Volume 65, Issue 5, 2021
Journal of Scientific Research
of
The Banaras Hindu University

Reliability Analysis of Cloud Computing Systems Serving Multi-Class Requests

Divya Srivastava[1], Rohit Sharma[2] and Anuj Srivastava[3]

[*1]Department of Computer Science and Engineering, Indian Institute of Technology Jodhpur srivastavadivya.90@gmail.com
[2]Department of Computer Engineering and Engineering, Dr. Ambedkar Institute of Technology for Handicapped, Kanpur rohit6615@rediffmail.com
[3]Department of Mathematics, Dr. Ambedkar Institute of Technology for Handicapped, Kanpur anuj_srivas1984@rediffmail.com

Abstract: **The current era of the internet has laid focus on Cloud Computing systems. We find the application of CCS in almost all the domains, such as the medical field, pharmacy sector, IT sector, etc. However, the reliability of CCS is of utmost importance for its smooth applicability. In this paper, we perform the reliability analysis of CCs using the ReliaCloud-NS simulation framework. We have carefully framed our test cases to cover the broad domain where CCS is used. Along with reliability analysis, the study on components of CCS, i.e., HDD, CPU, Bandwidth and memory, is performed, illustrating its failure in various VMs. We have tried our best to perform the analysis of CCs broadly to make this work valuable for almost all the domains where CCS is used.**

Index Terms: **Cloud Computing System, Reliability analysis, ReliaCloud-NS, Application of CCS, AWS.** Introduction

I. INTRODUCTION

The adoption of cloud technologies enables enterprises, big and small, to be agile, innovative, and competitive, and create differentiated customer experiences. The cloud adoption rate is growing faster than predicted. A report from Gartner says that the expected growth in the cloud market will be from $182.4 billion in 2018 to $331.2 billion in 2022, a growth rate of approximately 12.6% [1]. The question organizations are asking is what strategy they should adopt to move to the cloud.

Cloud computing reliability has gained popularity owing to rapid growth in demand for cloud computing services. As mentioned in [2], the cloud forms the heart of almost all internet-based activities; for example, it is used in social networking sites such as Facebook, Twitter, Snapchat, etc. It has found its application in almost every research sector, life sciences [3], natural sciences [4] or applied science. This growth in demand for cloud computing and the high demand on cloud services necessitates the requirement for quantitative ways for cloud services evaluation from a reliability standpoint efficiently.

Non-sequential Monte Carlo Simulation (MCS) is used to calculate the reliability of a CCS due to its high computational efficiency. The general steps followed by the non-sequential MCS algorithm are sampling, classification, calculation, and convergence. A non-sequential MCS-based simulation software ReliaCloud-NS has been used to performs reliability evaluations of carefully designed CCSs. The feature of ReliaCloud-NS to create and simulate complex CCS's is used to develop compute, memory, and storage-intensive clusters along with their suitable VM instances. We have also presented the simulation results, showing that ReliaCloud-NS can provide user-friendly charts and graphs that characterize CCS reliability.

In section 2 the literature review on CCS reliability with an outline of popular cloud simulation software and tools is given. Section 3 introduces and provides us with the idea of MCS-based Cloud reliability software ReliaCloud-NS and the description of VM types and clusters used for simulation. In section 4, the execution stage reliability of the system is obtained and analyzed. Numerical results depicting enhanced system reliability and comparison of results has been presented in section 5. Significant conclusions of the proposed work, along with the future scope is given in section 6.

II. RELATED WORK

CCS's plethora of works is done to model user requests and data transmission issues. However, one critical aspect of system failure in a CCS is hardware failure which is less addressed. The focus on failing hardware in CCS's begins in [5], where examination of hardware failures occurring in multiple data centers are thoroughly verified to identify the explicit rates of failure for various components, i.e., disks, CPUs, memory, and RAID controllers. The main conclusion of their work is that the largest source of failure in almost all data centers is disk

DOI: 10.37398/JSR.2021.650533

failure—the evaluation of intermittent hardware failures is countered in [6]. The review of failures occurring in DRAM, CPU, and disks in customer PC's continue in [7]. The most important outcome of this work is for recurring faults, as it suggests that an already failed PC component is likely to fail shortly soon. This paper also examines hidden failures such as DRAM's 1-bit failures[8]. A comprehensive failure evaluations at all levels of the CCS and systems overall reliability is obtained in [9]. To prevent failure of cloud computing system, authors in [10] proposed a service model for finite population clouds. It analyze the resource request stage of finite population cloud computing system. Then the losses incurred in the system due to impatient users are accounted from which the scheduling strategy is prepared to load balance the request stage of model.

The issue of network failures in Cloud Data Centers is addressed in [11,12]. These studies conclude to the fact that cloud data center networks and switches have high reliability, and faults in their load balancers are due to software failures. Network failure causes only minor faults, and redundancy in cloud resources is not always a positive thing. The study of resource utilization and general workloads in data centers and the study of various hardware components in it, including disks, CPU's and memory, was carried out in [13]. The facility of modeling and simulation of application schedulers for the multiple administrative domain distributed systems is provided by GridSim [14]. Their work is extended in [15] an which incorporates data grids in the simulations. Similarly, the simulation environment for evaluation of cluster, grid, and peer-to-peer heuristics and algorithms for distributed computing systems is SimGrid [16,17]. They also provide multiple user interfaces and APIs to assist researchers and developers in simulating scheduling heuristics and algorithms or even developing new applications for distributed computing systems. CloudSim presented in [18] forms an extensible toolkit for modelling cloud environments and performs evaluation of existing and new resource provisioning algorithms. Various other software packages are extensions of CloudSim, including CloudAnalyst, Cloudbus Toolkit, and NetworkCloudSim[19]. CloudAnalyst is a reliable tool for simulating hugely levelled cloud environments and presents insights on distributed applications and services among cloud infrastructure to optimize the application performance including providers using service brokers [20]. Cloudbus Toolkit is a group of softwares forming a system for enhancing the functionality of market-oriented resource management which involves support for federated clouds and energy-aware resource allocation to create green clouds [21]. NetworkCloudSim proposed in [22] is a tool for modeling the advanced application models, which inclues highly parallel applications utilizing message passing or scalable network models. EMUSIM [23] is an open-source integrated environment for emulation and simulation to model, evaluate, and validate cloud computing applications' performance. iCanCloud is a GUI-based open-source simulator used for modeling and simulating CCSs from a cost and performance perspective for a fixed set of applications running on specific hardware [24]. MAScloud is multiple agents and iCanCloud [25] based framework for optimizing the cost and performance of Cloud computing systems. GreenCloud is a simulation environment for energy-aware cloud computing data centers. It captures details for various components such as servers, switches, and links, realistic packet-level communications, and enables assessing power management techniques such as frequency and voltage scaling and the dynamic shutdown of network components [27]. [5] proposes a new, scalable algorithm based on non-sequential Monte Carlo Simulation (MCS) to evaluate the reliability of large-scale cloud computing systems (CCS) and develops appropriate performance measures. It also proposes a new iterative algorithm that leverages the MCS method to design highly reliable and utilized CCSs .The implementation, architecture, and use of ReliaCloud-NS allow users to evaluate a cloud computing system (CCS) and design a CCS to a pre-defined reliability level for both public and private clouds [28].

III. METHODOLOGY

In this work, we performed a detailed analysis for reliability of CCS using ReliaCloud-NS. We have categorized the user request for CCS into three categories of virtual machines. The virtual machines under these categories are then specified, and their reliability analysis is performed in the robust scenario.

A. ReliaCloud-NS

ReliaCloud-NS, proposed by Snyder et al. in [5] is a GUI-based web application for executing non-sequential Monte Carlo simulations (MCSs) to evaluate the CCS reliability. Along with assessing reliability, it enables the user to design a CCS for both public and private cloud to a specified reliability level. Their software comprises various types of CCS components, virtual machine allocation systems and simulation schemes. The basic functionality of ReliaCloud-NS includes designing individual components (like VM's, VM groups, clusters etc.), creating CCS, simulating CCSs for estimating reliability and result analysis obtained after simulation.

It is built over MVC architecture, enabling the non-sequential MCS algorithm to work in parallel mode using many cores. Each parallel trial can execute in its own thread and be simulated simultaneously. Non-sequential MCS in ReliaCloud-NS facilitates a flexible and efficient way to compute CCS reliability using a set of four different resources, i.e., CPU, memory, Bandwidth and Hard Disk Drive (HDD). These basic resources are combined to form Virtual Machines (VMs) and clusters. A VM is a fundamental reusable component of ReliaCloud-NS for requesting resources, while a cluster is a basic reusable component for allocating available resources.

ReliaCloud-NS components can be categorized in an hierarchial manner. The basic components are CPU, memory, Bandwidth and HDD. The components form VM or cluster. The

collection of individual VMs forms VM groups while the collection of clusters forms the cloud. VM groups can be considered as a simulation-facing resource-requesting component.

B. Proposed User-driven Simulation System

We proposed our simulation system to meet the demand of a real-life scenario, where the user wants access to the cloud for a specific type of resource. In real-scenario, among the four basic components, viz., CPU, memory, Bandwidth and HDD, user requirement focuses on one component as per its task. The task is either, computing or storage-intensive or memory-based. For example, cloud is used for saving user data, hence storage-intensive task.

Considering the above-mentioned real scenario, we frame our clusters and VMs accordingly to meet such requirements. Figure 1 shows our division of framework of clusters based on the incoming request.

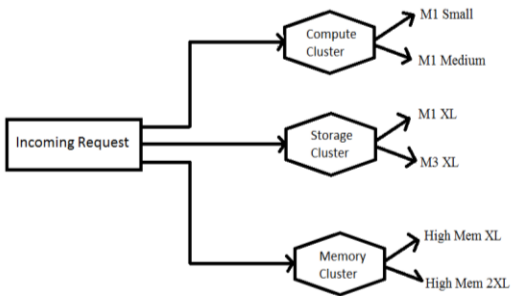

Fig. 1. Framework of clusters based on the incoming request.

We divide user requests into three clusters of resources: compute cluster, storage cluster, and memory cluster. Here, as their names suggest, compute cluster emphasizes computation capability, storage cluster ensures high storage capacity, and memory cluster focuses on large memory size needed during computation. Table 1 gives specifications for each of the cluster type.

Table 1: VM categories specifications used in our user-driven simulations

VM Category	VM Name	# of Cores	Memory (GB)	HDD Size (GB)	Bandwidth (Mbps)
Compute VM's	M1 Small	2	2	160	100
	M1 Medium	4	4	410	500
Storage VM's	M1 XL	16	16	1690	1000
	M3 XL	26	16	1690	1000
Memory VM's	High Mem XL	13	18	420	500
	High Mem 2XL	26	35	850	1000

IV. EXPERIMENT

Based on the above configuration of cluster and VM categories, we frame our experimental setup comprising test cases designed for each cluster. Three test cases of VMs categories are made for each of the clusters according to their type. Compute cluster is given 3 VM test cases, where test case 1, test case 2 and test case 3 comprises 20 VMs, 30 VMs, and 50 VMs, respectively, each belonging to compute VM category. Storage Cluster is given 3 VM test cases, where test case 1, test case 2 and test case 3 comprises 20 VMs, 30 VMs and 50 VMs, respectively, each belonging to the storage VM category. Memory Cluster is given 3 VM test cases, where test case 1, test case 2 and test case 3 comprises 20 VMs, 30 VMs and 50 VMs, respectively, each belonging to the memory VM category.

Each type of cluster has specific instances of each resource available with them depending upon its type. For example, Compute cluster has 250 cores, 300 GB Memory, 25000 GB HDD and 30000 Mbps Bandwidth available with them. Now, each of the test cases demands these resources. Test case 1 is for Compute VM category comprising 20 VM's request for 64 cores, 64 GB memory, 6200 GB HDD and 6800 Mbps Bandwidth. From these Available instances of resources and requested resources, the utilization is computed as mentioned in equation 1.

$$Utilization = \frac{RequestedResourceInstance}{AvailableResourceInstance} \quad (1)$$

Similarly, utilization is computed for the remaining 2 test cases of compute cluster. Table 3 gives insights for all the test cases, resource status and utilization corresponding to each test case for compute cluster.

Table 2: Details for test cases, resource status, and utilization for compute cluster

Compute_Cluster	Resources Status	# of Cores	Memory (GB)	HDD Size (GB)	Bandwidth (Mbps)
	Available Instances	**250**	**300**	**25000**	**30000**
20 VM's	Requested	64	64	62,00	68,00
	Difference	186	236	18,800	23,200
	Utilization	**0.2560**	**0.2133**	**0.2480**	**0.2267**
30 VM's	Requested	96	96	9300	10200
	Difference	154	204	15700	19800
	Utilization	**0.3834**	**0.3200**	**0.3720**	**0.3400**
50 VM's	Requested	160	160	15,500	17,000
	Difference	90	140	9,500	13,000
	Utilization	**0.6400**	**0.5333**	**0.6200**	**0.5667**

Similar procedure is applied for computing utilization for storage and memory cluster. Their details are given in table 4 and 5 respectively

Table 3: Details for test cases, resource status and utilization for storage cluster

Storage_Cluster	Resources Status	# of Cores	Memory (GB)	HDD Size (GB)	Bandwidth (Mbps)
	Available Instances	**1500**	**1000**	**100,000**	**60,000**
20 VM's	Requested	440	320	33,800	20,000
	Difference	1,060	680	66,200	40,000
	Utilization	**0.2933**	**0.3200**	**0.3380**	**0.3333**
30 VM's	Requested	660	480	50,700	30,000
	Difference	840	520	49,300	30,000
	Utilization	**0.4400**	**0.4800**	**0.5070**	**0.5000**
50 VM's	Requested	1,100	800	84,500	50,000
	Difference	400	200	15,500	10,000
	Utilization	**0.7333**	**0.8000**	**0.8450**	**0.8333**

Table 4: Details for test cases, resource status and utilization for memory cluster

Memory_Cluster	Resources Status	# of Cores	Memory(GB)	HDD Size(GB)	Bandwidth(Mbps)
	Available Instances	**1500**	**2000**	**50,000**	**60,000**
20 VM's	Requested	416	564	13,560	16,000
	Difference	1,084	1,436	36,440	44,000
	Utilization	**0.2773**	**0.2820**	**0.2712**	**0.2667**
30 VM's	Requested	624	846	20,340	24,000
	Difference	876	1,154	29,660	36,000
	Utilization	**0.4160**	**0.4230**	**0.4068**	**0.4000**
50 VM's	Requested	1,040	1,410	33,900	40,000
	Difference	460	590	16,100	20,000
	Utilization	**0.6933**	**0.7050**	**0.6780**	**0.6667**

V. RESULT

We compute reliability for the experimental setup described above. Three test cases are designed for VM's demand for resources from cluster as per their need according to their type. We create the total samples for each test case and compute the reliability of the system. Table 6 reports the total number of samples created for each test case of the VM type for clusters and the average reliability calculated for the total number of samples.

We also report the pictorial representation of results based on failure reports and reliability values. Here we depict, the number of failures accounted for each component (CPU, Memory, HDD, Bandwidth), Ratio of M1Small and M1Medium VM used and the reliability graph over the number of iterations. Figure 2-10 shows these results for each VM test case over all three clusters.

It can be seen from figure 2-4, for compute cluster, maximum failure is reported for HDD followed by CPU, Bandwidth and memory. In the storage cluster, as shown in Figures 5-7, regular fashion is CPU and HDD failure compared to Bandwidth and memory failure. It is also interesting to note that when the percentage of M3XL VM is more than M1XL, we encounter more CPU failure than HDD failure. Similar to compute cluster shown in figure 8-10, maximum failure is reported for HDD followed by CPU, Bandwidth and memory in the memory cluster.

Table 5: Total samples and average reliability for every VM test case of each cluster

CLUSTER NAME	VM Test Cases	Total Samples	Reliability
Compute_Cluster	20 VM's	236,385	0.999336
	30 VM's	218,837	0.999283
	50 VM's	210,925	0.999256
Storage_Cluster	20 VM's	230,148	0.999318
	30 VM's	209,141	0.999249
	50 VM's	182,269	0.999139
Memory_Cluster	20 VM's	235,533	0.999333
	30 VM's	235,320	0.999333
	50 VM's	219,526	0.999285

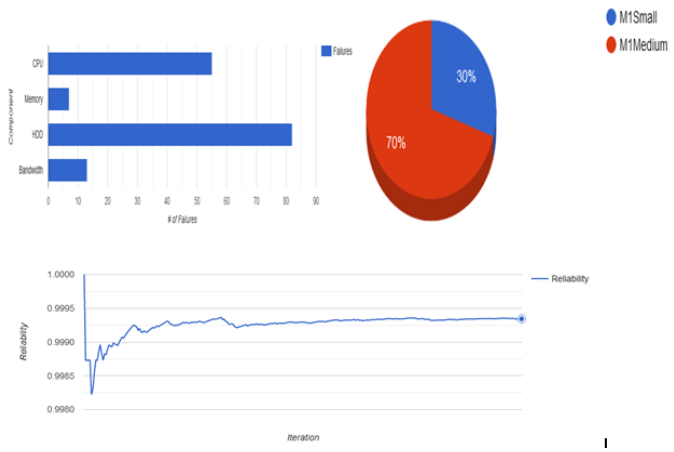

Figure 2 Pictorial representation of result for an instance of 20VM for Compute cluster

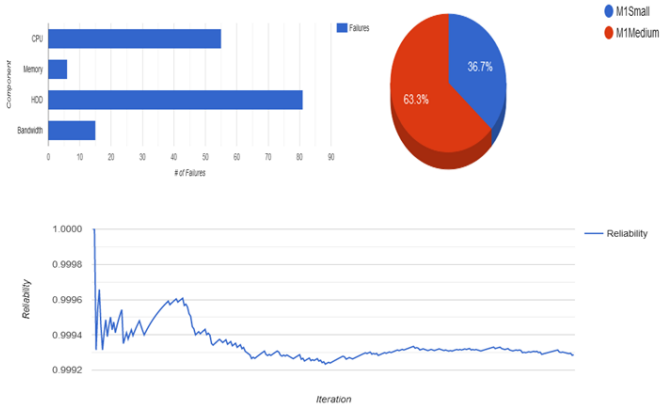

Figure 3 Pictorial representation of result for an instance of 30VM for Compute cluster

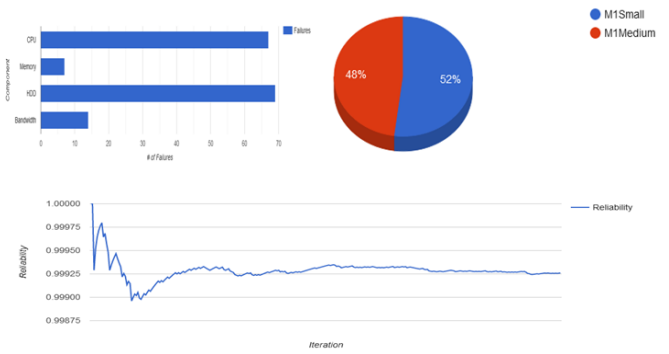

Figure 4 Pictorial representation of result for an instance of 50VM for Compute cluster

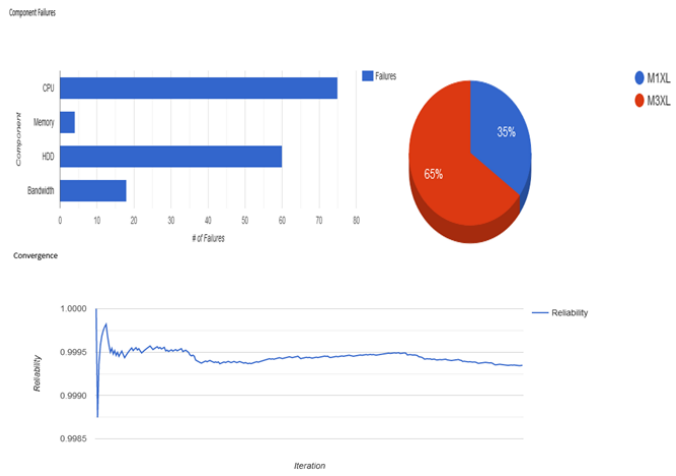

Figure 5 Pictorial representation of result for an instance of 20VM for the storage cluster

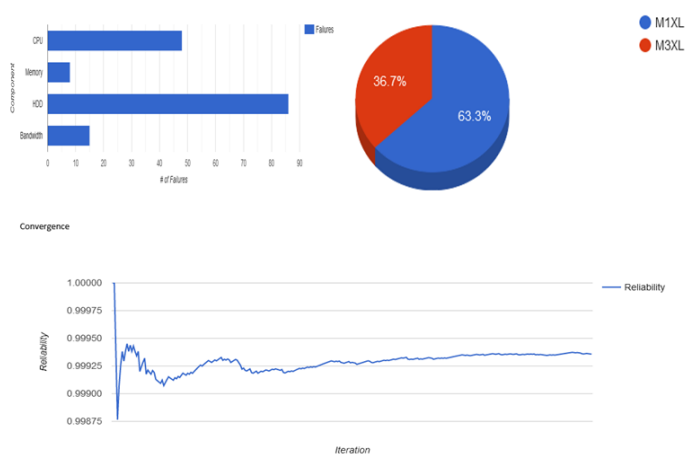

Figure 6 Pictorial representation of result for an instance of 30VM for the storage cluster

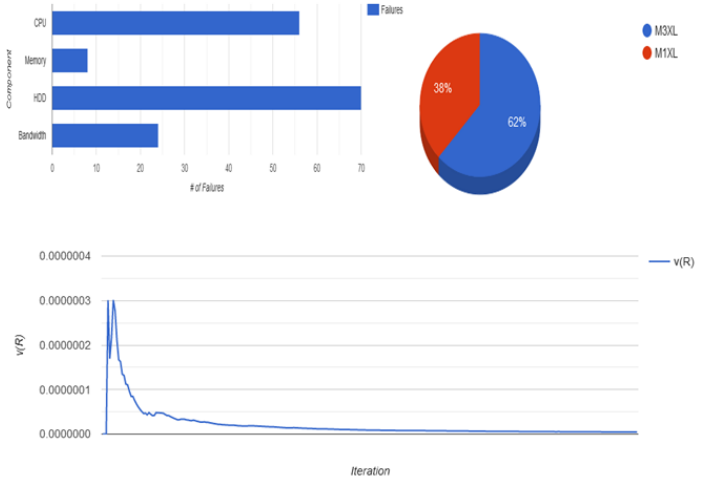

Figure 7 Pictorial representation of result for an instance of 50VM for the storage cluster

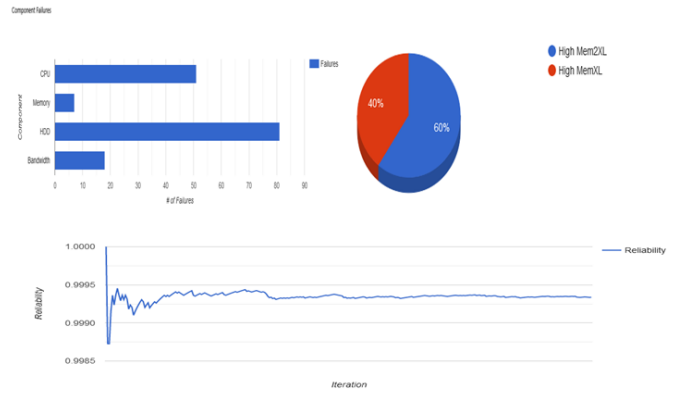

Figure 8 Pictorial representation of result for an instance of 20VM for memory cluster

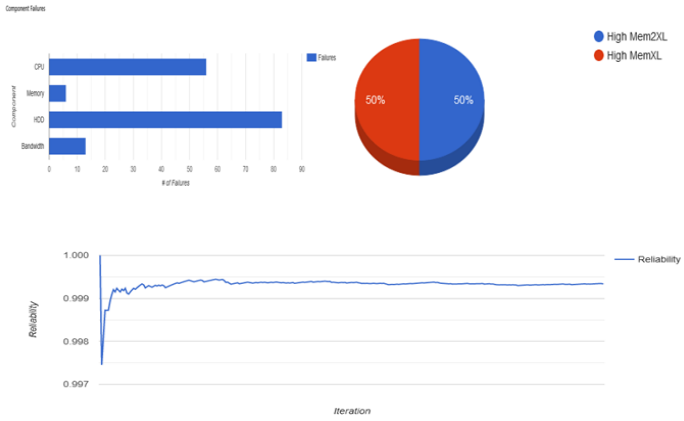

Figure 9 Pictorial representation of result for an instance of 30VM for memory cluster

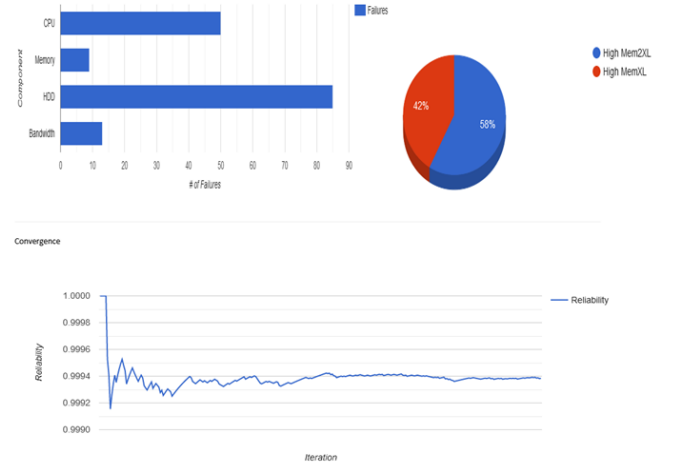

Figure 10 Pictorial representation of result for an instance of 50VM for memory cluster

VI. CONCLUSION

This paper performs a detailed analysis of the reliability of cloud computing systems for their applicability in various domains. We have framed our test cases so that it covers the broad domain where CCS is used. In this broad scenario, our paper reports the failure of its component, i.e., HDD, CPU, Bandwidth and memory, when tested across the various cluster and VM provided by ReliaCloud-NS. We have explored the various VMs available in ReliaCloud-NS and have studied their impact on failure on CCS components.

In future, we would like to explore more on the application of CCS in various domains. Application of CCS in the domain of life sciences, drug discovery, the medical field is the need of the hour. Hence, the future scope of our work is to focus on issues faced by life science researchers on using CCS. Here along with reliability, the sensitivity of data used is of significant concern.

REFERENCES

[1] Costello, K., & Rimol, M. (2020). Gartner Forecasts Worldwide Public Cloud End-User Spending to Grow 18% in 2021. *Gartner. Available online: https://www. gartner. com/en/newsroom/press-releases/2020-11-17-gartner-forecasts-worldwide-public-cloud-enduser-spending-to-grow-18-percent-in-2021 (accessed on 18 February 2021)*.

[2] Weinman, J. (2012). *Cloudonomics: The business value of cloud computing*. John Wiley & Sons.

[3] Breton, V., Lampe, N., Maigne, L., Sarramia, D., & Quang, B. T. (2016). Clouds in biosciences: A journey to high-throughput computing in life sciences. *Grid and Cloud Computing: Concepts and Practical Applications*, *192*, 71.

[4] Garg, V., Arora, S., & Gupta, C. (2011). Cloud computing approaches to accelerate drug discovery value chain. *Combinatorial chemistry & high throughput screening*, *14*(10), 861-871.

[5] Snyder, B., Ringenberg, J., Green, R., Devabhaktuni, V., & Alam, M. (2015). Evaluation and design of highly reliable and highly utilized cloud computing systems. *Journal of Cloud Computing*, *4*(1), 1-16.

[6] Rashid, L., Pattabiraman, K., & Gopalakrishnan, S. (2012, September). Intermittent hardware errors recovery: Modeling and evaluation. In *2012 Ninth International Conference on Quantitative Evaluation of Systems* (pp. 220-229). IEEE.

[7] Vishwanath, K. V., & Nagappan, N. (2010, June). Characterizing cloud computing hardware reliability. In *Proceedings of the 1st ACM symposium on Cloud computing* (pp. 193-204).

[8] Nightingale, E. B., Douceur, J. R., & Orgovan, V. (2011, April). Cycles, cells and platters: an empirical analysisof hardware failures on a million consumer PCs. In *Proceedings of the sixth conference on Computer systems* (pp. 343-356).

[9] Pham, C., Cao, P., Kalbarczyk, Z., & Iyer, R. K. (2012, June). Toward a high availability cloud: Techniques and challenges. In *IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN 2012)* (pp. 1-6). IEEE.

[10] Sharma, R., & Singh, R. (2021). A Highly Reliable and Cost-effective Service Model for Finite Population Clouds: Analysis and Implementation. *Arabian Journal for Science and Engineering*, 1-16.

[11] Ramos, R. M., Martinello, M., & Rothenberg, C. E. (2013, October). Slickflow: Resilient source routing in data center networks unlocked by openflow. In *38Th annual IEEE conference on local computer networks* (pp. 606-613). IEEE.

[12] Gill, P., Jain, N., & Nagappan, N. (2011, August). Understanding network failures in data centers: measurement, analysis, and implications. In *Proceedings of the ACM SIGCOMM 2011 Conference* (pp. 350-361).

[13] Birke, R., Chen, L. Y., Smirni, E., Birke, R., Chen, L. Y., & Smirni, E. (2012). Data centers in the wild: A large performance study. *IBM Research, Zurich, Switzerland*.

[14] Buyya, R., & Murshed, M. (2002). Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. *Concurrency and computation: practice and experience*, *14*(13-15), 1175-1220.

[15] Sulistio, A., Cibej, U., Venugopal, S., Robic, B., & Buyya, R. (2008). A toolkit for modelling and simulating data Grids: an extension to GridSim. *Concurrency and Computation: Practice and Experience*, *20*(13), 1591-1609.

[16] Casanova, H. (2001, May). Simgrid: A toolkit for the simulation of application scheduling. In *Proceedings First IEEE/ACM International Symposium on Cluster Computing and the Grid* (pp. 430-437). IEEE.

[17] Casanova, H., Legrand, A., & Quinson, M. (2008, April). Simgrid: A generic framework for large-scale distributed experiments. In *Tenth International Conference on Computer Modeling and Simulation (uksim 2008)* (pp. 126-131). IEEE.

[18] Buyya, R., Ranjan, R., & Calheiros, R. N. (2009, June). Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities. In *2009 international conference on high performance computing & simulation* (pp. 1-11). IEEE.

[19] Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., & Buyya, R. (2011). CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and experience*, *41*(1), 23-50.

[20] Wickremasinghe, B., Calheiros, R. N., & Buyya, R. (2010, April). Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications. In *2010 24th IEEE international conference on advanced information networking and applications* (pp. 446-452). IEEE.

[21] Buyya, R., Pandey, S., & Vecchiola, C. (2009, December). Cloudbus toolkit for market-oriented cloud computing. In *IEEE International Conference on Cloud Computing* (pp. 24-44). Springer, Berlin, Heidelberg.

[22] Garg, S. K., & Buyya, R. (2011, December). Networkcloudsim: Modelling parallel applications in cloud simulations. In *2011 Fourth IEEE International Conference on Utility and Cloud Computing* (pp. 105-113). IEEE.

[23] Calheiros, R. N., Netto, M. A., De Rose, C. A., & Buyya, R. (2013). EMUSIM: an integrated emulation and simulation environment for modeling, evaluation, and validation of performance of cloud computing applications. *Software: Practice and Experience*, *43*(5), 595-612.

[24] Nunez, A., Vazquez-Poletti, J. L., Caminero, A. C., Carretero, J., & Llorente, I. M. (2011, June). Design of a new cloud computing simulation platform. In *International Conference on Computational Science and Its Applications* (pp. 582-593). Springer, Berlin, Heidelberg.

[25] Núñez, A., Vázquez-Poletti, J. L., Caminero, A. C., Castañé, G. G., Carretero, J., & Llorente, I. M. (2012). iCanCloud: A flexible and scalable cloud infrastructure simulator. *Journal of Grid Computing*, *10*(1), 185-209.

[26] Núñez, A., Andrés, C., & Merayo, M. G. (2012, November). Mascloud: a framework based on multi-agent systems for optimizing cost in cloud computing. In *International Conference on Computational Collective Intelligence* (pp. 436-445). Springer, Berlin, Heidelberg.

[27] Kliazovich, D., Bouvry, P., & Khan, S. U. (2012). GreenCloud: a packet-level simulator of energy-aware cloud computing data centers. *The Journal of Supercomputing*, *62*(3), 1263-1283.

[28] Snyder, B., Green, R. C., Devabhaktuni, V., & Alam, M. (2018). ReliaCloud-NS: A scalable web-based simulation platform for evaluating the reliability of cloud computing systems. *Software: Practice and Experience*, *48*(3), 665-680.

***