IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-6, Issue-3 E-ISSN: 2347-2693

Efficient VM Allocation to Enhance Performance in Virtualized Cloud Environment

P.D. Pariyani[1*],H.B. Patel[2], B. Shrimali [3]

Dept. of Computer Engineering, LDRP-ITR, Gandhinagar, Gujarat, India
Dept. of Computer Engineering, LDRP-ITR, Gandhinagar, Gujarat, India
Dept. of Computer Engineering, LDRP-ITR, Gandhinagar, Gujarat, India

[*]Corresponding Author: poojapariyani18march@gmail.com, Mob.: 9726937693

Available online at: www.ijcseonline.org

Received: 12/Feb//2018, Revised: 20/Feb2018, Accepted: 15/Mar/2018, Published: 30/Mar/2018

Abstract— Cloud computing being one of the most progressive fields in computer science, there is a constant need of bringing about changes and advancements in the issues affecting the cloud computing applications. Such issues affecting the cloud computing applications include energy efficiency, improper utilization of resources, security and many more. so still found research gap in current technology like time, load distribution, balancing etc. So using proposed model Our work focuses on real time data cloud balancing, speed and data distribution etc. and also consider the enhancing the performance parameter of cloud computing applications. These barriers concern various levels such as virtualization, performance modeling, deployment, and monitoring of applications on virtualized IT resources. Finally, we analyze the effect of load balancing frequency, problem size, and computational granularity on the performance and scalability of our techniques.

Keywords— Cloud computing, Performance, Virtualization Component.

I. INTRODUCTION

Cloud computing has established itself as adequate means of providing resources on demand. The main benefits of cloud computing are the cost savings through its "pay-per-use " model, low investment costs and its rapid implementation of innovations. For cloud computing, the quality and reliability of the services become an important aspect, as customers have no direct influence on the services. In addition, virtualization support in cloud allows better flexibility and customization to specific application, software, and programming environment needs of HPC (High Performance Computing) users.HPC applications are typically tightly-coupled, and perform frequent inter-processor communication and synchronization.

In computing, virtualization means to create a virtual version of a device or resource, such as a server, storage device, network or even an operating system where the framework divides the resource into one or more execution environments. Types of Virtualization Technologies Full virtualization: In full virtualization, VMM will give the image of physical machine to the each VM request. In this way it executes the unmodified OS to execute the privileged instructions. Para virtualization: In Para virtualization, each VM has an abstraction of the hardware that is similar but distinguishable to the underlying physical hardware. Guest operating systems are modified to execute VMs. As a result, the guest operating systems are executing on a VM to provide a near-native performance. OS-level Virtualization: Unlike both Para-virtualization and full virtualization, OS-level virtualization does not depend on a hypervisor. Instead, it modifies the operating system securely to isolate multiple instances of an operating system within a host machine. Native Virtualization: In native virtualization, multiple unmodified operating systems are allowed to run along side one another. In this technique operating systems are capable of running on the host processor directly.

Virtual Machine can prompt execution corruption when an application experiences an expanding request bringing about an unforeseen ascent of the asset use. On the off chance that the asset prerequisite are not satisfied, the application can confront expanded reaction times, times-outs or disappointments. Guaranteeing dependable QOS characterized through SLAs built up between cloud suppliers and their clients for distributed computing conditions. With the point of accomplishing appropriate asset portion, proficiency and adaptability. Virtualization advancements have persistently developed to give distinctive abilities, for example, the execution of various applications in parallel. Two less investigated challenges are asset heterogeneity and multi-occupancy which are basic antiquities of running in cloud.

The main idea is periodic refinement of task distribution using measured CPU loads, task loads, and idle times. We analyse the impact of load balancing frequency, grain size, and problem size on achieved performance. The execution environment depends on VM to physical machine mapping. Cloud providers to create multiple virtual machine instances on a single server, thus improving the utilization of resources and increasing the Return On Investment (ROI).

Resource allocation is in form of virtual machine allocation.

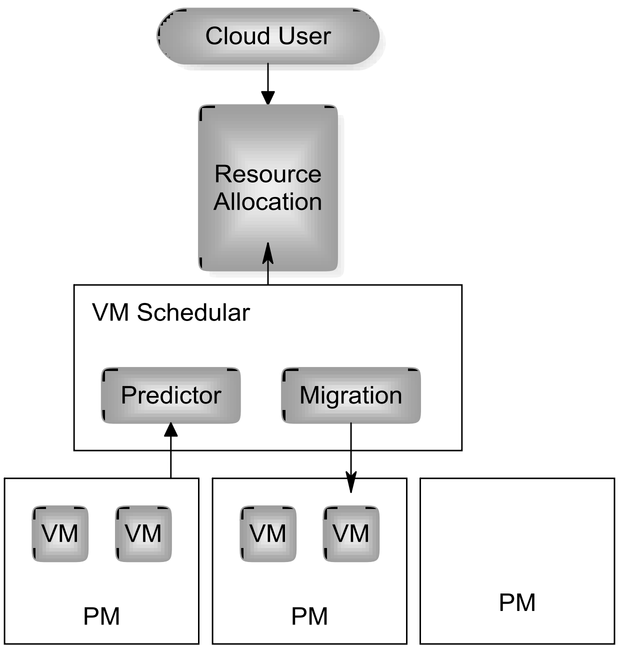

Figure 1: Resource Allocation

In the resource allocation cloud user may request different resources based on their needs , by using VM scheduler the resource can allocated. By using predictor the work load can allocate to physical machine which consist of no Virtual machines in it. Each machine consists of certain threshold value, if it exceeds the limit overload migration took place. In migration set of process can move to another machine.

Rest of the paper is organized as follows, Section I contains the introduction, Section II contain the related work, Section III describes the proposed method, Section IV experimental evaluation, section V concludes research work with future directions.

II. RELATED WORK

Some challenges[1] are insufficient network and I/O performance in cloud, resource heterogeneity, and unpredictable interference arising from other VMs. The approaches taken to reduce the gap between traditional cloud offerings and HPC demands can be classified into two broad categories – (1) those which aim to bring clouds closer to HPC and (2) those which want to bring HPC closer to clouds. We can make HPC applications more cloud friendly using a customized parallel runtime system. Moreover, our scheme uses a refined load balancing algorithm that reduces number of task migrations.

The objective [3] is to maximize the power provider's profit by minimizing both power and SLA violation. Kalman filter is applied to estimate the number of future requests to predict the future state of the system and perform necessary reallocations.

In [4] et all propose a system called claudia, where provisioning is based on performance indicators and elasticity rules defined by users. In both approaches number of instances vary reactively to incoming request rate, whereas our model proactively applies adaptive provisioning to deliver negotiated QoS to requests whose request arrival rate varies along with the time. Jung et all propose the Mistral system, which performs management at data centre host level to manage power consumption of resources and performance of applications.

In [5]It is issue that virtualization overheads manifest as extra resource usage by the hypervisor, particularly in the case of I/O workloads. This is virtualization architecture specific, while resolving VM placements that need to honour SLA guarantees. We propose VM placement approaches that consider performance SLAs and VM migration costs while optimizing VM placements over a minimal set of physical hosts. We would like to investigate the effect of consolidation of multiple VMs on KVM .Also, we would like to test these algorithms for real time elastic workloads .Multi-core VMs are allocated efficiently to meet the offered workload, and in a way to avoid any violations to the agreed SLA [7].

This involves the need to show SaaS administrations to anticipate the execution and general framework cost, and to appraise the required number of VM assets and their individual multi-center limit preceding the real arrangement. The quantity of required multi-center VM cases expected to fulfil the Quality of Service (QoS) parameters. The creators in proposed a confirmation control and booking calculations for asset assignment for SaaS suppliers to limit the aggregate cost and SLA infringement.Execute our recipes to be an indispensable piece of an extensive arrangement that can progressively scale assets productively for SaaS applications in publically accessible cloud stages as that of Amazon AWS.

III. Proposed Method

A. Problem Statement

Allocate task's from overloaded vm's to appropriate virtual machine to improve performance based on task frequency and memory requirement.

B. System Architecture

For our work, We consider a large-size task with a collection of PMs, Where VMs run on the top of PM according to the VM requests. Indeed, large data centres of Google, Microsoft, Yahoo and Amazon etc. Contain tens of thousands of PMs. Each VM is allocated to 1 PM, where as a PM can be allocated VMs through a hypervisor. Each PM is characterized by the CPU, size of RAM, storage hard drive and network bandwidth. Each client has applications running on VMs that require resources from the PMs defined in CPU capacity, amount of RAM, and network bandwidth. The requests from various clients in which are too connected to list of tasks for allocating task in to the VM allocator.

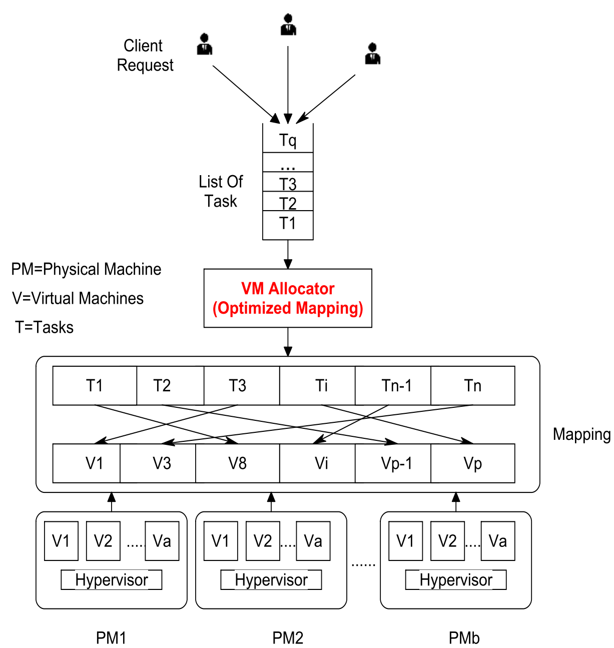

Figure 2. System Architecture

The VM allocator directed to one of the VMs. when final clients need new VM with required accomplishing their application, The VM allocator has functionality, to receive all the incoming requests to the list of task and distributed it evenly among the VMs in System model.VM allocator is consider the estimated load exceed the limit. Where task assign to the highest memory capacity of VM and frequency. and that have been multiple number of tasks.

C. Algorithm

The following steps are describes to the VM allocation to enhance performance.

Step 1: Identify list of all possible VCPUs where the task can be relocated.
Step 2: Remove all VCPUs From this list for which, after considering the task on the VCPU if the estimated load exceeds the limit.
Step3:OptimizedMappingIndex=alpha*(1/distance)+beta*(MIPS capacity of VCPU / MIPS requirement of task)+gamma*(Memory capacity of VCPU / Memory requirement of task).
(Where, alpha + beta + gamma = 1;alpha, beta, gamma belong to [0,1]).
Step 4: Assign the task on the VCPU having highest value of Optimized Mapping.
Step 5: Repeat Step 1 to Step 4 for all tasks for each VCPU of over Heap in such a way that VCPU becomes normal.

IV. EXPERIMENTAL EVALUTION

In the experimental evaluation we have considered various parameters like where P is the total number of VCPUs, Np is the number of tasks assigned to VCPU p, fp is the frequency for VCPU p. Show the example Here total five virtual machine and their name has p1,p2,p3,p4,p5 and virtual machine have assign to some task so there has n1 have 4 task and n2 5 task so on. task have use their different frequency like p1 have 100 MIPS and p2 have 200 MIPS and so on...now see the example.

Let's consider P=5 values there
No. of Vcpu P={p1,p2,p3,p4,p5}
No. of task on Vcpu Np={n1=4,n2=5,n3=3,n4=8,n5=6
Frequency for Vcpu f ={p1=100, p2=200, p3=150, p4=300, p5=250}

Total no of VM are five from p1,p2,p3,p4p5.And task assigned to VM are n1,n2,n3,n4,n5.there can be different frequency and their value is given to the above p1=100,p2=200.....etc.

$$O_p = T_{lb} - \sum_{i=1}^{Np} ti - t_{idle}^{p}$$

$$Tk_{avg} = \frac{\sum_{p=1}^{P}((\sum_{i=1}^{N_p} t_i + O_p) * f_p)}{P}$$

Table 1: calculation of task average

Op	Fq	Ti	Task	Tavg	Overloaded	Underloaded
5.95	100	10	4	539.5		539.5
4.95	200	20	5	1059.5	1059.5	
6.95	150	3	3	824.5		824.5
1.95	300	28	8	1499.5	1499.5	
3.95	250	24	6	1298.5	1298.5	
2.95	180	30	7	917.1		917.1
0.95	220	42	9	1076.9	1076.9	
7.95	120	18	2	671.4		671.4
8.95	130	4	1	740.35		740.35
6.95	280	8	3	1538.6	1538.6	

Here, calculation of task average and find the under loaded virtual machine and overloaded virtual machine.

Table 2: calculation of best core

Best Core	V1	V3	V6	V8	V9
T1	1295	1195	1595	1395	1795
T2	2590	2990	3190	2790	3590
T3	1942.5	2542.5	2092.5	1642.5	2692.5
T4	4185	5385	3585	4785	3285
T5	3487.5	2737.5	2987.5	3487.5	4487.5
T6	2151	2691	3591	2511	3231
T7	3289	3509	2849	3069	3949
T8	2034	2274	1914	1674	1314
T9	2203.5	1423.5	2593.5	1813.5	2333.5
T10	4746	5026	5306	3906	4466

Here, the calculation of best core considered only under loaded virtual machine to find best core.

Table 3: lowest value of best core

Task	VM	Best Core
T1	V3	1195
T2	V1	2590
T3	V8	1642.5
T4	V9	3285
T5	V3	2737.5
T6	V1	2151
T7	V6	2849
T8	V9	1314
T9	V3	1423.5
T10	V9	4466

Here, the list of task is given where best core value is lowest.

Table 4: task configuration

Task	Task Length
T1	2000
T2	4000
T3	3000
T4	2500
T5	1500
T6	2200
T7	1800
T8	1000
T9	1200
T10	2200

Table 5: VM Configuration

VM	MIPS
V1	3000
V2	300
V3	500
V4	1500
V5	2500
V6	2400
V7	1700
V8	1400
V9	1000
V10	3300

Table 6: task configuration

Task	Frequency	Memory	Task Length
T1	200	110	1000
T2	150	300	3000
T3	400	200	2000
T4	200	400	2500

T5	100	200	1500
T6	300	200	1200
T7	500	230	800
T8	600	520	1800
T9	300	420	1000
T10	100	150	2000

List of task is given where two parameter are used like frequency and memory .task id is given there are total 10 task is considered on that table.t1 has 200 frequency and 110 memory when ever task id is 1 and then after considered t2 has different frequency and memory and so on.

Table 7: VM Configuration

VM	Frequency	Memory	MIPS
V1	100	100	2000
V2	100	250	1000
V3	300	100	2500
V4	50	150	500
V5	200	150	3500
V6	220	100	2200
V7	300	200	1800
V8	100	200	1500
V9	110	250	1000
V10	250	230	3000

Table 8:Value of Constants

Alpha	Bita	Gamma
0.33	0.33	0.33
0.57	0.27	0.16
0.32	0.12	0.56
0.33	0.42	0.25
0.25	0.5	0.25
0.1	0.3	0.6
0.5	0.2	0.3
0.2	0.4	0.4
0.1	0.55	0.35
0.25	0.45	0.3

Here, α, β, γ are the preference defined as value of weight for the different parameters.Where $\alpha + \beta + \gamma = 1$.

Table:9 Calculation of OMI

Task	VM	Optimized Mapping Index
T1	V10	1.1355
T2	V8	1.157
T3	V2	0.762
T4	V7	0.788
T5	V7	1.775
T6	V10	0.95
T7	V10	0.45
T8	V7	0.373
T9	V7	0.726667
T10	V7	1.775

Table:10 calculation of highest OMI

OMI	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
T1	0.498	0.948	0.828	0.5655	0.813	0.696	1.128	0.798	0.9645	1.1355
T2	0.517	0.937	0.877	0.567	0.837	0.733	1.157	0.797	0.955	1.151
T3	0.342	0.762	0.402	0.467	0.512	0.378	0.682	0.622	0.765	0.751
T4	0.3055	0.39	0.7255	0.23175	0.54675	0.5575	0.788	0.368	0.42025	0.70175
T5	0.65	0.8375	1.65	0.4625	1.2125	1.25	1.775	0.775	0.8875	1.5625
T6	0.41	0.86	0.61	0.51	0.66	0.53	0.91	0.71	0.87	0.95
T7	0.22	0.41	0.300435	0.265652	0.325652	0.268435	0.43087	0.35087	0.420087	0.45
T8	0.169	0.278	0.296923	0.168718	0.268718	0.24359	0.373846	0.240513	0.285641	0.36359
T9	0.276	0.4017	0.643333	0.643333	0.501667	0.496667	0.726667	0.36	0.42	0.66
T10	0.675	0.975	1.575	0.55	1.225	1.215	1.775	0.875	1.02	1.61

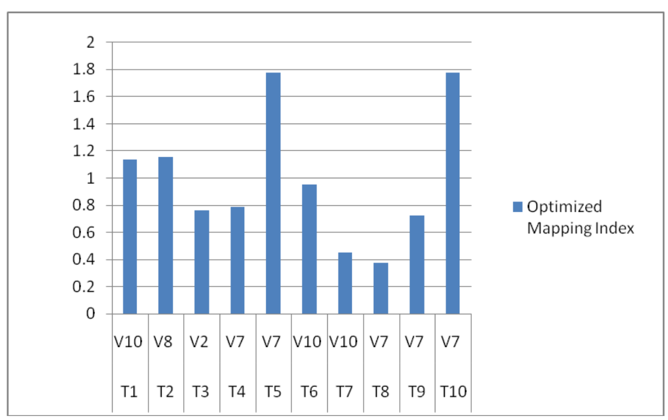

Fig.3

Table 11: Comparision of execution time

Execution Time(Best Core)	Execution Time(OMI)
4	0.33
2	2
3	0.
5	1.388
3	0.833
0.73	0.4
0.75	0.266
1	1
2.4	0.555
2.2	1.11

Here,Comaparison of existing best core algorithm and proposed optimized mapping algorithm with execution time parameter.

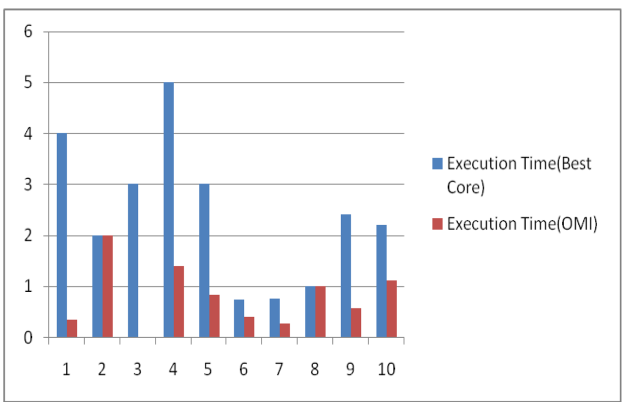

Fig.4

V. CONCLUSION AND FUTURE SCOPE

In this work, the optimized mapping algorithm determine on the base of current utilization of target virtual machine, MIPS specifications of task and main memory requirement of task.Task is assigned only to those vm whose limit is normal also task is assigned to only those vm's whose memory capacity and frequency is highest. The future work may lead to identify technique that reduce execution time & execution cost.

REFERENCES.

[1]Gupta, Abhishek, et al. "Improving hpc application performance in cloud through dynamic load balancing." *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*. IEEE, 2013..

[2]El Kafhali, Said, and Khaled Salah. "Performance analysis of multi-core VMs hosting cloud SaaS applications." *Computer Standards & Interfaces* 55 (2018): 126-135.

[3]Frey, Stefan, C. Reich, and C. Lüthje. "Key performance indicators for cloud computing SLAs." *The Fifth International Conference on Emerging Network Intelligence, EMERGING*. 2013.

[4]Calheiros, Rodrigo N., Rajiv Ranjan, and RajkumarBuyya. "Virtual machine provisioning based on analytical performance and QoS in cloud computing environments." *Parallel processing (ICPP), 2011 international conference on*. IEEE, 2011.

[5] Anand, Ankit, J. Lakshmi, and S. K. Nandy. "Virtual machine placement optimization supporting performance SLAs." *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*. Vol. 1. IEEE, 2013.

[6] B.Magensh Kumar "Vibrant resource allocation algorithem using virtual machine in cloud survey "2013.

[7] Anitha H M, P. Jayarekha , "*Security Challenges of Virtualization in Cloud Environment*", International Journal of Scientific Research in Computer Science and Engineering, Vol.6, Issue.1, pp.37-43, 2018

Authors Profile

Ms. P. D. Pariyani pursed Bachelor of Engineering from C. K. Pithawalla College of Engineering and Technology (Gujarat Technological University), Surat City, Gujarat in 2016. She is currently pursuing Master of Engineering at LDRP-ITR, Gandhinagar, Gujarat. Her main research work focuses on Cloud Computing based education.

Mr. H B Patel pursed Bachelor of Engineering from B.V.M Engineering College. He have pursued M.E from M.S.University, Baroda. He have completed his Ph.D from NIT Surat. He was former Dean at Sankalchand Patel University. He is currently working as H.O.D and professor at LDRP-ITR institute, Gandhinagar since 2017.

Ms. B Shrimali have pursued Bachelor of Engineering from Charotar Institute, Changa, Gujarat. She have pursued M.E from G.E.C-G, Gandhinagar, Gujarat. She is pursuing Ph.D from C.U.Shah University, Wadhwancity, Gujarat. She have total 12 years and 6 months of experience in teaching field. Currently she is working as lecturer at LDRP-ITR institute, Gandhinagar, gujarat. Her main research area focuses on cloud computing and language processing. She is also member of ISTE.