

High System Availability, Reliability, and Security in Cloud Environment

Jonathan Charity Talwana¹Gu Chun Hua¹Ulysse Rugwiro²

1. Department of Computer Science and Information Engineering.

2. East China University of Science and Technology - Shanghai City, China.

Abstract --*In IT industry, service provisioning is an emerging domain; in this context, cloud provisions must always be acquirable and procurable 24x7 over the internet with easy and secure access to cloud users. During accessing the cloud services, the uninterrupted availability is a significant concern for all service brokers and some of the services like recovery point, scheduling, load balancing, replication etc, are also applied to utilize performance. Thus, besides these services, there is prospect to research middleware and infrastructure or arbitrator solutions. The primary purposes of this research presented in this paper is to develop a model-based algorithm and a palatable system approach which provides high availability for cloud resources and services. The proposed availability-based model is calculated based on cloud resource uptime and downtime. The availability of cloud is based on the actual running time of datacenter and if any fault in a server is identified, load need to migrate from faulty server to another server immediately. During this migration process, unavailability that may transpire in a server considered as downtime of that server. The goal of using Weibull distribution in this research simulation is for reliability calculation and data analysis due to its versatile accuracy. The Exponential distribution purposely applied to model the behavior of units that possess constant failure.*

Keywords: Cloud Computing, High Availability, Fault Tolerance, Accessibility, Reliability.

I. INTRODUCTION

Cloud computing is a modern computing technology espoused from parallel, grid and distributed computing resources. The availability of a system is based on the actual running time of datacenter and if any fault is identified in a server, there is an obligation to immediately migrate a load of the faulty server to another server. During the migration process, unavailability that may transpire in a server considered as downtime of that server. However, the cloud computing environment offers low-cost computing resources as to subscription-based services.

These resources are elastically scalable and dynamically provisioned, the benefits are immense. New billed models have been introduced by cloud providers which allows clients to provide resources to use them anciently with a sign that is costly reductions. This computing paradigm is still vulnerable to a significant number of system failures and consequently, there is an increasing concern among clients regarding the resource reliability and availability of cloud computing services.

The Weibull distribution used in this research is to design various prediction of reliability and availability of system models. Weibull distribution (or 3-parameter Weibull) contains an additional parameter and it has the probability density function. The exponential distribution also the probability distribution technique represents the time between events in a poisson point process. Poisson point process is the process of occurrence of events which is unceasingly and independently at an equal average time.

NIST has defined cloud computing as a computing model for on-request network access to IT resources like network, applications, storage and servers which are placed at a shared pool. These resources can be readily accessible when requisite and released with least possible efforts^[1]. The best part of cloud computing is to work with all allied hardware and software malleable and continuously. Cloud computing tender's scalability pay-as-you-go, on-demand services, flexibility and rapid elasticity model^[2].

Virtualization is the mainpayback of cloud computing by which cloud users can create multiple instances of Virtual Machines (VM) and can accomplish various operating system requests with VMs without disturbing each other. Although all VMs managed on the identical hardware, they are entirely insulated and endeavor to boost deployment at its finest^[3].

At the present time, cloud computing is also essential for one of the preferred excitable subdivision, i.e., Mobile cloud computing. The driving objective of cloud and mobile cloud computing is to experience uninterrupted storage and handling capability from cloud data centers^[4].

In cloud computing, there are three primary service classification models for resistance (PaaS) (SaaS), and (IaaS) ^[5]. Figure1 highlights cloud service stack architecture. IaaS provides upfront assistance of core hardware like Memory, Storage, CPU and Network and all resources essential to accomplish activities.

Assistance of VMs, servers, load balancer, network to cloud users are available through IaaS. PaaS delivers accomplishment runtime, web server, and development tool to generate, deploy, and accomplish applications. The SaaS layer alliances with cloud users to tender CRM, Email, games, or any application assistance to cloud users who transmit demands to SaaS from web browsers, mobile applications, thin client, or terminal emulators.

The remainder of the paper contents are organized as follows; Section II focuses on related study of system reliability and availability for cloud computing and research gap for the proposed model. Section III describes the system model with working architecture. Section IV explains the Experimental setup with FT Cloud Sim for the proposed system model. Section V describes the analysis of simulation results for Exponential distribution and Weibull distribution and section VI discusses about conclusion of proposed model of this paper.

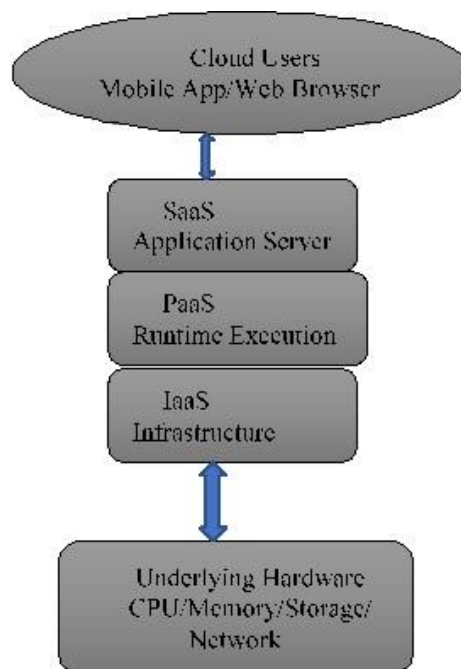


Fig.1 Cloud Service Stack Architecture

II. RELATED STUDY

Garg R. and Singh A.K. have recommended fault tolerant task scheduling model ^[6] for dependent and independent jobs using Weibull distribution technique rather than Poisson distribution technique for failure identification. They have innovated a notion of the checkpoint, if a failure ensues in any resource than by contemplating assessment, the system can return to a reliable point and henceforth recovery perceived for the improved system reliability.

Poola D, Ramapo hanarao K., and Buyya R. have recommended workflow deadline-based scheduling approach ^[7] to accomplish the job using spot and on-demand instance. This technique never permits ceasing execution prior to job accomplishment and concurrently assistances to slash cost. It also ensure to optimize functioning and cost reduces by 70%.

Kanso A and Lemieux Y have recommended High-Availability (HA) system ^[8] which offer services to deliver network access 99.999%. This technique is also known as five 9's system technique.

Toeroe M, Pawar N., and Khendek F. have realized a system to administer application-level availability and elasticity ^[9]. The system attains the maximum availability and accessibility when the outage rate is 5.25 minutes per year or smaller.

Frîncu M.E has recommended and realized a scheduling approach ^[10], which delivers a failure-prone system. Supplementary reduces restoration time in case if any issue occurs.

Sripanidkulchai K, Sahu S, Ruan Y, Shaikh, and Dorai C. have recommended High-availability system ^[11].They calculated availability based on downtime. The downtime was contemplated for the system unattain ability for several events more than 5 minutes over a year.

Araujo.J,Matos.R,Alves V, Maciel P,Souza F,Trivedi KS, et al . have instigated the software aging model for eucalyptus cloud ^[12].Using it, they establish the system accessibility and availability based on the offered quantum of time to accomplish cloud serves or jobs.

III. SYSTEM MODEL

System availability and reliability are key constrictions in cloud computing. If the cloud system is unreliable, then susceptibility will be extended irrefutably to cloud resource and break the security of cloud data center. This results untrusted and unsecured system where malware encroaches and generates security gaps and counterfeit undesirable influence into cloud users' astuteness.

Fig 2 illustrates the underlying cloud system structural design ^[13] with core units. In cloud architecture, the data center is the most significant unit, which generally contemplates solitary or complex hosts. On every host, a few instances of VM set up depending on application load. At another end of architecture, Cloudlets (user applications) are inserted from the top of the stack. Then, plotting of all the cloudlets with partnered VMs accomplished by middleware i.e. datacenter broker.

The problem with this approach even if you got suitable components, you could not get ultra-high reliability from a probability point of view for your system to work you have to get lucky, and all the three elements have to work. Even suitable components are not good enough in mathematical form;

$$R_{(t)serial} = R_{(t)1}R_{(t)2}R_{(t)3} = \prod_i R_{(t)i}$$

In this illustration, the presented three components which arranged in serials all the parts of 1,2,3 must work for better reliability.

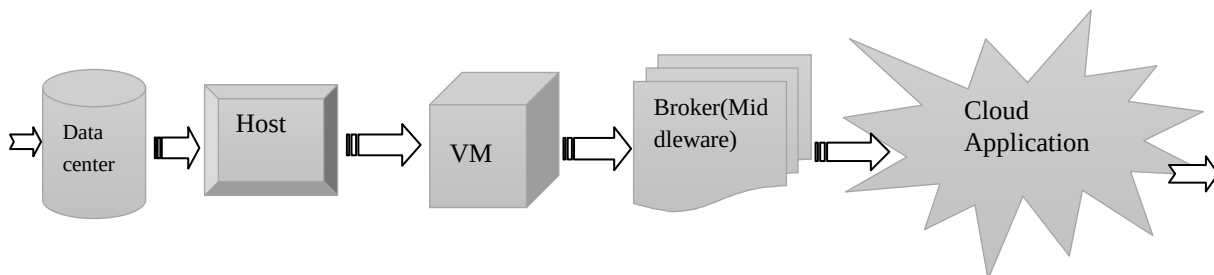


Fig.2 Cloud System Architecture

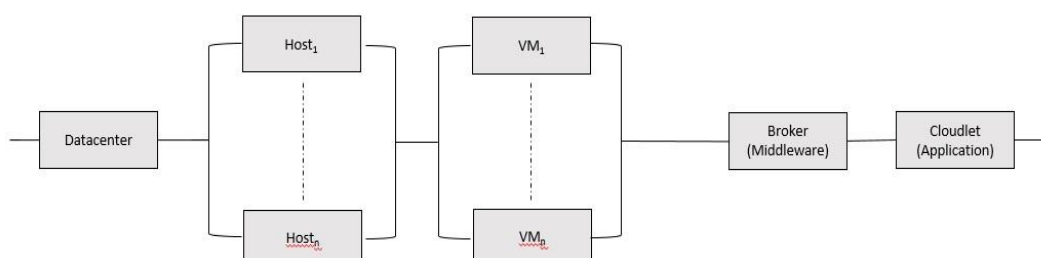


Fig.3Multiple-Host, Multiple-VM Cloud System

As illustrated in figure 3 illuminates a cloud system which can be available with multiple-Hosts and multiple-VMs. A layer exists between cloudlets and VMs known as broker. Broker apprehends the cloudlets from end users of the applications. Then, depending on several cloudlets and their necessities, broker demands to construct the required number of VMs and similarly based on the necessitated number of VMs, the number of Hosts essential to generate. Consequently, the architecture may incorporate several hosts as well as numerous VMs to accomplish the demanded cloudlets, and thereby redundancy increases the system availability.

A. SYSTEM AVAILABILITY AND RELIABILITY

System availability and reliability are characterized by the accuracy and security of the system, correspondingly. System availability can be specified as the required time the system is available or obtainable to end users to accomplish and manage jobs, while the reliability measured as the security and efficacy of the system^[14]. System availability for cloud computing is defined by equation(1)

$$A_{sys} = A_{datacenter} * A_{host} * A_{VM} * A_{broker} * A_{cloudlet} \quad (1)$$

Down host List and Up host List are host lists which include failed hosts and running hosts correspondingly. The A_{host} , A_{VM} , and $A_{cloudlet}$ are the Availability of the components which is calculated as per the following equation (2):

$$A = MTBF / (MTBF + MTTR) \quad (2)$$

Where,

MTTR = Mean Time To Repair.

MTBF = Mean Time Between Failure.

Furthermore, Mean Time Between Repair (MTTR) and Mean Time Between Failure (MTBR) can be calculated using equations (3) and (4) respectively:

$$MTTR = \text{Total Downtime} / \text{number of failure} \quad (3)$$

$$MTBF = \text{Total Uptime} / \text{number of failure} \quad (4)$$

Additionally, the proposed equation is to compute system availability to describe failure prone and extremely consistent cloud system with the help of downtime and uptime parameters. Downtime is the time period in which the system does not appear available for usage due to some reason and uptime is the time period in which the system is available or available to usage^[15].

$$\text{Availability} = \text{Uptime} / (\text{Uptime} + \text{Downtime}) \quad (3)$$

The availability of distinct host, VM, and cloudlet can be computed using the following equation (6):

$$\frac{MTBF_{host\ 1}}{MTBF_{host\ 1} + MTTR_{host\ 1}} \quad (4)$$

In the similar fashion, the availability of hosts in total A_{hosts} , VMs in total A_{VMS} , and $A_{Cloudlets}$ can be computed using the following equation (7):

$$A_{hosts} = \sum A_{hosti} / \text{Number of host} \quad (5)$$

where, $A_{host\ 1} = \text{Availi}$

In support of cloud system availability, the assessment phases were performed using Exponential and Weibull distribution for existing (ordinary) system model and suggested system model.

B. SYSTEM MODELS

For the efficiency, this paper introduces two different models for our system availability and accessibility; Exponential distribution and Weibull distribution, for probability theory.

C. WEIBULL DISTRIBUTION

Weibull distribution is an incessant distribution technique for probability theory. Based on various values of parameters, the Weibull distribution is used to design many predictions of reliability and availability of system models. Weibull delivery is obtained again by the following formula; with assuming $\gamma=0$ and assuming $\beta=C=Constant$ assumed value or: (8)

$$f(t) = \frac{c}{\eta} \left(\frac{t}{\eta}\right)^{c-1} e^{-\left(\frac{t}{\eta}\right)^c} \quad \text{where the only unknown parameter is the scale parameter, } \eta$$

Through Weibull distribution, the probability can be driven, thereby from that system reliability and the performance measured effectively

$$F(t) = (\gamma/\alpha) * (x/\alpha)^{\gamma-1} * \exp(-(x/\alpha)^\gamma) \quad (9)$$

; where $x \geq 0$

$$(1 - \prod_{Host} (1 - f(x)) * (1 - \sum_{host}(x))) * (1 - 1/\text{totalhost} * (\sum_{vm} f(x))) * (1 - (1/\text{totalhost}) * (\text{totalvm}) * (\sum_{cloudlet} f(x)))$$

D. EXPONENTIAL DISTRIBUTION

The exponential distribution is sometimes known as a contrary exponential distribution. The exponential distribution also referred to as the probability distribution technique that represents the time between actions in a Poisson point method, while the Poisson point process summarized as the process of occurrence of events which is continuously and independently at an equal average time.

$$f(x; \lambda) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases} \quad \text{Alternatively, this can define by } H(x) \text{ where } H(0) = 1$$

$$f(x; \lambda) = \lambda e^{-\lambda x} H(x) \quad (10)$$

Where $\lambda > 0$ is the parameter of the distribution or call rate parameter which is supported on the interval $(0, \infty)$ in that if random variable x has distribution, was written as $x \sim \text{Exp}(\lambda)$ since either the notation in the previous (using λ) is intended.

$$A_{sys} = A_{datacenter} * A_{host} * A_{VM} * A_{broker} * A_{cloudlet} \quad (11)$$

$$F(t) = 1 - e^{-\lambda t} = (1 - \prod_{Host} (1 - e^{-\lambda x}) * (1 - \sum_{host} (1 - e^{-\lambda x})) * (1 - 1/\text{totalhost} * (\sum_{vm} 1 - e^{-\lambda x})) * (1 - (1/\text{totalhost}) * (\text{totalvm}) * \sum_{cloudlet} (1 - e^{-\lambda x})))$$

The purpose of Exponential distribution allows for easy comparison between numbers for the scientific notation that would otherwise require counting zeros.

E. ALGORITHM

Input: number of Host, number of VM, number of Cloudlets, failure rate distribution

Output: Reliability of System

Step 1: Create host and Vm

Step 2: For each host repeat step 3 to 6

Step 3: As per failure distribution event generator generate host fail event for host h

Step 4: Migrate all running Vm from host h to other running host and add Vm migration time of VMi in totaldowntime_of VMi

Step 5: Remove host h from uphostList and add it in downhostList

Step 6: Event generator generates host repair event for host h
Remove host h from DownhostList and add host h in UphostList (running host) and calculate down time of host and add down time of host in totaldowntime_host.

Step 7: If any cloudlet failed due to any reason then calculate cloudlet downtime for cloudlet

Step 8: Calculate MTBR, MTBF

Step 9: Calculate A_{host} , A_{VM} and A_{cloudlet}

Step 10: Calculate system availability = $(A_{\text{host}} + A_{\text{VM}} + A_{\text{cloudlet}}) / 3$

IV. EXPERIMENTAL SETUP

The proposed experiment set up to execute system model in Weibull Distribution and Exponential Distribution; in this analysis, simulated on platform FT Cloud Sim were FT Cloud Sim offers an extendible associate to execute reliability improvement technique as a novel cloud service^[16]. For experimental setup, this paper simulation has included a data center with the several hosts using the 16-port fat-tree data center network and the matching number of VMs. Concurrently Linux operating system with x86 Architecture and for VMM (Hypervisor) Xen have been utilized.

V. RESULT ANALYSIS

For experimental simulations, worked on many metrics like; host downtime, host uptime, VM downtime, VM uptime, Cloud-let downtime and Cloud-let uptime. System Availability is computed by using two, and Host failure is derived based on Weibull Distribution technique and Exponential Distribution technique. Figure 4 and Figure 5 illustrate the comparison for system availability based on the proposed model and existing model for Weibull Distribution and Exponential, correspondingly.

For the System Evaluation, different quantities of hosts have been contemplated to comprehend the system and its reliability. It has also presumed 2 VMs on every Host to prepare outcome evaluation flawless.

Fig 5 compares the System availability quantities in terms of percentage for existing and suggested model, which clearly highlights the increased system availability and therefore reliability using proposed model.

Comparison between Fig 4 and Fig 5 exhibits a slight difference between gathered outcomes for System Availability in Weibull and Exponential Distribution.

By considering 1024 hosts, the System Availability attained is 0.998919160596718 and 0.999999126078982% for Existing Model and suggested model correspondingly for Weibull distribution method.

While using the Exponential distribution method, with the same number 1024 hosts, the System Availability is 0.998919168344889 and 0.999999126101682 for Existing Model and Proposed Model correspondingly. However, System Availability is to some extent superior in the Exponential distribution method. By progressively increasing the number of hosts and again gather outcomes for System Availability for both Exponential and Weibull Distribution method, accordingly, deliberating 16000 hosts, they have to receive System Availability of 0.999840672471314, and 0.999999983191246 correspondingly for Existing Model and Suggested Model for Weibull distribution.

At the same time, the System Availability for Exponential Distribution method is 0.999840594383115 and 0.999999983176605 for Existing and Suggested Model, correspondingly. Therefore, the System Reliability is greater in Weibull Distribution in contrast to Exponential Distribution method. Therefore, it is inferred that System Reliability influenced by a particular situation for Weibull and Exponential Distribution method.



Fig.4 System Availability Comparison for Weibull Distribution



Fig.5 System Availability Comparison for Exponential Distribution

In both, Exponential and Weibull Distribution methods, the suggested model performs well in comparison to the existing model. Significantly, in the suggested system model, this paper has considered availability depending on VM, cloud-let uptime and downtime and Hosts. Thus, when host crashes, at the same time participating VMs and cloudlets on that failed host will migrate and persist its accomplishment on other host which has excess resources. The existing system does not support this process.

Fig 6 and Fig 7 signify system availability as in percentage for the proposed and existing model correspondingly for Weibull Distribution. As illustrated in graphs, as the number of hosts enhances it proportionately influence the system availability. By observing results very carefully, it implies that the system available for the suggested model is comparatively improved with respect to the existing model. Supplementary the graphical curvature, engendered from the simulation outcomes, emphasizes the progressively worthy system availability for the suggested model.



Fig.6System Availability as per Proposed Model in Weibull Distribution

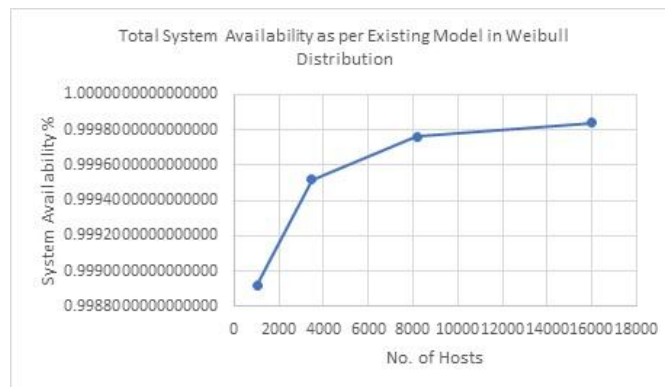


Fig.7System Availability as per Existing Model in Weibull Distribution

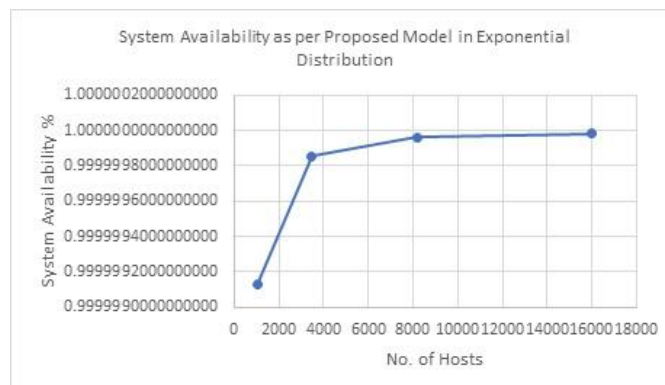


Fig.8System Availability as per Proposed Model in Exponential Distribution



Fig.9 System Availability as per Existing Model in Exponential Distribution

Fig 8 and Fig 9 shows the high system availability for suggested and Existing model correspondingly for Exponential Distribution. Contemplating simulation outcomes and graphs signifies that escalating the number of hosts and VMs will upsurge system availability for both models in Exponential Distribution. Nonetheless, as a broad study, evaluating graphical outcomes of both models, emphasizes the improved system reliability for the suggested model with respect to the existing model in Exponential Distribution.

VI. CONCLUSION

High system availability in cloud computing illustrates the consistency and fault tolerance in a system. Although, uninterrupted accessibility of a system is an important issue, it is possible to conclude high availability, can oblige the best result and assistances to cloud users. Comprehensive experiments and evaluations of existing approach and proposed approach signify noticeable inconsistency in system accessibility for the cloud environment. The proposed model in the paper tender high availability to the system, which ascertains the optimal reliability and availability for cloud users. Furthermore, the proposed approach also reduces system failure or system downtime, which ensures a failure-prone system. Therefore, the proposed system approach appeared much more adaptive and consistent in comparison to existing methods. The extension work of this research is still under process, and those work has not been elaborated/mentioned in this paper.

ACKNOWLEDGMENT

I appreciate my tutor Prof. Gu Chun Hua for the continuous support of my Ph.D. study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in my endeavors during the research and writing of this article.

REFERENCES

- [1] Mell P, Grance T: The NIST Definition of Cloud Computing. NIST Special Publication 800-145, U.S. Department of Commerce.
- [2] Lecznar M., Patig S. (2011) Cloud Computing Providers: Characteristics and Recommendations. In: Babin G., Stanoevska-Slabeva K., Kropf P. (eds) E-Technologies: Transformation in a Connected World. MCETECH 2011. Lecture Notes in Business Information Processing, vol 78. Springer, Berlin, Heidelberg
- [3] Xing Y., Zhan Y. (2012) Virtualization and Cloud Computing. In: Zhang Y. (eds) Future Wireless Networks and Information Systems. Lecture Notes in Electrical Engineering, vol 143. Springer, Berlin, Heidelberg
- [4] Bheda H.A., Thaker C.S. (2016) Scheduling for Distributed Applications in Mobile Cloud Computing. In: Satapathy S., Joshi A., Modi N., Pathak N. (eds) Proceedings of International Conference on ICT for Sustainable Development. Advances in Intelligent Systems and Computing, vol 409. Springer, Singapore
- [5] Chun, SH. & Choi, BS. Cluster Comput (2014) 17: 529. <https://doi.org/10.1007/s10586-013-0296-1>
- [6] Garg, R. & Singh, A.K. Arab J Sci Eng (2014) 39: 8775. <https://doi.org/10.1007/s13369-014-1455-2>
- [7] Poola D, Ramamohanarao K, Buyya R. (2014) Volume 29, Pages 523-533. <https://doi.org/10.1016/j.procs.2014.05.047>
- [8] Kanso A, Lemieux Y (2013) Achieving high availability at the application level in the cloud In: Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference On, 778–785. IEEE.
- [9] Toeroe M, Pawar N, Khendek F (2014) Managing application level elasticity and availability In Network and Service Management (CNSM), 2014 10th International Conference On, 348–351. IEEE.
- [10] Frîncu ME (2014) Scheduling highly available applications on cloud environments. Future Generation Comput Syst 32: 138–153.
- [11] Sripanidkulchai K, Sahu S, Ruan Y, Shaikh A, Dorai C (2010) Are clouds ready for large distributed applications? ACM SIGOPS Oper Syst Rev 44(2): 18–23.

- [12] Araujo J, Matos R, Alves V, Maciel P, Souza F, Trivedi KS, et al. (2014) Software aging in the eucalyptus-cloud computing infrastructure: characterization and rejuvenation. *ACM J Emerg Technol Comput Syst (JETC)* 10(1): 11.
- [13] Calheiros R, Ranjan R, Beloglazov A, Rose C, Buyya R using CloudSim Softw. *Pract. Exper.* 2011; 41:23–50. DOI: 10.1002/spe.995
- [14] Anderson J, Abhari R, et al. (2012) Reliability and Availability of Cloud Computing
- [15] Endo P, Rodrigues M, Gonçalves G, et al. (2016) *Journal of Cloud Computing Advances, Systems, and Applications* 2016:16. <https://doi.org/10.1186/s13677-016-0066-8>
- [16] Zhou A, Wang S, Sun Q, Zou H, Yang F, FTCloudSim: A Simulation Tool for Cloud Service Reliability Enhancement Mechanisms (2013), *ACM. Middleware 2013 Posters and Demos Track*, December 9–13, Beijing, China. Optimization.
- [17] Cloud computing, and grid computing 360-degree compared. In *Grid Computing Environments Workshop*, 2008. GCE'08 (pp. 1-10). Ieee. Foster, I., Zhao, Y., Raicu, I., & Lu, S.
- [18] Plummer, D. C., Cearley, D. W., & Smith, D. M. (2008). Cloud computing confusion leads to opportunity. *Gartner Report*.
- [19] Gong, C., Liu, J., Zhang, Q., Chen, H., & Gong, Z. (2010, September). The characteristics of cloud computing. In *Parallel Processing Workshops (ICPPW)*, 2010 39th International Conference on (pp. 275-279). IEEE.
- [20] Furht, B. (2010). Cloud computing fundamentals. In *Handbook of cloud computing* (pp. 3-19). Springer US.
- [21] Kaushal, V., & Bala, A. (2011). Autonomic fault tolerance using haproxy in a cloud environment. *Int. J. of Advanced Engineering Sciences and Technologies*, 7(2), 54-59.
- [22] Patra, P. K., Singh, H., & Singh, G. (2013). Fault Tolerance Techniques and Comparative Implementation in Cloud Computing. *International Journal of Computer Applications*, 64(14).
- [23] Bala, A., & Chana, I. (2012). Fault Tolerance Challenges, Techniques, and Implementation in Cloud Computing. *International Journal of Computer Science Issues (IJCSI)*, 9(1).
- [24] Tchana, A., Broto, L., & Hagimont, D. (2012, March). Fault Tolerant Approaches in Cloud Computing Infrastructures. In *ICAS 2012, The Eighth International Conference on Autonomic and Autonomous Systems* (pp. 42-48).
- [25] Hayashibara, N., Defago, X., Yared, R., & Katayama, T. (2004, October). The ϕ accrual failure detector. In *Reliable Distributed Systems*, 2004. Proceedings of the 23rd IEEE International Symposium on (pp. 66- 78). IEEE.
- [26] Gupta, I., Chandra, T. D., & Goldszmidt, G. S. (2001, August). On scalable and efficient distributed failure detectors. In *Proceedings of the twentieth annual ACM symposium on Principles of distributed computing* (pp. 170-179). ACM
- [27] Maier, G., Sommer, R., Dreger, H., Feldmann, A., Paxson, V., & Schneider, F. (2008, August). Enriching network security analysis with time travel. In *ACM SIGCOMM Computer Communication Review* (Vol. 38, No. 4, pp. 183-194). ACM.
- [28] Anjali. D. Meshram, A. S. Sambare, S. D. Zade, Fault Tolerance Model in Cloud Computing, *IJAIS Proceedings on 2nd National Conference on Innovative Paradigms in Engineering and Technology (NCIPET 2013)*, 2013.
- [29] Sagar C. Joshi, Krishna M. Sivalingam, fault tolerance mechanisms for virtual data center architectures, *K.M. Photon Netw Commun* (2014) 28: 154, 2014



Jonathan Charity Talwana is a Ph.D. candidate in Computer Applied Technology at East China University of Science and Technology. I received my MSc degree from ECUST, China.



Dr. Ulysse Rugwiro is a Ph.D. Computer Applied Technology at East China University of Science and Technology. He received his MSc degree at ECUST, China. He's working as a researcher in the field of Green Cloud Computing.



Gu Chun Hua is a Professor and Ph.D. supervisor in the School of Information Science and Engineering, East China University of Science and Technology. A senior member of the China Computer Federation. His main research interests include cloud computing and the internet of things.