*International Journal of Latest Trends in Engineering and Technology*
*Special Issue SACAIM 2016, pp. 43-48*
*e-ISSN:2278-621X*

# A STUDY ON WORKFLOW SCHEDULING ALGORITHMS IN CLOUD

Jason Prashanth Pinto[1], Afrin Hukkeri[2] and Santhosh B[3]

Abstract: Directed acyclic graphs (DAGs)model are used to represent scientific workflows. An efficient task scheduling algorithm is required since the tasks are dependent on each other. In cloud, the jobs are dynamically received and are submitted to the datacentre for execution of these jobs. Based on the dependencies of jobs, they are submitted. After the successful execution of all its parent jobs, a job is submitted. These jobs will be listed in the scheduler. The list of jobs that can be immediately executed will be in the scheduler. An efficient task scheduling mechanism is required since there is a limitation on the number of available resources. In this paper, we have compared several task scheduling algorithms with respect to makespan of jobs in the workflow. The algorithmsare analysed using the WorkflowSim simulator.
Keyword:scheduling, workflow, makespan, level, Max-Min, Min-Min, FCFS.

## I. INTRODUCTION

Cloud Computing is a latest trend in today's world. It provides on demand services like hardware, software, platform, infrastructure and storage etc. dynamically to the user according to the "pay per use" model by using virtualized resources over the internet. Cloud computing is able to host various applications such as business, social networks and scientific applications.

While Cloud computing provides various services like IaaS, PaaS and SaaS etc. to end users but due to novelty of cloud computing, it also suffers from many types of research issues such as security, performance, database management, virtual machine migration,server consolidation, fault tolerance and workflow scheduling etc. Among these workflow scheduling is major issue for scientific applications

## II. LITERATURE SURVEY

### Time Comparison Based Scheduling Heuristics

The main objective of scheduling algorithm is to achieve the best system throughput with proper resource utilization and high performance by obeying the user's specified QoS parameter. There are various time comparison based scheduling heuristics exists in the grid and Cloud computing environment, which schedules the tasks by comparing the arrival time or execution time of the tasks.

### First Come First Serve (FCFS)

First Come First Serve (FCFS) is the simple strategy and the most fundamental which involves the client-server interaction in grid scheduling. In grid scheduling, FCFS policy accomplishes the jobs based on their arrival time, which means that the job will be executed based on which job has arrived first in the queue. It doesn't consider any other biases or preferences. The simplicity and the fast execution behaviour of this algorithm are some of its advantages.

[1] *AIMIT, Beeri, Mangalore, Karnataka, India*
[2] *AIMIT, Beeri, Mangalore, Karnataka, India*
[3] *AIMIT, Beeri, Mangalore, Karnataka, India*

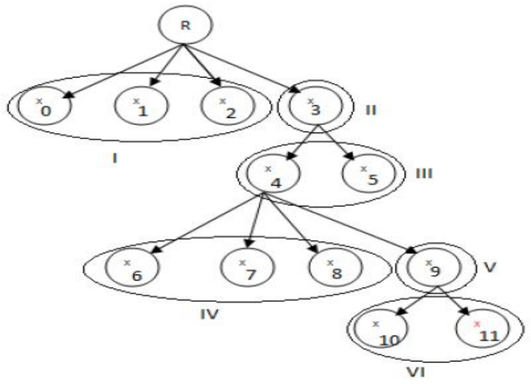

Figure1: Scheduling with FCFS approach

When FCFS scheduling is implemented as shown in Figure 3.2 by consideration of three virtual machines at a time only. Now for first schedule, FCFS scheduler will schedule task 0, 1 and 2 at three virtual machines paralely, so it will take X time to execute these three tasks. But for second schedule, only task 3 is available for execution because task 4 and task 5 can't be executed at this time since their parent has to be executed first. So it will take another X time to complete task 3 only. During this schedule, two virtual machines remains idle. Similarly, for third and sixth schedule, one virtual machine remains idle in each case and for each schedule X time is consumed. Fourth schedule also behaves in the same way as first schedule does and this will also consume X time. Fifth schedule behaves like second schedule and it will also take X time to complete the execution of the single task 9. So in this manner resources are not utilized properly and a total of 6X time is required to complete the whole workflow execution

**Min-min**

In Min-Min algorithm, the execution of the larger tasks is delayed so that a smaller task can be executed. In this algorithm, all the unmapped tasks are first sorted in the increasing order of their completion time. The task with the minimum completion time is scheduled to the available corresponding resources and this mapped task is removed from the unmapped task set. Until all the tasks from this list are mapped, this process repeats itself [1].

**Max-min**

In Max-Min algorithm, the execution of the smaller tasks is delayed so that a larger task can be executed. The Min-Min algorithm works in the similar way. But in Max-Min algorithm, the tasks are sorted in the decreasing order of their completion time. From the task list, the task with the maximum completion time is selected and the corresponding resource that is available is scheduled to it. Then the scheduled task has been removed from unmapped task set and the process repeats until all tasks of unmapped list is mapped

**Minimum Completion Time (MCT)**

In Minimum Completion Time algorithm, the task with the least completion time is assigned randomly to a machine. This algorithm behaves somewhat like Min-Min algorithm. MCT algorithm considers only one task at a time, unlike Min-min algorithm in which all the unmapped tasks are considered during each mapping decision [2].

**MaxChild**

In this algorithm, the task which has maximum number of Childs will be scheduled first, so that maximum number of tasks can be available for the next schedules and resource are utilized properly.

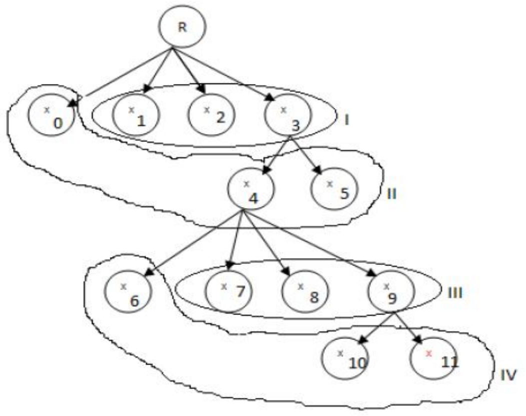

Figure 2: Scheduling with MaxChild approach

MaxChild Scheduler will schedule task 3 prior to all the other tasks at level 1on any of three virtual machines along with any 2 tasks from the remaining task of that level. So X time is consumed for these three tasks of level one. Now at the second schedule, MaxChild scheduler will schedule task 4 and task 5 of level 2, along with the remaining one task (task 0) of level 1. To execute these three tasks another X time is consumed. Similarly, at the third schedule MaxChild scheduler will schedule task 9 prior to all the other tasks of level 3 along with task 8 and task 9. And in the final schedule, it will schedule task 10 and task 11 of level 4 along with the remaining one task of level 3. So in this way resource are utilized properly and the given workflow will be executed completely in 4X time

## III. EXPERIMENTAL RESULTS

The paper uses simulation to test and verify theefficiency and correctness of the scheduling algorithm

### Simulation setup

The proposed algorithm is simulated in a simulation toolkit workflowsim[3].We have created a datacentre with the following properties and created 20 virtual machines. Here we have considered 5 virtual machines with processing capability 100,300,200,500 and 600 MIPS respectively

### Results

Experimental result of MINMIN with respect to Cybershake_50 data set

| Cloudlet ID | STATUS | DatacentreID | VM ID | Time | Start Time | Finish Time | Depth |
|---|---|---|---|---|---|---|---|
| 50 | SUCCESS | 2 | 4 | 0.18 | 0.1 | 0.28 | 0 |
| 2 | SUCCESS | 2 | 4 | 121.9 | 0.28 | 122.18 | 1 |
| 22 | SUCCESS | 2 | 3 | 181.12 | 0.28 | 181.4 | 1 |
| 5 | SUCCESS | 2 | 4 | 66.5 | 122.18 | 188.68 | 2 |
| 25 | SUCCESS | 2 | 4 | 49.4 | 188.68 | 238.08 | 2 |
| 6 | SUCCESS | 2 | 4 | 1.68 | 238.08 | 239.77 | 3 |
| 26 | SUCCESS | 2 | 4 | 1.28 | 239.77 | 241.05 | 3 |
| 9 | SUCCESS | 2 | 3 | 122.42 | 181.4 | 303.82 | 2 |
| 27 | SUCCESS | 2 | 4 | 65.5 | 241.05 | 306.55 | 2 |
| 10 | SUCCESS | 2 | 4 | 1.65 | 306.55 | 308.2 | 3 |
| 28 | SUCCESS | 2 | 4 | 2.3 | 308.2 | 310.5 | 3 |
| 23 | SUCCESS | 2 | 4 | 71.27 | 310.5 | 381.77 | 2 |
| 31 | SUCCESS | 2 | 3 | 83.44 | 303.8387.26 | | 2 |
| 24 | SUCCESS | 2 | 3 | 2.06 | 387.26 | 389.32 | 3 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 32 | SUCCESS | 2 | 3 | 1.36 | 389.32 | 390.68 | 3 |
| 35 | SUCCESS | 2 | 1 | 415.6 | 0.28 | 415.88 | 1 |
| 29 | SUCCESS | 2 | 4 | 90.4 | 381.77 | 472.16 | 2 |
| 36 | SUCCESS | 2 | 4 | 38.55 | 472.16 | 510.71 | 2 |
| 30 | SUCCESS | 2 | 4 | 2.18 | 510.71 | 512.9 | 3 |
| 37 | SUCCESS | 2 | 4 | 1.28 | 512.9 | 514.18 | 3 |
| 33 | SUCCESS | 2 | 3 | 128.12 | 390.68 | 518.8 | 2 |
| 40 | SUCCESS | 2 | 4 | 63.05 | 514.18 | 577.23 | 2 |
| 34 | SUCCESS | 2 | 4 | 2.13 | 577.23 | 579.36 | 3 |
| 41 | SUCCESS | 2 | 4 | 2.57 | 579.36 | 581.93 | 3 |
| 46 | SUCCESS | 2 | 3 | 79.4 | 518.8 | 598.2 | 2 |
| 7 | SUCCESS | 2 | 1 | 214.03 | 415.88 | 629.91 | 2 |
| 47 | SUCCESS | 2 | 1 | 4.3 | 629.91 | 634.21 | 3 |
| 8 | SUCCESS | 2 | 1 | 4.4 | 634.21 | 638.61 | 3 |
| 38 | SUCCESS | 2 | 4 | 72.88 | 581.93 | 654.81 | 2 |
| 44 | SUCCESS | 2 | 3 | 88.8 | 598.2 | 687 | 2 |
| 39 | SUCCESS | 2 | 3 | 2.74 | 687 | 689.74 | 3 |
| 45 | SUCCESS | 2 | 3 | 2.5 | 689.74 | 692.24 | 3 |
| 3 | SUCCESS | 2 | 0 | 595.69 | 122.18 | 717.88 | 2 |
| 4 | SUCCESS | 2 | 3 | 2.8 | 717.88 | 720.68 | 3 |
| 42 | SUCCESS | 2 | 4 | 96.62 | 654.81 | 751.43 | 2 |
| 43 | SUCCESS | 2 | 4 | 2.27 | 751.43 | 753.69 | 3 |
| 48 | SUCCESS | 2 | 1 | 154.07 | 638.61 | 792.67 | 2 |
| 49 | SUCCESS | 2 | 4 | 2.02 | 792.67 | 794.69 | 3 |
| 11 | SUCCESS | 2 | 2 | 929.9 | 0.28 | 930.18 | 1 |
| 20 | SUCCESS | 2 | 4 | 61.08 | 930.18 | 991.26 | 2 |
| 21 | SUCCESS | 2 | 4 | 2.6 | 991.26 | 993.86 | 3 |
| 14 | SUCCESS | 2 | 3 | 82.8 | 930.18 | 1012.98 | 2 |
| 15 | SUCCESS | 2 | 4 | 2.17 | 1012.98 | 1015.15 | 3 |
| 12 | SUCCESS | 2 | 1 | 164.57 | 930.18 | 1094.75 | 2 |
| 13 | SUCCESS | 2 | 4 | 1.47 | 1094.75 | 1096.21 | 3 |
| 18 | SUCCESS | 2 | 2 | 254.8 | 930.18 | 1184.98 | 2 |
| 19 | SUCCESS | 2 | 4 | 2.4 | 1184.98 | 1187.383 | |
| 16 | SUCCESS | 2 | 0 | 551.1 | 930.18 | 1481.28 | 2 |
| 1 | SUCCESS | 2 | 4 | 0.28 | 1481.28 | 1481.56 | 3 |
| 17 | SUCCESS | 2 | 3 | 3.14 | 1481.28 | 1484.42 | 3 |
| 0 | SUCCESS | 2 | 4 | 0.4 | 1484.42 | 1484.82 | 4 |

Experimental result of MAXCHILD with respect to Cybershake_50 data set

| Cloudlet ID | STATUS | Data center ID | VM ID | Time | Start Time | Finish Time | Depth |
|---|---|---|---|---|---|---|---|
| 50 | SUCCESS | 2 | 4 | 0.18 | 0.1 | 0.28 | 0 |
| 2 | SUCCESS2 | | 4 | 121.9 | 0.28 | 122.18 | 1 |
| 3 | SUCCESS | 2 | 4 | 99.28 | 122.18 | 221.47 | 2 |
| 22 | SUCCESS | 2 | 1 | 301.87 | 0.28 | 302.15 | 1 |
| 7 | SUCCESS | 2 | 4 | 107.01 | 221.47 | 328.48 | 2 |
| 4 | SUCCESS | 2 | 4 | 2.33 | 328.48 | 330.81 | 3 |
| 11 | SUCCESS | 2 | 3 | 371.96 | 0.28 | 372.24 | 1 |
| 23 | SUCCESS | 2 | 4 | 71.27 | 330.81 | 402.08 | 2 |
| 25 | SUCCESS | 2 | 3 | 59.28 | 372.24 | 431.52 | 2 |
| 27 | SUCCESS | 2 | 4 | 65.5 | 402.08 | 467.58 | 2 |
| 9 | SUCCESS | 2 | 1 | 204.03 | 302.15 | 506.18 | 2 |
| 5 | SUCCESS | 2 | 0 | 399 | 122.18 | 521.18 | 2 |
| 8 | SUCCESS | 2 | 0 | 13.2 | 521.18 | 534.38 | 3 |
| 31 | SUCCESS | 2 | 4 | 69.53 | 467.58 | 537.11 | 2 |

```
29    SUCCESS    2          3     108.48    431.52     540       2
14    SUCCESS    2          4     69     537.11     606.11     2
35    SUCCESS    2          2     623.4    0.28      623.68     1
16    SUCCESS    2          3     110.22    540      650.22     2
24    SUCCESS    2          3     2.06     650.22     652.28     3
26    SUCCESS    2          3     1.54     652.28     653.82     3
28    SUCCESS    2          3     2.76     653.82     656.58     3
10    SUCCESS    2          3     1.98     656.58     658.56     3
6    SUCCESS    2           3     2.02     658.56     660.58     3
32    SUCCESS    2          3     1.36     660.58     661.94     3
30    SUCCESS    2          3     2.62     661.94     664.56     3
15    SUCCESS    2          3     2.6     664.56     667.16     3
18    SUCCESS    2          4     84.93     606.11     691.05     2
36    SUCCESS    2          3     46.26     667.16     713.42     2
33    SUCCESS    2          1     213.53     506.18     719.71     2
38    SUCCESS    2     4     72.88     691.05     763.93     2
40    SUCCESS    2          3     75.66     713.42     789.08     2
20    SUCCESS    2          2     183.25     623.68     806.93     2
44    SUCCESS    2          4     74     763.93     837.93     2
17    SUCCESS    2          4     2.62     837.93     840.55     3
19    SUCCESS    2          4     2.4     840.55     842.95     3
37    SUCCESS    2          4     1.28     842.95     844.23     3
34    SUCCESS    2          4     2.13     844.23     846.36     3
39    SUCCESS    2          4     2.28     846.36     848.65     3
41    SUCCESS    2          4     2.57     848.65     851.21     3
21    SUCCESS    2          4     2.6     851.21     853.81     3
45    SUCCESS    2          4     2.08     853.81     855.9     3
46    SUCCESS    2          3     79.4     789.08     868.48     2
47    SUCCESS    2          4     2.15     868.48     870.63     3
42    SUCCESS    2          1     193.23     719.71     912.95     2
43    SUCCESS    2          4     2.27     912.95     915.21     3
12    SUCCESS    2          0     493.69     534.38     1028.07     2
13    SUCCESS    2          4     1.47     1028.07     1029.54     3
48    SUCCESS    2     2     231.1     806.93     1038.02     2
1    SUCCESS    2           4     0.28     1038.02     1038.31     3
49    SUCCESS    2          3     2.42     1038.02     1040.44     3
0    SUCCESS    2           4     0.4     1040.44     1040.84     4
```

The summary of the experimental results are given in the bellow table

Table 1: MakeSpanof scheduling algorithms wrt the data sets Cybershake_50 and Montage_100

| Scheduling Algorithm | Cybershake_50.xml dataset | Montage_100.xml dataset |
|---|---|---|
| FCFS | 1249.39 | 1024.66 |
| MINMIN | 1484.82 | 724.43 |
| MAXMIN | 1111.31 | 724.25 |
| MAXCHILD | 1040.84 | 719.81 |

The experimental results clearly showsMaxChild algorithm is more efficient with respect to makespan compared to FCFS, MINMIN and MAXMIN.

## IV.CONCLUSIONS

One of the main objectives in the above Scheduling algorithms is Minimizing Makespan. In comparison to algorithms such as FCFS, MAXMIN and MINMIN, the MAXCHILD algorithm was found to be the most efficient algorithm with respect to Makespan. After a job is submitted to the resource, if the resource becomes unavailable it may affect the makespan. These algorithms can further be improved by considering multiple objective functions, fault tolerance.

## V.ACKNOWLEDGEMENT

The authors would like to thank the dedicated research group in the area of Cloud & Grid Computing, wireless networks at the Dept of Computer Science, AIMIT, St Aloysius College, Mangalore, India, for many simulating discussions for further improvement and future aspects of the Paper. Lastly but not least the author would like to thank everyone, including the anonymous reviewers.

## REFERENCES

**[1]** D. Tracy et. al. “A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems,” *Journal of Parallel and Distributed computing*, vol.61, no. 6, pp. 810-837, 2001.

[2] F. Dong and S.G. Akl, “Scheduling Algorithms for Grid Computing: State of the Art and Open Problems,” Technical Report of the Open Issues in Grid Scheduling Workshop, School of Computing, University Kingston, Ontario, January 2006

[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 Workflow Management Systems for Grid Computing,” *journel of Grid Computing*, Springer, pp. 171-200, 2005.

[5] Vijay Prakash, masters thesis An Efficient Workflow Scheduling Approach in Cloud Computing,thapar university patiala

[6] P. Senkul and I. Toroslu, “An Architecture for Workflow Scheduling under Resource Allocation Constraints,” *Information Systems*, vol. 30, no. 5, pp. 399-422,2005.

[7] H. Fard, R. Prodan, J. Barrionuevo, and T. Fahringer, “A multi-objective approach for workflow scheduling in heterogeneous environments,” In *Proceedings of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*, pp. 300-309. IEEE Computer Society, 2012.

[8] D. Amalarethinam and F. Selvi, “A minimum makespan grid workflow scheduling algorithm,” In *International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1-6, IEEE, 2012.

[9] S. Ghanbari and M. Othman, “A priority based job scheduling algorithm in cloud computing,” *Proceeding In International Conference on Advances Science and Contemporary Engineering 2012 (ICASCE 2012)*, pp. 778-785, 2012.

[10] W. Chen, R. Silva, E. Deelman, and R. Sakellariou, “Balanced Task Clustering in Scientific Workflows,” In *proceedings IEEE 9th International Conference on eScience*, pp. 188-195. 2013