ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
Indian Journals of Science and Technology, *Vol 9(19), DOI: 10.17485/ijst/2016/v9i19/89211, May 2016*

A Systematic Approach to Deal with Noisy Neighbour in Cloud Infrastructure

Ansuyia Makroo* **and Deepak Dahiya**

School of Engineering and Technology, Ansal University, Gurgaon - 122003, Haryana, India;
ansuyia@gmail.com, deepakdahiya@ansaluniversity.edu.in

Abstract

Background/Objectives: One of the major challenges of the multitenant cloud model is performance unpredictability because of resource contention. The objective of this paper is to propose an approach to deal with noisy neighbours in a shared cloud infrastructure and reduce their affect on the other tenant applications. **Methods/Statistical Analysis:** Multiple tenant applications are deployed on cloud VMs which share the underlying system resources. Noisy neighbour applications take up more resources and leave other applications in turmoil state that leads to lack of predictability of the performance of other applications. The proposed system actively monitors the resource consumption of the tenant applications based on some identified parameters. **Findings:** Monitoring the resource consumption of the applications helps categorize the resource greedy applications as noisy neighbours. The other tenant applications which do not get their fair share of the resources are victim applications. Once the noisy neighbours have been identified, the next step is to deal with noisy neighbours by either migration of victim applications on a separate node on the cloud or borrowing resources from other nodes or implementing a quota system for resource allocation in cloud. **Applications/Improvements:** The pragmatic study of the behaviour of tenant applications based on their resource consumption pattern would help researchers to better focus on techniques to improve the Quality of Service of a cloud.

Keywords: Cloud Computing, Multitenant, Noisy neighbour, Quality of Service, Virtual Machine

1. Introduction

Cloud computing[1] is an evolving area of research in academia and industry, which promotes economies of scale by providing virtualised scalable computing resources shared across multiple customers over internet. The pay per use[2] cloud outsourcing model has been immensely beneficial to small and medium scale enterprises to save on capital expenditure on upfront initial investment as well as maintenance of software and hardware computing resources. Cloud offers a multi-tenant[3] architecture to its users where multiple independent instances of user applications operate in a shared environment. A multi-tenant cloud architecture should ideally provide a very secure and exclusive virtual computing environment to the tenant applications but it is practically not feasible. A cloud consists of clusters[4] that further consists of one or more physical data centers which are virtually partitioned into a number of Virtual Machines (VMs). Tenant applications are deployed on VMs that are logically isolated, but physically integrated. The tenant applications which share the physical infrastructure[5] are in reality competing for shared underlying resources. In the multitenant infrastructure of a cloud, the greedy nature of one or more tenants can affect the performance of the other tenant applications. Noisy neighbour[6] is a phrase used to describe a cloud computing infrastructure co-tenant that monopolizes bandwidth, disk I/O, CPU and other resources, and can negatively affect other user applications cloud performance. The hypervisor virtualises the cloud physical resources in such a way that the virtualisation details are abstracted from tenant applications giving them an illusion of exclusive access to underlying physi-

*Author for correspondence

cal resources. Because of the scalable[1] nature of cloud, a resource greedy application consumes more resources than their fair share, thereby impacting the co-tenant applications. In such a scenario, the application's performance of a tenant application deployed on the same physical infrastructure as one of the noisy neighbours would suffer. Hence, a consequence of multitenant model of cloud is complete lack of performance predictability[9]of a given VM, since it depends on the other tenants of the service. Thus contention for resources is one of the main reasons for performance unpredictability in the cloud impacting the Quality of Service (QoS) of cloud[1] . In this paper, the authors have proposed a technique to deal with noisy neighbours in a shared cloud environment. The proposed model is based on monitoring and capturing the metrics based on the dynamic consumption of the system resources by the cloud tenant applications to facilitate proactive action.

2. Motivation and Problem Definition

The growing dependency of businesses applications on cloud infrastructure, will soon make it a fifth utility service[8]. Considering the immense business dependency on cloud, we can visualise, how popular the cloud computing model is going to be and how it is going to revolutionize the way computing resources have been utilised. One of the biggest obstacle to the growth of cloud computing is non-deterministic performance[1]. Multiple applications deployed on VMs share the system resources. Noisy neighbour applications take up more resources and leave other applications in turmoil state that leads to lack of predictability of the performance of other applications. A very important task in the cloud is to properly schedule the user's tasks on the VM in such a way that noisy applications do not impact the performance of other applications hosted on cloud.

This leads us to the following problem definition i.e.

To design a system that improves the performance and QoS in a cloud based infrastructure by handling the problems caused by noisy neighbours.

This problem definition leads us to the following broad objectives summarized below:

- Environment setup that simulates the noisy neighbours causing starvation of victim applications
- Implement the proposed solutions on the environment
- Derive meaningful conclusions based on the co-relation between identified parameters and proposed approaches

The rest of the paper is organized as follows: Section 1 gives an introduction to background. Section 2 includes the motivation and the problem definition. Section 3 discusses the related work done in the identified area. Section 4 describes the research framework and the proposed system in detail. Section 5 describes the experimental setup. Finally, Section 6 summarizes the conclusion; limitations and the future scope of work.

3. Related Study

Cloud computing model provides metered[10] computing resources as a service which allows the users to deploy software applications and data on a network of remote servers shared across multiple clients. Based on tenancy, there are two models of cloud computing: single-tenant and multi-tenant[11]. Single-tenant model[12] is the traditional model in which a customer deploys an instance of its application on a physically isolated hardware environment. However, the foundation of cloud computing is multitenant architecture in which multiple tenants share the virtually partitioned common infrastructure which helps in resource optimisation, thereby significantly saving the cost on software licenses and software maintenance. Cloud computing model offers computing as a service based on the principles of virtualisation[12] and abstraction[32]. Virtualisation enables creation a pool of storage and computing resources by partitioning the physical resources using virtualization technologies such as Xen[15], KVM[16]and VMware[17]. Virtualized servers allow co-hosting multiple independent[16] servers as virtual machines (VMs), on the same physical machine thereby improving the utilization rates of all the available servers and storage assets in the data center due to which cloud computing offers services below the costs of a medium-sized datacenter[1]. Multiple instances of client applications deployed on VMs share the physical hardware resources but the implementation details and complexities are abstracted from user applications so that they are not directly tied to the underlying physical hardware. Multitenant virtualized cloud environment where each tenant has an illusion of infinite capacity, could lead to performance unpredictability. One of the major challenges[20] in multi-tenant environment of cloud computing is dynamic provisioning and resource allocation to achieve the desired performance. However from cloud customer's point of view, such a high contention environment can be a problem as its application could be a highly targeted attack victim example when one tenant's I/O requests dominate in a shared I/O network[21], the I/O response times of the other applications may suffer. Noisy neighbour applications put large demands on the server, storage, database or network, thereby, impacting the performance of other neighbour applications that share those resources. A similar situation can arise in a corporate datacenter as well but in this case the user would have control over its internal datacenter. However, in a multi-tenant cloud environment, external users has no control over it. Increasing adoption of virtualization technologies for enterprise applications has brought out the need for establishing a uniform framework for comparing different technologies to identify the performance bottlenecks on the virtualization stack and their impact on different applications. It is necessary to have a test bed[20,43] that helps in analyzing the various components[19] and their associated behaviour.

4. Proposed Research Framework

In this paper, authors have proposed a system design that improves the quality and performance of the services in a cloud by handling the problems caused by noisy neighbours. The figure no. 1 shows the proposed research framework. A cloud has number of physical servers. Each of the server is virtually divided into a number of Virtual Machines. The Virtual machines (VM) are emulated computers running on a hypervisor[33]. A hypervisor virtually partitions the cloud physical resources resources such as memory and processor to the VMs. The hypervisor layer is either on top of physical hardware layer or operating system or software layer. The client jobs on cloud are actually deployed on cloud VMs. However, the underlying resources are shared within the VMs on the cloud. The details of virtualisation are abstracted from the application, giving them an illusion of exclusive access to the resources. Hence, a resource greedy job deployed on a VM may consume more resources on the underlying hardware than its fair share. In a noisy neighbour scenario one or more virtual machines on the physical infrastructure consumes very large amounts of disk I/O thereby impacting the performance of the co-tenant virtual machines.

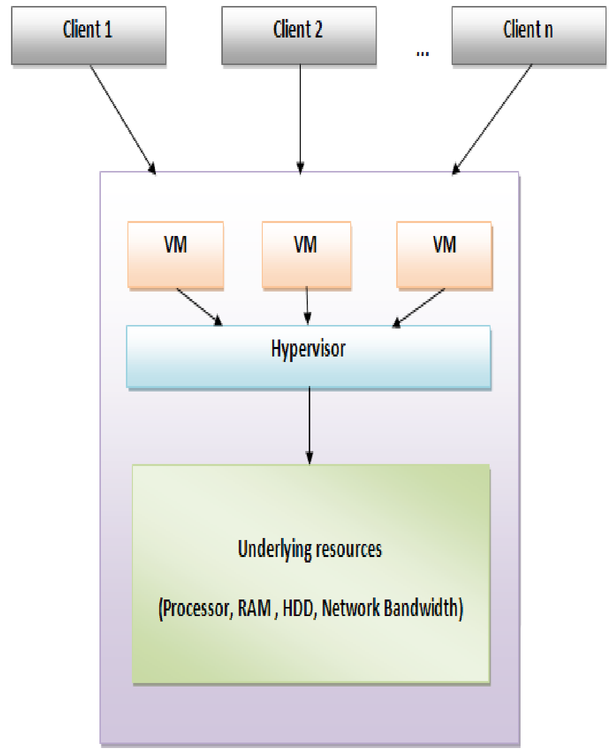

Figure no. 1 Proposed Research Framework

As per the proposed research framework, the cloud tenant applications are categorized into two main categories

- Noisy neighbour- Noisy neighbours are resource greedy applications. In the cloud realm, the term "noisy neighbours[6]" is defined as a cloud computing co-tenant that dominates the bandwidth, disk I/O, and CPU within a multitenant cloud infrastructure, thus negatively impacting the other cloud co-tenants and causing cloud network performance to diminish.
- Victim - Applications with less resource hunger which are impacted by the noisy co-tenants.

The proposed model can be sub-divided in two major tasks:

1. Identify noisy neighbour
2. Deal with the noisy neighbour.

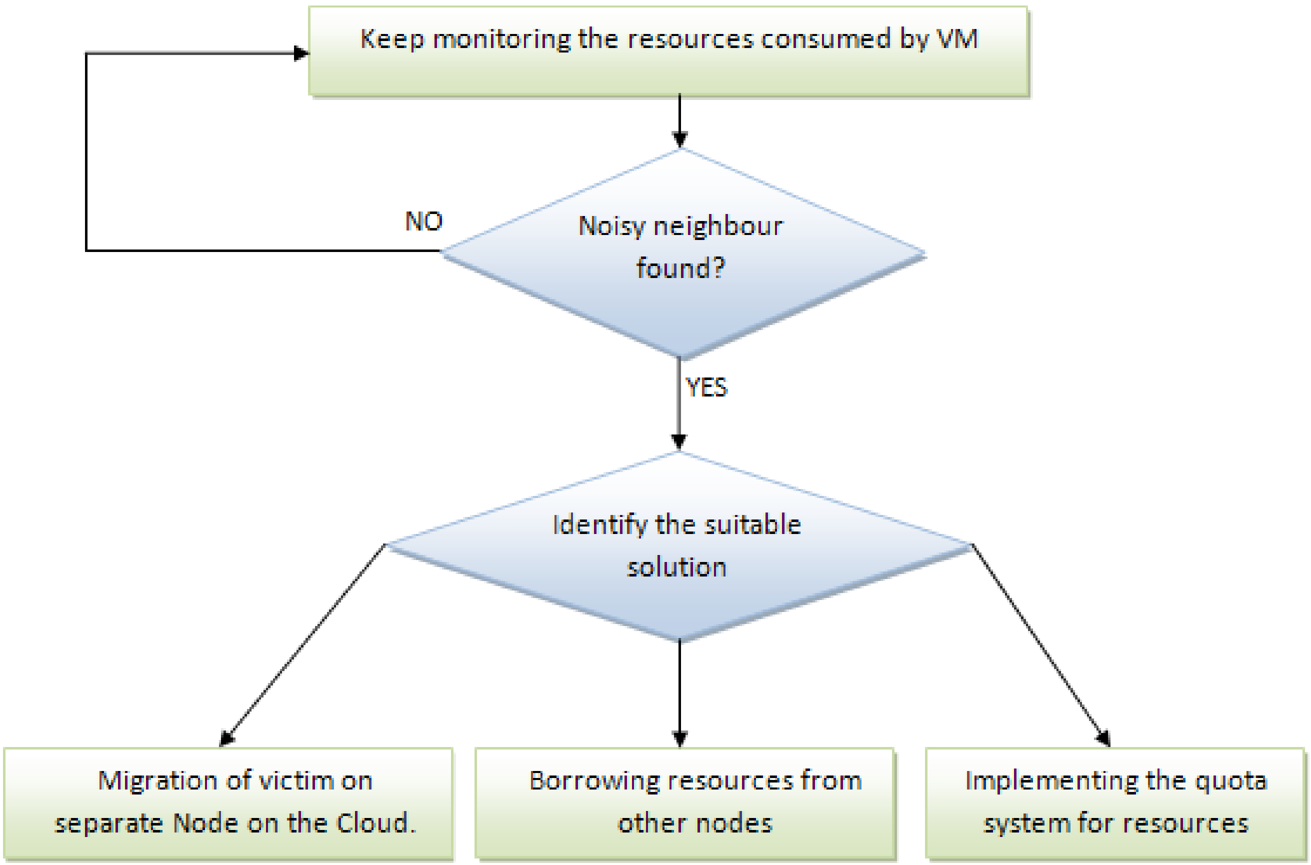

Figure no. 2 Flowchart depicting the high-level view of the proposed system

4.1. Identify the Noisy neighbour

The noisy neighbour introduces the problem of performance unpredictability[1] for the other tenant applications sharing the same physical infrastructure because of its resource greedy nature. So in order to provide QoS[1] to the cloud client, it is necessary to deal with the noisy neighbour and the first step to do this, is to identify the noisy neighbour. The authors have proposed the following approaches to identify a noisy neighbour:

1. Monitoring the cloud applications based on identified parameters
2. Heuristics based approach to classify the applications

The proposed approaches are discussed below in detail.

4.1.1. Monitoring the Cloud Applications based on Identified Parameters

The proposed model is based on monitoring and capturing the following metrics based on the consumption of the system resources by the applications deployed on VMs dynamically to facilitate proactive action.

Table no. 1 Cloud parameters to be measured

S. No.	Parameters to be Measured
	Memory usage
	Network bandwidth consumption
	Storage Usage
	I/O operations
	IO Error
	Memory R/W Error (5xx Errors can occur)

The virtual machines are running on top of a hypervisor deployed on the physical hardware (hosts). The concept of cloud is based on increasing the resources utilization of the physical hardware. So, the hypervisor shares the limited physical resources like CPU cycles, memory, disk bandwidth, network bandwidth etc. In such an environment of resource contention, the problem happens when a guest application must wait for its turn to use the physical resource. This is the "noisy neighbor" problem. To identify a noisy neighbour, the application is monitored continuously to capture the metrics defined in Table 1. When any of the metric crosses the pre-defined threshold a noisy neighbour is identified. Noisy neighbour consumes the memory, network bandwidth, storage and has high volume of I/O operations and impacts the performance of the neighbour applications thereby causing IO Error, Memory R/W Error in other applications.

4.1.2. Heuristics based Approach to Classify the Applications

The historical[39,42] data can help in determining the noisy neighbours. Based on the usage pattern of the application, it can be predicted that an application which has been resource greedy in the past will be resource greedy in the future too. Better visibility into performance heuristics will help in the ongoing management of the cloud applications.

4.2. Dealing with Noisy Neighbour

The noisy neighbour impacts the other tenant applications on shared infrastructure by consuming majority of available resources[40,41] and degrading their performance. So, once the noisy neighbour is identified using one of the discussed approaches, the next step is to deal with the noisy neighbour. The authors have the following approaches to deal with the noisy neighbours:

1. Cloning or migration of smaller applications on separate node on the Cloud.
2. Borrowing resources from other nodes.
3. Implementing quota system for resources

The proposed approaches are discussed in detail below.

4.2.1. Cloning or Migration of Victim Applications on Separate Node on the Cloud

In the proposed Victim migration policy, migration of the resource starved victim applications is triggered when a noisy neighbour is identified in a shared physical infrastructure. We chose the victim application rather than noisy neighbour for migration because of the low migration overhead. The new home node chosen for the application can be based a number of parameters like physical proximity, performance, current resource utilisation etc.

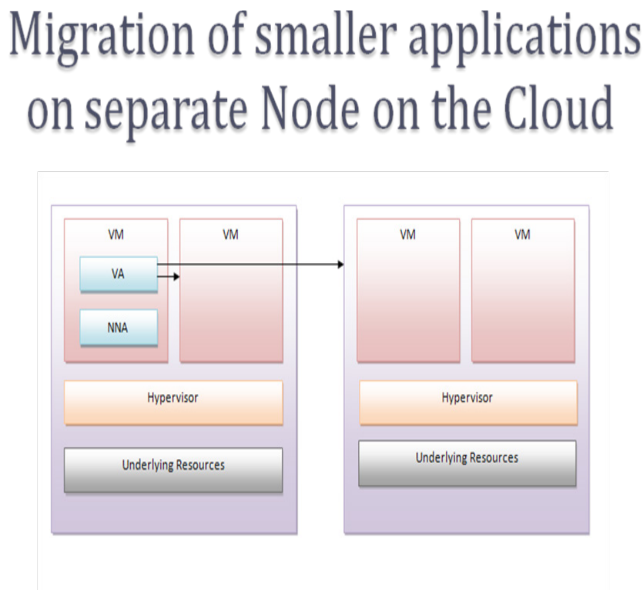

Figure no. 3 Migration of victim applications on separate node on the cloud

A lot of research has been done in the area of VM migration and a number of VM migration policies have been proposed. The VM migration can be done for a number of different reasons such as such as improving the power efficiency, satisfying performance requirements, improving the network performance and communication costs. There are a number of challenges around VM migration across cloud nodes. An application has a large volumes of data. So, an application migration across a node would involve transfer of the associated large volumes (typically 1-30GB in size) over limited shared bandwidth. This can degrade the network performance as VM migration may consume significant amount of network bandwidth. Also, multi-tier application with data layer, user interface layer, business logic layer can have its layers deployed on separate VMs. In the case of such applications, the complexity of migration would increase. A number of approaches for live VM migration have been proposed. The live migration is carried out in two phases which include switching the control to the destination node and transfer of data to the destination node. Based on which phase is carried out first, the live migration can be broadly classified into two types, pre-copy and post-copy. In the pre-copy VM migration, suited for read intensive workloads, the memory is migrated first followed by execution migration. In the post-copy VM migration, suited for write intensive workloads, the order of the above mentioned phases is reversed. The first phase of pre-copy VM migration memory invloves transfer of RAM contents iteratively such that in subsequent iterations, the dirtied content since previous iteration are transferred. The next phase of pre-copy involves transfer of application execution from source to the destination. This is achieved by stopping the execution of application instance on the source, copying its current state viz. CPU state, registers, virtual devices state, last memory pages etc to the destination and then resuming the application instance execution on the destination. The post-copy migration37,38 also follows a similar process but reverses the order of phases. The first phase of post-copy involves transfer of application execution from source to destination followed by the second phase which involves memory transfer. On-demand paging is used to transfer the data from source to destination node thereby causing page faults, degradation of performance and increase in VM migration time. Clark et al.22 have proposed a live VM migration tool which is capable of migrating live VMs between LAN-connected nodes. The model is based on the assumption that a VMs persistent image is stored on a shared network-accessed storage system. A similar model for VM migration on LAN connected platforms was proposed by Lagar-Cavilla et al. 30 based on cloning live VMs. The proposed mechanism also assumes the source and destination share a copy-on-write storage system storing the VM images. Sapuntzakis et al.23 proposes a system for virtual machine live migration that tracks the similarities between the master image and the transferred image to significantly reduce the volumes of data transfer. Hirofuchi et al.24 proposes a post-copy VM live migration mechanism that prioritises the recently accessed data blocks for migration over others. On the other hand, Bradford et al proposes a post-copy migration technique of VM disk image transfer at the same time as VM's live in-memory state transfer.

4.2.2. Borrowing Resources from other Nodes

When a noisy neighbour, impacts the other tenant applications by consuming most of the available resources, there is a performance degradation of the other resource starved applications. In such a case of massive over-provisioning, the resources can be borrowed from other nodes.

One of the most used resource sharing methods for computational grids currently is Beowulf clusters computers that uses Beowulf Allocator as[34] a resource allocator

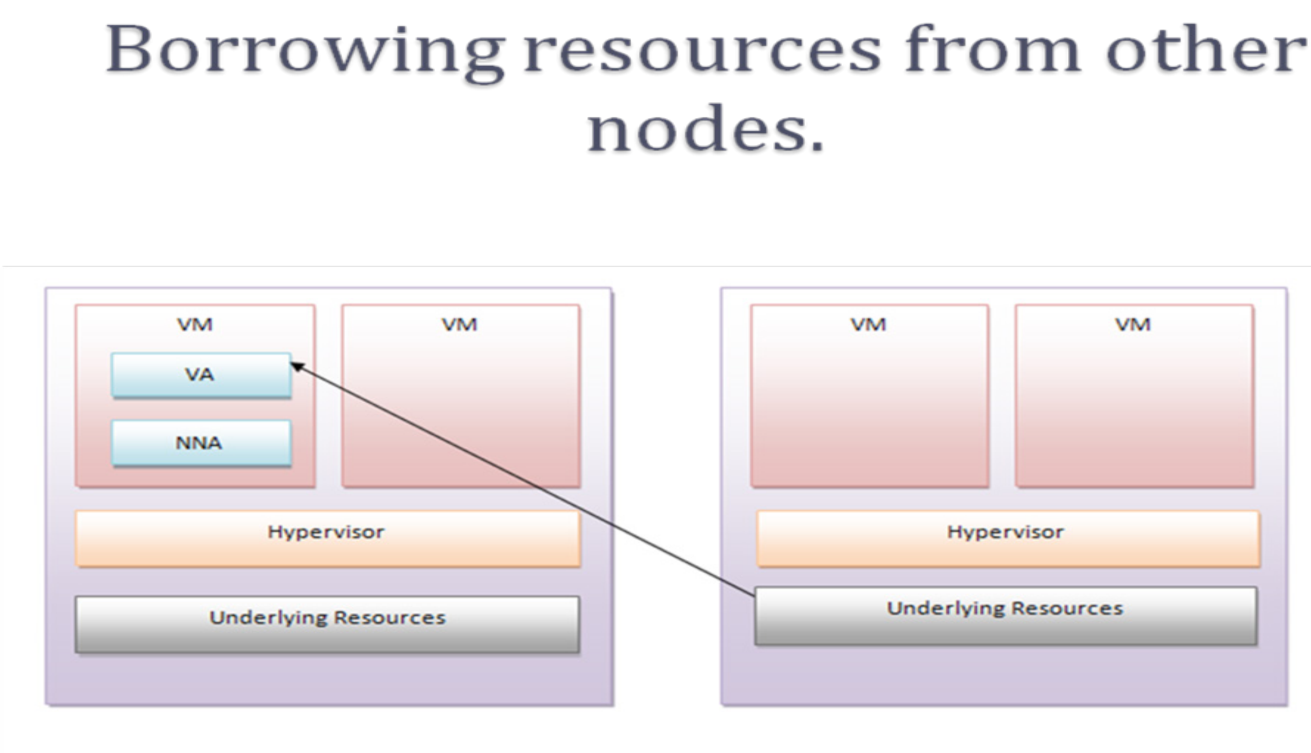

Figure no. 4 Borrowing resources from other nodes that allocates resources on any cluster in the grid. In case, there are not enough resources on the cluster, it automatically borrows resources from other clusters in the grid. SETI@home[25] ("SETI at home"), a cycle stealing application is another example of resource sharing.

It is an Internet-based public volunteer computing project hosted by the Space Sciences Laboratory, at the University of California, Berkeley, in the United States making use of processor time that would otherwise be unused. Grid Infrastructures that use meta-schedulers like Globus[35], Condor[35] , InVIGO[36] and PUNCH [28] provide access to unlimited computation resources without incurring the full cost of ownership. Some systems which allows dynamic sharing (by borrowing and lending) of resources between multiple clusters are VioCluster[27], Cluster-OnDemand[27].

4.2.3. Quota system for Resource Allocation in Cloud

We can define quota of computing resources like storage, network band etc. that restricts applications from using resources of other applications in use. This would not completely ensure that resource greedy applications cannot take over the resources used by smaller applications with less resource hunger but increase the probability of this happening less. Resource quotas are helpful to allocate, track and limit resource utilisation. Cloud quota system would keep a track on tenant application's usage of system resources, allowing the cloud administrator to set limits on the usage of these resources. Quota limits can be set per application. Allocated resource quotas for users and groups with quota management features would also help control resource allocation across clusters, defined by users and groups and analyse cloud usage patterns. Windows Azure[28] web sites imposes quotas on computing resources like File System/Storage, Egress Bandwidth, CPU Time, Memory and Database at data-center and regional level. Windows Azure portal provides a dashboard that helps users to track their usage an diagnose future usage. OpenNebula[29] has a powerful and extensible built-in monitoring subsystem and resource quota management to allocate, track and limit resource utilisation.

5. Experimental Setup

The proposed system has been built using component based software development techniques that can be extended and enhanced in future. The private cloud environment has been setup using Ubuntu Enterprise Cloud (UEC)[31] and Eucalyptus[32]. UEC is a stack of applications from Canonical included with Ubuntu Server Edition. UEC includes Eucalyptus along with a number of other open source software. Eucalyptus is an open source software that provides a platform for creation and management of private/public cloud.

Therefore, the scope of our work is to test the designed approach for dealing with noisy neighbour in a cloud environment. To test the designed approach, the first step is to create a cloud environment testbed with noisy neighbours deployed on it which causes resource starvation for other deployed applications. In order to test the proposed design, the authors have created a Java based application deployed on private cloud setup to monitor the defined metrics. The authors have setup the environment using Linux Shell having 3 Machines with i3 Processor and 8 GB RAM and 500 GB storage and Hypervisor KVM, Database MySQL.

Table 2. Tools and technologies used

Operating System	**Ubuntu Server Edition , Ubuntu Desktop Edition**
Cloud Management Suite	Eucalyptus
Programming Languages	J2EE ,XHTML, JavaScript, XML , Shell Script
Database	MySQL
Virtualization Software	KVM

The first step to deal with a noisy neighbour is to identify a noisy neighbour. The authors have developed a Java based application that can be deployed on a cloud based infrastructure that provides the following capabilities

- Monitoring the cloud applications based on identified parameters
- Heuristics based approach to classify the applications

The application would continuously monitor the applications deployed on cloud based infrastructure to the metrics viz. Memory usage, Network bandwidth consumption, Storage Usage, I/O operations, IO Error, Memory R/W Error (5xx Errors) to facilitate proactive action and to persist the metrics based on time stamp for each application to help in future guidance

6. Conclusion and Future Scope

Cloud computing has made it possible for enterprises to outsource their computing resources requirement in a more economic and flexible manner. A consequence of multitenant model of cloud is complete lack of performance predictability because of resource contention. In this paper, the authors have proposed a model to monitor the resources consumed by cloud hosted applications, identify and deal with noisy neighbour and victim applications. This would help in actively and proactively deal with noisy neighbours and reduce their affect on the other tenant applications on a shared infrastructure Monitoring the resource consumption of the applications would help categorize the applications based on their resource consumption pattern. This would help to focus research to improve the QoS of a cloud by moving in the direction of analysis of behaviour . In future, authors plan to do a comparative analysis of the proposed methodology and incorporate the security and fault tolerance mechanisms features in the same. Authors also plan to carry out a comparative analysis of advanced migrating techniques and use them to enhance the proposed system.

7. References

1. Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, Zaharia M. A view of cloud computing. Communications of the ACM. 2010 Apr 1; 53(4):50–8.
2. Vaquero LM, Rodero-Merino L, Caceres J, Lindner M. A break in the clouds: towards a cloud definition. ACM SIGCOMM Computer Communication Review. 2008 Dec 31; 39(1):50–5.
3. Bezemer CP, Zaidman A. Multi-tenant SaaS applications: maintenance dream or nightmare? Proceedings of the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE), ACM. 2010 Sep 20; 88–92.
4. Mahajan K, Makroo A, Dahiya D. Round robin with server affinity: a VM load balancing algorithm for cloud based infrastructure. Journal of information processing systems. 2013; 9(3):379–94.
5. Rochwerger B, Vazquez C, Breitgand D, Hadas D, Villari M, Massonet P, Levy E, Wolfsthal Y, Nagin K, Galis A, Llorente IM. An architecture for federated cloud computing. Cloud Computing. 2010 Apr: 391–411.
6. Available from: https://www.linkedin.com/pulse/oracle-cloud-engineering-erp-architecture-21st-century-dennis-jolluck as on 2,March,2016
7. Jeyakumar V, Alizadeh M, Mazieres D, Prabhakar B, Kim C. EyeQ: Practical network performance isolation for the multi-tenant cloud. In Presented as part of the 2012 Jun.
8. Dikaiakos MD, Katsaros D, Mehra P, Pallis G, Vakali A. Cloud computing: Distributed internet computing for IT and scientific research. Internet Computing, IEEE. 2009 Sep; 13(5):10–3.
9. Li J, Guo R, Zhang X. Study on service-oriented Cloud conferencing. 2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), IEEE. 2010 Jul 9; 6:21–5.
10. Mell P, Grance T. The NIST definition of cloud computing.
11. Shen Z, Subbiah S, Gu X, Wilkes J. Cloudscale: elastic resource scaling for multi-tenant cloud systems. Proceedings of the 2nd ACM Symposium on Cloud Computing, ACM. 2011 Oct 26; 5.
12. Grobauer B, Walloschek T, Stocker E. Understanding cloud computing vulnerabilities. Security and privacy, IEEE. 2011 Mar; 9(2):50–7.
13. Zhang Q, Cheng L, Boutaba R. Cloud computing: state-of-the-art and research challenges. Journal of internet services and applications. 2010 May 1; 1(1):7–18.
14. Zhang L, Luo Y, Tao F, Li BH, Ren L, Zhang X, Guo H, Cheng Y, Hu A, Liu Y. Cloud manufacturing: a new manufacturing paradigm. Enterprise Information Systems. 2014 Mar 4; 8(2):167–87.
15. Barham P, Dragovic B, Fraser K, Hand S, Harris T, Ho A, Neugebauer R, Pratt I, Warfield A. Xen and the art of virtualization. ACM SIGOPS Operating Systems Review. 2003 Dec 1; 37(5):164–77.
16. Fenn M, Murphy MA, Martin J, Goasguen S. An evaluation of KVM for use in cloud computing. Proceedings of 2nd International Conference on the Virtual Computing Initiative, RTP, NC, USA. 2008 May.
17. Zhang H, Jiang G, Yoshihira K, Chen H, Saxena A. Intelligent workload factoring for a hybrid cloud computing model. 2009 World Conference on Services-I, IEEE. 2009 Jul 6; 701–8.
18. Yang X, Nasser B, Surridge M, Middleton S. A business-oriented Cloud federation model for real-time applications. Future Generation Computer Systems. 2012 Oct 31; 28(8):1158–67.
19. Lakshmi J. System Virtualization in the Multi-core Era-a QoS Perspective (Doctoral dissertation, Ph. D. dissertation,

Supercomputer Education and Research Center, Indian Institute of Science).
20. Wang X, Liu Z, Li J, Yang B, Qi Y. Tualatin: Towards network security service provision in cloud data centers. 2014 23rd International Conference on Computer Communication and Networks (ICCCN), IEEE. 2014 Aug 4; 1–8.
21. Shue D, Freedman MJ, Shaikh A. Performance isolation and fairness for multi-tenant cloud storage. Presented as part of the 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI 12); 2012; 349–62.
22. Al-Kiswany S, Subhraveti D, Sarkar P, Ripeanu M. VMFlock: virtual machine co-migration for the cloud. Proceedings of the 20th International Symposium on High Performance Distributed Computing, ACM. 2011 Jun 8; 159–70.
23. Sapuntzakis CP, Chandra R, Pfaff B, Chow J, Lam MS, Rosenblum M. Optimizing the migration of virtual computers. ACM SIGOPS Operating Systems Review. 2002 Dec 31; 36(SI):377–90.
24. Hirofuchi T, Ogawa H, Nakada H, Itoh S, Sekiguchi S. A live storage migration mechanism over wan for relocatable virtual machine services on clouds. Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, IEEE Computer Society. 2009 May 18; 460–5.
25. Anderson DP, Cobb J, Korpela E, Lebofsky M, Werthimer D. SETI@ home: an experiment in public-resource computing. Communications of the ACM. 2002 Nov 1; 45(11):56–61.
26. Adabala S, Chadha V, Chawla P, Figueiredo R, Fortes J, Krsul I, Matsunaga A, Tsugawa M, Zhang J, Zhao M, Zhu L. From virtualized resources to virtual computing grids: the In-VIGO system. Future Generation Computer Systems. 2005 Jun 30; 21(6):896–909.
27. Ruth P, McGachey P, Xu D. Viocluster: Virtualization for dynamic computational domains. 2005 IEEE International In Cluster Computing, IEEE. 2005 Sep; 1–10.
28. Available from: https://dzone.com/articles/azure-websites-quotas-scaling as on 2,March.2016
29. Staring AJ, Karagiannis G. Cloud computing models and their application in LTE based cellular systems. 2013 IEEE International Conference on Communications Workshops (ICC), IEEE. 2013 Jun 9; 750–5.
30. Satyanarayanan M, Bahl P, Caceres R, Davies N. The case for vm-based cloudlets in mobile computing. Pervasive Computing, IEEE. 2009 Oct; 8(4):14–23.
31. Wardley S, Goyer E, Barcet N. Ubuntu enterprise cloud architecture. Technical White Paper. 2009 Aug.
32. Available from: https://help.ubuntu.com/community/UEC?action=show&redirect=Eucalyptus as on 2,March,2016
33. Voorsluys W, Broberg J, Venugopal S, Buyya R. Cost of virtual machine live migration in clouds: A performance evaluation. In Cloud Computing, Springer Berlin Heidelberg. 2009 Dec 1; 254–65.
34. Hu F, Evans JJ. Power and environment aware control of Beowulf clusters. Cluster Computing. 2009 Sep 1; 12(3):299–308.
35. Foster I, Kesselman C, Nick JM, Tuecke S. Grid services for distributed system integration. Computer. 2002 Jun; 35(6):37–46.
36. Adabala S, Chadha V, Chawla P, Figueiredo R, Fortes J, Krsul I, Matsunaga A, Tsugawa M, Zhang J, Zhao M, Zhu L. From virtualized resources to virtual computing grids: the In-VIGO system. Future Generation Computer Systems. 2005 Jun 30; 21(6):896–909.
37. Hines MR, Gopalan K. Post-copy based live virtual machine migration using adaptive pre-paging and dynamic self-ballooning. Proceedings of the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, ACM. 2009 Mar 11; 51–60.
38. Uddin M, Memon J, Alsaqour R, Shah A, Abdul Rozan MZ. Mobile Agent based Multi-layer Security Framework for Cloud Data Centers. Indian Journal of Science and Technology. 2015; 8(12):1–10.
39. Hussain Madni SH, Abd Latiff MS, Coulibaly Y, Abdulhamid SM. An Appraisal of Meta-Heuristic Resource Allocation Techniques for IaaS Cloud. Indian Journal of Science and Technology. 2016; 9(4):1–14.
40. Shyamala K, Sunitha Rani T. An Analysis on Efficient Resource Allocation Mechanisms in Cloud Computing. Indian Journal of Science and Technology. 2015; 8(9):814–21.
41. Durairaj M, Manimaran A. A Study on Security Issues in Cloud Based E-Learning. Indian Journal of Science and Technology. 2015; 8(8):757–65.
42. Sugumar R, Sheik Imam SB. Symmetric Encryption Algorithm to Secure Outsourced Data in Public Cloud Storage. Indian Journal of Science and Technology. 2015; 8(23):1–5.
43. Mahajan K, Dahiya D. A cloud based deployment framework for load balancing policies. 2014 Seventh International Conference on Contemporary Computing (IC3). 2014 Aug 7; 565–70. IEEE.