ISSN 2320-9984
Volume 10 Issue 4 | December 2022 ISSN: 2320-9984 (Online)
International Journal of Modern Engineering & Management Research
Website: www.ijmemr.org

Dynamic Resource Allotment in Cloud Computing with Predetermined Waiting Queue

Dhanpal Singh
M.Tech. Research Scholar
Computer Science and Engineering
Shriram Group of Institutions
Jabalpur, (M.P.) India
Email: dhanpal530@gmail.com

Sapna Jain Choudhary
Assistant Professor
Department of Computer Science and Engineering
Shriram Group of Institutions
Jabalpur, (M.P.) India
Email: choudharysapnajain@gmail.com

Abstract— *For delivering the cloud services over the internet, cloud computing has become proficient infrastructural model for hosting cloud services. Server virtualization is key technology that enables cloud computing as a service, which authorizes dynamic sharing of physical resources. Virtualization introduces the problem of virtual machine placement that increases the overheads in load balancing. Existing infrastructure needs the strategy for the VM Placement as it may create poor allocation and load balancing issues. In most of the cases, due to lack of input parameters physical machines are partially loaded that creates issue of fragmentation which leads to insufficient resources that causes more utilization of physical machines in any infrastructure. We did extensive survey in the said domain and found that, In load balancing approach, when VM placement is done without measuring its lifetime, that creates fragments on PM. So we propose dynamic priority based spill over technique and add the concept of short life/ long life container for solving the fragmentation issue.*

Keywords:—*VM Placement, Load Balancing; Fragmentation, Data center, Cloud Computing*

1. INTRODUCTION

Cloud computing as a novel and entirely internet-based approach provides a highly available, scalable, and flexible computing platform for a variety of applications and has brought about great benefits to both enterprises and individuals [1]. Computing is being changed to a service based model whereby access to these services depend on users' requirements without regard to where the services are hosted or how they are delivered [2]. Such computing model offers many types of services, such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). With the spread of cloud computing, cloud workflow systems are designed to facilitate the cloud infrastructure to support huge scale distributed collaborative e-business and e-science applications [3].

An ever-growing number of corporations, companies and individuals are running applications of all sorts in the cloud. Such a huge demand for services, at different scale, led to the development of sophisticated load balancing techniques [5] to efficiently utilize resources and reduce resource wastage in cloud environments [4]. One of the ways to balance the load is scheduling tasks in the cloud. Task scheduling is an important part of any distributed system like P2P networks, Grid and cloud [6]. It assigns tasks to suitable

resources for execution. However, it is an NP-Complete to schedule tasks to obtain the least make span [7]. The main goal of a task scheduling algorithm is scheduling all subtasks on a given number of accessible resources in order to minimize make span without violating precedence constraints [8]. The most vital side of cloud computing environments is load balancing. Efficient load balancing scheme assures effective resource utilization by provisioning of resources to the cloud user's on-demand basis in pay-as-you-say-manner. Load balancing may even support prioritizing users by applying appropriate scheduling criteria [9]. As load balancing algorithms depend on current situation of system, it is considered a dynamic issue to balance its load.

Meta-heuristic approaches are inclusively applied to find optimal or near-optimal solutions for the scheduling problem. Owing to implicit parallelism and intelligence, genetic algorithms (GAs) have been applied to solve some large-scale, nonlinear resource scheduling [10] and good results have been obtained from their use in task scheduling.

A. Cloud Service Models

Cloud Software as Service (SAAS): **-** It is also known as ''On demand Software'' and it is a software licensing and it provide the software to consumer on subscription base.

Applications: Business / Multimedia, Web Service

Examples: You tube, Google Apps

Cloud Platform as Service (PAAS):-In this type of service, the consumer can deploy, the user generated or developed applications which is create by using programming or tools given by provider, on the cloud infrastructure.

Applications: Software Framework (Java/.Net), Data /File Storage

Examples : AWS, Microsoft Azure

Cloud Infrastructure as Service (IAAS): **-** This is a capability provided to the consumer by which, it can provision processing, storage, networks and other fundamental computing resources where the consumers can deploy and run the software.

Applications: Hardware Resources(CPU, Memory, Disk)

Examples: Go Grid, Amazon EC2, Data Centers. b. Cloud Deployment Models

Public Cloud:-This public cloud is available for every organization.

Private Cloud:-This cloud is available only for particular organization or company.

Community Cloud: - In this type of cloud deployment model, the infrastructure of the cloud system is commonly used by many of the organizations and supports a specific community with shared concerns.

Hybrid Cloud:-It is a composition of two or more different clouds that is private or community or public. Element of the hybrid cloud are tightly coupled. Load Balancing algorithms can be of 3 categories are as

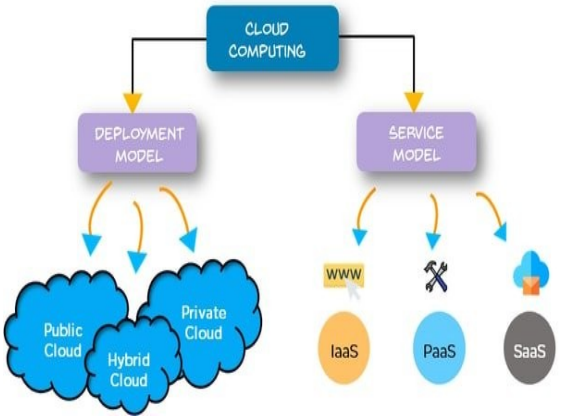

Figure 1: Cloud Service Model

i. ***Sender Initiated:*** If the load balancing algorithm is initiated by the sender.

ii. ***Receiver Initiated:*** If the load balancing algorithm is initiated by the receiver.

iii. ***Symmetric:*** It is the combination of both sender initiated and receiver initiated. Normally, the loads were distributed evenly, uniformly, overloaded, minimally among the nodes to the system.[5] [6][7][8]

B. Load Balancing Algorithms (Cloud Computing)

Min-Min Load Balancing Algorithm

This is simple static algorithm and offers excellent performance in task scheduling. The cloud service manager find the completion time of every task. The new task has been waiting in a queue for execution. This algorithm assigns the task to the resource based on which task has minimum execution time to complete.

The pseudo code is

following Procedure Minmin(Task Ti)

{

Find execution_Completion_Time of each task

Store the execution_Completion_Time of task Ti

in orderQueue

repeat

{

for each task Ti in orderQueue

{

obtain minimum completionTime from orderQueue;

assign task to vm;

update the execution_Completion_Time;

}

}

Until orderqueue empty;

}

This algorithm works well when the task has minimum execution time however if task has maximum execution time then the task must be wait with undefined time. This will lead the starvation problem. This algorithm is best in the situations where the number of tasks with minimum completion time.

Max-Min Load Balancing Algorithm

This algorithm is following identical procedure of Min- Min algorithm. This algorithm calculates the execution completion time of all tasks. The maximum completion time is taken and assigned to the corresponding resources. This algorithm is best in the situations where the amount of tasks with maximum completion time and it take away the starvation. The task minimum completion time has been waiting in ordered queue until the other maximum completion time task must be completed. Here we can understand that this algorithm performs well in a static environment and both the algorithm has their merits and demerits based on the environments. The performance doesn't depend on the algorithm chosen but indeed the environment taken. Min-Min and Max-Min algorithms are equally performed on the static cloud environment.

Round - Robin Algorithm

It is the static load balancing algorithm and elects the node randomly for allocate a job. This algorithm assigned the resource in circular order and without using priority of the task. It's not suitable if any virtual machine is heavy loaded and some virtual machines are lightly loaded because the running time of all processes is not aware in advance. This algorithm is not preferred because prior prediction of execution time is not possible. The round robin algorithm is given below 9,10.

Create Q1, Q2;

Q1 used to store ready process

Q2 used to store blocked process

New process place to end of Q1

If task time interval finished then

Move to end of Q1.

If I/O request or swapped out request is made by process

then

Move process from Q1 to Q2.

If I/O operation is completed or ready to move from

blocked processes then

Move process from Q2 to Q1.

Genetic Load Balancing Algorithm

This algorithm implement in dynamic cloud environment and it used soft computing approach. This algorithm is experimental from the natural development. This algorithm provides better performance compare to RR and FCFS algorithm. The advantage of this algorithm is easily handle a vast search space, applicable to complex objective function and may avoid being trapping into local optimal solution11.

GA's implementation mechanism is based on three steps:

Selection Operator: Selects the initial population randomly.

Crossover Operator: Find fitness pair of individuals for crossover.

Mutation Operator: A small or low probability value is called mutation value. These bits are toggled from 0s to 1s or 1s to 0s. The output is new pool of individuals ready for crossover.

HBB-LB Algorithm

This is nature inspired algorithm for self -organization. HB consists of a queen and foragers, where forager is of two types; employed and unemployed. The foragers are informed about the food available nearby by waggle dance by scout bee (unemployed), the dance is to give the information to the other foraging bees about the distance, quality, direction and other information which is useful in getting the food12. This algorithm has similar principle in balance the work of the virtual machine. The HBB algorithm calculates the virtual machine workload then it decides whether it is overloaded, light weighted or balanced. The high priority of the task is off from the overload virtual machine and tasks are waiting for the light weight virtual machine. These tasks are known as scout bee in the next step. Honey Bee Behaviour inspired Load Balancing technique reduces the response time of VM and also reduces the waiting time of task13-15. 4

Ant Colony based Algorithm

The ant colony algorithm is based on the behaviour of the real ants. The ant can notice the optimal path where the source food is available. The Ants while seeking a path from their colony in search of food secrete a chemical called pheromone on the ground thus leaving a trail for other ants to follow the path. But this chemical evaporates with time. This approach aims to give efficient distribution of workload among the node. The ant maintains the record of every visited node for better making decision in future. The ant would deposit pheromones during their movement for other ant to select next node. The intensity of pheromones can vary on the bases of certain factors like distance of food, quality of food etc. When the job gets successful the pheromones is updated. Each ant build their own individual result set and it is later on built into a complete solution. The ant continuously updates a single result set rather than updating their own result set. By the ant pheromones trials, the solution set is continuously updated 16-18.

The basic algorithm is given below:

Step 1: Initialize the pheromone.

Step 2: Placing all the ants at the beginning of the VMs.

Step 3: Until all the ants have found food (solution) DO.

Each ant should follow Do

Choose a VM for new task

Check for the pheromone intensity

End Do

End Do

Find the best so far

Update the pheromone

Step 4: End

Opportunistic Load Balancing (OLB) Algorithm

It is the static load balancing algorithm and this algorithm does not consider the current workload of the virtual machine. It keeps each node busy although they are already loaded with the task. This algorithm schedules the new task in randomly available virtual machine without checking workload of that machine. It provides load balance schedule without good result. The task executed slowly because it does not calculate execution time of the VM. This algorithm is not suitable in improving resource utilization19.

Game Theory Algorithm

This dynamic algorithm works in public cloud environment. This algorithm is partition the cloud into three class namely idle, normal and overload based on the load degree. The public cloud includes many nodes and it located at different places. This partition helps us to manage the large cloud. The load balancing started after the portioning, with the main controller deciding which cloud partition should receive the job and partition load balancer formulates the job assigning to the nodes. If the cloud partition is normal then task complete locally and if the cloud partition load status is not normal, this job should be transferred to another. When the environment is huge and compound these divisions streamline increase the proficiency in the public cloud environment. This load balancing refining the presentation and maintain stability. The challenge of this algorithm is to predict the job arrival, capabilities of each node in the cloud 20,21.

Stochastic Hill Climbing Algorithm

A variant of Hill Climbing algorithm Stochastic Hill Climbing (SHC) and it gives the solution for optimization problem. This procedure classified into two methods called complete and incomplete. Complete method which guarantees a correct answer either by proving that no such assignment exists or by finding a possibly valid assignment to the variable. On the other hand, incomplete method doesn't guarantee correct answers for all the given inputs. A Hill Climbing algorithm- Stochastic Hill Climbing algorithm is based on the incomplete method for solving optimization problems. SHC is a local optimization algorithm that continuously moves in the upward direction for increasing the value. If no neighbour has a higher value then it will automatically stop. This basic idea of this operation is repeat the solution until found or stopping no neighbour has a high value. So it has two main components a candidate generator that maps one solution candidate to a set of possible successors, and evaluation criteria which ranks each valid solution such that improving the evaluation leads to better solutions21,5.

A2LB Algorithm

The dynamic A2LB algorithm addresses the resource utilization, maximum throughput and minimum response time issues. The overloaded virtual machine was distributed by cloud service provider to available resource for utilize in a right manner. This approach

helps to balance the workload of all virtual machines. A2LB consists of three agents called load, channel and migration agent. Load and channel agents are static agents whereas migration agent is an ant, which is a special category of mobile agents. The reason behind deploying ants is their ability to choose shortest/best path to their destination. The ant start searching a food randomly, thus they may follow different paths to the same source, however with passage of time, density of pheromone on the shortest path increase and thus all follower ants start following that path resulting in increase of pheromone density even further. The ant moves from source to destination for collecting necessary information or carryout a task. It is not necessary come back to their source rather they destroy themselves at the destination only thereby reducing unnecessary traffic on the network. This algorithm would require searching for under loaded servers and resources, ant agents suit the purpose and fulfil it appropriately without putting additional burdenon network4,1.

Firefly Algorithm

The dynamic firefly scheduling algorithm is relies on flashing characteristics of fireflies and its application in optimizing the schedule process to the cloud network. This approach concerned about workload balance, thus they used following three rules in the cloud system.

All fireflies are unisex so that one firefly can attracted to other fireflies of their sex6.

Attractiveness is proportional to their brightness, thus for any two flashing fireflies, the less bright one will move towards the brighter one. The attractiveness is proportional to the brightness and they both decrease as their distance increases6.

The brightness of a firefly is affected or determined by the landscape of the objective function.

The brightness proportional to the value of the objective functions in maximization problem. The load balancing operation going to be initiated effectively based on load index value which was calculated.

2. MEDIUM LEVEL LOAD BALANCING MECHANISM

The new innovative Load Balancing algorithm is to balance the load in medium level. The Server is having 100 rps. The Client A can accept only 50 rps. After reaching the half of the requests from Server automatically redirect the requests to the Client B, if it reaches half load then redirect to Client C and so on. The Medium Level Load Balancing algorithm will give to increase client satisfaction and maximize resource utilization.

2. Objective

Objective of this work is to introduce and evaluate the proposed scheduling and load balancing algorithm by considering the capabilities of each virtual machine (VM), the task length of each requested job, and the interdependency of multiple tasks. Performance of the proposed algorithm is studied by comparing with the existing methods.

2.1 Existing Work

The management and scheduling of resources in cloud environment is complex, and therefore demands sophisticated tools for analyzing scheduling algorithms before applying them to real systems [3]. There are many papers that address the problem of scheduling in distributed systems, like Grid and multiprocessor systems, whereas, there is a few research on this problem in cloud. The multi objective character of the scheduling problem in cloud environments makes it difficult to solve, especially in the case of complex tasks [11]. The major research contributions in this field are summarized below.

A load balancing task scheduling algorithm based on weighted random and feedback mechanisms is proposed by Qian et al. [12]. In the first stage of the algorithm, the chosen cloud scheduling host selects resources by demands and sort them based on static quantification. Secondly, the algorithm randomly choses resources from sorted list based on their weight; then it obtains corresponding dynamic information to make load filter and sort the remaining. At last, it reaches, in a self-adaptively manner, the right system load through feedback mechanisms. The experimental results show that the algorithm avoids the system bottleneck effectively and achieves balanced load as well as self-adaptability to it.

Malik et al. [5] developed an efficient priority based round robin load balancing technique which prioritizes various tasks to virtual machines on the basis of various metrics, such as required resources or processors, number of users, time to run, job type, user type, software used and cost. Then, it conveys them to various available hosts in round robin fashion. This approach seems to improve the system capability by enhancing various parameters such as fault tolerance, scalability and overhead, and by minimizing resource utilization and response time. This technique has been simulated and tested over CloudSim, which is a widely used tool to test cloud based techniques.

In another research [13], a hybrid ACHBDF (Ant Colony, Honey Bee with Dynamic Feedback) load balancing method for optimum resource utilization in cloud computing is presented. The developed ACHBDF method uses combined strategy of two dynamic scheduling methods with a dynamic time step feedback method. The proposed ACHBDF utilizes the quality of ant colony method and honey bee method in efficient task scheduling. The used feedback strategy checks system load after each phenomenon in a dynamic feedback table to help migrating tasks more efficiently in less time. An experimental analysis comparing existing ant colony optimization, honey bee method and ACHBDF, showed superiority of ACHBDF.

M. Aggarwal et al.[1], has figured that Load balancing require more efficiency for the data dispersed on one or more physical machine. This also makes the problem of data locality on physical machine. For solving the efficiency problem, this paper proposed CPU utilization based priority method and architectures clarifies the task of developing load balancers that focuses on data locality. In proposed model, it comprised the set of calculate nodes and a global task scheduler. The task scheduler pull together status information from the nodes and allocates information based on task. In this approach, priority based load balancer plays vital role in balancing load for overall strategy. This proposed technique depends on less CPU utilization higher priority and higher CPU utilization less priority policy. If waiting time of VM is more than migration time then migration of job at remote node.

D. Chitra Devi et al. [4], states the problem for Scheduling of the Nonpreemptive task in the Cloud Computing Environment. For nonpreemptive task, round robin Policy does not consider the resource capabilities, priority and length of task as well as The Weighted Round Robin failed to judge the length of the tasks to select the suitable VM. For Solving the Problem author introduce Improved Weighted Round Robin algorithm. This proposed methodology aimed to the efficient scheduling as well as load balancing algorithm. For this they considered the capabilities of VM, task length of each requested job and the interdependency of multiple task. Improved weighted round robin focuses on the allocation of the jobs based on VM's capabilities such as capacity of processing the load on VM and priority based on length of incoming tasks.

In Mishra, Mayanket al. [5], the problem occurs in estimation of VM resource requirement and in VM placement strategy to achieve efficient resource utilization of PMs.

This paper introduces Novel vector base approach. In this planar resource hexagonal is defined for position of PM or VM. There are four vectors such as capacity vector, resource utilization vector, remaining capacity vector and resource requirement vector. In this remaining capacity vector is obtained from total capacity vector minus resource utilization vector. From above defined vector VM placement is done.

K R Remesh Babu et al. [6],Load balancing aims to equally distribution of load among VM. Therefore nodes doesn't remain idle or overloaded. The proposed approach of bee colony algorithm based on foraging behaviour of honey bees is adopted for effective balancing of nodes. In this algorithm, it calculates the load of VM through standard deviation it assumes for load balancing criteria. If load balance is required, it is done on grouping of under loaded or overloaded. Then after the tasks are assigned based on this.

In Hou, Weigang, et al [7], the issue of Fragmentation in Cloud infrastructure is pointed, this paper suggests BVF algorithm at the first level solution thereby using Maximum profit Defragmentation (MaxPD) algorithm if BVF fails. In MaxPD the existing jobs are migrated and newly arrived jobs are satisfied at physical machine. In Minimum cost Migration (MinCM) algorithm, selection of existing job at minimum migration cost.

Anton Beloglazov and Raj Kumar Buyya [1] proposal clearly states the quick development sought after for computational power driven by present day benefit applications consolidated with the move to the Cloud figuring model have prompted the foundation of huge scale virtualized information focuses. Such server farms expend gigantic measures of electrical vitality bringing about high working expenses what's more, carbon dioxide emanations. Dynamic combination of virtual machines (VMs) utilizing live movement and changing inactive hubs to the rest mode permit Cloud suppliers to upgrade asset use and decrease vitality utilization. Be that as it may, the commitment of giving high caliber of administration to clients prompts the need in managing the vitality execution exchange off, as forceful combination may prompt execution corruption. Because of the changeability of workloads experienced by present day applications, the VM situation ought to be enhanced consistently in an online way. To comprehend the ramifications of the online nature of the issue, we lead aggressive investigation and demonstrate focused proportions of ideal online deterministic calculations for the single VM movement and element VM solidification issues. Besides, we propose novel versatile heuristics for element solidification of VMs in light of an investigation of verifiable information from the asset use by VMs. The proposed calculations fundamentally diminish vitality utilization, while guaranteeing an abnormal state of adherence to the Service Level Agreements (SLA).

Whereas, Sobers Bazarbayev, William H. Sanders [2] statement states the following proposal on virtual machines scheduling in cloud workstations. Associations of all sizes are moving their IT foundations to the cloud due to its cost effectiveness and accommodation. On account of the on-request nature of the Infrastructure as a Service (IaaS) mists, a huge number of virtual machines (VMs) might be sent and ended in a solitary extensive cloud server farm every day. A substance based booking calculation for the situation of VMs in server farms. We exploit the way that it is conceivable to discover indistinguishable circle obstructs in various VM plate pictures with comparative Working frameworks by planning VMs with high substance closeness on similar hosts. That permits us to lessen the measure of information exchanged while sending a VM on a goal have.

In this paper, we first present our review of substance likeness between various VMs, in view of a vast set of VMs with various

working frameworks that speak to the greater part of mainstream working frameworks being used today. Our examination demonstrates that substance closeness between VMs with the same working framework and close form numbers can be as high as 60%. We likewise appear that there is near zero substance comparability between VMs with distinctive working frameworks. Second, in light of the above outcomes, we composed a substance based planning calculation that brings down the arrange activity related with exchange of VM circle pictures inside server farms. Our exploratory outcomes demonstrate that the measure of information exchange related with sending of VMs and exchange of virtual plate pictures can be brought down by over 70%, coming about in huge reserve funds in server farm organize usage and clog.

The proposal by Alexander Ngenzi, Selvarani and Suchitra Nair [3] states the Cloud asset administration has been a key element for the cloud datacenters improvement. Numerous cloud datacenters have issues in comprehension and executing the procedures to oversee, designate and move the assets in their premises. The outcomes of uncalled for asset administration may come about into underutilized and wastage of assets which may likewise come about into poor administration conveyance in these datacenters. Assets like; CPU, memory, hard plate and servers should be very much distinguished and oversaw. Dynamic Resource Management Algorithm (DRMA) might confine itself in the administration of CPU and memory as the assets in cloud datacenters. The objective is to spare those assets which might be underutilized at a specific timeframe. It can be accomplished through Implementation of appropriate calculations. Here, Bin pressing calculation can be utilized whereby the best fit calculation is sent to get results and contrasted with select reasonable calculation for proficient utilization of assets.

In Tony T Tran, Peter Yun Zhang [4] they have stated that a calculation for asset mindful booking of computational occupations in a substantial scale heterogeneous server farm. The calculation intends to distribute diverse machine designs to employment classes to achieve an effective mapping between work asset ask for profiles and machine asset limit profiles. a three-arrange calculation. The primary stage utilizes a queuing model that treats the framework in an accumulated way with pooled machines and employments spoke to as a liquid stream. The last two phases utilize combinatorial advancement strategies to take the arrangement from the main stage and apply it to a more precise portrayal of the server farm. In the second stage, occupations and machines are discretized. A straight programming model is made to acquire an answer for the discrete issue that boosts the framework limit. The third and last stage is a booking approach that uses the arrangement from the second stage to direct the dispatching of arriving occupations to machines. Utilizing Google workload follow information, we demonstrate that our calculation beats a benchmark voracious dispatch approach. Calculation can give mean reaction times up to a request of greatness littler than the benchmark dispatch arrangement.

Whereas. Ziqian Dong, Ning Liu and Roberto Rojas-Cessa [5] has stated that a model of errand planning for a distributed computing server farm to break down vitality proficient errand planning. We figure the assignments of undertakings to servers as a whole number programming issue with the goal of limiting the vitality devoured by the servers of the server farm. We demonstrate that the utilization of an eager errand scheduler limits the limitation benefit time while limiting the quantity of dynamic servers. As a down to earth approach, we propose the most-proficient server-first assignment booking plan to limit vitality utilization of servers in a server farm. Most-effective server-first calendars undertakings to

a base number of servers while keeping the server farm reaction time inside a most extreme requirement. Additionally demonstrate the solidness of most-proficient server-first conspire for errands with exponentially appropriated, autonomous, and indistinguishably circulated landings. Recreation comes about demonstrate that the server vitality utilization of the proposed most -effective server first planning plan is 70 times lower than that of an irregular based assignment planning plan.

3. EXISTING LOAD BALANCING POLICIES

There are various load balancing algorithms used in cloud computing. In this study, the following three algorithms have been studied, which can be implemented in the Cloud Analyst simulator [4].

A. Round Robin Algorithm (RR)

This is the simplest algorithm that uses the concept of a quantum of time or interval (Figure 1).

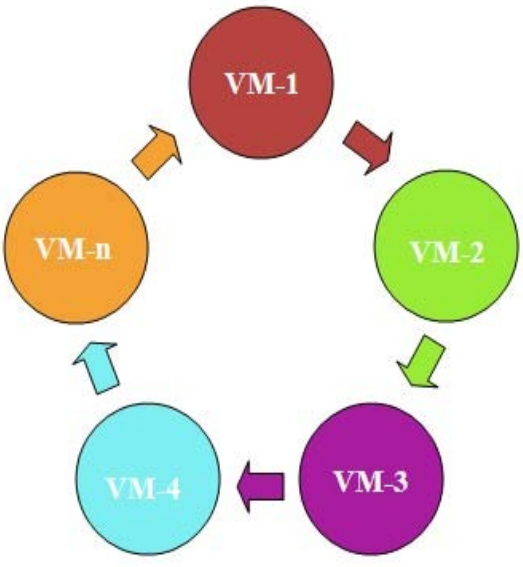

Figure 2: Round Robin Algorithm (RR).

Here time is divided into several sectors, and each node is given a specific time quantum or time interval, and in this quantum the node will perform its operations.

In Round Robin, scheduling a time quantum plays a very important role, because if the time slice is very large, then the Round Robin scheduling algorithm is the same as the FCFS planning [4].

The disadvantage of the method is that, although the algorithm is very simple, but to determine the quantum size, it generates an additional load on the scheduler. In addition, it has higher context switches that increase the turnaround time, and low throughput.

B. Active Load Balancing Monitoring (AMLB)

This algorithm has a dynamic character. It stores information about each VM virtual machine and the number of requests that are currently assigned to each VM. When the request is distributed by the new VM and if there are several VMs, the first recognized one is selected, and the AMLB returns the VM identifier to the data center controller. The data center controller warns the AMLB about the new distribution and sends the request to the virtual machine known under this VM identifier (Figure 3).

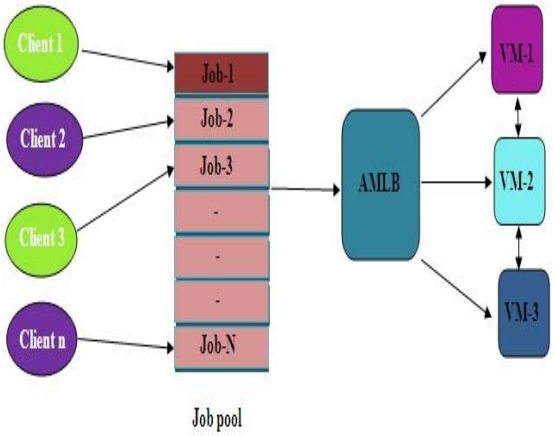

Figure 3: AMLB Algorithm (AMLB).

The disadvantage of the algorithm is that AMLB always finds the least loaded VM to assign a new incoming request, but does not check whether it was used earlier or not (therefore some VM is used intensively, and some are still not involved).

C. Throttled Load Balancing Algorithm (TLB)

In this algorithm, the load balancer maintains a table of virtual machine indexes, as well as their states (Available or Busy). The client / server first makes a request to the data center to find a suitable virtual machine (VM) to perform the recommended task (Fig. 3).

The data center requests a load balancer to distribute the virtual machine. The load balancer scans the index table from above until the first available virtual machine is found or the index table is completely scanned.

If a virtual machine is found, the data center passes the request to the virtual machine identified by the identifier. In addition, the data center confirms the load balancing of the new distribution, and the data center appropriately revises the index table.

When processing a client request, if the corresponding VM is not found, the load balancer returns "-1" to the data center. The center request is processed by the data center.

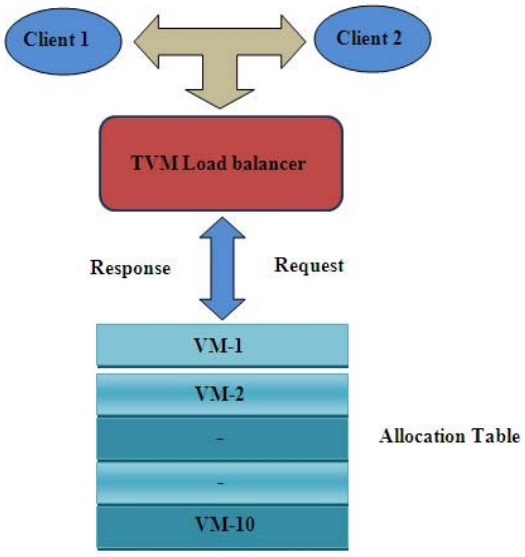

Figure 4: Throttled Algorithm (TLB).

When processing a client request, if the corresponding VM is not found, the load balancer returns "-1" to the data center. The center request is processed by the data center.

2.2 Gap Identification

- We aim to propose a system which handles the problem of resource utilization and solve fragment issue. We use short life/long life container for solving the fragment issue.

Table 1: Defined the Survey on Existing Papers and Evaluated the Gap from the same

No	Paper	Related work on	Problem in existing work
1	Dynamic Load Balancing Based on CPU Utilization and Data Locality in Distributed Database Using Priority Policy[1]	Load balancing	If waiting time is more than migration time then Data migrated to remote node.
2	A Provident Resource Defragmentation Framework for Mobile Cloud Computing[7]	Defragmentation	In BVF algorithm, the job having bigger volume is scheduled first. so the waiting time for small volume job increases.
3	Load Balancing in Cloud Computing Environment using Improved Weighted Round Robin Algorithm for No preemptive Dependent Tasks[4]	Load balancing	Due to nature of round robin algorithm, jobs are migrated frequently and also load balancing is not handled.
4	Load Balancing Of Tasks In Cloud Computing Environment Based On Bee Colony Algorithm[6]	Load balancing	Bee colony algorithm removes the tasks from overloaded VMs and submitted it to the under loaded VM. So migration cost and down time is large for this algorithm.

- Most of the time due to lack of input parameters [short life VMs, longlife VMs] placement always done on Physical machine which is partially loaded. Which results into so many partially loaded Physical machines and creates issue of fragmentation which leads to insufficient resources that causes more utilization of physical machines in any infrastructure? For solving this problem, we use short life and long life container in our architecture. We define threshold point for calculation of completion time of VM. If the lifetime of incoming job is smaller than the threshold value then it is defined as a short life job so that jobs store in short life container otherwise it is long life job and store inlong life container. Using our architecture resource utilization is maximize and fragment issue seems to be solve.
- The architecture of the system is simple, flexible and easy. All the incoming jobs are queued in VM. In VM queue, it stores all the short time and long time jobs. Also, everyone of the jobs from VM queue are transferred to DC manager. There is a threshold value which is ascertained in view of VM MIPS in DC manager. In the event that the life time of incoming job is littler than threshold value then it is characterized as a short life VMelse it is long life VM. Monitor stores all the data about CPU, stockpiling, and RAM and gives all data to the DC administrator. DC manager take plan of arrangement as indicated by the data.
- We aim to propose a system which handles the problem of resource utilization and solve fragment issue. We use short life/long life container for solving the fragment issue.
- Most of the time due to lack of input parameters [short life VMs, long life VMs] placement always done on Physical machine which is partially loaded. Which results into so many partially loaded Physical machines and creates issue of fragmentation which leads to insufficient resources that causes more utilization of physical machines in any infrastructure? For solving this problem, we use short life and long life container in our architecture. We define threshold point for calculation of completion time of VM. If the lifetime of incoming job is smaller than the threshold value then it is defined as a short life job so that jobs store in short life container otherwise it is long life job and store in long life container. Using our architecture resource utilization is maximize and fragment issue seems to be solve.
- The architecture of the system is simple, flexible and easy. All the incoming jobs are queued in VM. In VM queue, it stores all the short time and long time jobs. Also, everyone of the jobs from VM queue are transferred to DC manager. There is a threshold value which is ascertained in view of VM MIPS in DC manager .In the event that the life time of incoming job is littler than threshold value then it is characterized as a short life VM else it is long life VM. Monitor stores all the data about CPU, stockpiling, and RAM andgives all data to the DC administrator. DC manager take plan of arrangement as indicated by the data of monitor and VM queue. System Architecture.

VM Queue: VM queue store all incoming VMs. All short time VM sand long time VMs store in VM queue.

***DC Manager*:** DC manager analyse next few jobs (VMs) in VM queue. Using DC manager we characterize incoming job is short time job or long time job using threshold point. DC

manager predefine threshold value. If the life time of incoming job is smaller than the threshold value then it is characterized as a short life job else it is long life job. Monitor works with DC manager and provides all information like status of CPU, RAM, storage etc to the DC manager of each individual physical machine. DC manager collect the information about the both VM and PM.

Monitor: Monitor monitors status of CPU, RAM, storage etc. of PM and provide this in formation to the DC manager for better decision making.

4. CONCLUSION AND FUTURE WORK

After doing rigorous survey on various issues in resource utilization, we found that load balancing, VM placement and fragmentation are the greatest issue in cloud computing. So we propose dynamic priority based spill over technique and add the concept of short life/long life container for solving the fragmentation issue. Our architecture maximize the resource utilization also we minimize fragmentation issue with using shortlife/longlife container at physical machine in our architecture.

REFERENCES:

[1] Amandeep, ‘‘Analysis of Load Balancing Techniques in Cloud Computing’’, International Journal of Computers & Technology, vol.4, no.2, pp.2277-3061,March-April,2013.(references)

[2] Angona Sarker, Ali Newaz Bahar, SM Shamim, “A Review on Mobile Cloud Computing”, International Journal of Computer Applications (0975– 8887)

[3] Aarti Singha, Dimple Junejab, Manisha Malhotraa, “Autonomous Agent Based Load Balancing Algorithm in Cloud Computing”- International Conference on Advanced Computing Technologies and Applications (ICACTA- 2015)

[4] B. Subramani, ‘‘A New Approach For Load Balancing In Cloud Computing’’, IEEE, vol.2, pp. 1636-16405, May 2013.

[5] Buyya, Rajkumar. “Market-oriented cloud computing: Vision, hype, and reality of delivering computing asthe5th utility”. In Proceedings of the 2009 9th IEEE/ACM.

[6] Geethu Gopinath P, Shriram K Vasudevan, “An in- depth analysis and study of Load balancing techniques in the cloud computing environment”-. 2nd International Symposium on Big Data and Cloud Computing(ISBCC’15).

[7] Ruhi Gupta. “Reviewon Existing Load Balancing Techniques of Cloud Computing.” International Journal of Advanced Research in Computer Science and Software Engineering Volume 4, Issue 2, February 2014.

[8] Jinhua Hu, “A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment”, 3rd International Symposium on Parallel Architectures, Algorithms and Programming 978-0-7695-4312-3/10©2010 IEEE(published)

[9] Jayant Baliga, Robert W. A. Ayre, Kerry Hinton, and Rodney S.1Tucker, Fellow IEEE, “Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport”.

[10] S. Kapoor, and C. Dabas, Clusterbased load balancing in cloud computing, Proc. Eighth International Conference in Contemporary Computing (IC3),2015, 76-81.

[11] Klaithem Al Nuaimi, “A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms”, 2012

IEEE Second Symposium on Network Cloud Computing and Applications 978-0-7695-4943-9/12

[12] Mohit and Jitender Kumar, International Journal of Advanced Research in Computer Science and Software Engineering, "A Survey of Existing Load Balancing Algorithms in Cloud Computing".

[13] S.Mohinder, R. Ramesh, D. Powar, ‘‘Analysis of Load Balancers in Cloud Computing’’, International Academy of Science, Engineering & Technology, vol.2, May2013.

[14] Poulami Dalapati1, G. Sahoo, “Green Solution for Cloud Computing with Load Balancing and Power Consumption Management”- International Journal of Emerging Technology and Advanced Engineering,Vol3:2013

[15] Saurabh Kumar Garg and Rajkumar Buyya, “Green Cloud computing and Environmental Sustainability”. (references)

[16] Sidhu A,S. Kinger, “Analysis of Load Balancing techniques in Cloud Computing”, Council for innovative research international Journal of Computer & Technology, vol.4, March-April2013.

[17] Sumalatha M.R,C. Selvakumar, T.Priya, R.T. Azariah, and P. M. Manohar, "CLBC-Cost effective load balanced resource allocation for partitioned cloud system", Proc. International Conference on Recent Trends in Information Technology (ICRTIT), 2014, 1-5.

[18] The Amazon Elastic Compute Cloud (AmazonEC2), http://aws.amazon.com/ec2/

[19] YilinLu, “A Hybrid Dynamic Load Balancing Approach for Cloud Storage”, 2012 International Conference on Industrial Control and Electronics Engineering 978-0-7695-4792-3/12©2012 IEEE.

[20] Yuvapriya Ponnusamy, S Sasikumar, “Application of Green Cloud Computing for Efficient Resource Energy Management in Data Centres”, International Journal of Computer Science and Information Technologies, Vol3:2012.

[21] R. Divya, VE. Jayanthi “International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878 (Online), Volume-9 Issue-1, May 2020” Dynamic Resource Scheduling Cloud using Enhanced Queuing Model