ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Special Issue 7, October 2015

HLF: A Dynamic Scheduling Algorithm for Task Scheduling in Cloud Computing

Santhosh B[1], Prachi Kaneria A[2] , Dr. Manjaiah D.H [3]

Dept. of MCA, AIMIT, St. Aloysius College (Autonomous), Mangalore, Karnataka, India[1]

Dept. of MSc, AIMIT, St. Aloysius College (Autonomous), Mangalore, Karnataka, India [2]

Dept. of CS, Mangalore University, Mangalore, Karnataka, India[3]

ABSTRACT: Scientific workflows are represented by using directed acyclic graphs (DAGs)model. Since tasks are dependent on each other it requires an efficient task scheduling algorithm .In this paper, we have focused on implementation of task scheduling algorithm that reduces the overall makespan of jobs in the workflow. The proposed algorithm has been evaluated by using WorkflowSim simulator.

KEYWORDS: scheduling, workflow,makespan, level

1. INTRODUCTION

Buyya et al. [1] defines Cloud as a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements. To make this possible cloud requires an effective resource management and scheduling policy. The response to the user from the cloud must satisfy the SLA. Because of the heterogeneity of the resources and different types of SLA's between user and provider resource management and scheduling is considered as a NP-Hard problem. In this paper our primary objective is to reduce the total execution time ie total makespan of jobs in the workflow.To design such an scheduling algorithm requires optimizing the resources being allocated. Resource allocation requires optimisation of the costs associated with it. The Taxonomy of Workflow is showed in the figure 1. In our experiment we have considered DAG representation of the workflow structure.

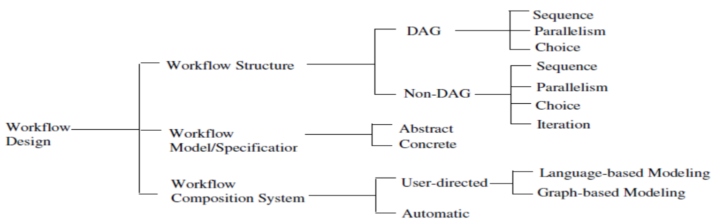

Fig 1: Taxonomy of Workflow Design [4]

II. PROBLEM DESCRIPTION

The Workflow Engine manages jobs based on their dependencies. It releases the jobs only when all of its parent jobs completed execution successfully. The Workflow Engine will only release free jobs to the Scheduler[2]. Scheduler will have the list of jobs which can be executed immediately. Since there is a limitations of number of resources available it requires an efficient task scheduling mechanism.

The figure 2 shows the the FCFS scheduling implementation on workflows. Here we have considered three virtual machines at a time which can execute three tasks parallel. schedule 1 the tasks 0,1 and 2 executed parallel which takes

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Special Issue 7, October 2015

x time to complete , at schedule level V and Vi there is only one task totally it requires 6x time required to complete the entire workflow. To reduce the completion time we have considered maxlevel , ie selecting the job which has maximum level at every job selection. The fig 3 shows the implementation of this. In fig 3 only level V has two tasks which can complete the entire workflow in 5x time. If we compare fcfs and maxlevel , maxlevel results with minimum completion time. And also it is showed that if we select jobs which has highest level for scheduling uses the resources efficiently. In fcfs in level V and V1 there is a wastage of 2vms and in maxlevel in level v has only one vm wastage

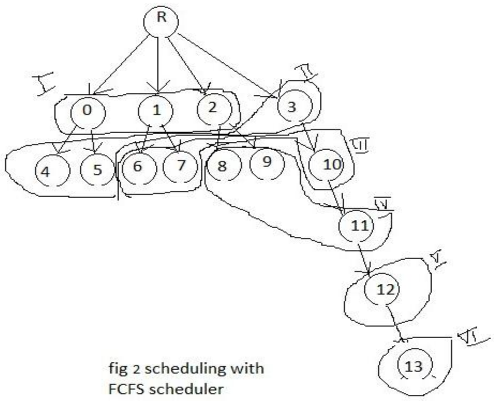

fig 2 scheduling with FCFS scheduler

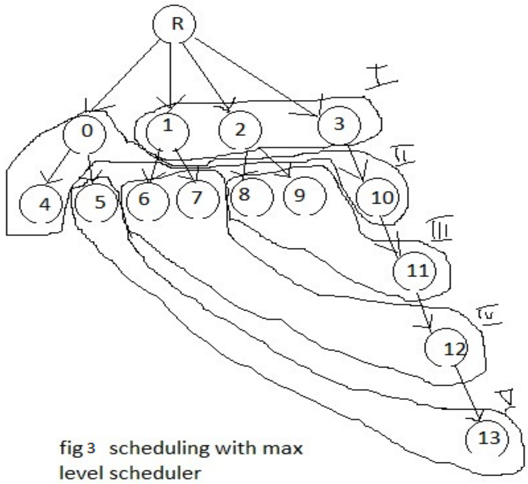

fig 3 scheduling with max level scheduler

III. PROPOSED ALGORITHM : HLF

The proposed algorithm takes the advantage of maxmin algorithm. Always it selects the job with highest level, and if there are multiple jobs exists in highest level ,it selects the job with maximum length and assigns to the vm where the processing time is minimum. Thus the algorithm always guarantees that makespan is less than or equal to maxmin algorithm.

1.size= size(cloudlet_list);

2. For i= 0 to size-1
    maxlevel= maxlevel(not_selected_cloudlet_list(cloudlet_list));
    cloudletlist temp_cloudletlist_list=empty;
    Cloudlet cloudlet=null;

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Special Issue 7, October 2015

```
 for i=0 to size-1
          If   level(Cloud_list.get(i))==maxlevel  && !selected(Cloud_list.get(i))
                Add  Cloudlet_list.get(i) to temp_cloudletlist_list;
          End if
 End for
 if size(temp_cloudletlist_list)>1
            cloudlet= maxcloudlength(temp_cloudletlist_list)
 else
            cloudlet= temp_cloudletlist_list.get(0)
 end if
 setselected(cloudlet)=true;

 vm=select_Idle_MaxMIPS(vm_available_list )
 Set vm=bussy
 submit cloudlet to vm
End for
```

IV. SIMULATION RESULTS

To test working of the algorithm we have used CyberShake_30, CyberShake_50 and modifiedCyberShake_50 workflows. In modifiedCyberShake_50 we have assigned shorter length for some of the jobs having maximum level. Makespan is used to evaluate the performance of the algorithm. The Makespan of a workflow is the time elapsed from its submission to the cloud until the completion of its last task. For the multi-workflow scheduling experiments, we have considerd the metric Total Makespan, is the time elapsed from the submission to the completion of instances submitted concurrently to the cloud. The workflow CyberShake_30 tested with two vms with mips spead 100 and 800, CyberShake_50 tested with three vms with mips spead 800,500,400 and modifiedCyberShake_50 with two vms having mips spead 800 and 500 .

The tool used to simulate the experiments is WorkflowSim. WorkflowSim extends the CloudSim simulator and support for large scale clustering, provisioning and scheduling studies. It provides a higher layer of workflow management. The result of the experiments conducted is shown in table . table values denotes makespan in seconds.

	CyberShake_30 Vm=2, 100,800	CyberShake_50 Vm=3, 800,500,400	modifiedCyberShake_50 vm=2 800,500
FCFS	1996.18	951.6	383.31
MCT	1996.18	951.6	383.31
MINMIN	1972.8	950.08	421.44
MAXMIN	1467.23	931.33	377.21
HLF	1448.82	931.33	372.95

Table :Total make span of the algorithms

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Special Issue 7, October 2015

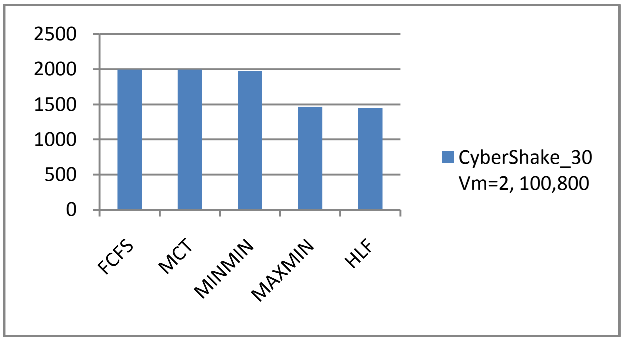

Fig: MakeSpan using CyberShake_30 workflow

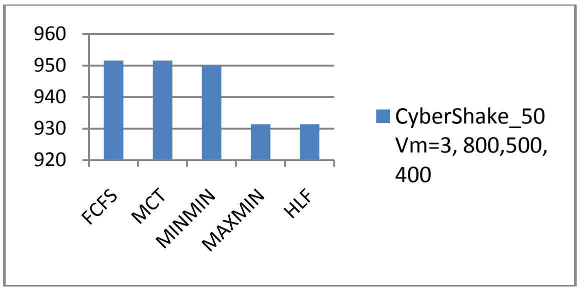

Fig: MakeSpan using CyberShake_50 workflow

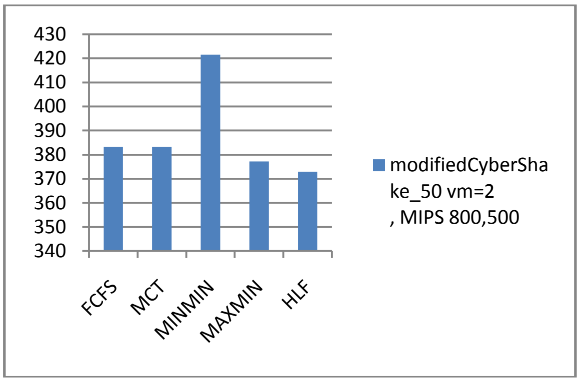

Fig: MakeSpan using ModifiedCyberShake_30 workflow

The above fig 1,2 and 3 shows the bar chart of the result with different workflows. The result clearly shows the proposed algorithm results in makespan minimum or equal to MaxMin

V. CONCLUSION

The experimental results shows the proposed algorithm minimises the MakeSpan. MaxMin algorithm always selects the jobs with the highest length and assigns to the best best available resource (fastest). In the proposed algorithm a job in the maximum level is selected, if there are multiple jobs then job with highest length is selected and assigned to the fastest resource. The algorithm could be improved by assigning the job to the resource which gives minimum completion time. The algorithm further can be improved by optimising multiple factors like cpu, memory ,network bandwidth etc.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Special Issue 7, October 2015

REFERENCES

[1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the5th utility," Future Generation Computer Systems , vol. 25, no. 6, pp. 599–616, 2009.
[2] Weiwei Chen; Deelman, E., "WorkflowSim: A toolkit for simulating scientific workflows in distributed environments," *E-Science (e-Science), 2012 IEEE 8th International Conference on* , vol., no., pp.1,8, 8-12 Oct. 2012
[3] P. Senkul and I. Toroslu, "An Architecture for Workflow Scheduling under Resource Allocation Constraints," *Information Systems*, vol. 30, no. 5, pp. 399-422,2005.
[4] J. Yu and R. Buyya, "A taxonomy of scientific workflow systems for grid computing," *SIGMOD Record*, Vol. 34, no. 3, pp. 44-49, 2005.
[5] J. Yu and R. Buyya, "Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms," *Scientific Programming*, vol. 14, no. 3, pp. 217–230, 2006.