RESEARCH PAPER | Computer Science | Volume : 5 | Issue : 8 | August 2015 | ISSN - 2249-555X

Centralized Load Balancing Mechanism in Federated Cloud

KEYWORDS: Broker Manager, Federated Cloud, Load Distribution, Dispatcher

C.S. RAJARAJESWARI
Research Scholar, Bharathiar University, Tamil Nadu, India.

Dr. M. ARAMUDHAN
Associate Professor and Head, Department of Information Technology, PerunthalaivarKamarajar Institute of Engineering and Technology,Karaikal, India.

ABSTRACT *Cloud computing provides an efficient, cost effective and dynamic services to cloud users based on the QoS requirements. Federated cloud is the amalgamation of different cloud service providers. Load balancing among the cloud service providers is one of the complex tasks in federated cloud. Proposed Centralized Federated Load Balancing (CFLB) algorithm in thispaperprovides an efficient way to balance the workload among the cloud service providers in federated cloud through Broker Manager. Experimental result proves that the proposed work improves the performance of federated cloud model.*

Introduction

Load balancing mechanism is needed to distribute equal workloads to the cloud service providers to achieve optimum outset. Load balancing strategies may be either static or dynamic. Static strategies use information about the average performance of the system and the transfer decisions are independent of the current system state. Dynamic strategies use system state information to make load distribution decisions. Due to uneven job arrival patterns and unequal computing capabilities, some cloud service providers may be overloaded while others may be underutilized. Hence, load balancing is required to redistribute the workload among the available resources in order to achieve optimal resource utilization, maximize throughput, minimum response time and avoid overload.

In federated cloud model, load balancing techniques are used to balance the workloads of cloud service providers by distributing the workload among the brokers. Almost all the load balancing techniques are centralized decision making for forwarding the requests for execution in the federated cloud. The federated cloud architecture proposed by the authors in (Rajarajeswari et al., 2014) suggested the need of load balancing at the level of Broker Manager to promote QoS.

Related Work

In algorithms (Xu et al., 2009) the processes are divided between all processors. Each process is assigned to the processor in a round robin order. The process allocation order is maintained locally independent of the allocations from remote processors. Though the work load distributions between processors are equal but the job processing time for different processes are not same. So at any point of time some nodes may be heavily loaded and others remain idle.

Load balancing algorithm (Werstein et al., 2010) can also be based on least connection mechanism which is a part of dynamic scheduling algorithm. It needs to count the number of connections for each server dynamically to estimate the load.

Equally spread current execution algorithm (Nitika et al., 2012) process handle with priorities. it distribute the load randomly by checking the size and transfer the load to that virtual machine which is lightly loaded or handle that task easy and take less time , and give maximize throughput. It is spread spectrum technique in which the load balancer spread the load of the job in hand into multiple virtual machines.

Throttled algorithm (Nitika et al., 2012) is completely based on virtual machine. In this client first requesting the load balancer to check the right virtual machine which access that load easily and perform the operations which is give by the client or user. In this algorithm the client first requests the load balancer to find a suitable Virtual Machine to perform the required operation.

Min-Min Algorithm (Kokilavani, 2011) begins with a set of all unassigned tasks. First of all, minimum completion time for all tasks is found. Then among these minimum times the minimum value is selected which is the minimum time among all the tasks on any resources..

Max-Min is almost same as the min-min algorithm except the following: after finding out minimum execution times, the maximum value is selected which is the maximum time among all the tasks on any resources. (Ray et al., 2012)

Y. Fang et al (2010) discussed a two-level task scheduling mechanism based on load balancing to meet dynamic requirements of users and obtain high resource utilization. It achieves load balancing by first mapping tasks to virtual machines and then virtual machines to host resources thereby improving the task response time, resource utilization and overall performance of the cloud computing environment.

Modified Federated Cloud Architecture

At the start, users submit request to Broker Manager (BM) (Rajarajeswari et al., 2014). Dispatcher component is used to schedule the incoming requests of Broker Manager. Since the cloud traffic is dynamic and unpredictable, dispatcher distribute the incoming workload to the brokers with the help of Broker Monitoring Agent (BMA).

BMA monitors the workload of all the brokers and inform the status to broker manager. BMA also maintains a BLI (Broker Load Index) table which records the information such as broker-id, length of jobs in a waiting queue, length

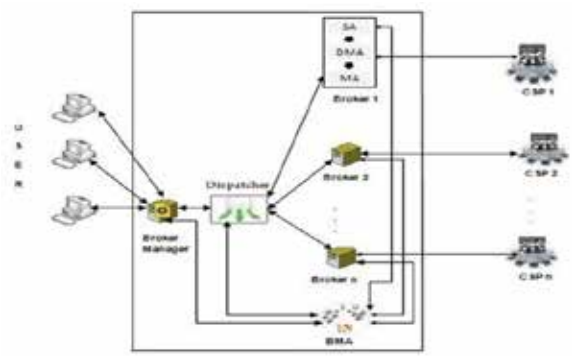
Volume : 5 | Issue : 8 | August 2015 | ISSN - 2249-555X

of jobs in service etc. for each broker. At each time t, BMA counts the number of request in the queue, and update into the load index table.

Dispatcher component uses this BLI table to schedule the incoming jobs. Dispatcher decision is based on the total time needed for the completion of the previous assigned request in the brokers. Dispatcher component is used to allocate or reallocate the incoming user request to the selected broker.

The functional model of the federated cloud broker architecture is shown in Fig. 1.

Fig 1: Federated Cloud Architecture



Centralized Federated Load Balancing Model

Broker Manager disseminates the incoming workload based on the current status and workload of brokers not on the past. BM uses the concept of stochastic Markov process, to select the service providers. A random process satisfying the Markov property can make predictions of the future process based on the present conditions but not on the past is called Markov process.

Existing federated cloud load balancing algorithm use the historical information about the cloud service provider to allocate the incoming request. Since the cloud processing is highly dynamic, historical based submission leads to incorrect solution.

The following Centralized Federated Load Balancing (CFLB) algorithm explains the processing sequence of centralized load balancing algorithm in federated cloud.

1. Cloud user submits a request to the broker manager(BM)
2. BM transfer a request to dispatcher(DIS)
3. Based on the current status of Cloud Service Providers (CSP) DIS chooses the CSP who matched with the user request using Markov process.
4. DIS send a Status Request (SR) message and the user request parameters to all CSP found in step 3
5. Each CSP send a Response status (RS) message to the LD. RS message consists of the status of CSP and acknowledgement message. Acknowledgement message consist of any one of the following values

Accept: CSP calculates the Execution Time (ET) (Job Waiting Time + Job Execution Time) of the incoming request. If ET is greater than the response time value given by user request then the CSP send a accept message to DIS

Reject: IF ET is less than response time value given by user then the CSP send a reject message to the DIS.

6. DIS receives a RS message from CSP's.
7. Based on the RS message load dispatcher selects the CSP whose acknowledgement status as "Accept"
8. DIS sort the CSP based on ET.
9. If DIS receives the same ET for more than one CSP value then select any one CSP at random
10. Assign a user task to the first CSP in the sorted list.

Algorithm 1: Centralized Federated Load Balancing

Experimental Results

On comparing the average job execution time of 3000, 6000 and 9000 jobs between the proposed CFLB and the existing historical based load balancing, the execution time is less in the CFLB when compared to other case. The following table shows the comparison of average execution time of the proposed and existing work.

Table 1: Average job Execution Time

No. of Jobs (Average job execution Time (in ms))	Proposed CFLB algorithm	Existing Load balancing algorithm
3000	142.35	166.66
6000	321.54	333.33
9000	538.47	554.66

Conclusion

CFLB algorithm reduces the average execution time of jobs and also improves the performance federated cloud architecture than compared to the existing federated cloud model. Load balancing at the level of brokers for efficient performance of the federated cloud architecture is considered to be the future part of the work.

REFERENCE

[1] P. Werstein, P.H. Situ, H and Z. Huang (2010). "Load balancing in a cluster computer". IEEE International Conference on Parallel and Distributed Computing, Applications and Technologies. PP:569-577. | [2] M. Nitika,G. Shweta and G. Raj (2012). "Comparative analysis of load balancing algorithms in cloud computing". International Journal of Advanced Research in Computer Engineering & Technology, 1(3), PP: 34-38. | [3] Y. Fang,F. Wang and J. Ge (2010). "A task scheduling algorithm based on load balancing in cloud computing. Web Information Systems and Mining" Lecture Notes in Computer Science. PP: 271-277. | [4] C.S. Rajarajeswari and M. Aramudhan, "Differentiated services at Application level for SLA resource provisioning management", Elsevier International conference on Mathematical Sciences, Chennai, India, 2014, PP:639-642. | [5] T. Kokilavani (2011). "Load balanced min-min algorithm for static meta-task scheduling in grid computing". International Journal of Computer Applications, 20(2), PP: 43-49. | [6] S. Ray and A.D. Sarkar(2012). "Execution analysis of load balancing algorithms in cloud computing environment". International Journal on Cloud Computing Services and Architecture, 2(5), PP:1-13. | [7] Zhong Xu, Rong Huang, (2009), "Performance Study of Load Balancing Algorithms in Distributed Web Server Systems", CS213 Parallel and Distributed Processing Project Report. | [8] C.S. Rajarajeswari and M. Aramudhan (2014). "Ranking model for SLA resource provisioning management". International Journal of Cloud Applications and Computing, IGI Global, 4(3), PP:68-80.