W., Yu, S., & Zhang, Z. (2015). A decentralized cloud firewall framework with resources provisioning cost optimization. IEEE Transactions on Parallel and Distributed Systems, 26(3), 621-631.

Liu, X., Li, S., & Tong, W. (2015). A queuing model considering resources sharing for cloud service performance. The Journal of Supercomputing,71(11), 4042-4055.

Liu, X., Tong, W., Zhi, X., Zhiren, F., & Wenzhao, L. (2014). Performance analysis of cloud computing services considering resources sharing among virtual machines. The Journal of Supercomputing, 69(1), 357-374.

Liu, X., Zha, Y., Yin, Q., Peng, Y., & Qin, L. (2015). Scheduling parallel jobs with tentative runs and consolidation in the cloud. Journal of Systems and Software, 104, 141-151.

Rimal, B. P., Jukan, A., Katsaros, D., & Goeleven, Y. (2011). Architectural requirements for cloud computing systems: an enterprise cloud approach. Journal of Grid Computing, 9(1), 3-26.

Sousa, E., Lins, F., Tavares, E., Cunha, P., & Maciel, P. (2015). A modeling approach for cloud infrastructure planning considering dependability and cost requirements. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 45(4), 549-558.

Tian, Y., Lin, C., Chen, Z., Wan, J., & Peng, X. (2013). Performance evaluation and dynamic optimization of speed scaling on web servers in cloud computing. Tsinghua Science and Technology, 18(3), 298-307.

Trivedi, K. S., & Sahner, R. (2009). SHARPE at the age of twenty two. ACM SIGMETRICS Performance Evaluation Review, 36(4), 52-57.

Vilaplana, J., Solsona, F., Teixidó, I., Mateo, J., Abella, F., & Rius, J. (2014). A queuing theory model for cloud computing. The Journal of Supercomputing, 69(1), 492-507.

VMware vLockstep, VMware Inc. vSphere ESX and ESXi Info Center. http://www.vmware. com/products/esxi-and-esx/overview.html, July 2017.

Wang, L., Ranjan, R., Chen, J., & Benatallah, B. (Eds.). (2011). Cloud computing: methodology, systems, and applications. CRC Press.

Xia, Y., Zhou, M., Luo, X., Pang, S., & Zhu, Q. (2015a, January). A stochastic approach to analysis of energy-aware DVS-enabled cloud datacenters. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 45(1), 73-83.

Xia, Y., Zhou, M., Luo, X., Pang, S., & Zhu, Q. (2015b, April). Stochastic modeling and performance analysis of migration-enabled and error-prone clouds. IEEE Transactions on Industrial Informatics, 11(2), 495-504.

Yang, B., Tan, F., & Dai, Y. S. (2013). Performance evaluation of cloud service considering fault recovery. The Journal of Supercomputing, 65(1), 426-444.

Zhang, S., Qian, Z., Luo, Z., Wu, J., & Lu, S. (2016). Burstiness-Aware Resource Reservation for Server Consolidation in Computing Clouds. IEEE Transactions on Parallel and Distributed Systems, 27(4), 964-977.