

Performance Evaluation of SUMEGHA Cloud Computing Environment: Methodologies & Tool



Cyril Raj, Abey Jacob

Abstract—Cloud ecosystem basically offers Platform as a Service (PaaS), Infrastructure as a Service (IaaS) and Software as a Service (SaaS). This paper describes the testing process employed for testing the C-DAC cloud SuMegha. Though new tools for the testing cloud are emerging into the market, there are aspects which are suited for manual testing and some which can be speeded up using automated testing tools. This paper brings out the techniques best suited to test different features of Cloud computing environment. It offers a comparison of several tools which enhance the testing process at each level. The authors also try to bring out (recommend) broad guidelines to follow while setting up a cloud environment to reduce the number of bugs in the system.

Index Terms—Cloud Computing, Performance, Testing tools.

I. INTRODUCTION

SuMegha Cloud is developed by CDAC and it offers numerous facilities for computing functions. Currently we have a tendency to give the Platform as a Service (PaaS), Infrastructure as a Service (IaaS), Storage as a Service (SaaS) and Software as a Service (SaaS). High-Performance Computing (HPC) permits researchers and academicians to unravel advanced Scientific, engineering and business applications that need terribly high figure capability, high capacity storage, higher bandwidth and low latency interconnections. To achieve this, Scientists and engineers either have to wait for a long queue to access the shared cluster resources or acquire pricey computing hardware (clusters) for their own premises. Cloud computing is a beneficial model for, on the fly access from a common pool and can customizable the HPC resources (e.g., servers, storage, networks, applications, software, and services) which can be freely provisioned as and when required. For analysis teams, cloud computing can give comfortable access to stable, HPC clusters and storage, without procuring and retain highly advanced computing peripherals. High-Performance Computing (HPC) helps researchers, engineers and academicians to unravel advanced scientific and business problems that need terribly high figure capacity, storage, large bandwidth and highly optimized interconnectivity.

Scientists and researchers have to wait for a long time to get access from a pooled cluster resources or has to procure dearly-won physical systems (clusters) at own research centre. Computing facility over cloud is a reliable model, for on the fly access from a common pool of reconfigurable highly intensify computing resources (e.g., servers, storage, networks, applications, software, and services), which can be easily accessible in on demand basis. For application and analysis teams, computing over the cloud can facilitate on the fly access to this stable, highly intensified computing resources and storage, by not procuring and retain the innovatory computing nodes. SuMegha cloud is an associate in nursing IaaS Cloud that can facilitate virtual instruments and cluster on the fly for researchers and scientists. IaaS Web access portal links you with the GUI to various offerings for Creating, Destroying and saving the virtual instruments. It conjointly provides single-click cluster with MPI or HADOOP platform change thereon. SuMegha cloud is a PaaS Cloud for facilitating single-access on-the-fly virtual cluster with the MPI facility enabled on it. IaaS internet based web portal provides the Graphical Interface to various services like making and crashing the virtual cluster. If wish to try, it's just to pick out the required environment (i.e. MPI) from the given list and Machine Size (i.e. Small/Medium/Large), along with the entire number of nodes in a cluster and the duration required for running the virtual cluster. SuMegha facilitates for SaaS (Software as a Service) within a variety of the "Job Submission Portal" and alternative numerous usability depended "Problem finding setting Portals". This web access hosted service facilitates online access for keying serial and parallel assignments/executables to HPC cloud computing setting that is obtainable as IaaS of SuMegha. This conjointly provides observance for task standing and viewing/downloading the out comings/faults knowledge. Web access portal is made for the engineering application developers in the cloud for accessing the virtual machines and for executing their tasks, wherein not moving to the cumbersome of scripting on command lines.

II. SUMEGHA CLOUD

The three main types of offerings by C-DAC SuMegha Cloud are IaaS, PaaS and SaaS. The layers of SuMegha stack are as shown in fig1.

Revised Manuscript Received on October 30, 2019.

* Correspondence Author

ABEY JACOB*, Principal technical Officer with Centre for Development of Advanced computing (CDAC), Bangalore - India.

Dr. V CYRIL RAJ, Join Registrar (E&T) at Dr. M.G.R University, Chennai - India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>.



Performance evaluation of SUMEGHA Cloud Computing Environment: Methodologies & Tool

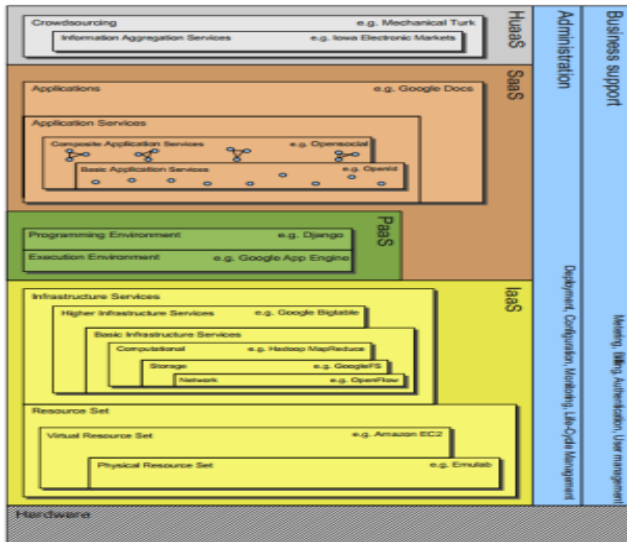


Figure 1 Layers of SuMegha stack

Individual components at each layer need to be tested, and then Integration Testing should be conducted to ensure the quality of the Cloud computing system. Specifically, with respect to Cloud computing environment, testing should be carried out for cloud-specific factors such as elasticity. The SPEC Open Systems Group (OSG) Cloud working group has been investigating application workloads suitable for benchmarking. It lists four key metrics relevant in the context of cloud benchmarking based on the report published by Steering Committee, SPEC OSG Cloud Computing Working Group in 2012. The main aspects to be tested in the Cloud Computing environment from the user's viewpoint include the following

- Elasticity
- Provisioning interval, or the lag between when a resource is requested and when it is actually available.
- Agility or the ability of the provider to track the needs of the workload.
- Scale-up or the improvement in response times with an increased amount of resources.
- Elastic speed-up or the development in turnout as a further resource is supplementary on the fly.
- Throughput
- Response time
- Variability

III. SUMEGHA CLOUD STACK COMPONENTS

Cloud Stack of SuMegha is conceived in such a manner to cater to the needs of scientific environments and application ecosystems. The stack contains most reliable versions of most appropriated Cloud elements needed to make the non-public scientific cloud.

The SuMegha Cloud consists of the following primary components:

- Hypervisor (Xen)
- Middleware (Nimbus)
- Web Portal for user interface

Pre-built golden images with CentOS, MPI & HADOOP ecosystem and facilities for submitting executables through a portal to virtual HPC clusters.

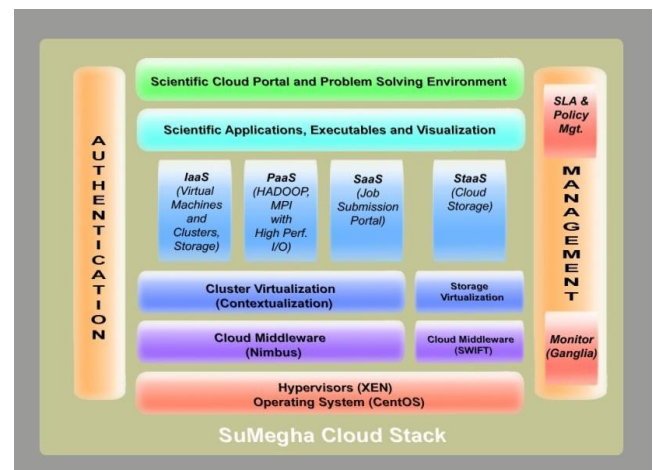


Figure 2 SuMegha Cloud Stack

The architecture of SuMegha introduces the concept of two nodes, akin to Client-Server architecture: Virtual Machine Manager (VMM) Node and the Service Node (SN), as shown in Figure 3. The minimum requirement for SuMegha installation is a desktop machine, where both VMM and Service Node can be installed. The VMM Node has to be setup initially and then the Service Node and finally configure both VMM and Service node. The below Figure depicts the Basic Architecture of SuMegha.

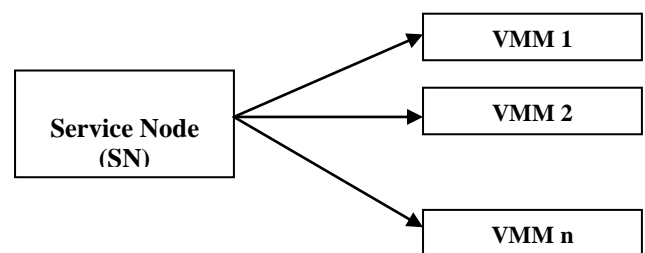


Figure 3 Basic Architecture of SuMegha

SuMegha Cloud can have a multi-node installation. There shall be only one Service Node, which acts as a controller. VMM Nodes can be installed on multiple machines and configured with a Service Node. SuMegha Cloud Lab kit then installs the Cloud components automatically in an interactive mode. The administrator should be aware of the network details (ex: Hostname, IP address, DNS, Gateway) which has to be provided during installation. After knowing about the SuMegha Cloud, I was asked to install and configure. For the installation, the CD was provided by the SuMegha Cloud. With the help of CD and installation manual, I was able to install the SuMegha Cloud Lab Kit.

The below Figure.4 depicts SuMegha offerings which include the GUI, IaaS, PaaS and SaaS long with the Image size.

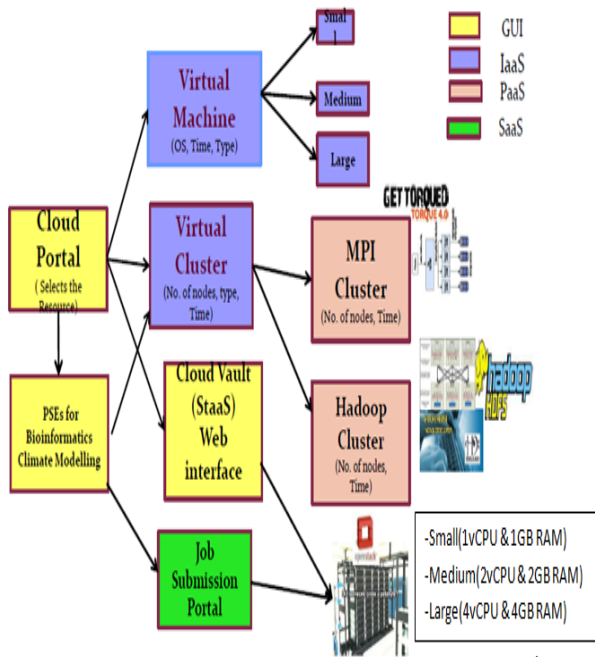


Figure: 4 SuMegha Offerings

In order to program a Virtual Machine Manager (VMM) node and Service Node for SuMegha Cloud Lab Kit in a new system, it should possess the following minimum requirements.

VIRTUAL MACHINE MANAGER (VMM) NODE

| | |
|-------------------------|--|
| O S | CentOS Version > 6.2 |
| Central Processing Unit | Minimum 1 or more 64-bit x86 CPU(s), 1.5 GHz or above, 2 GHz or rapid multi-core CPU suggested |
| RAM | At least 4 GB |
| Disk Space | At least 60 GB; least 2GB for /boot partition |
| Network | Internet Connectivity |

SERVICE NODE

| | |
|-------------------|-------------------------------------|
| Operating System | CentOS Version > 6.2 |
| RAM | 4 GB |
| Storage Space | At least 100 GB |
| Interconnectivity | Fair speed Internet Connection |
| Software | Oracle JAVA 1.7, Python (2.6 – 3.0) |

SCIENTIFIC CLOUD FOR HPC

High-Performance Computing (HPC) permits researchers and academicians to resolve sophisticated scientific problems, algorithms and business problems which need really large cipher capacity, immense storage, bigger system of measurement and least delayed network. Researchers and academicians have to wait for a longer time to get connected with pooled cluster machines or have to procure these high-end devices at a higher cost for them.

Cloud computing is a practical replica for accessing the resources from a distributed pool of reconfigurable HPC hardware machines along with relevant software packages

(e.g., servers, storing space, interconnectivity, running suits, software, and services) on the fly. For analysis teams, cloud computing is capable to present a simple one-shot opportunity to a stable, high-performance clusters and storing space, without procuring and maintain dearly-won hardware (clusters) at their organization. C-DAC scientific cloud is associate in nursing IaaS Cloud that facilitates virtual machine and virtual cluster on the fly as and when needed. Scientific Cloud Web access portal has the Graphical User Interface, which simplifies the Creation, liquidation and preserving the virtual machines. It conjointly serves the platform to make a simple cluster with MPI or HADOOP thereon.

Highlights

- User-friendly GUI to the scientific Cloud.
- On the fly Creation and liquidation of Virtual Machine and Cluster, as and when required.
- Option to preserve the Virtual Machine.
- Executing the Virtual Machine Images.
- Facility to capture record the Virtual machines and Cluster.
- SSH-based key authenticated access to the Virtual machines and Cluster.

INTENDED USERS

CDAC Scientific Cloud is meant for the scholars and men of science who wants the tiny cluster for research and executing the MPI and HADOOP projects. Since it's capable to cater to the requirements of a Virtual Machine, it's able to leverage for server functionality too.

BENEFIT TO END USERS

For end-users, it can provision for functional servers, group of HPC and storing resources as IaaS. On the fly, facility for MPI & Map Reduce clusters for executing high computing requirements and large data-generating programs.

- Secured access using SSH commands to Virtual machines.
- Periodic updating all instances in the web portal.

FEATURES TO BE TESTED

This section describes the various items that required to be tested in our cloud computing. The network interconnectivity infrastructure level comprises of the routers, switches, gateways and firewalls. Suggesting and executing a suitable solution for "Benchmarking and Performance Monitoring of Scientific Cloud."

- Methodology for Performance Evaluation
- Finding Optimal Performance
- Benchmarking and Standardization

TESTING METHODS & TOOLS

For each of the above features describe, what are the testing method and tools available? Many standard non-cloud testing tools can be re-used for performance, stress and benchmarking of Cloud Computing Environment.

Performance evaluation of SUMEGHA Cloud Computing Environment: Methodologies & Tool

AVAILABLE BENCHMARKING SUITS

- PerfKit Benchmark from Google
- LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator)
- GEAGUL-C v1.0 (Grid Software- Enabling Applications for Grid Computing Using GLOBUS and C-Language)
- GOPAEAG-v1.0 (Globus and Object-oriented Programming Approach to Enable Applications for Grid Computing)
- SPAGMOS-v1.0 (Software Probes for Assessment of Grid Middleware Overheads)

SUGGESTED SOLUTION: PERFKIT BENCHMARKER

PerfKit Benchmark is associated in nursing benchmarking tool accustomed live and to equate the cloud performances. PerfKit Benchmark is legalized underneath the Apache 2 legal terms & conditions. The goal is to form associate in nursing an active benchmarking structure which it constitutes. However, Cloud creators square measure developing applications, assessing Cloud options and nursing how to build use cases for each cloud. PerfKit Benchmark calculates the starting to finish time to provide resources within the cloud, additionally to reportage on the foremost customary metrics of top accomplishment, e.g.: Delay, amount of data passing, execution time, IOPS. PerfKit Benchmark minimizes the complication in handling benchmarks on assisting cloud service vendors by combined and basic commands. It's conceived in such a manner to handle via servicer enabled command-line scripts.

PERFKIT INSTALLATION AND CONFIGURATION
PerfKit Benchmark is a free to use publically available tool. For Benchmarking SuMegha we have used a trial version for 60 days and it can be updated later can be downloaded from the GitHub. Before we run the Benchmark we are supposed to create an account either in Google, Amazon, Microsoft Azure, Alicloud, Digital Ocean, Rackspace. For Benchmarking the account was created on Google Cloud.

After the installation and configuration of PerfKit, instances are created. User can either select the default images which are available or can create the image and upload. User can also select the machine size for the image. Machine-size can be from 1vCPU to 32vCPUs. With the selected Machine-size the memory of the image is also noted. Once the instances are created it is possible to know the detailed description of the instance.

Benchmarking is done by capturing and analyzing factors like how much CPU is utilized, Disk Bytes, type of Disk Operations, Network Bytes and Network Packets used. The screenshots based on these parameters are depicted for the instance created.

CPU Count=1(SMALL)

| Iteration | 25000 | 30000 | 35000 | 40000 | 45000 | 50000 |
|-----------|-------|-------|-------|-------|-------|-------|
| VM | 0.25 | 0.3 | 0.32 | 0.43 | 0.45 | 0.51 |
| Console | 0.08 | 0.08 | 0.1 | 0.13 | 0.15 | 0.14 |

CPU Count=2(MEDIUM)

| Iteration | 25000 | 30000 | 35000 | 40000 | 45000 | 50000 |
|-----------|-------|-------|-------|-------|-------|-------|
| VM | 0.0 | 0.0 | 0.0 | 0.0 | 0.01 | 0.01 |
| Console | 0.07 | 0.08 | 0.1 | 0.11 | 0.11 | 0.14 |

CPU Count=4(LARGE)

| Iteration | 25000 | 30000 | 35000 | 40000 | 45000 | 50000 |
|-----------|-------|-------|-------|-------|-------|-------|
| VM | 0.05 | 0.02 | 0.0 | 0.0 | 0.2 | 0.0 |
| Console | 0.07 | 0.09 | 0.1 | 0.1 | 0.12 | 0.14 |

When both Small and Medium VMs are Running

| Iteration | 25000 | 30000 | 35000 | 40000 | 45000 | 50000 |
|-----------|-------|-------|-------|-------|-------|-------|
| VM(S) | 0.26 | 0.23 | 0.27 | 0.3 | 0.37 | 0.4 |
| VM(M) | 0.3 | 0.14 | 0.01 | 0.01 | 0.01 | 0.01 |
| Console | 0.07 | 0.09 | 0.1 | 0.11 | 0.12 | 0.14 |

When both Medium and Large VMs are Running

| Iterations | 25000 | 30000 | 35000 | 40000 | 45000 | 50000 |
|------------|-------|-------|-------|-------|-------|-------|
| VM(M) | 0.26 | 0.0 | 0.03 | 0.02 | 0.02 | 0.12 |
| VM(L) | 0.06 | 0.0 | 0.04 | 0.04 | 0.0 | 0.02 |
| Console | 0.07 | 0.09 | 0.1 | 0.11 | 0.11 | 0.14 |

When both Large and Small VMs are Running

| Iteration | 25000 | 30000 | 35000 | 40000 | 45000 | 50000 |
|-----------|-------|-------|-------|-------|-------|-------|
| VM(L) | 0.26 | 0.0 | 0.02 | 0.0 | 0.01 | 0.0 |
| VM(S) | 0.19 | 0.22 | 0.39 | 0.31 | 0.31 | 0.36 |
| Console | 0.07 | 0.07 | 0.11 | 0.11 | 0.12 | 0.14 |

When only Small Size VMs are Running

| Iteration | 2500 | 3000 | 3500 | 4000 | 4500 | 5000 |
|-----------|------|------|------|------|------|------|
| VM(S) | 0.25 | 0.34 | 0.21 | 0.42 | 0.38 | 0.52 |
| VM(S) | 0.25 | 0.25 | 0.28 | 0.36 | 0.35 | 0.36 |
| Console | 0.07 | 0.08 | 0.1 | 0.1 | 0.12 | 0.14 |

When only Medium Size VMs are Running

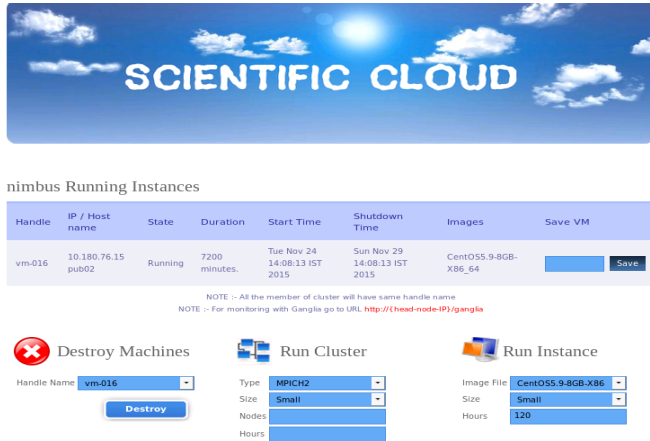
| Iteration | 25000 | 30000 | 35000 | 40000 | 45000 | 50000 |
|-----------|-------|-------|-------|-------|-------|-------|
| VM(M) | 0.04 | 0.02 | 0.02 | 0.02 | 0.0 | 0.01 |
| VM(M) | 0.09 | 0.13 | 0.12 | 0.22 | 0.11 | 0.19 |
| Console | 0.07 | 0.08 | 0.09 | 0.11 | 0.13 | 0.14 |



When only Large Size VMs are Running

| Iteration | 2500 | 3000 | 3500 | 4000 | 4500 | 5000 |
|-----------|------|------|------|------|------|------|
| VM(L) | 0.03 | 0.05 | 0.06 | 0.06 | 0.02 | 0.04 |
| VM(L) | 0.01 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Console | 0.06 | 0.08 | 0.09 | 0.10 | 0.12 | 0.14 |

Figure Screenshot of Running VM



The below Figure depicts a Screenshot of the CPU Utilization for the instance of 2vCPU

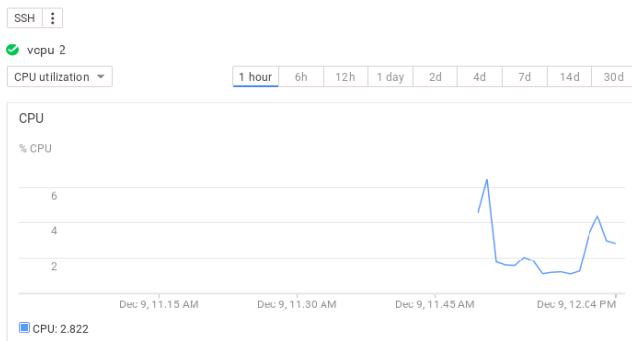


Figure: Screenshot of CPU Utilization for 2vCPU

The below Figure depicts a Screenshot of the Network Packets for the instance of 2vCPU

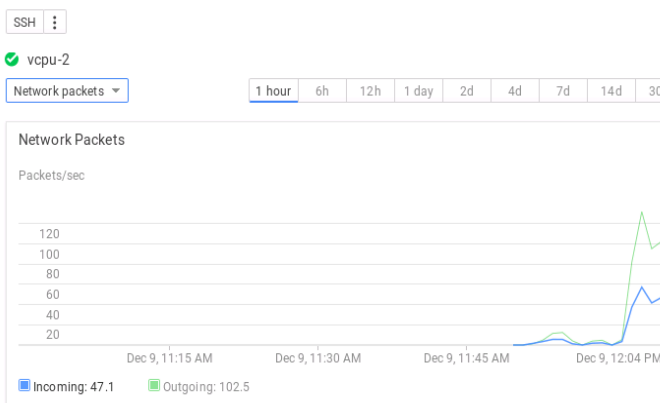


Figure: Screenshot of Network Packets for 2vCPU

The below Figure depicts the Screenshot of the CPU Utilization for the instance of 8vCPU

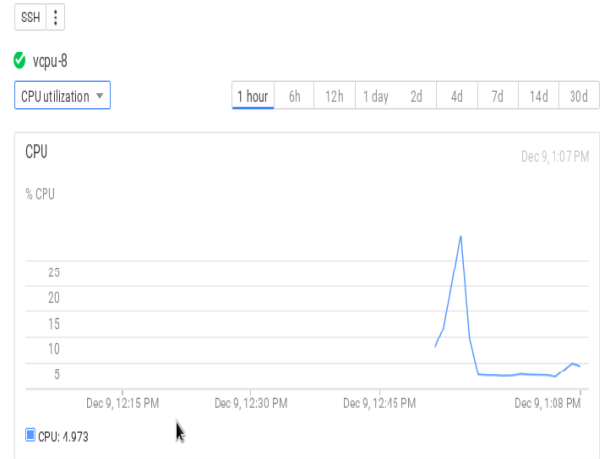


Figure: Screenshot of CPU Utilization for 8vCPU

The below Figure depicts the Screenshot of the Network Bytes for the instance of 8vCPU



Figure: Screenshot of Network Bytes for instance of 8vCPU

IV. CONCLUSION

The paper discussed the testing methodology used for testing various aspects of IaaS, PaaS and SaaS. It describes the popular kinds of bugs that appear in the Cloud Ecosystem and recommends some guidelines to reduce the number of bugs in the system. Popular evaluation methods adopted for validating the cloud (SuMegha) are described and their shortcomings for testing a cloud computing system are identified. The effort required to build new tools for testing the cloud vs. the practical implication on testing an operational cloud is yet quite debatable.

ACKNOWLEDGEMENT

Am grateful to Ministry of Electronics & Information Technology, for funding the SuMegha cloud project. We are also extremely thankful to Executive Director - CDAC, Group Heads and Ms. Mangala for their valuable guidance and encouragement.



Performance evaluation of SUMEGHA Cloud Computing Environment: Methodologies & Tool

Also thanking SuMegha development team for providing this opportunity to carry to perform the benchmarking experiments along with Ms. Nandini S, who assisted me in carrying out this research work.

REFERENCES

1. <http://www.sumegha.in/>
2. What's Inside the Cloud? An Architectural Map of the Cloud Landscape, <http://www.di.ufpe.br/~redis/intranet/bibliography/middleware/lenk-what-2009.pdf>
3. <http://clds.sdsc.edu/sites/clds.sdsc.edu/files/wbdb2012/papers/WBDB2012Paper40Jost.pdf>
4. SuMegha Cloud Kit: Create Your Own Private Scientific Cloud by Vineeth S Arackal, Arunachalam B, Kalasagar, Sumit Kumar, Mangala N, Sarat Chandra Babu, Prahlada Rao, Sukeshini. <https://pos.sissa.it/270/024/pdf>
5. https://pdfs.semanticscholar.org/533f/d669ab5e0fa925dab55bca2d43c936f9a9b3.pdf?_ga=2.204569424.1593676386.1561018071-691968923.1558962402
6. Yatendra Singh Pundhir, "Cloud Computing Applications And Their Testing Methodology", Bookman International Journal of Software Engineering, Vol. 2 No. 1, Jan-Feb-Mar 2013
7. Michael Lynch, Thomas Cerqueus, Christina Thorpe, "Testing a cloud application: IBM SmartCloud inotes: methodologies and tools", International Workshop on Testing the Cloud, IISTA, July 2013, Switzerland.
8. William Jenkins, Sergiy Vilkomir, Puneet Sharma, "Framework for Testing Cloud Platforms and Infrastructures ", 2011 International Conference on Cloud and Service Computing
9. Dr. Rahul Malhotra, Prince Jain, "Testing Techniques and its Challenges in a Cloud Computing Environment", The SIJ Transactions on Computer Science Engineering & its Applications (CSEA), Vol. 1, No. 3, July-August 2013
10. Eric Bauer, Randee Adams, "Reliable Cloud Computing – Key Considerations", <http://www2.alcatel-lucent.com/techzine/reliable-cloud-computing-key-considerations/>
11. Sue Poremba, "Open Source Cloud Computing Testing Tools", Cloud Expo, 25 April 2013, <http://www.cloudcomputingexpo.com/node/2625872>
12. Alexandru Iosup, Simon Ostermann, Nezhir Yigitbasi, Radu Prodan, Thomas Fahringer, Member and Dick Epema "Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing" IEEE published in 2010
13. Niloofar Khangahi and Reza Ravanmehr "Cloud Computing Performance Evaluation: Issues And Challenges" IJCCSA (International Journal on Cloud Computing: Services and Architecture) ,Vol.3, No.5, October 2013
14. Dhivya, Padmaveni "Dynamic Resource Allocation using Virtual Machines for cloud computing Environment" IJREAT (International Journal of Research in Engineering & Advanced Technology), Volume 2, 2014.
15. Rupali Shelke, Rakesh Rajani "Dynamic resource allocation in Cloud Computing" International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 10, October – 2013
16. www.Nimbus-zero-cloud
17. [www.GooglePerfkit Benchmark](http://www.GooglePerfkitBenchmark)
18. Hanyu Pei ; Beibei Yin ; Min Xie, "Dynamic Random Testing Strategy for Test Case Optimization in Cloud Environment" 2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW). Pages: 148 - 149
19. Sheng-Jen Hsieh; Shyan-Ming Yuan; Guo-Heng Luo; Hsiao-Wei Chen "A flexible public cloud based testing service for heterogeneous testing targets" The 16th Asia-Pacific Network Operations and Management Symposium 2014, Pages: 1 - 3
20. Pillutla Harikrishna ; A. Amuthan, "A survey of testing as a service in cloud computing", 2016 International Conference on Computer Communication and Informatics (ICCCI) 2016, Pages: 1 - 5
21. Wei Wang ; Ningjing Tian; Sunzhou Huang; Sen He; Abhijeet Srivastava; Mary Lou Soffa; Lori Pollock, "Testing Cloud Applications under Cloud-Uncertainty Performance Effects", IEEE 11th International Conference on Software Testing, Verification and Validation (ICST) 2018, Pages: 81 - 92
22. HongHui Li; XiuRu Li; Huan Wang; Jie Zhang; ZhouXian Jiang, "Research on Cloud Performance Testing Model", IEEE 19th International Symposium on High Assurance Systems Engineering (HASE) 2019, Pages: 179 - 183

23. Nitin Naik, "Connecting google cloud system with organizational systems for effortless data analysis by anyone, anytime, anywhere", IEEE International Symposium on Systems Engineering 2016, Pages 1 – 6

AUTHORS PROFILE



ABEY JACOB had completed B.E (ECE) and M.Tech (CSE) and at present working as a Principal technical Officer with Centre for Development of Advanced computing (CDAC), Bangalore - India. He is currently pursuing PhD in cloud computing, at CSE Department of Dr.M.G.R University, Chennai - India. His areas of interest are Grid Computing, Cyber Forensic and Cryptography. He has published couple of papers in National and International Journals.



Dr. V CYRIL RAJ had completed M.E, PhD and presently working as Join Registrar (E&T) at Dr.M.G.R University, Chennai - India. Formerly he is the Dean (E&T) & HoD for Computer science Department. He has guided more than 20 PhD students and his areas of interest are Bioinformatics Data Mining, Grid Computing, Cloud & Mobile Computing, Robotics etc. He had published a lot of papers in International and National level journals and also authored many books.