



## An Intelligent Recommender System for Cloud Usage Data Using Predictive Analysis

Prof.M.B.Lonare<sup>1</sup>, Amit Kumar<sup>2</sup>, Krishna Chaitanya<sup>3</sup>, Shivam Kumar Rathee<sup>4</sup>, Yuvraj Singh<sup>5</sup>

<sup>1,2,3,4,5</sup>Department of Computer Engineering, Army Institute of Technology, Pune, India

---

**Abstract**—Cloud computing allows tenants to rent resources in a pay as-you-go fashion. It offers the potential for a more cost effective solution than in-house computing by obviating the need for tenants to maintain complex computing infrastructures themselves. To achieve this benefit, the right amounts of computing resources need to be given to the applications running in the cloud. The amount of resources needed is rarely static, varying as a result of changes in overall workload, the workload mix, and internal application phases and changes. To avoid problems, the amount of resources allocated to applications should be adjusted dynamically, which brings two main challenges: (1) deciding how much resource to allocate is non-trivial since application resource needs often change with time and characterizing runtime application behaviour is difficult; (2) application resource needs must be predicted in advance so that the management system can adjust resource allocations ahead of the needs. Furthermore, resource-management systems should not require prior knowledge about applications, historical data such as application behaviour profiles, and running the resource management system itself (including its prediction algorithms) should not be costly.

---

**Keywords--** Cloud Computing, OpenStack, Data Analysis, DevStack, Linear Regression, Nova, Cinder

### I. INTRODUCTION

Cloud computing enables provisioning the user with a utility of cloud which might be a platform, software or an infrastructure as whole. Varying infrastructural and service tools may significantly impact the performance of the cloud, its overall usage and in turn, the cost an industry is expending out to purchase that cloud. Monitoring such changes is essential for the analysis of the relationship between the usage of the cloud and its users. Monitoring infrastructural resources is essential for the building up of frameworks that enables Service Level Agreements based on applications QoS requirements. Unless performance guarantees at the level of hardware resources like CPU, Memory and I/O Devices are not given, it becomes mandatory to have necessary monitors in place for the infrastructural resources. Both Cloud provider and clients are the beneficiaries of resource monitoring. Cloud providers have to monitor the current status of allocated resources in order to handle future requests from their users efficiently and to keep an eye on malicious users by identifying anomalous behaviour. The analysis of this data would help the client to buy the cloud with the optimised value of data required for the operations and cost cutting operation would be successfully implemented. The advent of cloud computing in the 20th century, initially lead to a small scale of transfer and sharing data between various users. Soon as the time passed by, storage and security of the cloud became the most important aspects to be taken care of and many industries started using clouds for their information storage. The concept of storing data in a remote location was new to the world, and this technology was supported by construction of huge data centres underground or in buildings, supported with high transfer rates of data using fibre optics. The industries today, use data centres which constitute of millions of hard disks storage and store trillion bytes of data. On a local level, whenever a person purchases a cloud worth a particular cost, the usage is limited to his or her needs. Sometimes the user requires fewer amounts for storage and sometimes more. The irregular use of data leads to the excess purchasing of the cloud which directly suggests that a user might be purchasing a higher amount but is using it less.

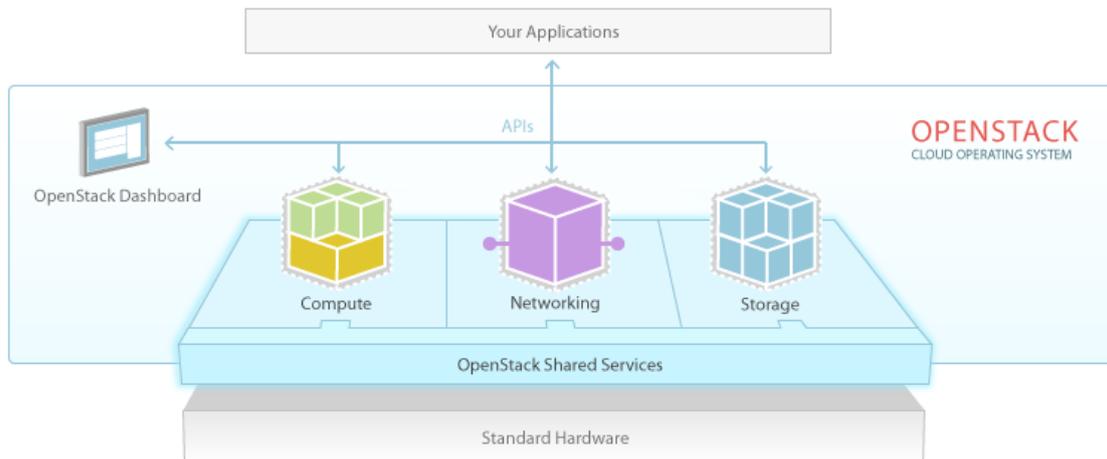
### II. PROBLEM DESCRIPTION

In a Cloud, hosted applications such as a multi-tier websites may run on group of VMs that span multiple physical hosts. These VMs form a resource pool. Due to initial placement and load balancing, the actual deployment of these VMs can show an arbitrary topology on physical nodes. As the number of VMs increase, the cloud infrastructure is divided into sub clusters, each of which is responsible for resource allocation of one application. In this dissertation, we aim to design, implement and evaluate a resource management mechanism that indicates the user usage of the cloud over the period of time using metrics provided by OpenStack and Ubuntu and using data analysis provide the user an optimal solution. The goal of this project is to create an intelligent resource management scheme in a cloud platform. Uses the combined cloud utilization data, we want to create a usage model for a recommender system. Using this recommender system, the relevant modification is carried on the specific instance. A resource utilization report is presented to the user along with the model recommendations. The analysis step then uses a machine learning algorithms to make an Online Prediction Model. Using the prediction model, the needs of the particular instance are extrapolated and the necessary modifications are made for the resource allocation. Make a front-end for the user to manage his resource requirements based on the recommendations by the usage records.

### III. LITERATURE SURVEY

#### 3.1 OpenStack

OpenStack is open source cloud computing software that provides infrastructure as a service cloud deployment for public and private cloud. OpenStack provides a significant alternative for organizations which don't wish to go for commercially provided cloud as OpenStack is open source. OpenStack is organized around three main modules i.e. compute, storage and networking. Along with these three, dashboard become an important component in providing interface to administrators and users for provisioning and release of resources. OpenStack network provides necessary services which are used for communication with in virtual machine i.e. inter-VM and external to virtual machines.



**Figure 1. OpenStack Architecture**

OpenStack currently consist of seven different service code projects to make it modular, each having its different code name for project. The seven OpenStack's projects along with their respective code name are listed in **table 1**

Table 1. OpenStack Projects

<i>S.No.</i>	<i>Project</i>	<i>Name</i>
1	Compute	Nova
2	Object Storage	Swift
3	Identity	Keystone
4	Dashboard	Horizon
5	Block Storage	Cinder
6	Network	Quantum
7	Image Service	Glance

**Compute** (codenamed Nova): OpenStack compute originated from NASA's Nebula platform. Compute provides the service for provisioning and un-provisioning of virtual machines on demand basis. Many companies provide compute services commercially bases on OpenStack's compute moreover, compute is used internally by many organizations including where it originated i.e. NASA. Nova is main core part of OpenStack essential for IaaS. It is responsible for running virtual servers. It is used to host and manage cloud computing systems. Nova originated at NASA Ames Research Laboratory. Its design is built keeping in mind the ease of adding new features, scaling with workload, fault tolerance and compatibility of its APIs with other systems like Amazon EC2.

#### 3.2 Linear Regression

Linear regression is a technique used for prediction and forecasting. Its main purpose is to find dependency between a dependent and one or more independent variables. It predicts value of dependent variable for given values of independent variable. Assumptions are made before applying linear regression and these include constant variance i.e. almost same error in values and linearity dependency which is that a linear change is observed in dependent

variable when a linear change is introduced in independent variables.

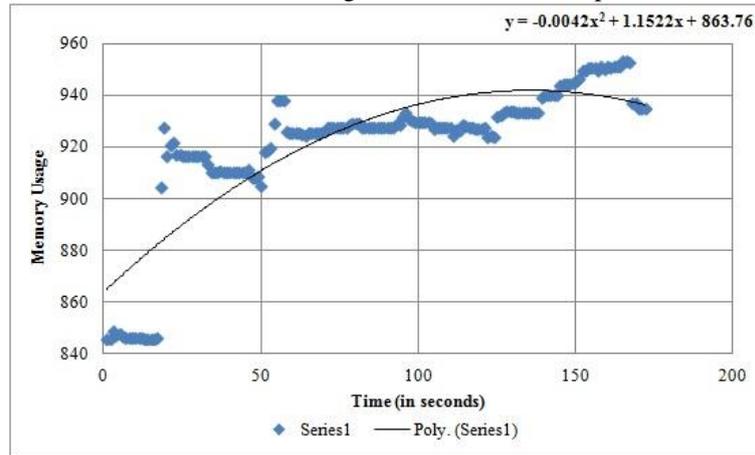


Figure 2. Regression equation between Memory usage and Time

#### IV. EXPERIMENTAL SETUP

This project requires very high hardware support which has atleast 12GB of RAM, xecore processors and storage of more than 1TB; this is minimum requirement of hardware required to implement this project. On a newly installed stable version of any Linux flavor, example install Ubuntu 14.04LTS and then to install OpenStack on a single system. We have to created a separate user and list this user in sudoers and provide it with all the privileges of super user without setting any password, we will then login to that newly created user and later installed git in this system. Further we will fork the DevStack project from github. DevStack is a series of scripts which when executed on a Linux system installs all the required softwares for bringing up the OpenStack environment with their latest version from the git master. Before running the script present is DevStack we need to set passwords for various application required for OpenStack such as administrative password, mysql password, rabbit MQ password, service password used by keystone for authentication, then run the script it will setup OpenStack environment in an hour or half with the installation in final phase it will show with various IP for accessing services provided by OpenStack such as horizon dashboard, for keystone etc.

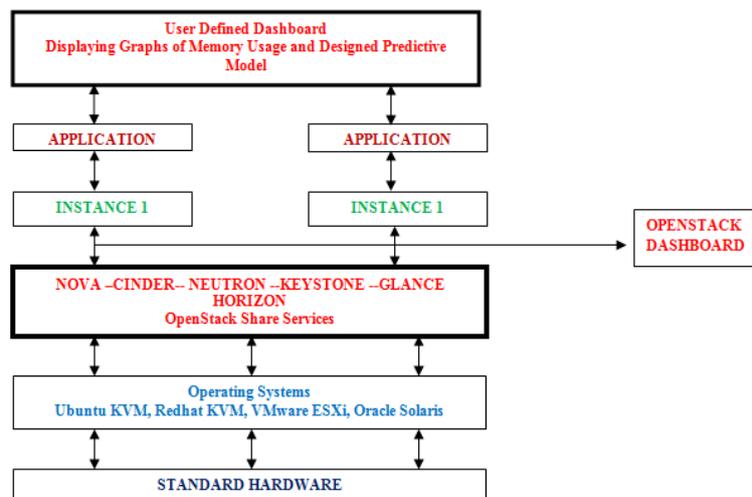


Figure 3. SYSTEM ARCHITECTURE

#### V. IMPLEMENTATION

The project is implemented on OpenStack platform, which we have OpenStack on a new installed Linux system in order to record resource usage by instances running on OpenStack we launched three instances on OpenStack for that firstly we

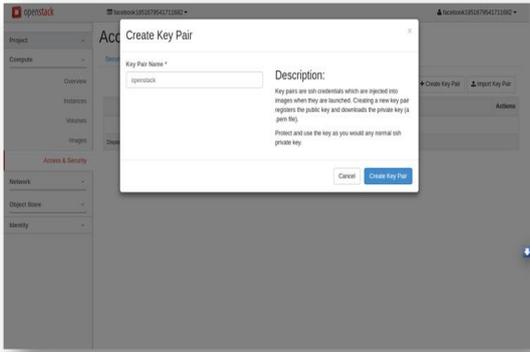


Figure 4. Key Pair Generation

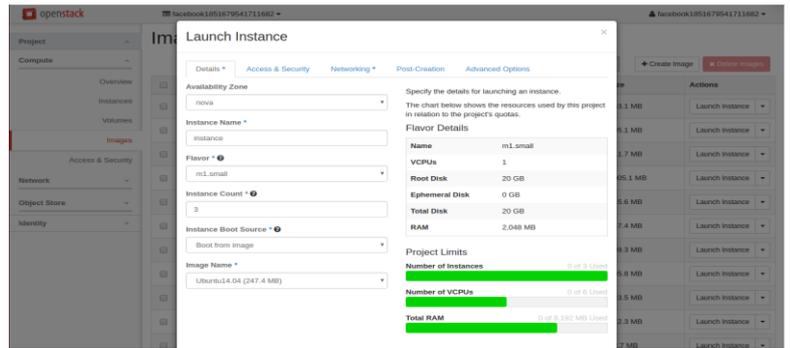


Figure 5. Launching an instance

Created a key pair so that we can access our instances from any terminal connected to internet if we have the private key with us .Further, created an internal network for our instances after that launched our three instances . The access rules and security rules for the instances were defined. The connection to one of the instance using terminal on the local system was established with ssh connection and private key that was downloaded earlier and in one of the Ubuntu instances running on OpenStack , various programs such as of matrix multiplication, minimum spanning tree, regex were executed.

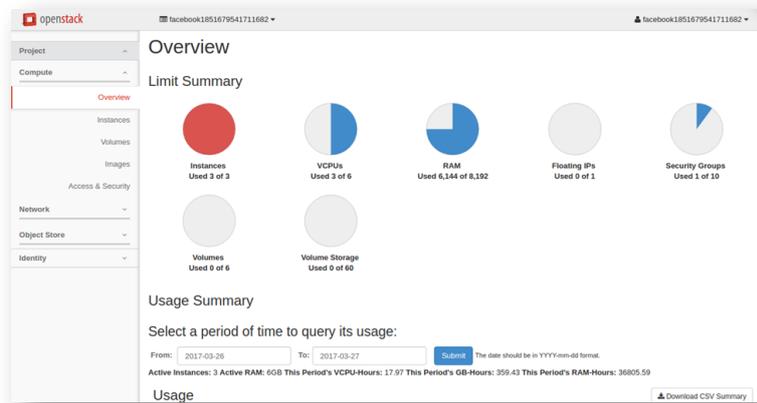


Figure 6. OpenStack dashboard - Horizon

The resource usage of that instance was extracted by running a bash script on that instance. After getting the numerical values of resource usage of that system, linear regression graph of cpu usage with time and memory usage with time was plotted and further using linear regression, a predictive model was constructed separately for cpu usage and memory usage.

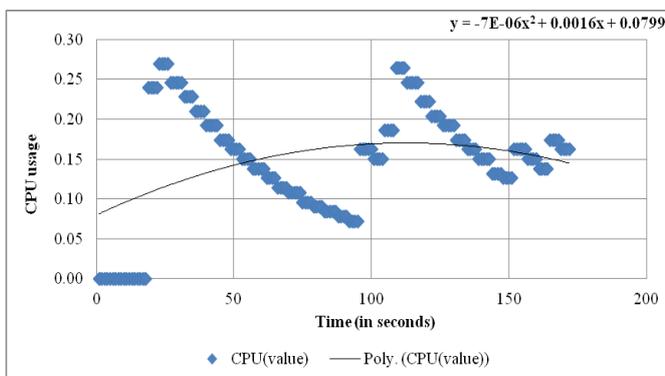


Figure 7. CPU usage vs Time

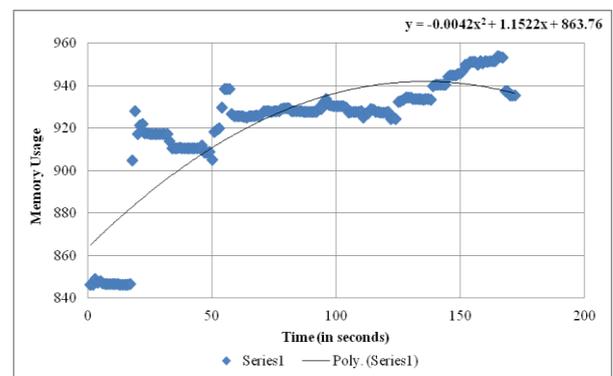


Figure 8. Memory usage vs Time

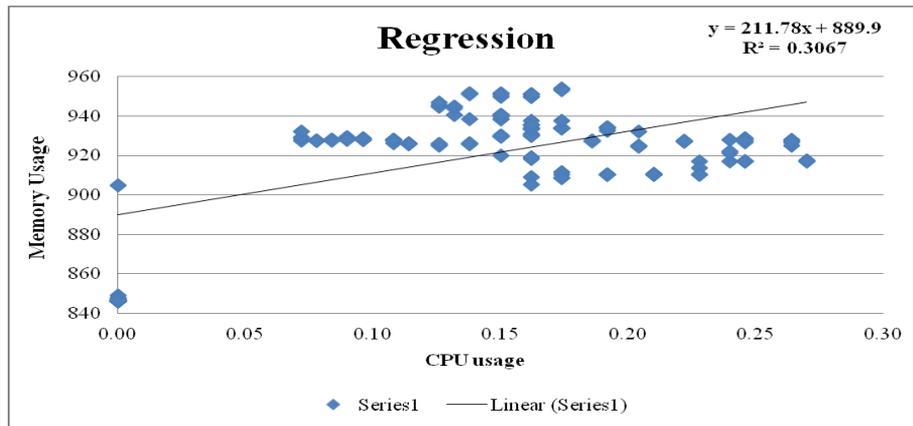


Figure 9. Regression model of Memory and CPU

The following statistics are generated once the regression model is applied . Using the values of particular item sets such as Standard Error, Residual (also known as degree of freedom), the prediction is found at a particular value of .25 CPU usage and the corresponding value is generated i.e. 942.848. The lower bound and upper bound values are also generated which are the recommendal values to the user so that he or she can buy the cloud for this memory usage.

Regression Statistics							
Multiple R	0.554						
R Square	0.307						
Adjusted R Square	0.303						
Standard Error	22.600						
Observations	172						
ANOVA							
	df	SS	MS	F	Significance F		
Regression	1	38412.401	38412.401	75.204	3.2773E-15		
Residual	170	86832.024	510.777				
Total	171	125244.425					
	Coefficients	Standard Error	t Stat	P-value	Lower 95%	Upper 95%	
Intercept	889.903	3.999	222.532	1.5552E-211	882.009	897.797	
X Variable 1	211.780	24.421	8.672	3.2773E-15	163.573	259.988	
Prediction (desired) 95.00%	Alpha value = 5.00%				Mean(CPU) = 0.15		
					Std. Dev(CPU) 0.071		
Regression equation is = 889.903+211.780*CPU							
Prediction find at	Point prediction	T-value	St. Error	Margin of error	Lower bound	Upper bound	Interval width
0.25	942.848	1.974	22.803	45.014	897.835	987.862	90.027

## VI. REFERENCES

1. Niloofar Khanghahi, Reza Ravanmehr, 2011, Performance Monitoring and Analysis of Cloud Computing Environment. Available at <http://ieeexplore.ieee.org/document/6998759>
2. Yi Han, Jeffrey Chan, Christopher Leckie, 2013, Department of Computing and Information Systems, University of Melbourne. Analysing Virtual Machine Usage in Cloud Computing. Available at <http://ieeexplore.ieee.org/document/6655723>
3. V. Goswami, S. S. Patra, G. B. Mund, 2012, Performance Analysis and Optimal Resource Usage in Finite Population Cloud Environment. Available at <http://ieeexplore.ieee.org/document/6449902>
4. Mohammadreza Mesbahi, Amir Masoud Rahmani, 2016, Load Balancing in Cloud Computing: A State of the Art Survey. Available at <http://www.mecs-press.org/>
5. V.Vinothina, Dr.R.Sridaran, Dr.Padmavathiganapathi, 2012, A Survey on Resource Allocation Strategies in Cloud Computing. Available at [www.ijmlc.org/papers/382-C0068](http://www.ijmlc.org/papers/382-C0068)
6. Swapnil M Parikh, 2013, A Survey on Cloud Computing Resource Allocation Techniques. Available at [www.ijmlc.org/papers/6780076](http://www.ijmlc.org/papers/6780076)
7. Siqi Nie, Qiang Ji, 2014, Feature Learning Using Bayesian Linear Regression Model. Available at <http://doi.ieeecomputersociety.org/10.1109/ICPR.2014.267>

8. Weijia Song, Zhen Xiao , Qi Chen , 2013, Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment. Available at <http://doi.ieeecomputersociety.org/10.1109/TPDS.2012.283>
9. Rohit Kamboj, Anooa AryA, 2014, OpenStack: Open Source Cloud Computing IaaS Platform. Available at <https://www.ijarcsse.com/docs/papers/Volume 4/5 May2014/V4I0745.pdf>