Proceedings of
INTERNATIONAL CONFERENCE ON INNOVATIVE TRENDS IN ENGINEERING RESEARCH (ICITER-2016)
International Journal of Innovations in Engineering, Research and Technology,IJIERT-ICITER-16,ISSN:2394-3696
26th June,2016

SLA AND IDLE SERVER MONITORING ALGORITHM WITH FEEDBACK IN CLOUD ENVIRONMENT

[PAPER ID ICITER-D191]

SANA J. SHAIKH,
M.E. Computer Department,
SAE, Pune University, Pune, India shaikh.sana669@gmail.com

PROF.S.B.RATHOD
Assistant Professor, Computer Department,
SAE, Pune University, Pune, Indiasbrathod.sae@sihgad.edu

ABSTRACT:

The scheduling algorithm plays vital role in day-today life. The load balancer can map task to resource that based on some particular objectives. The main objectives of load balancing are resource utilization and task completion. Cluster formation is done based on properties and processing power of server and assign task to first phase. In First phase, Service Level Agreement (SLA) algorithm determines priority of tasks, cost estimation and assign task to the respective cluster to second phase. In second phase, the Idle-server monitoring algorithm applies to check server is idle or not and result is forwarded to third phase which check whether task is get processed or not and reassignment of task will be done and analyze the result. The main aim is to understand the processing power and numbers of tasks are going to be processed by server to maximize throughput. This paper shows that maximum throughput by introducing Quality-of-Service in cloud environment.
KEYWORDS: Cloud computing, Quality of Service, Load balancing scheduling techniques, Load balancing algorithm.

INTRODUCTION:

The cloud load balancing is one type of load balancing method that is performed in cloud computing environment. Load balancing is process of distributing or dividing workloads across multiple computing system or resources. A load balancing reduces cost and maximizes availability of resources which is associated with document management systems. In order to suit user requirements, it uses a precise method to map the tasks to appropriate cloud resources, though by default maximum strategies are static in nature[6].

Whenever cluster formation is done then the cluster of server should be session-aware, so that any client connect to any cluster of servers at any time , the user gets unpredicted experience.[10] This is usually achieved with in-memory database or shared database. In distributed resources, scheduling problem is process that maps and manages the implementation of independent tasks. In order to meet the users specific need, process can provide appropriate resources to ensure that the workflow can be successfully completed.[6] Cloud Computing is state which gives proper and on-demand network access to shared pool of computing resources like network, storage, servers and services that are to be rapidly released with the efficient way in minimum management.[7]

At present, cloud computing is suffering from some challenges like security, QoS, Power Consumption and Load Balancing etc. Currently, as there is an increase in technology and consumer demands, there is excessive workload which calls for the need of the load balancer.[6] To balance the task properly the task should be get prioritize so that the tasks can be handled properly. The priority of task is depend upon the processing power of ant server or system. The processing power is calculated depend upon the hardware configuration such as input and output functionalities of system [6] [7].

The concept of balancing the load on the server on cloud has an important effect on performance.[10] The uneven distribution of load among the servers result in server overloading and may lead to crashing of servers. This degrades the performance of server. Load balancing is technique that distributes the load equally among the servers which avoid the overloading of server, server crashes and performance degrades. Load Balancing is an important factor that good response time, effective resource utilization. Thus the effective load balancing is needed. [6][10]

RELETED WORK:

This section describes the related work of QoS scheduling algorithm [6] in cloud environment. The main challenge of cloud computing is distribution of workload in well balanced manner. So the distribution should be done among the different nodes so that resources should be properly utilized. To optimize this problem, good load balancer should be used [1]. In distributed workflow, the process that can provide the appropriate resources to ensure that the workflow can be successfully completed in order to meet users need. In other words, the workflow scheduling algorithms are workflow instances of system instances by relevant rules and relational allocation of idle system resources so that the workflow can be easily implemented. The scheduling algorithms mainly have two types as: Market driven algorithm [6][9] and Performance driven algorithm[6][9].

The Performance Driven algorithm can optimize the performance of system without considering the cost and map the workflow tasks to resources according to policies. There are two representative algorithms of Performance driven algorithm as: Heterogeneous Earliest Finish Time algorithm[6] and throughput maximizing strategy[6]. The Market Driven scheduling algorithms manage resource allocation of any task and it considers the cost. The representative algorithms are Backtracking [5][9], Generic Algorithm[2][9], LOSS and GAIN algorithm[3][9], Deadline allocation algorithm (Deadline Distribution Algorithm)[4][9] and QoS based deadline allocation scheduling algorithm[6].

As we know the cloud has greatly simplified capacity provisioning process, it poses several challenges in the area of Quality-of Service (QoS) management. Quality of Service demoted the performance level, reliability and availability offered by infrastructure and application [9].

The cloud computing is technique where group of servers are distributed in data center that allows centralized data storage and online access to computing resources or services. As the request enters, it has to be distributed equally among the servers other- wise results in server overloading, performance degrades and not effective utilization of resources.[9] Effective load balancing technique improves response time of the task as well as utilizes the resources effectively.

A. BACKTRACKING:

Backtracking [9] is general algorithm that finds all the solution to some computational problem, notably constraints satisfaction problems, which incrementally builds candidates (backtracks) to the solutions and it determines that candidate cannot possibly be completed to valid solutions. Backtracking can be applied for different problems that admit the concept of partial candidate solution and relatively quick test of whether it can possibly be completed to valid solutions. Backtracking [5] is important method for solving problems such that crosswords, Sudoku and many other puzzles. It is most popular and convenient technique for parsing. But when the problem is large then it is very difficult to backtrack each problem to find solution and sometimes it becomes very time consuming job so the backtracking is not efficient for large problems.[5]

Disadvantages is that if data is large then it is very difficult to backtrack each problem to find solution and so this process is too time consuming and not reliable.

B. GENERIC ALGORITHMS:

By applying the principle of evolution, genetic algorithm provides robust search technique that allows a high-quality solution to be derived from a large space in given polynomial time. The Genetic Algorithm [2][9] always combines the exploitation of the best solutions from the past searches with the exploration of new regions of the solution space and solution of any problem in search space can be represented by individuals. So this algorithm is very popular . The fitness function in population determines the quality of individuals. A disadvantage of this scheduling algorithm is complex and time consuming so it is not reliable.

C. QDA SCHEDULING ALGORITHM:

A QoS- based Deadline Allocation algorithm [6]; QDA in short, considers cloud computing environment and the characteristics of workflow. The QDA algorithm [6] refers the main sub-deadline allocation criteria of Comprised Time Cost Scheduling Algorithm. The

CTC algorithm [4] uses QoS utility function value as a service resource selection condition and it takes user performance into account. QDA algorithm[6] takes set of work instances of task as input, perform the scheduling and generate output as instance set. First check whether any uncompleted task is present if yes then get first priority of execution and if no then the main task is get divided into some instances and depend upon performance evaluation techniques predict the Expected Execution Time[6] of various instances executing in each resource node and also calculate average execution time for different tasks.

After all this calculation, QDA algorithm[6] then assume the execution of task is done as Stream-mode, in this each task is get executed in FIFO manner. Depend upon this Utility function is get derived by considering the utility functions and the candidate set is created in ascending order. The allocation of all sub-tasks to its corresponding service resources is done and one round scheduling is executed.

Disadvantages are it uses the First In First Out and Stream based mode approach so that it takes time for execution because it uses sequential approach for execution and second is each time it divide task into some instances then this instances get executed one after anther and the previously executable tasks get first chance so this is not reliable.

PROPOSED METHOD:

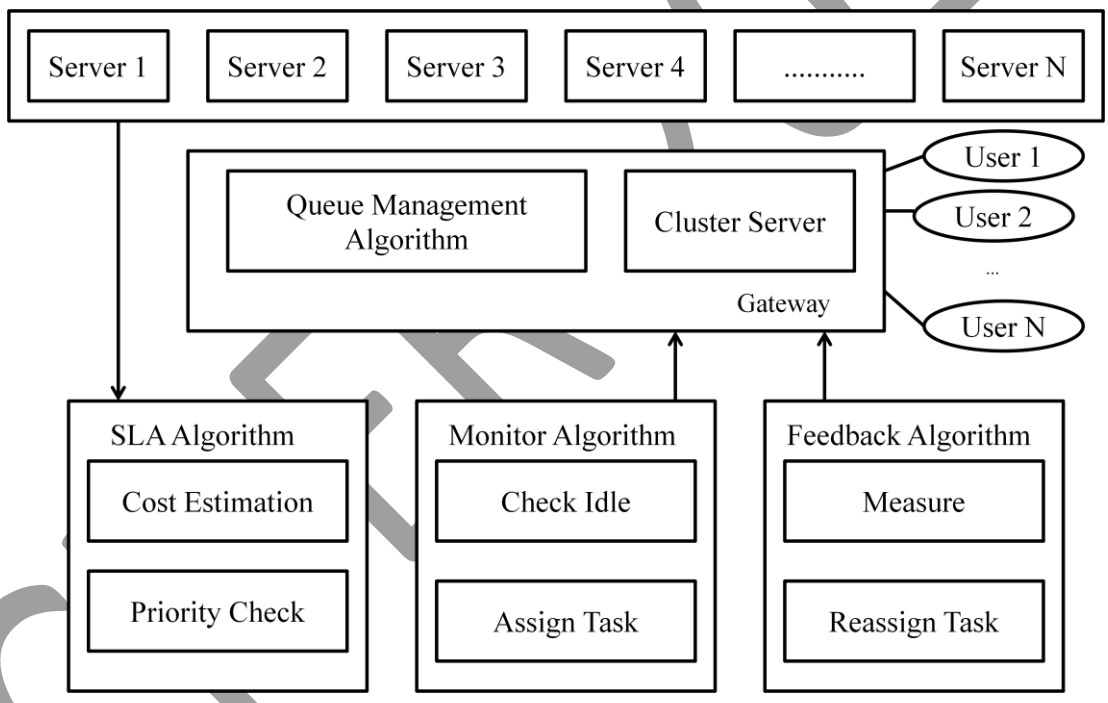


Fig 1. Proposed Methodology

Based on above discussion, the algorithms are having some disadvantages. The backtracking algorithm is not efficient and the generic algorithm is not reliable means it is complex and time consuming scheduling algorithm and the QoS scheduling algorithm proposed in this paper to overcome them. Currently, due to the increased usage of cloud, there is a tremendous increase in workload.

The uneven distribution of load among the servers results in server overloading and may lead to the server crash. This affects the performance. Cloud computing service providers can attract the customers and maximize their profit by providing Quality of Service (QoS). Providing both QoS and load balancing among the servers are the most challenging research issues.

Hence, in this paper, the framework is designed to offer both QoS and balancing the load among the servers in cloud. This paper proposes

three algorithms. First of all, the servers with different processing power are grouped together and forms different clusters. In the first stage, Service Level Agreement (SLA) based scheduling algorithm determines the priority of the tasks and assigns the tasks to the respective cluster. In the second stage, the Idle-Server Monitoring algorithm balances the load among the servers within each cluster and in third stage it measures any incomplete task is present into queue or not. if present then this task is get processed.
The proposed architecture provides better response time, waiting time, effective re- source utilization and balance load among the server as compare to other existing algorithm.

A. SLA BASED SCHEDULING ALGORITHM

In Service Level Agreement Algorithm , as per the priority of task , scheduling is done means whatever the input is accepted form user get executed in priority manner. The highest priority will get first chance. For computing the priority of task the some factors to be get considered as deadline, cast and task length.

B. IDLE-SERVER MONITORING ALGORITHM

The Idle Server Monitoring Algorithm run within each cluster to monitor servers and it checks any idle server in cluster. If this algorithm found any idle sever into cluster then it assign task to that identified server. If this algorithm does not found any idle server then the task is put into the Queue and maintain the status.

C. FEEDBACK ALGORITHM

The Feedback Algorithm perform monitoring of task and reassignment of task to server. Fr monitoring the task it will check queue continuously, if task is present into queue then cluster formation should be done to check the priority of task and as per the priority the task get distributed as per processing power. Then this algorithm checks for idle server into particulate cluster. If found then task get executed successfully.

D. K-MEANS CLUSTERING ALGORITHM

k-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed apriori. The main idea is to define k centers, one for each cluster. These centers should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest center. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as barycenter of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new center. A loop has been generated. As a result of this loop we may notice that the k centers change their location step by step until no more changes are done.

The working of proposed Architecture is as follows:

Step 1: The server 's cluster formation of done depend upon its processing power and Memory as High, Low and Medium. For that K-Means Clustering Algorithm used.

Let X={X1, X2, X3,.....,Xn} be the set of Data Points and V={V1, V2, V3.....n} be the set of Centers.

i. Randomly select cluster Centers and calculate the difference between data point and each cluster center.
ii. Assign the data point to cluster center whoes distance from cluster center is minimum of all the cluster centers.
iii. Recalculate the cluster center by using the formula :

$$v_i = (1/c_i) \sum_{j=1}^{c_i} x_i$$

Where Ci = Represent the number of data point s in i^{th} Cluster.

iv. Recalculate the distance between each data point and new obtained cluster centers.
v. If no any data point is reassign then stop, otherwise Repeat from Step ii.

Step 2: When task X_i arrives , SLA algorithm determine the priority of task by considering

Deadline:

$$EET = MI / MIPS \quad (1)$$

Where,

EET= Expected Execution Time

MI= Size of task

MIPS = Execution Speed.

Cost :

$$\sigma = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(\frac{\mu_i}{\mu} - 1\right)^2}$$

Where,

i = Load time of services.

μ = Average load time of all services.

m = total number of all services

Step 3**:** Task assignment is done based on their task priority as:

i. If X_i is a high priority, it is assigned to high power cluster.

ii. If X_i is an average priority, it is assigned to average power cluster.

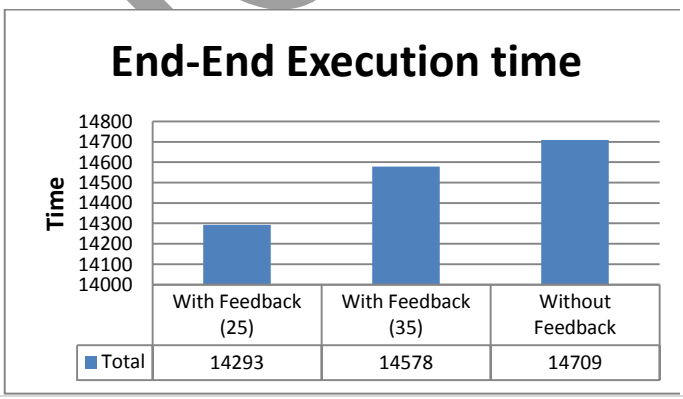
iii. If X_i is a low priority task, it is assigned to a low power cluster.

Step 4 : The working of Idle Server Monitoring Algorithm is done as: Check Idle server is present into in each cluster, if found then task is passed to server and if not the task is put into Queue.

Step 5 : The working of Feedback Algorithm is as follows: Check unprocessed task into queue, if found then do step 2 and step 3

Step 6: Steps 3 to step 5 repeated for each incoming task to get final result.

RESULT:



This section describes the performance of proposed algorithm. The result shows three different execution time of system depending upon threshold value. The first bar shows when threshold value is 25 it takes 14293 ms time for execution. When threshold value is 35 ms it takes 14578 ms for execution and without threshold value it takes 14709. With threshold value it will switch task if execution time of task exceeds threshold value. Si it gives better performance.

CONCLUSION:

In cloud computing environment, load balancing and scheduling are very wide concepts. In this paper we are specifically focused on load balancing by task shifting. In cloud computing, designing an algorithm with an aim to perform load balancing to resource in an optimized way has been a complicated task. The main aim of proposed algorithm is to satisfy both SLA and Load balancing among the servers. In this report, SLA based algorithm and Idle-Server Monitoring Algorithm and feedback algorithms are elaborated. SLA based scheduling algorithm schedules the tasks to the respective cluster based on processing power such as processor speed and memory and Idle-Server Monitoring Algorithm check whether any server in cluster is idle mode or not and feedback algorithm can shift task from over loaded server to idle server.

FUTURE WORK:

In future, we can add more I/O parameters. This will also work at kernel lavel where clustering in disk management is used. In which the latancy time should be calculated.

REFERENCES:

i. Madhurima Rana, Saurabh Bilgaiyan, Utsav Kar - *A Study on Load Balancing in Cloud Computing Environment Using Evolutionary and Swarm Based Algorithms*, 2014

ii. Yu J, Buyya R. *Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms.* Scientific Programming Journal, 2006, 14(3 /4): 217-230.

iii. Sakellriou R, Zhao H, Tsiakkour E, et al. *Scheduling workflows with budget constraints,* 05-22 Pisa, Italy:University of Pisa, Dipartimento di Informatica, 2005:347-357.

iv. Yu J, Buyya R, Than CK.A *cost-based scheduling of scientific workflow applications on utility grids. The 1st International Conference*

on E-Science and Grid Computing. Washington, DC: IEEE Computer Society, 2005: 140-147.

v. Menasc D A, Casalicchio E. *A framework for resource allocation in grid computing. Proceedings of 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems*. Washington, DC: IEEE Computer Society, 2004:259-267.

vi. Huifang Li, Siyuan Ge, Lu Zhang. A QoS- *based Scheduling algorithm for Instance-intensive Workflow in Cloud Environment*. 26th Chinese Control and Decision Conference (CCDC), 2014:4094-4099.

vii. Mark D. Ryan, ―*Cloud computing for Enterprise Architectures: Concepts*, Principles and Approaches‖, 2013

viii. Liu Ke, Jin Hai, Chen Jinjun, Liu Xiao, Yuan Dong, Yang Yun.A *compromised-time-cost scheduling algorithm in SwinDeW-C for instance-intensive cost-constrained workflows on a cloud computing platform*. International Journal of High Performance Computing Applications, 2010, 24(4): 445-456.

ix. A.Malcom Marshall ,Dr.S.Gunasekaran . *A Survey on QoS Constraint Based Scheduling Algorithms for cloud Workflows*. CONFERENCE PAPER · MARCH 2014 DOI: 10.13140/2.1.3958.7203

x. Ektemal Al-Rayis, Heba Kurdi. *Performance Analysis of Load Balancing Architectures in Cloud Computing*. 2013 European Modelling Symposium. 978-1-4799-2578-0/13.