# Dynamic load balancing in cloud using extended hungarian method

S. Jyothsna[1] and Bipin Bihari Jayasingh[2]
[1]CVR College of Engineering/Information Technology Department, Hyderabad, India
Email:jyothsna.sundaragiri@gmail.com
[2]CVR College of Engineering/Information Technology Department, Hyderabad, India
Email: bbjayasingh9@rediffmail.com

***Abstract:*** **Cloud Computing is a type of service based computing or utility computing. Our objective is to develop an effective load balancing algorithm using hungarian method to minimize response time and to increase the resource utilization. Load balancing in cloud need to consider the allocation of virtual machine and scheduling tasks on the virtual machine depending on various client and application given objectives. Hungarian method is a kind of assignment problem which works efficiently for static assignments. We are extending hungarian algorithm to work efficiently in dynamic environment like cloud. The availability status and load on each virtual machine are to be updated periodically and assigning tasks dynamically.**

***Index Terms--*** **cloud computing, load balancing, tasks, virtual machine(VM),hungarian method.**

## I. INTRODUCTION

A Cloud can be considered as an enormous collection of resources. Cloud computing is providing easy accessibility for required services and users can also deploy applications at competitive costs. Large data centers provide services through various cloud platforms. Cloud computing provides services at infrastructure level are called as infrastructure as a service.

The cloud service provider (CSP) makes on demand provisioning of hardware like processing power, I/O, large amounts of storage etc.Users accesses the services of cloud through a virtual machine, where number of virtual machines share a single physical server. Cloud computing provides a service oriented platform for cloud users.

The following cloud architecture is a three tier architecture where the number of clients are conneted to the data center(DBServer). Here the data center is the collection of host machines. The clients are sending their requests through client machines or virtual machines. The estimated cost to process the requests depends on the processing power of the data center and how the requests are scheduled so that the load on each virtual machine is distributed evenly to avoid delay and improve throughput.

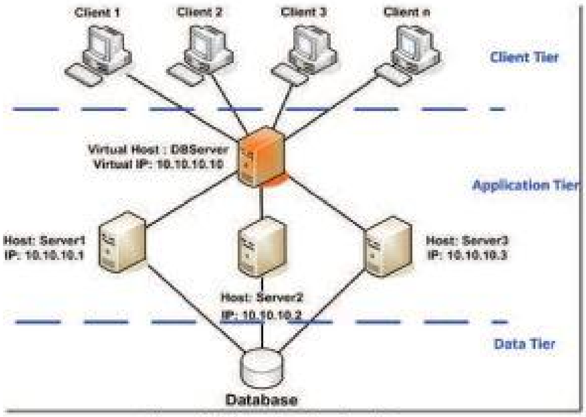

Figure 1. Three tier cloud architecture

### A. *Load balancing in cloud*

The load balancing in cloud computing is dynamic i.e cloud cannot rely on the prior knowledge whereas it takes into account run-time statistics. It is complex to maintain the stability of processing so many jobs in the cloud computing environment. The cloud provider installs heterogeneous resources. The resources are flexible to users in dynamic environment. In this scenario the requirements of the users are granted flexibility (i.e. they may change at run-time). Algorithm proposed to achieve load balancing in dynamic environment can easily adapt to run time changes in load. Dynamic environment is difficult to be simulated but is highly adaptable with cloud computing environment.

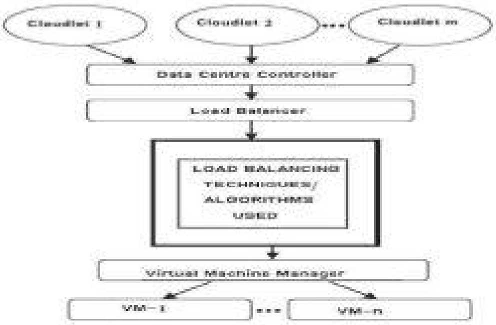

Figure 2. load balancing in cloud

Figure 2 shows the load balancing in cloud where the load balancer schedules tasks(cloudlets) to the corresponding virtual machines(VM-1...VM-n) using the appropriate algorithm such that the cost of the processing

doi:10.32377/cvrjst1115

should be minimized and servicing the number of users should be maximized. Here the virtual machine manager is responsible to allocate and monitor the virtual machines. Data center controller is processing the tasks(cloudlets) using the load balancer algorithm. In designing this paper the optimum load balancing algorithm to minimize the cost is discussed.

## II. Related Work

### A. Cloud Service Scheduling

Scheduling of cloud services can be categorized at user level and system level. User level scheduling deals with problems raised by service provision between providers and customers. The system level scheduling handles resource management within data center. Data center consists of many physical machines. Millions of tasks from users are received and assignment of these tasks to physical machine is done at data center. This assignment or scheduling significantly impacts the performance of cloud. In addition to system utilization, other requirements like QoS, SLA, resource sharing, fault tolerance, reliability, real time satisfaction etc, should be taken into consideration.

### B. User Level Scheduling

Market-based and auction-based schedulers are suitable for regulating the supply and demand of cloud resources. Market based resource allocation is effective in cloud computing environment where resources are virtualized and delivered to user as a service. Service provisioning in Clouds is based on Service Level Agreements (SLA). SLA represents a contract signed between the customer and the service provider stating the terms of the agreement including non-functional requirements of the service specified as Quality of Service (QoS), obligations, and penalties in case of agreement violations. Thus there is a need for scheduling strategies considering multiple SLA parameters and efficient allocation of resources. The focus of model is to provide fair deal to the users and consumers, enhanced quality of service as well as generation of optimal revenue.

### C. System Level Scheduling

System level scheduling is scheduling virtual machines to the corresponding physical machines. While scheduling virtual machines, the scheduler needs to consider the capacity of each physical machine and some threshold value must be chosen for each physical machine depending on the capacity. The load of each physical machine must be updated for every assignment and when the load reaches the threshold value,The migration or transfer of virtual machines must take place. The transfer may be static or dynamic. Static transfer considers the transfer of volume only where as dynamic transfer has to consider the state of virtual machine along with volume. Downtime of virtual machine is important to be considered to improve the performance of cloud. Dynamic transfer is complex to implement with no or less downtime but static transfer have more downtime. This needs to be considered in server scheduling.

## III. Mathematical Model

### A. Hungarian Method

The Hungarian method is a combinatorial optimization algorithm which was developed and published by Harold Kuhn in 1955. This method was originally invented for the best assignment of a set of persons to a set of jobs. It is a special case of the transportation problem. The algorithm finds an optimal assignment for a given "n x n" cost matrix. Assignment problems deal with the question how to assign n items (e.g., jobs) to n machines (or workers) in the best possible way […]. Mathematically an assignment is nothing but a bijective mapping of a finite set into itself […]"

The Hungarian algorithm can also be used for a maximization problem in which case we first have to transform the matrix. For example a company wants to assign different workers to different machines. Each worker is more or less efficient with each machine. The efficiency can be defined as profit. Higher the number, is higher the profit.

For Solving the assignment problem create one table based on the value of data, we call it as cost matrix. With the determined optimal solution we can compute the maximal profit.

- Worker1 => Machine2 - 9
- Worker2 => Machine4 - 11
- Worker3 => Machine3 - 13
- Worker4 => Machine1 - 7

Steps

1. Find the minimum from each row and subtract the minimum from all the rows.

2. Find the minimum from each row and subtract the minimum from all the rows.

3. Assign one zero ("0") to each row and column.

3. Cover all zeros with a minimum number of lines.

4. Create additional zeros if required.

5. Map all zeroes then this will give the minimum cost.

Existing work was using balanced assignment i.e the assignment was square matrix, the number of sources and destinations are equal. This type of assignment problems can be considered as balanced assignment. This can be represented as a square matrix, but in cloud environment this may not work as the cloud is servicing number of requests with minimum resources compared to the requests.

doi:10.32377/cvrjst1115

The following table gives the general Hungarian assignment, where number of tasks(souces) are equal to the virtual machines(destinations) so that this can be represented as a square matrix. The servicing time is represented as the cost matrix.

### *B. Hungarian Method for load balancing in cloud*

Load balancing problem can be represented as a two dimensional cost matrix. Resources are represented in a column and tasks who need resources are represent in a row.

The hungarian method can be used for assignment of tasks to virtual machines. Based on the virtual machine's current load and capacity the tasks are to be assigned to the corresponding virtual machines dynamically. Whenever the cost matrix of an assignment problem is not a square matrix, that is, whenever the number of sources is not equal to the number of destinations, the assignment problem is called an unbalanced assignment problem. In such problems, dummy rows (or columns) are added in the matrix so as to complete it to form a square matrix. The dummy rows or columns will contain all costs elements as zeroes.

Periodically checking the availability and load on each virtual machine by a load balancing algorithm then assigns the tasks to the corresponding virtual machine if its capacity is more than the required capacity to process the request and its availability status is free.

```
repeat {

Step 1: Initialize the capacity(VC[i]) of each virtual machine(VM).
        VC[i] = VM Capacity in MIPS

Step 2 : Initialize the Length of each Task
        TL[j] = Length of Request in MI

Step 3: Initialize the availability status of each VM=available, $Vm_a[i]_{=available}$

Step 4: Find Execution Time for each Request to VM if $VM_a[i]$=available
        Ex_Time[i][j]= TL[j]/VC[i] Seconds

Step 5 : Construct Expected Cost Matrix ECM[V,R]

Step 6 : Find out Minimum Execution Time from each row and subtract it from entire row.

Step 7 : Find out Minimum Execution Time from each column and subtract it from entire column.

Step 8 : Cover all zeroes with minimum number of lines.
If , No.of zero = number of resources then go to step 9,
Else, go to step 8.

Step 9 : Find minimum value which is not covered by any row and column and Subtract it from uncovered row and add it from value which is twice covered by line.

Step 10 : This matrix and original matrix are to be compared and choose best resources(VM) for the assignment of tasks and update the assignment status of VM as busy.

} until there is no job left unassigned

Step 11: The above process should be repeated infinitely.
```

As the cloud is providing service based computing for n number of users for all the time until there is no resource available, but ideally the resource must be available all the time as the cloud is a collection of enormous pool of resources as per definition. And once after the request is serviced again the availability status of the virtual machine must be updated to be available. so the load balancer algorithm should update the availability status and load on each resource ( virtual macine) periodically.

## IV. Implementation And Results

When we apply the above algorithm on the test data it shows the following results.

**Steps 1&2:** According to the considered capacities and task lengths, the following assignments were taken

| VM capacity (MIPS) | Task Length(MI) |
|---|---|
| 120 | 9000 |
| 150 | 8000 |
| 180 | 6000 |
| 200 | 5000 |

**Steps 3&4:** According to the estimated execution times calculated on the corresponding virtual machines , The following Execution time matrix is formed. Here there are four tasks and three vms, so it is an unbalanced assignment

| Task/VM | Task1 | Task2 | Task3 | Task4 |
|---|---|---|---|---|
| VM1 | 30 | 40 | 50 | 40 |
| VM2 | 40 | 30 | 25 | 30 |
| VM3 | 40 | 35 | 45 | 40 |

**Step 5:** Add a dummy row as the number of resources(VMs) are less than the number of Tasks to be processed

**Step 6** Now Apply Hungarian for finding optimum assignment, subtract min cost value from each row.

doi:10.32377/cvrjst1115

| Task/VM | Task1 | Task2 | Task3 | Task4 |
|---|---|---|---|---|
| VM1 | 0 | 12 | 6 | 16 |
| VM2 | 15 | 5 | 0 | 8 |
| VM3 | 0 | 5 | 18 | 11 |
| Dummy VM | 0 | 0 | 0 | 0 |

**Step 7 :** Here column wise reduction is not required as each column contains zero, draw minimum number of lines to draw all zeroes.

| Task/VM | Task1 | Task2 | Task3 | Task4 |
|---|---|---|---|---|
| VM1 | 0 | 12 | 6 | 16 |
| VM2 | 15 | 5 | 0 | 8 |
| VM3 | 0 | 5 | 18 | 11 |
| Dummy VM | 0 | 0 | 0 | 0 |

**Step 8**: Number of lines !=order of the matrix, Hence not optimal, again choose minimum for uncovered elements

**Step 9:** Here it is 5, subtract 5 from all uncovered elements

| Task/VM | Task1 | Task2 | Task3 | Task4 |
|---|---|---|---|---|
| VM1 | 0 | 7 | 1 | 11 |
| VM2 | 10 | 0 | 0 | 3 |
| VM3 | 0 | 0 | 13 | 6 |
| Dummy VM | 0 | 0 | 0 | 0 |

**Step 10:** Cover all zeroes with minimum number of line

**Step 11:** Number of resources=Number of lines, Then compare this matrix with original matrix and choose optimal assignment

| | Task1 | Task2 | Task3 | Task4 |
|---|---|---|---|---|
| VM1 | **30** | 42 | 36 | 46 |
| VM2 | 40 | 30 | **25** | 33 |
| VM3 | **30** | 35 | 48 | 41 |
| Dummy VM | 0 | 0 | 0 | **0** |

According to hungarian methods the above assignment gives the optimum assignment so that the overall cost can be minimized.

## V. Conclusions

This work is applying the hungarian algorithm for load balancing in cloud environment. As the cloud resources are dynamic in nature and tasks are also dynamic, the existing hungarian algorithm is extended by adding availability status for each resource periodically and the available resources may not be equal to the required tasks to be completed so the problem can be formulated as unbalanced assignment.. In this algorithm expected cost matrices(ECM) are generated and based on the results of the hungarian algorithm the load balancer knows about which task have to schedule by which VM. Here the priorities of tasks are not considered. Using extended hungarian method we can allocate resources dynamically for huge amount of tasks dynamically and also the hungarian method is simple to implement and works in balanced as well as in unbalanced conditions.

## References

[1] Disha Patel and Ms.Jasmine Jha,"Hungarian Method Based Resource Scheduling algorithm in Cloud Computing",*IJARIIE-ISSN(O)-2395-4396 -Vol-2 Issue-3 2016*

[2] Jameer. G. Kotwal, Tanuja S. Dhope "Unbalanced Assignment Problem by Using Modified Approach",International Journal of Advanced Research in Computer Science and Software Engineering-Volume 5, Issue 7,July 2015

[3] Seematai S. Patil, Koganti Bhavani — Dynamic Resource Allocation using Virtual Machines for Cloud Computing EnvironmentI ISSN: 2249 –8958,IJEAT-2014

[4] Fei Teng, "Resource allocation and schedulingmodels for cloud computing", Paris, 2011

[5] Daniel, D., Lovesum, S.P.J., "A novel approach for scheduling service request in cloud with trust monitor",IEEE, 2011

[6] Ahuja, R., De, A., Gabrani, G., "SLA BasedScheduler for Cloud for Storage & ComputationalServices", IEEE, 2011

[7] Harold W. Kuhn, "The Hungarian Method for the assignment problem",Naval Research Logistics Quarterly, 83–97, Kuhn's original publication, ISSN-15206750, Springer-Berlin Heidelberg, 2010

[8] Han Zhao, Xiaolin Li, "AuctionNet: Market oriented task scheduling in heterogeneous distributed environments", IEEE, 2010

doi:10.32377/cvrjst1115