A Comparative Analysis of Task Scheduling Approaches for Cloud Environment

Anurag Jain
Assistant Professor, Computer Science Department,
Geeta Institute of Management and Technology
Kurukshetra, India
Email id: er.anuragjain14@gmail.com

Rajneesh Kumar
Professor, Computer Science Department,
MMEC, Maharishi Markandeshwar University
Ambala, India
Email id: drrajneeshgujral@mmumullana.org

***Abstract*—Cloud computing is still an emerging technology whose future depends on optimum resource utilization, efficient task management & effective installation of infrastructure. It will assist in attaining improved fault tolerance ratio and satisfaction of service level agreement in a better manner. Load balancing approaches plays a vital role in achieving higher resource utilization & user satisfaction ratio. It can be done through task scheduling, resource allocation and task migration. Objective of this paper is to evaluate several suitable strategies for task scheduling in cloud environment using cloud analyst simulator.**

Keywords—Cloud Analyst, Cloud Computing, Join Idle Queue, Join Shortest Queue, Load Balancing, Minimum Completion Time, Minimum Execution Time, Task Scheduling

I. INTRODUCTION

Cloud computing means service delivered through internet. This service can be in the form of software and hardware both. Due to ubiquitous, scalable and pay per usage feature of cloud computing, its user base is growing exponentially. When services are provided as software then delivery model is called Software as a service. When services are provided as a platform then delivery model is called Platform as a service. When services are provided as hardware then delivery model is called Infrastructure as a service. Based upon the implementation of the service delivery model, we can have public, private and hybrid cloud. Due to unpredictable behavior of job arrival and huge exchange of data over network, sometimes resources are not utilized in balance way and this results in delay or failure of service. To avoid this situation there is a need of efficient load balancing algorithm at various levels [1, 2].

In this paper, authors have done the thorough analysis of task scheduling approaches used in cloud environment for load balancing purpose. Cloud Analyst simulator has been used to do the comparison on various metrics suitable for cloud environment. Organization of paper is as follows: Section II covers the fundamentals of load balancing, its types, and metrics for cloud environment. Section III gives an overview of simulator used. Section IV gives the detail description of task scheduling approaches suitable for load balancing in cloud environment. Results and comparative analysis of task scheduling techniques (that are discussed in section IV) has been given in section V. This has been done using cloud analyst simulator. Conclusion and future scope has been mentioned in section VI.

II. LOAD BALANCING BASICS

Load balancing involves distribution of jobs amongst multiple computers, processes, disk or other resources so that resources are utilized in optimal manner and computation time also get minimized. It can be done at data center level, host level and virtual machine level. Cloud computing future depends on effective installation of infrastructure, efficient utilization of resources and dynamic transformation of load. So, there is a strong need of efficient and effective load balancing strategy which satisfies the need of cloud environment. Load balancing can be done through task scheduling, resource allocation and task migration.

A. Load Balancing Types

Broadly load balancing strategies for cloud environment can be divided into following categories [3]:

- Centralized Approach: This approach is suitable for homogeneous and non dynamic environment. In these types of techniques, single scheduler manages all the resource allocation and task assignment activities.
- Distributed Approach: In these types of techniques, every node not only maintain their individual load vector but also maintain the information about load and resource of adjacent nodes. In case of overloading, a node can transfer its load to any under loaded node. This approach is suitable for distributed and dynamic environment.
- Mixed approach: To get the benefits of centralized and distributed approach, mixed mode approach is used. Scheduling is implemented at two levels. One approach is implemented at first level and other approach is implemented at second level. This is the most appropriate approach for cloud environment [4].

B. Metrics

Efficiency of a load balancing approach can be analyzed on the following parameters [5]:

- Response time: It shows the time taken by a scheduling approach to respond a task in the system. Its least value is desirable.

978-9-3805-4421-2/16/$31.00

- Cost: It involves the migration cost, processing cost and storage cost. An efficient algorithm always tries to minimize this factor.
- Throughput: This parameter tells us about the number of tasks completed per unit of time. Its higher value is desirable.
- Scalability: This parameter shows how algorithm handles the variation in number of tasks. If algorithm handles the jobs efficiently in such environment then it is called scalable.
- Fault tolerance: It shows the algorithm's potential to handle the situation of fault and its recovery power from the failure.

III. CLOUD ANALYST SIMULATOR

Cloud analyst is a cloud simulator through which user can test the various task scheduling approaches in the cloud environments. It has a graphical user interface which makes it easy to configure the simulation environment. It can also generate the results in the form of graphs. It has been developed using cloudsim architecture.

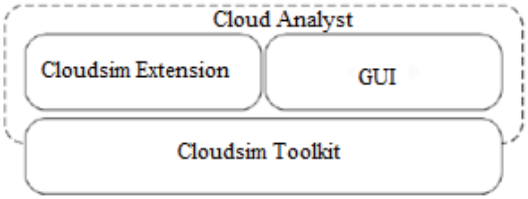

Fig. 1. Cloud Analyst Structure [6]

A. Components of Cloud Analyst

- VMLoadBalancer: This module plays a very crucial role as it simulates the various load balancing approaches which are used for task scheduling or task migration from one VM to another.
- Simulation: This component handles the management of various parameters related to simulation, generates & executes the simulation.
- UserBase: By generating the varying traffic as per the configuration of simulation parameter, this component simulates a user base.
- CloudAppServiceBroker: This module simulates the service brokers which controls the routing of traffic among user bases and data centers.
- Internet: This module simulates the Internet & simulates the traffic routing actions.
- Internet Characteristics: This component is responsible for management of internet characteristics during simulation. It includes the latencies & accessible bandwidths among regions and present performance level data for the data centers.
- Data Center Controller: This component manages the activities related to data centre.
- GUI Package: This package provides graphical user interface, which work as the front end manager for managing screen transitions, application, and additional user interface activities [6].

IV. TASK SCHEDULING APPROACHES

This section includes the in depth analysis of various tasks scheduling approaches:

Yi Lu et al. in [7] have discussed the Join Idle Queue (JIQ) scheduling approach for load balancing. Authors have implemented a two level scheduling. To realize the concept of two levels scheduling, authors have used the distributed scheduler. Number of schedulers is very less in comparison to number of virtual machines. Every scheduler maintains a queue of idle virtual machines. On receiving a task, scheduler first consults its idle queue. If it finds any virtual machine which is idle then it immediately assigns the task to that virtual machine and removes that virtual machine from its idle queue. If it does not find any idle virtual machine then it randomly allot that task to any virtual machine. Virtual machine after job completion, update about its status to any of the randomly chosen idle queue associated with a scheduler. This approach has separated the task of discovery of idle servers from the task of job assignment to virtual machine.

Algorithm JIQ()

{

Every Scheduler maintains the list of idle virtual machines in its idle queue;

While (there is a task received by cloud broker)

{

Cloud-broker forwards the task randomly towards any of the scheduler;

On receiving a task, scheduler checks its idle queue;

If (idle queue is not empty) then

{

Scheduler removes the idle VM from the queue and assigns the received task to that VM;

}

else

{

Scheduler assigns the task to any randomly selected VM;

}

If (any virtual server get idle) then

{

VM randomly selects scheduler;

Add itself to the idle queue of that scheduler;

}

}

}

Varun Gupta et al. in [8] have discussed the Join Shortest Queue (JSQ) scheduling approach for load balancing in distributed environment. This approach uses only single scheduler, which dispatch the task towards that virtual machine whose queue length is small. Scheduler maintains a VM allocation table which stores the queue length corresponding at each virtual machine. This helps scheduler to redirect the received task towards a suitable virtual machine. No queues are maintained at scheduler level. Queues are maintained only at virtual machine level.

Algorithm JSQ()

{

Scheduler initializes the VM allocation table;

while (there is a task received by JSQ scheduler)

{

Scheduler forwards the task towards that VM whose queue length is smallest & update VM allocation table;

If (any virtual machine completes the task)

{

Update the VM allocation table accordingly;

}

}

}

G. Ritchie et al. in [9] have discussed the Minimum Completion Time (MCT) approach for load balancing. Tasks are assigned to resources in first come first serve manner. The virtual machine which requires minimum completion time for current task is scheduled first. For assignment of task to virtual machine, MCT Scheduler accesses the VM allocation table. VM allocation table stores the virtual machine id, virtual machine power, number of tasks in queue & completion time of that virtual machine. This approach is dynamic in nature as it considers the current load of virtual machine.

Algorithm MCT()

{

Populate the VM allocation table;

While (there is a task received by MCT scheduler)

{

Choose that virtual machine from VM allocation table whose completion time is lowest;

Assign the received task to selected machine;

Update the VM allocation table;

}

}

R. Armstrong et al. in [10] have discussed the Minimum Execution Time (MET) approach for task scheduling. In this approach, tasks are assigned to resources in first come first serve manner. The virtual machine which requires minimum Execution Time (ET) for current task is scheduled first. MET Scheduler accesses the VM allocation table to assign task to virtual machine. VM allocation table stores the virtual machine id and virtual machine power. Virtual machine with more processing power can execute the task fast. So this centralized load balancing approach is static in nature which neither considers the present load nor considers the task size.

Algorithm MET()

{

Populate the VM allocation table;

While (there is a task received by MET scheduler)

{

Choose that virtual machine from VM allocation table whose processing power is highest;

Assign the received task to selected machine;

}

}

V. RESULTS & ANALYSIS

A. Simulation Configurationt

Simulation Duration: 5 hrs

Service Broker Policy: Optimize response time

Cost Model:

- Virtual machine usage cost per hour = $ 0.1
- Memory usage cost per second = $ 0.05
- Data storage cost on cloud per second = $ 0.1
- Data transfer cost per 1GB (from cloud to internet or vice versa) = $ 0.1

Grouping Model:

- No of concurrent users from a single user base= 500
- No of concurrent requests a distinct application server instance can sustain= 50
- Length of executable instruction per request= 200 bytes

Resource Scheduler Policy to Virtual Machine: Time-shared

Data Center hardware configuration:

- No of processors on physical machine= 2
- Processing power= 5000 MIPS
- Storage capacity= 50000MB
- Primary memory= 512 MB
- Internal bandwidth= 10000 MBPS

VM specification in Data Center:

- RAM = 128 MB
- Architecture = x86
- Operating system = Linux
- Storage quota= 5000 MB
- Virtualization technique = Xen
- Bandwidth = 1000 MBPS

User Base Specification:

TABLE I. USER BASE CONFIGURATION [11]

User Base	Region	No of Requests	Data Size	Peak Hour Start Time	Peak Hour End Time	No of users
UB1	0	50	80	11	13	20000
UB2	1	45	120	17	19	100000
UB3	2	60	160	18	20	60000
UB4	3	75	145	3	5	40000
UB5	4	35	105	19	21	80000
UB6	5	30	115	14	16	120000

B. Results

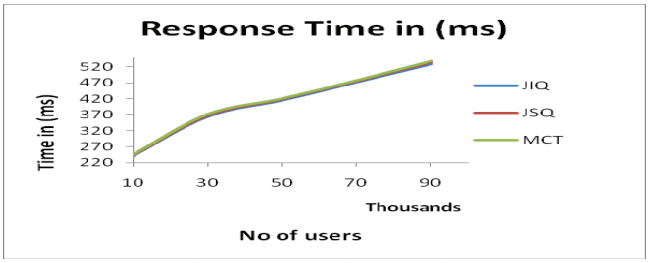

Fig. 2. Response Time Comparison

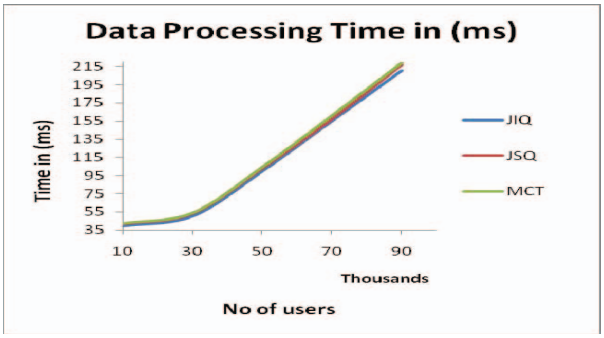

Fig. 3. Data Processing Time Comparison

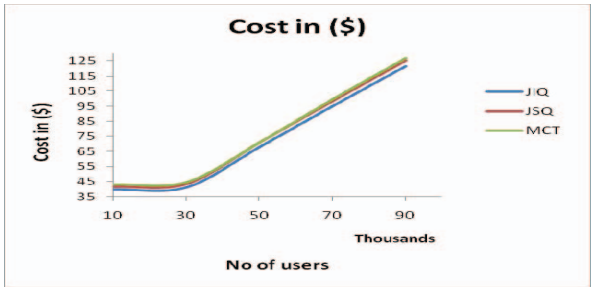

Fig. 4. Cost Wise Comparison

C. Analysis

TABLE II. ANALYSIS OF DISCUSSED APPROACHES

Scheduling Approach	Remarks
JIQ	• Uses multiple schedulers. • Distributed in nature • Does not consider the present load of VM. • Does task scheduling in first come first serve manner. • Uses a Queue to store the record of idle VM.
JSQ	• Centralized in nature. • Considers the present load in terms of queue length. • Does task scheduling in first come first serve manner. • Uses Single Scheduler. • Uses VM allocation table to store VM id & corresponding queue length.
MCT	• Centralized & Dynamic in nature as it considers the present load in terms of queue length & VM power to estimate the task completion time. • Does task scheduling in first come first serve manner. • Uses Single Scheduler. • Uses VM allocation table to store VM id, VM power & corresponding queue length.
MET	• Centralized & Static in nature as it does not consider the present load of VM. • Does task scheduling in first come first serve manner. • Uses Single Scheduler • Uses VM allocation table to store VM id, VM power.

D. Regression Analysis(R)

Regression analysis has been done as this helps one to establish the relationship between the variables so that the value of dependent variable can be predicted from the given values of independent variables. It has been done using Microsoft excel. No. of users are considered as an independent variable while response time, data processing time and cost is considered as dependent variable. It has been found that the value of R in all cases is very close to one. Also there is not much deviation between predicted value and actual value in all the cases. This indicates about the correctness of the result [12].

TABLE III. REGRESSION VALUE ANALYSIS

Regression Analysis	JIQ	JSQ	MCT
Value of R for Response Time	0.980872	0.982156	0.980958
Value of R for Data Processing Time	0.980805	0.980185	0.980473
Value of R for Cost	0.97363	0.974742	0.97326

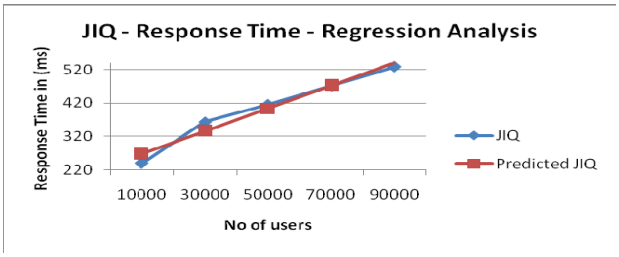

Fig. 5. Response Time Regression Analysis for JIQ Approach

Fig. 6. Response Time Regression Analysis for JSQ Approach

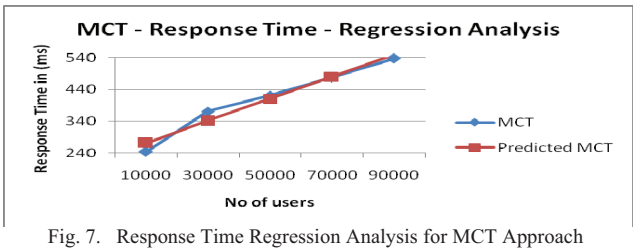

Fig. 7. Response Time Regression Analysis for MCT Approach

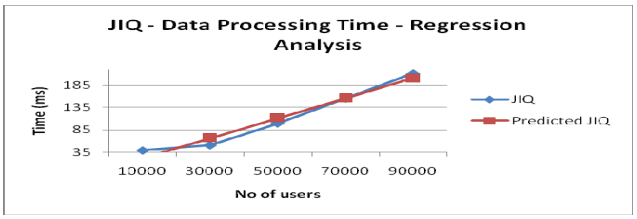

Fig. 8. Data Processing Time Regression Analysis for JIQ Approach

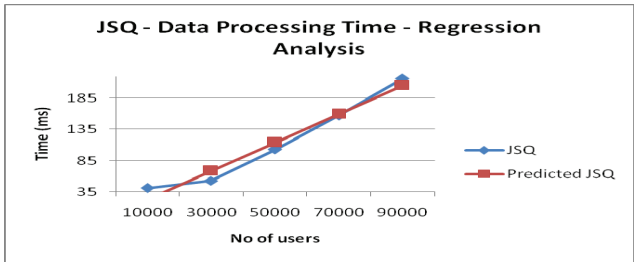

Fig. 9. Data Processing Time Regression Analysis for JSQ Approach

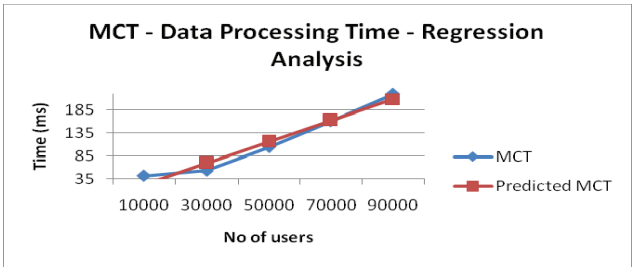

Fig. 10. Data Processing Time Regression Analysis for MCT Approach

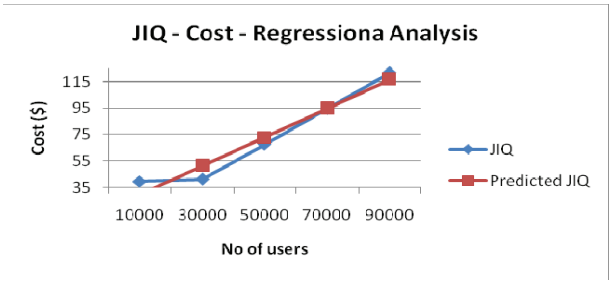

Fig. 11. Cost Regression Analysis for JIQ Approach

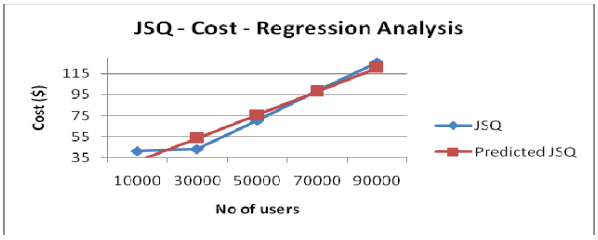

Fig. 12. Cost Regression Analysis for JSQ Approach

Fig. 13. Cost Regression Analysis for MCT Approach

VI. CONCLUSION & FUTURE SCOPE

Load balancing is one of the key challenge in cloud computing. It will help in achieving better fault tolerance ratio and higher customer satisfaction. In this paper, authors have discussed the pros and cons of task scheduling algorithm for load balancing in cloud environment through simulation. It has been found that Join Idle Queue approach is better for cloud environment among the discussed approaches. Results have been validated by regression analysis.

Combination of algorithms is an important task. So as a future work authors are planning to propose a new load balancing approach by merging some of the existing approaches to get the best one.

REFERENCES

[1]. A. Jain and R. Kumar, "A Taxonomy of Cloud Computing", International Journal of Scientific and Research Publications, Vol. 4, No. 7, pp. 1-5, July 2014.

[2]. P. Sasikala, "Cloud computing: Present Status and Future Implications' International Journal of Cloud Computing, Vol. 1, No. 1, pp. 23-36, 2011.

[3]. A. Khiyaita, M. Zbakh, H. El Bakkali, and D. El Kettani, "Load balancing Cloud Computing: State of Art", In Proceedings of 2nd National Days of Network Security and Systems (JNS2), Marrakech, Moroccopp, 106-109, April 2012.

[4]. K. A. Nuaimi, N. Mohamed, M. A. Nuaimi and J. Al-Jaroodi, "A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms", In Proceedings of IEEE Second Symposium on Network Cloud Computing and Applications (NCCA), London, UK, pp. 137-142, December 2012.

[5]. Amandeep, V. Yadav and F. Mohammad, "Different Strategies for Load Balancing in Cloud Computing Environment: A Critical Study", International Journal of Scientific Research Engineering & Technology (IJSRET), Vol. 3, No. 1, pp. 85-90, April 2014.

[6]. B. Wickremasinghe, R. N. Calheiros and R. Buyya, "Cloudanalyst: A Cloudsim-Based Visual Modeller for Analysing Cloud Computing Environments and Applications", Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA 2010), Perth, Australia, pp. 446-452, April 2010.

[7]. Y. Lu, Q. Xie, G. Kliot, A. Geller, J. Larus and A. Greenberg, "Join-Idle-Queue: A Novel Load Balancing Algorithm for Dynamically Scalable Web Services", Elsevier Science Publishers, Vol. 68, No. 11, pp. 1056-1071, November 2011.

[8]. V. Gupta, M. Harchol-Balter, K. Sigman and W. Whitt, "Analysis of Join-the-Shortest-Queue Routing for Web Server Farms", International Symposium on Computer Modelling, Measurement and Evaluation, Elsevier Science Publishers, Vol. 64, No. 9, pp. 1062-1081, October 2007.

[9]. G. Ritchie and J. Levine, "A Fast, Effective Local Search for Scheduling Independent Jobs in Heterogeneous Computing Environments", Journal of Computer Applications, Vol. 25, No. 5, pp. 1190-1196, 2005.

[10]. R. Armstrong, D. Hensgen and T. Kidd, "The Relative Performance of various mapping algorithms is independent of sizable variances in run-time predictions", In Proceedings of Seventh Heterogeneous Computing Workshop (HCW 98), Orlando, USA, pp. 79-87, 1998.

[11]. http://www.internetworldstats.com as on August 2015.

[12]. S. R. Jena, S. Padhy and B. K. Garg, " Performance Evaluation of Load Balancing Algorithms on Cloud Data Centers", International Journal of Scientific & Engineering Research, Vol. 5, No. 3, pp. 1137-1145, March-2014.