Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 4, April 2015, pg.663 – 669

RESEARCH ARTICLE

A Two Phase Fair Dynamic Job Assignment Strategy for Cloud Environment

Pankaj Bhatt[1], Avita katal[2], Pooja Pokhriyal[3], Susil Mishra[4]
[1]Dept.of CSE, at GEHU – Dehradun, India, [2]Dept.of CSE, at GEHU – Dehradun, India
[3]Dept.of CSE, at GEHU – Dehradun, India, [4]Dept.of CSE, at GEHU – Dehradun, India
[1] pankaj03bhatt@gmail.com; [2] avita207@gmail.com; [3] poojapokhriyal30@gmail.com; [4] sushiljune@gmail.com

Abstract— The Scheduling optimization mechanism in heterogeneous cloud is a NP-hard Type problem where all incoming jobs are to be arranged and mapped in such an efficient order with respect to available resources so that it could provide maximum, fair and flexible utilization of resources. As cloud system is almost heavily loaded all the time with a large number of requests that are coming at a regular or irregular interval from different sources. So it becomes necessary to take care so that a proper pattern for job assignment and grouping among jobs is created otherwise it may cause a strong risk to system performance resulting in sudden failure of system resources , as well as some serious problems like starvation and problems due to job communication. In this paper we have provided a solution for proper task ordering, arrangement of jobs and further for scheduling in dynamic cloud environment. We have used a different strategy for job scheduling in cloud environment which involves Linear Job Rearrangement Scheme for best Job ordering and Granularity Size based fair dynamic Job scheduling policy which is strongly beneficial for overall system balanced performance

Keywords— two phase, dynamic job scheduling, fair job assignment, cloud scheduling

I. INTRODUCTION

One of the most important features of Cloud Computing is the use of Virtualization technique to provide user on Demand resources as pay per use model. Cloud has become a major solution provider for flexible, dynamic and on demand computing infrastructure through virtualized resource provision.

Scheduling is the mechanism of scheduling the requests or jobs of user according to the availability of resources in cloud environment in accordance with condition, requirement and uniqueness of the tasks.

Scheduling algorithms have specific goals and they include high Throughput, efficient Resource utilization, quality of services, less energy consumption etc.

The process of scheduling in cloud basically consists of three levels namely:

- **Resource Discovery and Filtering**– This level is concerned with the discovery of resources and collection of status information by Data Center Broker.
- **Resource Selection** – This level is responsible for selection of jobs based on certain parameters which is the most crucial and deciding factor.
- **Task submission –** This level is concerned with job submission to the resource selected..

The paper is divided into 5 sections. Related work is briefly discussed in section 2. Section 3 presents the proposed Strategy followed by Experimentation and results in section 4. Section 5 includes the conclusion and the future work.

© 2015, IJCSMC All Rights Reserved

II. Related work

Scheduling in cloud environment is one of the most challenging and interesting areas of research. Currently several researchers are working to discover a better scheduling approach. A number of scheduling algorithms for cloud environment had already been proposed with respect to cloud and grid systems, it is very often seen that the scheduling policy is designed to meet certain specific requirements of the system. In distributed systems task scheduling usually has the goal of distributing the load on processors and maximizing their utilization while minimizing the execution time for global tasks.

A Content Based Scheduling Algorithm has been is proposed in [1] that is capable to reduce the network traffic associated with transfer of VMs from storage area to host area and thus improve the system performance by maintaining the load balance. A Deadline based Resource provisioning and scheduling algorithm is discussed in paper [2] in which it is clarified that in cloud environment the execution of workflow is planned by two stages first one is the resource provisioning phase and second one is schedule generator phase. The selection of resources and mapping of task is done according to user defined quality of service (QoS).

Traditionally FCFS, SJF, Greedy approach and prioritization based approach is used for task scheduling but as the research in cloud is going towards the increasing level, it is discovered that the efficiency in cloud scheduling is not only measured just by the task completion time but also considers the balanced response i.e. System throughput. Clearly saying it considers that up to how much extent a cloud system is capable of handling job assignment overhead, communication overhead, starvation etc

An efficient TPD scheduling is introduced in paper [3] which is based on the task selection and priority distribution strategy to manage scheduling efficiently. In paper [5] authors have taken into consideration the Best fit strategy for request assignment in Dynamic cloud. A Linear scheduling approach is proposed for task and available resources which is designed to maximize the resource utilization and hence improve the overall system performance. In paper [6] authors have suggested that by maintaining priority queue for task ordering the problem of starvation can be solved.

Task scheduling is generally of two types: Static and Dynamic scheduling. Static scheduling works on the concept of pre-scheduling where there is no matter for dependency for current system and also have apriori knowledge of system resources to perform scheduling action. On the other hand Dynamic Scheduling algorithms take decision concerning load balancing based upon the current state of the system. Authors in paper [7] have proposed a three phases scheduling approach for cloud environment. In this study it is suggested that to reach the load balance and reduce the execution time, best task ordering must be maintained. For this at the initial phase a BTO scheduler is introduced which can be different according to characteristics of task and requirement of overall scheduling policy. In paper [8] authors have suggested a Dynamic Job Grouping based scheduling for fine grained tasks on global grids, where a granularity size factor is introduced to setup groups of jobs.

The conclusion of all the related work studied is that in cloud scheduling the optimization is only effective if specific best task ordering is maintained which results in maximizing the utilization of the available resources. Best task ordering may be different according to the broker's own understanding of the specific cloud system.
In our approach we are providing a Job Scheduling strategy in which first job are selected and filtered. Then groups are created based on Granularity size and are scheduled to the corresponding resources.

III. Proposed Methodology

In our proposed algorithm we focus on scheduling of jobs with the aim to reduce the risk of job assignment overhead condition and maximize the utilization of cloud resources in order to achieve the minimum job execution time and cost.

In the first phase for the best task ordering, the proposed system introduces a simple job arrangement strategy which provides a better flexibility for job selection pattern to filter the all incoming requests. This is significant in terms of reducing the risk of job communication and assignment overhead. In second phase a Granularity Size based dynamic job grouping strategy is used for distributing the tasks among available resources in order to perform a fast and better utilization of available virtual machines which in turn reduces the overhead time by making jobs groups.

© 2015, IJCSMC All Rights Reserved

A. *First Phase: Linear Job Rearrangement Scheme (LJRS)*

According to Linear Job Rearrangement Scheme for job rearrangement all incoming requests are collected at a regular interval and rearranged in a list in ascending order of MI / cloudlet length . A threshold value (In MI - million instruction) based distribution of cloudlet (jobs) is performed and two sub lists are generated. A third list of cloudlets (cloudlet_ListArranged) is generated in which one lowest MI cloudlet from first sub list and one highest MI cloudlet from second sub list is selected and this selection of cloudlets is done up to the length of the two sorted lists.

The Linear Job Rearrangement Scheme can be made more effective by regulating the pattern of distribution of cloudlet, number of sub list generation and selection pattern of cloudlet. Since the cloudlets are selected for execution from the entire sub list the basic problem of prioritization i.e. starvation is solved here.

The Linear Job Rearrangement Scheme of job selection for best task ordering can be differ according to cloud broker's own interest. This scheme can be used as a job priority base filtration according to broker understanding. Fig. 3.1 shows a sample of Linear Job Rearrangement scheme for a set of 20 jobs/cloudlets.

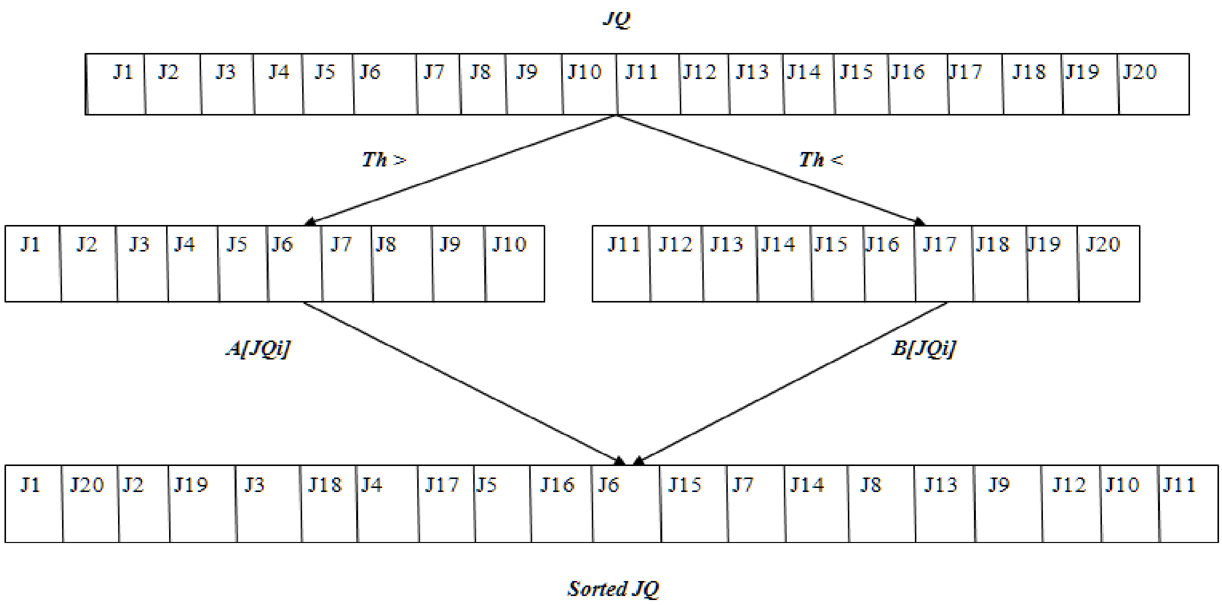

Fig .3.1.An Example of Linear Job Rearrangement Scheme

B. *Second Phase: Granularity Size Base Task Scheduling.*

The cloudlets from the output cloudlet_ListArranged from Linear Job Rearrangement Scheme are used to form groups with the help of Granularity size. Granularity size is a user defined parameter which is used to measure the total number of jobs that can be completed within a specified period of time.

1. Set a Granularity Size and Calculate the Total MIPS (Millions of Instructions per Second) of each available resource. i.e.
   Total MIPS = Resource MIPS * Granularity Size
2. The scheduler Groups the jobs by adding MI/ length of jobs from cloudlet_ListArranged and comparing the resulting jobs Total MI with the resource Total MIPS i.e.
   Resource Total MI >= Total MI of Jobs
3. Assign the unique ID to all the groups created from the above step and place the created groups to the Group job List
4. Continue this process of group assignment of jobs until the selection of all cloudlets from cloudlet_ListArranged is done.
5. Send all the groups to their respective target resource for execution.

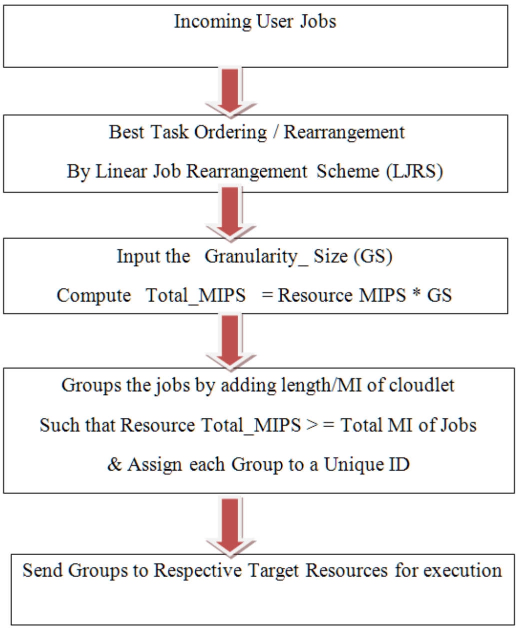

Fig. 3.2 Two Phases Fair Dynamic Job Scheduling

C. *Algorithm Pseudocode*

```
Input Cloudlet List   cloudlet List
Where Ci: cloudlet, Li (in MI): cloudlet length 0< i< n n :number of cloudlets.
Sort (cloudlet List) returns sorted List cloudlet_ListSorted // using Collection. sort()

  Pseudo code Rearrange Cloudlets ()
 // Threshold value calculation
 Rearrange Cloudlet (cloudlet_ListSorted)
   For (i=0;i<n;i++)
       add_Length= add_Length + Li
    Threshold= add_Length /n

/*Two cloudlet Lists formed one having values less than threshold and one having values more
than threshold.*/
 For (i=0; i<n; i++)
If (Li< Threshold)
Add Ci to cloudlet_ListSorted1
Else add Ci to cloudlet_ListSorted2
End For

// Grouping:  Linear Job Rearrangement Scheme; final list created
While ( j< cloudlet_ListSorted1.Size & & cloudlet_ListSorted2.Size)
Add cloudlet_ListSorted1.Cj and cloudlet_ListSorted2.Ck to cloudlet_ListArranged
j=j+1
k=k+1
 End of while

Pseudo code Cloudlets_Submission ()
// Grouping and Submitting to VM
// set Granuality size and Vm Total MIPS
Set GRANUALITY  = c                                   // c is a constant
For ( i=0; i< cloudlet_ListArranged .Size; i++)
VM_totalMIPSindex = c * VM_MIPSindex
   While (i< cloudlet_ListArranged .Size &&   VM_totalMIPSindex  > cloudlet_ListArranged.Ci.Li)
                  cloudlet_ListArranged. Ci.vm=Vmiindex;
Add cloudlet_ListArranged .Ci to Group_List


//Group_list final list to be submitted to VM
VM_totalMIPSindex = VM_totalMIPSindex -  cloudlet_ListArranged .Ci.Li
i=i+1
End of while
Submit Group_listgid to vmindex
Group_Id = Group_Id+1
Index = (Index + 1) % vm_Created_List.Size;
End For
```

IV. EXPERIMENTATION AND RESULTS

The experiment is conducted on CloudSim (3.0.3) simulator, where it is clearly the correctness of the algorithm "Two Phase Fair Dynamic Job Assignment" is verified. The comparison is done with sequential assignment which is inbuilt in Cloudsim and the proposed scheduling algorithm. The algorithm is implemented by using Granularity Size of 3.The configuration of datacenter created is shown below in Table 1 There are 1 processing elements and 2 hosts. Virtual Machines used in this experiment is shown in table 2.System Configuration is shown in Table 3.

Table 1: Configuration of Hosts

RAM (MB)	10240
Processing Power (MIPS)	110000
VM Scheduling	Time shared

Table 2: Configuration of VMs

Virtual Machine	VM 1	VM 2
RAM (MB)	5024	5024
Processing Power (MIPS)	22000	11000
Processing Element (CPU)	1	1

Table 3: System Configuration

Processor	Intel®Core™i5-3210M CPU, 2.5GHz
RAM	4 GB
System type	64 bit operating system
Operating System	Windows 7

Table 4: Comparison of Task Completion Time

No. Of Cloudlets	**Sequential Algorithm (ms)**	**Proposed Algorithm (ms)**
25	**227.23**	**216.05**
50	**1001.70**	**901.42**
75	**2388.29**	**2145.43**
100	**4570.03**	**4063.69**
125	**7537.84**	**6753.94**
150	**11559.05**	**10469.94**

Table 5: Comparison of Execution Cost

No. Of Cloudlets	**Sequential Algorithm**	**Proposed Algorithm**
25	**681.69**	**648.15**
50	**3005.10**	**2704.26**
75	**7164.87**	**6436.29**
100	**13710.03**	**12191.07**
125	**22613.52**	**20261.82**
150	**34677.15**	**31409.82**

Performance with Time: It is evident from results that proposed algorithm gives better task completion time.

Performance with cost: The cost of Task execution with proposed Two Phase Fair dynamic algorithm is reduced as compared to the sequential algorithm.

Fig 4.1 and Fig 4.2 shows the Proposed algorithm task completion comparision and cost comparision graph with respect to sequential algorithm.

© 2015, IJCSMC All Rights Reserved

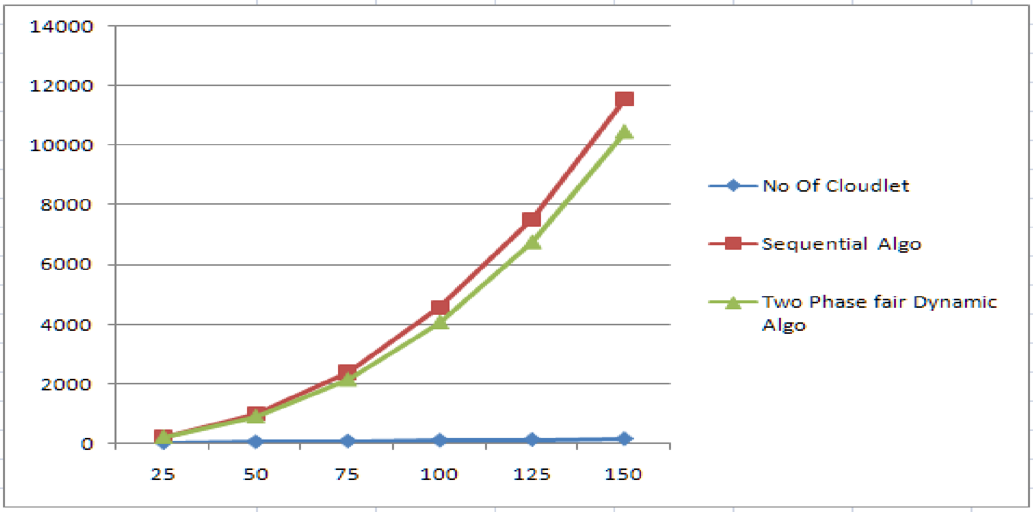

Fig. 4.1. Task completion comparison graph

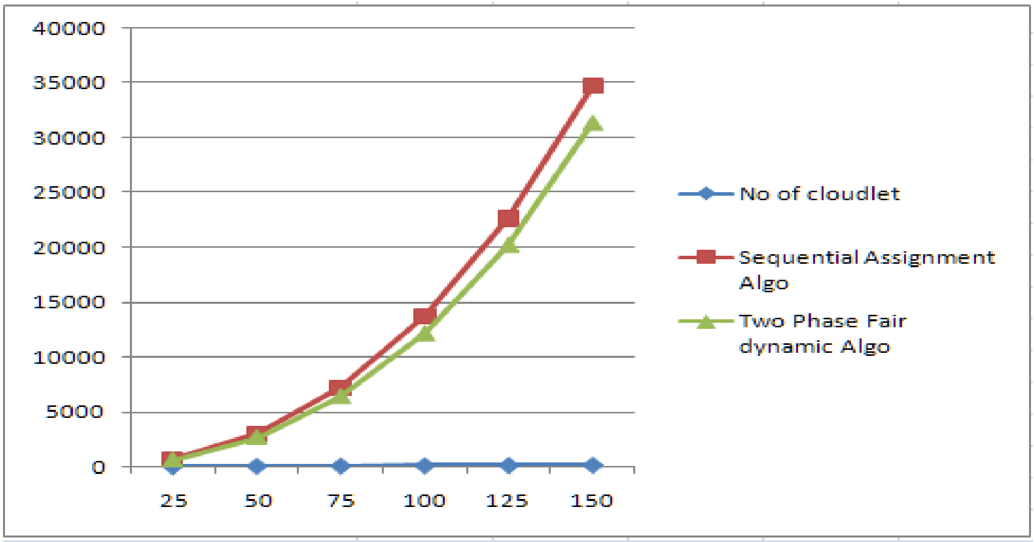

Fig. 4.2 Cost Comparison graph

The experimental results vary every time as the job selection priority and granularity size selection change according to broker's designed Best task ordering policy. It is observed that the proposed "Two Phase Fair Dynamic Algorithm" improves Task completion time and cost of execution as compared to Sequential Assignment. The results improve with increase in task count and it also provides a better approach to deal with job communication overhead and job assignment problem along with providing flexibility and reliability for system performance.

V. CONCLUSION AND FUTURE WORK

A cloud system most of the time has to manage heavy traffic load so there must be a requirement of such type of scheduling mechanism which manages jobs transition time and reduces energy consumption in a more frequent and effective way. This can be only achievable by a balancing scheduling approach. Up to a good extent our proposed algorithm is capable to achieve such type of flexibility. This work suggested that at the initial stage of scheduling by implementing Best Task Ordering strategy smartly according to the specific cloud environment the overall throughput of cloud scheduling is improved. To handle large number of jobs more frequently the granularity size base dynamic group formation is the fastest and easiest way to map job groups to respective target resources for execution. It is observed that our proposed approach is better than other traditional algorithm with respect to a overall balance scheduling performance. In future the proposed algorithm can be

© 2015, IJCSMC All Rights Reserved

made further more effective by implementing a more appropriate dynamic load balancing pattern which could be beneficial for much improved system performance and also capable to reduce energy consumption.

REFERENCES

[1] *S.Bazarbayev,M.Hiltunen,K.joshi,W.sander,R.Schlichting, "Content based scheduling Of virtual Machines (VMs) in the cloud", in 2013 IEEE 33rd International conference on distributed Computing systems*

[2] M.A.Rodriguez and R. Buyya , "*Deadline based Resource Provisioning and Scheduling Algorithm for Scientific Workflows on Clouds*", the Cloud Computing and Distributed Systems (CLOUDS) Laboratory, Department of Computing and Information Systems, The University of Melbourne, IEEE 2013

[3] Dr.V.Vaithiyanathan, R.Arvindh Kumar, S.Vignesh, B.Thamotharan, B.Karthikeyan *"An Efficient TPD Scheduling Algorithm for Cloud Environment"*, International Journal of Engineering and Technology (IJET), ISSN : 0975-4024, Vol 5 No 3 Jun-Jul 2013.

[4] K. Thomas Erl, Zaigham Mahmood,and Ricardo Puttini, *"Cloud Computing Concepts, Technology & Architecture,"* Second Printing: September 2013

[5] Abirami S.P, S Ramanathan, *"Linear scheduling strategy for Resource allocation in cloud environment"*, International Journal on Cloud Computing: Services and Architecture(IJCCSA),Vol.2, No.1,February 2012

[6] I. A.Moschakis, H D karatza, *"Performance and cost evolution of gang scheduling in cloud Computing system with job migration and Starvation Handling*", IEEE 2011

[7] Shu_ching, kuo_qin yan Shun_sheng wang ,Ching_wei chen, *"A three Phases scheduling in a hierarchical cloud Computing Network"*,Third International Conference on Communications and Mobile Computing in IEEE 2011.

[8] Nithiapidary Muthuvelu, Junyang Liu, Nay Lin Soe, Srikumar Venugopal,Anthony Sulistio and Rajkumar Buyya, *"A Dynamic Job Grouping-Based Scheduling for Deploying Applications with Fine-Grained Tasks on Global Grids*", At the Australasian Workshop on Grid Computing and e-Research (AusGrid2005).

© 2015, IJCSMC All Rights Reserved