Score-Based First Come First Serve Budget Constraint Workflow Scheduling Algorithm for Cloud System

Makhan Singh
Department of Computer Science & Engineering,
UIET, Panjab University, Chandigarh
(singhmakhan@rediffmail.com)

***Abstract*—**Cloud computing is the new paradigm in distributed environment that provides services and resources over the internet to the users. The user can access the services from cloud based on the Service Level Agreement (SLA), which define their needed QoS parameter on a pay per use Basis. Workflow scheduling is one of the key issues in the workflow management that maps and manages the execution of inter-dependent tasks on the distributed resources. It allocates desirable resources to workflow tasks such that the execution can be completed to satisfy objective functions defined by users. Proper scheduling can have significant impact on the performance of the system. Failure of workflow application is another major concern to guarantee availability and reliability of critical services as well as application execution. So, there is need to implement reliable execution of workflows in Cloud environment. Since there is large data and compute nodes to process data. In traditional scheduling, users prefer to minimize the completion time of their jobs. However, in Cloud environment the cost is an important performance factor. So there is need to implement the workflow scheduling algorithm in cloud environment that will reduce the execution cost and also allow the reliable execution of the workflow application. At present, the workflow scheduling algorithms only focus on certain QoS parameters which are mainly cost during the allocation of virtual machines to workflow applications. Sometimes resources (virtual machines) are unreliable at data centers hence they frequently results into failure when workflow applications are scheduled on these resources. The user workflow application may contain sensitive data that cannot tolerate failure of resources on which it is scheduled. So, there is a need to propose workflow scheduling algorithm that reduces the failure rate of workflow applications and also achieves QoS constraints imposed by Cloud user. In this paper a Score based deadline constraint workflow scheduling algorithm has been implemented. This algorithm reduces the failure rate and cost of workflow applications. It allocates those resources to workflow application that are reliable and reduce the execution cost and complete the execution within user specified deadline.

Keywords*—Workflow Scheduling; Score; Virtual Machine; Budget Constrain; First Come First Serve.*

I. INTRODUCTION

Cloud computing is paradigm that provides demand service resources like software, hardware, storage, networks, platform and infrastructure. Due to the advantage of cost effectiveness, on demand resource provision, easy to sharing, scalability, reliability, cloud computing has grown in popularity with research community for deploying scientific applications such as workflows. In workflow management one of major challenges is Workflow scheduling [1], especially in the Cloud and Grid workflow systems. It is a process that maps and manages the execution of tasks in workflow on different distributed resources and satisfies the constraints defined by the user. The workflow scheduling in cloud is divided into two main categories, Best effort Based scheduling algorithms which tries to optimize the execution time and ignoring the other factors such as execution cost, other QoS constraints. Another one is QoS constraint Based scheduling algorithms which tries to optimize the performance under QoS constraints for e.g. cost minimization under the deadline constraint or time minimization under the budget constraint.

II. RELATED WORK

In literature many researcher have proposed different algorithms for workflow scheduling in clouds like PCP (partial critical path)[2], SCFP (shortest cloudlet to fastest processor)[3], Genetic[4], HCOC (Hybrid Cloud Optimized Cost scheduling algorithm)[5], DBD-CTO (Deadline and Budget distribution based Cost-Time Optimization) [6] etc. schedule the workflow application under the QoS constraints. The entire proposed algorithms in literature define their own strategy for scheduling the workflow based application. Many of the algorithms use the priority of hardware resources in workflow application for assigning the resources. The Microsoft assign score value to show capability of hardware resource. The existing algorithms do not measure the capability of hardware resource while assigning the priority to it. They assign the priority on processing power of machines. We implement the score concept of Microsoft to assign the priority to hardware resources. Microsoft is using module known as Window System Assessment Tool (WINSAT) in Windows Vista, Windows 7, and Windows 8 to calculate the capabilities of hardware it is running in terms of Windows Experience Index (WEI) score [7] In Cloud environment this concept can be used to measure the capabilities of hardware resources as Score which represents the minimum performance of the system. To reduce the failure of workflow applications in performance based scheduling high performance machines are used. This will satisfy the users imposed deadline constraint with minimum execution cost. On studying the literature we have found that this type of work is not performed earlier. So there is a need to use score based workflow scheduling that satisfies the deadline constraints imposed by user and need of workflow execution through reliable machines.

III. PROPOSED SCORE BASED ALGORITHM

In our scheduling algorithms, the user specifies an objective function deadline constraint and the algorithm attempt to minimize the execution cost under the deadline constraint. This is an easy way for the users to define their requirement and propose a fast and high performance solution. In traditional algorithms for scheduling user mainly focus on reduction of execution time of the jobs, However in cloud cost is an important factor. As the user jobs has a deadline constraints imposed by the user itself before which the jobs must be finished, but over the Cloud early completion of jobs incurs more cost. So our performance criterion to reduce the execution cost for workflow application and execute them before user specified deadline.

A. Various parameters used in the Algorithm:

- **Score:** The concept of score is taken from the Microsoft window [8]. Similar approach is used over here. In which a score is defined on the basis of machine capability. Every hardware component gets an individual score. The machines final score is calculated by considering the minimum sub-score of the every hardware component. The final score shows the minimum overall performance of machine which is totally based on the different hardware elements of system such as processing power, storage space and RAM. The lowest sub-score among the different component are considered as final score of machine which is shown in the Table 1.

As shown in the Table 1final score of the machine is determined by the lowest sub-score of the component. Workflow application (software score) performance requirement is defined on the basis of number of instructions in the Workflow application tasks. Higher is the number of instructions in the tasks higher is the Performance Requirement of Workflow application. In suggested algorithm consider those Hardware Resources whose final score is equal to or more than the tasks score and execute the tasks on those Hardware Resources within specified deadline with minimum cost.

Table 1: Score Calculation of Hardware.

Component	**Metrics**	**Sub-score**	**Final score (Lowest Sub-score)**
Processor	Calculation per second (MIPS)	6	4
Memory (RAM)	Memory operations per second	5	
Storage space	Data transfer rate	4	

- **Deadline:** Suppose user specifies deadline to be achieved is D. This specified deadline D is achieved by distributing this over the workflow application. On considering the specified deadline D, sub-deadlines of tasks are calculated and schedule the tasks within that sub-deadline. With this strategy user specified deadline D is achieved for the workflow application. Consider there are T_1, T_2 and T_3 Tasks in workflow application with 1000, 5000, 4000 instructions respectively and user specified deadline is 100 hours for complete workflow execution. Now depending upon user specified deadline the sub deadlines for T_1, T_2 and T_3 Tasks are calculated as 10 hours, 40 hours and 50 hours respectively. The proposed algorithm schedules the tasks as per their sub-deadlines such that entire workflow application gets executed within 100 hours.

- **Cost:** Cost of the virtual machines is set on the basis of their configuration. A higher configuration machine gets the higher cost. A more cost is involved in executing the workflow application on the high performance machine as compared to executing the workflow application on the low performance machine.

- **Failure rate:** Failure of workflow application is the proportion of absolute number of times virtual machines failed to execute the workflow application and the total number of times virtual machines effectively executed the workflow application. The suggested algorithm considers only those virtual machines having more performance than required and discards the machines with lower performance so that the failure rate of workflow applications is reduced. The term score is used to measure the performance.

B. Score Based FCFS Budget Constrained Workflow Scheduling Algorithm

First Come First Served (FCFS) algorithm assigns the workflow tasks to virtual machines on first come first serve basis in such a way that user defined budget constraint is satisfied. Score concept has also been introduced and only those machines are selected for scheduling which satisfy minimum task score. Then performance of score based FCFS budget constrained workflow scheduling algorithm has been compared with basic FCFS budget constrained workflow scheduling algorithm with respect to their execution time, execution cost and failure rate. Step by step description of basic FCFS (First Come First Served) budget constrained workflow scheduling algorithm is presented below:

```
Algorithm BASIC_FCFS (T, VM, B , $C_i$)
  // T is Work Flow Tasks List, VM is Virtual Machines List,
     B is Budget, $C_i$ is Virtual Machine Costs.
Step1:-Submit list of workflow tasks
         T :={ $T_1, T_2, \dots, T_n$ }.
Step2:-  Get available resources from data center
         VM :={ $VM_1, VM_2, \dots, VM_n$ }.
Step3:-  Assign Budget B to workflow tasks T.
Step4:-  Repeat while T != NULL
         {
         Step 4.1:- Pick VM from list.
         Step 4.2:- Pick next VM from list.
         Step 4.3:- If (VM Cost < task Budget)
```

```
            {
        Step 4.3.1:- Assign workflow task to VM.
        Step 4.3.2:- Select next task from list and
                     goto Step 4.
            }
        Else
        Step 4.3.1:- Goto Step 4.2.
        }
Step 5:-  Return map to simulation
```

The figure 1 shows the flow diagram of Basic FCFS Budget Constrained Workflow Scheduling Algorithm.

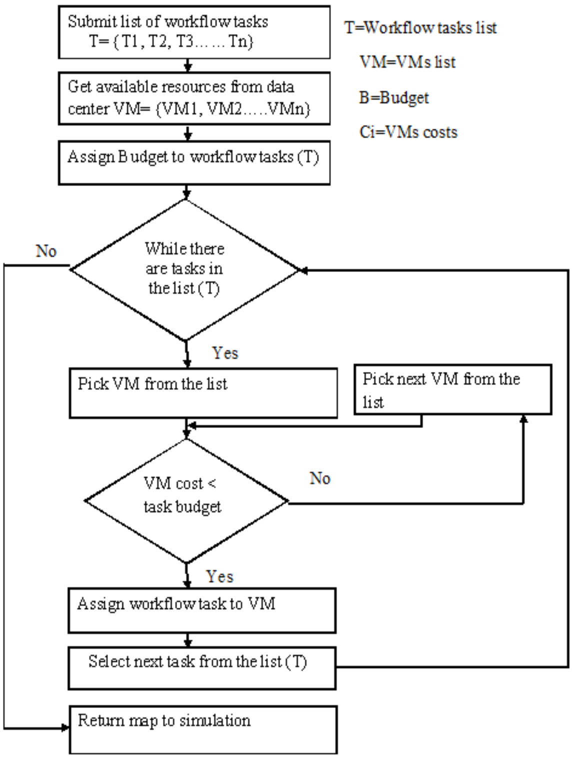
Submit list of workflow tasks
T={T1, T2, T3……Tn}
Get available resources from data center VM= {VM1, VM2…..VMn}
Assign Budget to workflow tasks (T)
While there are tasks in the list (T)
No
Yes
Pick VM from the list
Pick next VM from the list
VM cost < task budget
No
Yes
Assign workflow task to VM
Select next task from the list (T)
Return map to simulation
T=Workflow tasks list
VM=VMs list
B=Budget
Ci=VMs costs

Figure1: Basic FCFS Budget Constrained Workflow Scheduling Algorithm

Step by step description of score based FCFS (First Come First Served) budget constrained workflow scheduling algorithm is presented below:

Algorithm SCORE_BASED_FCFS (T,VM, B , C_i , S_v , S_t)

```
// T is Work Flow Tasks List, VM is Virtual Machines List, B
is Budget, $C_i$ is Virtual Machine Costs, $S_v$ is VM Score, $S_t$ is
Task Score.
Step1:-  Submit list of workflow tasks
         T :={ $T_1, T_2, \dots, T_n$ }.
Step2:-  Get available resources from data center VM
         :={ $VM_1, VM_2, \dots, VM_n$ }.
Step3:-  Assign Budget B to workflow tasks T.
Step 4:- Obtain Scores of VMs ($S_v$).
Step 5:- Obtain task scores ($S_t$) based on instruction length.
Step6:-  Repeat while T != NULL
         {
         Step 6.1:- Pick VM from list that satisfies
                    the task score.
         Step 6.2:- Pick next VM from list that
                    satisfy the task score.
         Step 6.3:- If (VM Cost < task Budget)
                {
                Step 6.3.1:- Assign workflow task
                             to VM.
                Step 6.3.2:- Select next task from
                             list and goto Step 6.
         }
         Else
         Step 6.3.1:- Goto Step 6.2.
         }
Step 7:- Return map to simulation.
```

The figure 2 shows the flow diagram of score based FCFS Budget Constrained Workflow Scheduling Algorithm.

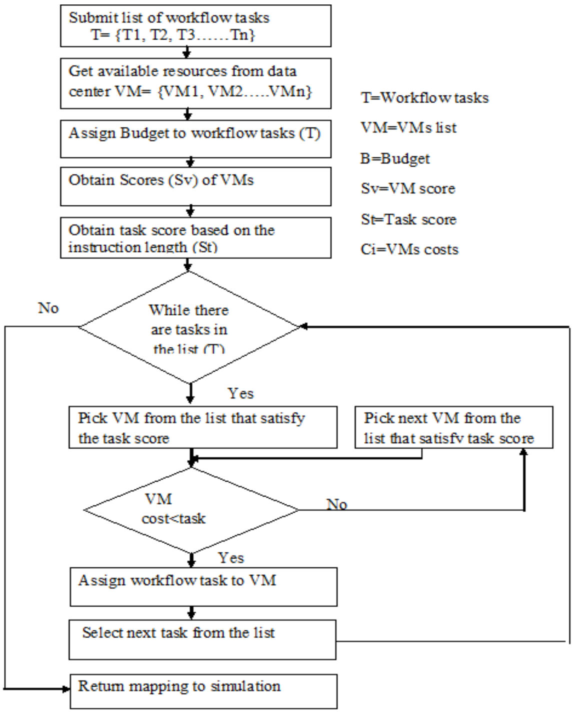
Submit list of workflow tasks
T= {T1, T2, T3……Tn}
Get available resources from data center VM= {VM1, VM2…..VMn}
Assign Budget to workflow tasks (T)
Obtain Scores (Sv) of VMs
Obtain task score based on the instruction length (St)
While there are tasks in the list (T)
No
Yes
Pick VM from the list that satisfy the task score
Pick next VM from the list that satisfy task score
VM cost<task
No
Yes
Assign workflow task to VM
Select next task from the list
Return mapping to simulation
T=Workflow tasks
VM=VMs list
B=Budget
Sv=VM score
St=Task score
Ci=VMs costs

Figure 2: Score based FCFS Budget Constrained Workflow Scheduling Algorithm

IV. RESULTS

The simulation environment CloudSim has been used for performing experiments and the performance analysis of score based workflow scheduling algorithms. The parameters for Cloud Simulator are set as per Table 2.

In the experimental results performance of score based FCFS budget constrained workflow scheduling algorithm has been compared with basic FCFS budget constrained workflow scheduling algorithms with respect to their execution time, execution cost and failure rate.

Table 2: Cloud Simulator Parameter

Type	Parameter	Value
Datacenter	Number of Datacenter	1
	Number of Hosts	4
	Type of Manager	Time-Shared
VM	Total Number of VMs	10
	MIPS of PE(processing element)	1000-21000
	Number of PEs per VM	1
	VM Memory(RAM)	256-2048 MB
	Storage Space	1-21 GB
	Score	1-10
Cloudlet	Number of Workflow Application	1
	Number of Cloudlets(Tasks)	5-20

This test case in Table 3 shows the effect on execution time of FCFS budget constrained workflow scheduling algorithm when it is executed with score and without score concept by varying the number of cloudlets.

Table 3: Showing Execution Time and Number of Cloudlets

FCFS Budget Constrained Workflow Scheduling Algorithm		
Number of Cloudlets	Execution Time	
	FCFS (score)	FCFS(basic)
5	2130	2560
10	3480	4860
15	5578	6470
20	7710	8206
25	9138	9654

Simulation results shown in Figure 3 indicate that score based FCFS budget constrained workflow scheduling algorithm exhibit less execution time for the workflow application as compared to basic FCFS budget constrained workflow scheduling algorithm.

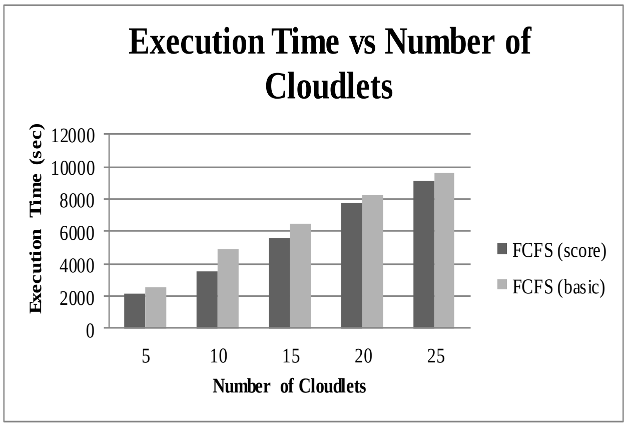

Figure 3: Showing Execution Time vs. Number of Cloudlets

This test case in Table 4 shows the effect on execution cost of FCFS budget constrained workflow scheduling algorithm when it is executed with score and without score concept by varying the user specified budget.

Table 4: Showing Execution Cost and User Budget

FCFS Budget Constrained Workflow Scheduling Algorithm		
User Budget	Execution Cost	
	FCFS (score)	FCFS(basic)
3000	2570	2360
6000	5316	5190
9000	8290	7800
12000	11065	10300
15000	14100	13238

Results obtained in Figure 4 indicate that score based FCFS budget constrained workflow scheduling algorithms, although incur more cost compared to their basic counterparts, but still execute the workflow application within user specified budget.

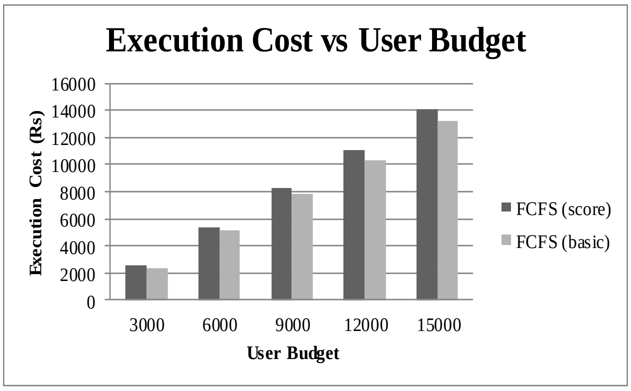

Figure 4: Showing the Execution Cost vs. User Budget

The test case in Table 5 shows the effect on failure rate of FCFS budget constrained workflow scheduling algorithm when it is executed with score and without score concept by varying the number of iterations.

Table 5: Showing Failure Rate and Number of Iterations

FCFS Budget Constrained Workflow Scheduling Algorithm		
Number of Iterations	Failure Rate	
	FCFS (score)	FCFS(basic)
10	0	0.1
20	0	0.3
30	0.1	0.45
40	0.1	0.45
50	0.12	0.5

Result obtained in Figure 5 shows score based FCFS budget constrained workflow scheduling algorithm exhibit less failure rate as compared to basic FCFS budget constrained workflow scheduling algorithm.

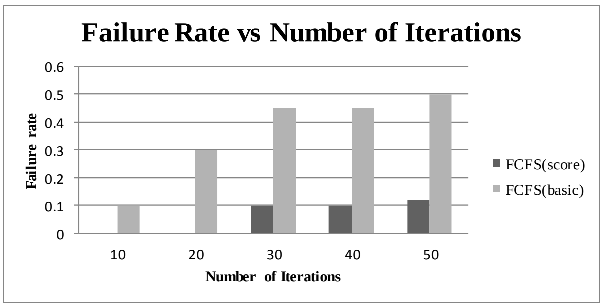

Figure 5: Showing Failure Rate vs. Number of Iterations

V. CONCLUSION

The proposed a Score based deadline constraint algorithm for workflow application in which workflow scheduling is done to encourage the formation of solution to achieve the deadline constraint and cost minimization and compared with basic FCFS and score based FCFS algorithm. Score based developed workflow scheduling algorithm is more efficient, reduces the execution cost of the workflow application, satisfies the user deadline which is very important criteria to be considered while scheduling of the workflow application and reduces the failure of workflow applications as these applications are very critical sometimes cannot afford the failure.

REFERENCES

[1] J. Yu and R. Buyya, "Grid Computing," A Taxonomy of Workflow Management Systems for Grid Computing, vol. 34, no. 3, pp. 171-200, September 2005.

[2] M. Naghibzadeh, S. Abrishami, "Deadline-constrained workflow scheduling in software as a service Cloud," Science Direct, vol. 19, no. 3, pp. 680-689, November 2011.

[3] M.Naghibzadeh, S.Abrishami, "Deadline-constrained workflow scheduling algorithms for Infrastructure as a Service Clouds," Eslevier, vol. 29, pp. 158-169, 2012.

[4] Jia Yu , R.Buyya, "Scientific Workflow Applications with Deadline and Budget Constraints using Genetic Algorithms," ACM, vol. 14, no. 3,4, pp. 217-230, December 2006

[5] Lining Xing, Weiming Zhang, Meidong Xiao Yanli hu, "knowledge-based ant colony optimization algorithm(KBACO) for grid workflow scheduling," in ACM, 2010.

[6] Luiz Fernando, Bittencourt Edmundo, Roberto Mauro Madeira, "A Cost Optimization Algorithm for Workflow Scheduling in Hybrid Clouds," Springer, vol. 2, no. 3, pp. 207-227, December 2011.

[7] "Windows System Assessment Tools", http://www.en.wikipedia.org/wiki/ Windows_System_Assessment_Tool

[8] Linlin Wu, Suraj Panday Rajkumar Buyya, "A Particle Swarm Optimization-Based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments," in IEEE, 2010.