International Journal of Latest Trends in Engineering and Technology
Special Issue SACAIM 2017, pp. 331-335
e-ISSN:2278-621X

MULTI-DIMENSIONAL RESOURCE SCHEDULING ALGORITHM FOR CLOUD DATACENTERS

Santhosh B[1], Chetan Hebsur[2] & Jithesh S[3]

Abstract- In distributed environments load balancing is essential for efficient operations. Load Balancing strategies are of huge significance in improving the reliability and performance of resources in data centres. Allocating and migrating virtual machines (VMs) which are reconfigurable and taking into consideration integrated features of hosting physical machine (PMs) are one of the challenging problems in scheduling resource in cloud data centres. Therefore load balancing due to its challenges and importance for the cloud has become a major research area. Algorithms were suggested to provide efficient mechanisms and algorithms for assigning the clients requests to available Cloud nodes. In this paper, we investigate the different multi dimensional resource scheduling algorithms used to achieve scheduling of various task and balancing load in Cloud Computing.

Keywords- scheduling, integrated load balancing, imbalance value, cloud computing

1. INTRODUCTION

Cloud load balancing is the distribution of workload over various available computing resources. Cloud load balancing increases resource availability and decreases costs associated with document management systems. Load balancers can perform a range of specialized runtime workload distribution functions, such as:

- Asymmetric Distribution **-** Computing resources with higher processing capacities are issued larger workloads.
- Workload Prioritization –Based on priority levels various scheduling processes are carried out.
- Content-Aware Distribution –As per the content of each request, they are provided with various computing resources

1.1 Importance of Load Balancing

Cloud computing benefits users with "cost, flexibility and availability of service users."[8]

The rise in the demand of Cloud services are directly the result of these advantages. Due to this high demand various technical issues like high availability and scalability in Internet of Services (IoS) and Service Oriented Architectures-style applications arise. Dynamic local workload are allocated across all nodes[10] by the load balancer which allows cloud computing to "scale up to increasing demands"[9]which solves a major concern in the issues stated.

1.2 Challenges in load Balancing

Modified resource allocation techniques and better improved strategies through efficient job scheduling are the main provisions of the load balancer. The load can be network load, CPU load, memory capacity or delay. Load balancer performs the distribution of load among various nodes in a distributed system while also being able to avoid a situation where some of the nodes are idle or less loaded while others are overloaded and to improve both utilization of resources and response time of jobs. At any instant of time load balancer ensures that equal amount of work is done by every node in the network or all the processors in the system. This is the important factor to be considered during the resource allocation but this has become more difficult especially in elastic cloud computing where the user can dynamically request for the resource.

The performance prediction also plays an essential role in load balancing. But cloud environment is highly variable and unpredictable. To increase resource utilization, providers try to oversubscribe as many users to a shared infrastructure. This results in resource contention and interference. Other factors that contribute to unpredictability of the environment include heterogeneity within the same instance type and administrative action (e.g., eviction) to maintain the service level. These make it extremely difficult to predict the performance variability and track down its causes.

2. PROBLEM STATEMENT

In a cloud data center, within certain time period there are M Physical Machines (PMs), also called hosts, which configuration may be heterogeneous. CPU, memory and network bandwidth of each host are considered as multi-dimensional resources. In Infrastructure as a Service (IaaS), each user requests a Virtual Machine (VM) represented in a vector r=(vC, vM, vN) where

[1] Asst Professor, Dept Of MSc-ST, AIMIT, St Aloysius College, Mangalore
[2] Student, MCA, AIMIT, St Aloysius College, Mangalore
[3] Student, MCA, AIMIT, St Aloysius College, Mangalore

vC, vM, vNis CPU, memory and network bandwidth requirement respectively. All virtual machines on a physical machine share CPU, memory and network bandwidth capacities provided by the physical machine.

2.1 Problem

Allocating and migrating virtual machines (VMs) which are reconfigurable and taking into consideration integrated features of hosting physical machine (PMs) are one of the challenging problems in scheduling resource in cloud data centres. This problem can also be defined as given a set of n requests (VMs) and a set of m identical machines (PMs) PM1, PM2, ...,PMm, each request has a processing time, the objective of load-balance is to balance load on every machine while they are being assigned requests.

3. LITERATURE SURVEY

Virtual machines demand is changing over time very highly in Cloud computing environment. So in this paper we consider dynamic load balancing scheduling when total number of physical servers is fixed. In this case, dynamic load balancing is conducted by allocating virtual machines to minimize current total imbalance-value.

3.1 ZHCJ Algorithm:

Wood et al. [1], introduced a few virtual machine migration techniques. One integrated load balance measurement is applied as follows:

$$V = \frac{1}{(1-CPU_u)(1-MEM_u)(1-NET_u)}$$

Where cpu, net and mem are the corresponding utilizations of that resource for the virtual or physical server. The higher the utilization of a resource, the greater the volume; if multiple resources are heavily utilized, the above product results in a correspondingly higher volume. The volume captures the degree of (over) load along multiple dimensions in a unified fashion and can be used by the mitigation algorithms to handle all resource hotspots in an identical manner. The algorithm always chooses physical machines with lowest referred V value and available resource to allocate virtual machines.

3.2 ZHJZ Algorithm:

Zheng et al [3]. proposed integrated load-balancing measurement as following:

$$B= \frac{aN1_iC_i}{N1_m C_m} + \frac{bN2_iM_i}{N2_m M_m} + \frac{cN3_iD_i}{N3_m D_m} + \frac{dN4_iNet_i}{Net_m}$$

The referred physical server m is selected firstly. Then other physical servers iis compared to server m. N1i is the CPU capability, N2i is for memory capability, N3i is for hard disk. Ci, Mi is for average utilization of CPU and memory respectively, Di is for transferring rate of hard disk, Neti is for network throughput. a, b, c, d is for weighting factor of memory, network bandwidth, CPU and hard disk respectively.

The algorithm selects a physical machine , and calculates the value and chooses lowest referred B value in different physical machines and available resource to allocate virtual machines.

3.3 LIF Algorithm:

WenhongTian et al [5]. Proposed new algorithm based on demands characteristics (for example, CPU intensive, high memory, high bandwidth requirements etc.), always selects lowest integrated imbalance value in different physical machines (as stated in equation (2-3)) and available resource to allocate virtual machines.

LIF algorithm considers imbalance values integrated CPU, memory and network bandwidth utilization, and the following parameters are considered:

Average CPU utilization $CPU_i^U$of a single server i: is averaged CPU utilization during observed period. For example, if the observed period is one minute and utilization of CPU is recorded every 10 seconds, and then $CPU_i^U$is the average of six recorded values of server i.

Average utilization of all CPUs in a Cloud datacenter. Let CPU_i^nbe the total number of CPUs of server i,

$$CPU_u^A=\frac{\sum_{i=1}^{N} CPU_i^U CPU_i^n}{\sum_{i=1}^{N} CPU_i^U} \qquad (1)$$

Where N is the total number of physical servers in a Cloud data center and CPU_i^nrepresents the number of CPUs in physical server i. Similarly, average utilization of memory, network bandwidth of server i, all memories and all network bandwidth in a Cloud data center can be defined asMEM_i^u,NET_i^u, MEM_u^A, NET_u^Arespectively.

Datacenter-wide integrated imbalance value ILBi, of server i. In statistics variance is used as a measurement of how far a set of numbers are spread out from each other, therefore it is widely used. Using variance, an integrated load imbalance value (ILBi) of server iis defined

$$\frac{(Avg_i - CPU_u^A)^2+(Avg_i - MEM_u^A)^2+(Avg_i - NET_u^A)^2}{3} \qquad (2)$$

Where

$$Avg_i = \frac{(CPU_i^u + MEM_i^u + NET_i^u)}{3} \qquad (3)$$

is average utilization of multi-dimensional resource in a physical machine, also called integrated load in LIF

3.4 Rand Algorithm:

Randomly assigns requests (virtual machines) to physical machines which have available resource.

3.5 Round Robin (RR):

The round-robin is one of most used algorithm for scheduling (for example by Amazon EC2 and Eucalyptus [4]), in which PM's are allocated VM's in turns. Simplicity in implementation is the advantage of this algorithm.

4. PROPOSED SOLUTION

The proposed solution is the modification of LIF algorithm .while calculation integrated load imbalance value (ILBi) of server i The variance is used as a measure of how far a set of numbers are spread out from each other in statistics. Using variance, an integrated load imbalance value (ILBi) of server i is defined

$$\frac{(CPU_u^U - CPU_u^A)^2 + (MEM_u^U - MEM_u^A)^2 + (NET_u^U - NET_u^A)^2}{3} \qquad (6)$$

Modified LIF algorithm is stated as follows

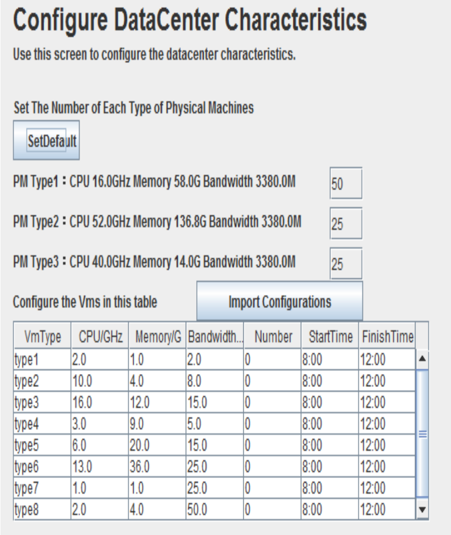
Algorithm**:** Modified Integrated Load First()
Input**:** Placement request r = **(**id, vC, vM, vN);
Status of current active tasks and PMs.
Output**:** Placement Scheme for r and IBL_tot.

```
Initialization: LowestAvg = number;
For i=1:N Do
If request r can be placed on PMi  ;
Then
    Compute Avgi utilization value of  PMi it using equation 1 and 6
          If  Avgi < LowestAvg
            Then
          LowestAvg = Avgi;
           Else
          Endif
     Else         //find next PM
Endfor
If LowestAvg == large number L//cannot allocate
  Put r into waiting queue or reject
Else
  allocatedPMID = i;
  Place r on PM with allocated PMID and compute IBL_tot
```

Fig 1 : Modified LIF Algorithm

5. EXPERIMENTAL RESULTS:

Algorithms are tested using the cloud tool –cloudsched. Experiment is conducted using the following DataCenter characteristics



Configure DataCenter Characteristics

Use this screen to configure the datacenter characteristics.

Set The Number of Each Type of Physical Machines

PM Type1 : CPU 16.0GHz Memory 58.0G Bandwidth 3380.0M 50

PM Type2 : CPU 52.0GHz Memory 136.8G Bandwidth 3380.0M 25

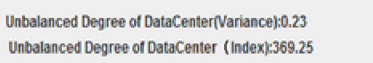
PM Type3 : CPU 40.0GHz Memory 14.0G Bandwidth 3380.0M 25

Configure the Vms in this table

VmType	CPU/GHz	Memory/G	Bandwidth...	Number	StartTime	FinishTime
type1	2.0	1.0	2.0	0	8:00	12:00
type2	10.0	4.0	8.0	0	8:00	12:00
type3	16.0	12.0	15.0	0	8:00	12:00
type4	3.0	9.0	5.0	0	8:00	12:00
type5	6.0	20.0	15.0	0	8:00	12:00
type6	13.0	36.0	25.0	0	8:00	12:00
type7	1.0	1.0	25.0	0	8:00	12:00
type8	2.0	4.0	50.0	0	8:00	12:00

Fig 2: Data Center characteristics

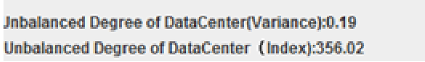
Random Algorithm



```
Unbalanced Degree of DataCenter(Variance):0.23
Unbalanced Degree of DataCenter（Index):369.25
```

Fig 3: Result using Random load balancing algorithm

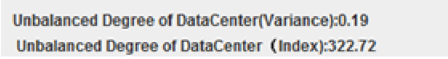
Round-Robin Algorithm



```
Jnbalanced Degree of DataCenter(Variance):0.19
Unbalanced Degree of DataCenter（Index):356.02
```

Fig 4: Result using Round-Robin load balancing algorithm

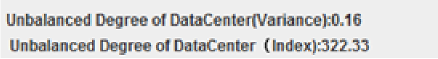
ZHJZ Algorithm



```
Unbalanced Degree of DataCenter(Variance):0.19
Unbalanced Degree of DataCenter（Index):322.72
```

Fig 5: Result using ZHJZ load balancing algorithm

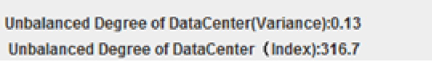
LIF Algorithm



```
Unbalanced Degree of DataCenter(Variance):0.16
Unbalanced Degree of DataCenter（Index):322.33
```

Fig 6: Result using LIF load balancing algorithm

Modified LIF Algorithm



```
Unbalanced Degree of DataCenter(Variance):0.13
Unbalanced Degree of DataCenter（Index):316.7
```

Fig 7: Result using Modified LIF load balancing algorithm

Cloudsched simulator generates different requests as follows: the total numbers of arrivals (requests)can be randomly set; all requests follow Poisson arrival process and have exponential length distribution; therefore to test the algorithm , it is executed six times and its average is taken as follows

	Unbalanced degree of Datacenter
Random Algorithm	0.24
Round-Robin Algorithm	0.20
ZHCJ Algorithm	0.21
ZHJZ Algorithm	0.19
LIF Algorithm	0.17
Modified LIF Algorithm(proposed algorithm)	0.15

Fig 7: The average of six experiments

6. CONCLUSION

Essential requirements of a dynamic resource scheduler is to have low Computational complexity, require little information about the system state, and be robust to changes in the traffic parameters. To meet these requirements, in this paper, we introduce a dynamic resource scheduling algorithm (Modified LIF) in Cloud data center by considering multi-dimensional resource. Modified LIF makes current total imbalance value of all servers in a Cloud datacenter the lowest..

7. ACKNOWLEDGEMENT

The authors would like to thank the dedicated research group in the area of Cloud & Grid Computing , wireless networks at the Dept of Computer Science, Mangalore University, Mangalore, India, for many simulating discussions for further improvement and future aspects of the Paper. Lastly but not least the author would like to thank everyone, including the anonymous reviewers

8. REFERENCES

[1] T. Wood et. al., Black-box and gray-box strategies for virtual machine migration, Proceedings of Symp. on Networked Systems Design and Implementation (NSDI), 2007

[2] W. Zhang, Research and Implementation of Elastic Network Service, Ph.D. Dissertation, National University of Defense Technology, China, 2000 (in Chinese)

[3] H. Zheng, L. Zhou, J. Wu, Design and implementation of load balancing in web server cluster system, Journal of Nanjing University of Aeronautics & Astronautics, Vol. 38, No. 3, Jun. 2006

[4] eucalyptus, www.eucalyptus.com

[5] Wenhong Tianet.al LIF: A Dynamic Scheduling Algorithm for Cloud DataCenters Considering Multi-dimensional Resources ,Journal of Information & Computational Science 10:12 (2013) 3925–3937

[6] A. Gulati, G. Shanmuganathan, A. Holler, Cloud-scale resource management: Challenges and techniques, Proceedings of HotCloud 2011, Portland, OR, USA, June 14-17, 2011

[7] A. Singh, M. Korupolu, D. Mohapatra, Server-storage virtualization: Integration and load balancing in data centers, Proceedings of the 2008 ACM/IEEE Conference on Supercomputing, 2008,1-12

[8] B. Wickremasinghe et al., CloudAnalyst: A CloudSim-based tool for modelling and analysis of large scale cloud computing environments, Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA 2010), Perth, Australia, April 20-23, 2010

[9] E. Arzuaga, D. R. Kaeli, Quantifying load imbalance on virtualized enterprise servers, Proceedings of WOSP/SIPEW’10, San Jose, California, USA, January 28-30, 2010

[10] W. Tian, Adaptive dimensioning of cloud data centers, Proceeding of the 8th IEEE International Conference on Dependable, Automatic and Secure Computing, DACS 2009, 2009