Design of Task Scheduling Model for Cloud Applications in Multi Cloud Environment

P.K. Suri[1] and Sunita Rani[2(✉)]

[1] Department of Computer Science and Applications, Kurukshetra University, Kurukshetra, India
pksurikuk@gmail.com
[2] Department of CSE & IT, B.P.S.M.V., Khanpurkalan, Sonepat, Haryana, India
sunita.bpsmv@gmail.com

Abstract. Task Scheduling is important part in cloud computing environment for heterogeneous resources. Task scheduling is to allocate tasks to the best suitable resources to increase performance in terms of some dynamic parameters. The proposed scheduling model is constructed for cloud applications in multi cloud environment and implemented in three phases (minimization, grouping & ranking and execution) and considered average waiting time, average turnaround time, completion time and makespan as performance parameters. In this scheduling model, execution time of tasks in cloud applications is generated through normal distribution and exponential distribution. Ranking of tasks is based upon shortest job first strategy (SJF) and results are compared with other ranking method based upon first come first serve (FCFS) and largest processing time first (LPTF). The proposed scheduling model gives better performance as per defined performance parameters.

Keywords: Cloud computing · Scheduling model · Task grouping · Ranking · DAG (Directed Acyclic Graph) · SJF

1 Introduction

Cloud computing is parallel and distributed computing that is collection of inter-connected, dynamic and virtualized resources [1]. Resource Management is a very important issue in cloud computing and some various important factor such as cost, performance, functionality are affected by resource management [2]. Cloud resource management is mainly concerned with two aspects i.e. resource allocation and task scheduling. Resource allocation is to allocating resources to the needed applications as per the availability of resources. Task scheduling is to schedule jobs on the allocated resources achieving maximum profit, efficient resource utilization and to meet user's QoS requirement. Cloud computing resource management model has interconnected shared resources and interdependent, interrelated tasks that fall into workflow application model [3]. Workflow application model can be represented by Directed Acyclic Graph (DAG), where nodes represent task and edges represent interdependency and relationship between these tasks.

S. Kaushik et al. (Eds.): ICICCT 2017, CCIS 750, pp. 11–24, 2017.
https://doi.org/10.1007/978-981-10-6544-6_2

In this paper, authors present a cloud task scheduling model in multi cloud environment based on workflow application models is presented. Cloud applications are represented as Directed Acyclic Graph. Execution time of all tasks corresponding to all resources is not known with certainty and the execution time of tasks is probabilistically generated through normal distribution and exponential distribution. This task scheduling model is works in three phases, in the first phase find minimum execution time of tasks, in second phase makes groups and assign ranking as per the SJF strategy and in the third phase execution of these groups are performed. This task scheduling model is compared with other algorithm that follows different ranking strategy i.e. FCFS (First Come First Serve) strategy and LPTF (Largest Processing Time First) strategy. Simulation is performed in MATLAB and a proposed scheduling model outperforms the other two models and minimizes waiting time, completion time and makespan.

This paper is organized as follows: Sect. 2 discusses the related work regarding this scheduling model. Section 3 presents proposed scheduling model and scheduling algorithm and its strategy. Section 4 provides the simulation results on the proposed scheduling algorithm using MATLAB. Section 5 gives the conclusion of this paper.

2 Related Work

Various resource allocation and task scheduling algorithms that are based upon grouping of tasks or jobs are proposed in cloud computing. A dynamic job- grouping scheduling strategy is [4] developed that considered processing requirements of jobs, granularity size and transmission of job groups to the required resources. Granularity size is used for measurement of total jobs executed within a specified time on a particular resource [4, 5]. Granularity is also defined as number of jobs to be grouped at a particular time [5].This scheduling strategy maximized the resource utilization and reduced overhead time of communication and processing of each jobs. An adaptive and parameterized job grouping scheduling algorithm is [5] proposed for grid jobs. Jobs are grouped according to the processing requirements of jobs, resource policies, network conditions and user's QoS requirements. In this scheduling algorithm all jobs are assumed for independent applications and [5] algorithm reduced the total processing time and cost of computational grid applications. An immediate mode scheduling of independent jobs in computational grids [6] that discussed the job allocation and considered makespan, flowtime, resource utilization and matching proximity parameters for performance measurement. This scheduling mode allocated the jobs to available resources for execution as these entered in the system. Fine grained grouping scheduling [7] that grouped lightweight jobs into coarse-grained jobs. Bandwidth-aware job grouping scheduling algorithm [8] proposed that grouped independent jobs with small processing requirements into suitable job groups with large processing requirements and scheduled according to network conditions. This scheduling strategy reduced the total job processing time. An algorithm [9] discussed in which jobs are grouped according to the ability of the resources according to the processing capabilities of resources. Scheduling algorithm [10] proposed that have task grouping, prioritization of bandwidth awareness and SJF algorithm and reduced processing time, waiting time and overhead. In this tasks are generated using Gaussian distribution and resources are created using random distribution. The smoothing concept for organization of tasks in heterogeneous multi-cloud environment [11] represented cloud model as DAG (Directed Acyclic Graph) and scheduling performed in two pass. In the first pass, tasks are divided into batches and in second pass batches are executed. A cost-based job grouping scheduling algorithm for grid computing environment [12] proposed where Job prioritization is done on the base of cost and then job grouping is performed. The fine grained jobs are grouped to coarse grained jobs and minimized the processing time and cost and achieved full utilization of resources. [13] proposed resource allocation algorithm for cloud computing system that have combined two algorithms, one is based on priority and second is based on earliest deadline first scheduling. This scheduling algorithm presented the task migration. In this scheduling, firstly assigned the priority of tasks and allocated the resources according to their priority and migrated the resources whenever they miss their deadline. This approach reduced the execution time and waiting time of preempt-able tasks. A task scheduling algorithm for heterogeneous multi-cloud environment that is based on Min-Min and Max-Min [14] proposed and this scheduling algorithm is executed with two phases and tested synthetic and benchmark data sets. This represented cloud as DAG (Directed Acyclic Graph) and performance is measured in terms of makespan and cloud utilization. A description of resource management and scheduling of cloud computing [15] presented and discussed cloud computing model and resource management model with virtual machine allocation and scheduling issues. Probabilistic availability based task scheduling algorithm (PATSA) [16] proposed where resources are scheduled on the basis of probabilistic availability of resources. This is based on assigning priority for each node by using rank value and application model is represented as DAG. Adaptive deadline based dependent job scheduling (A2DJS) algorithm for cloud computing [17] proposed that consisted job manager and data center. The job manger pointed the dependences among the tasks and eliminated ambiguity and data center comprised of job scheduler. This scheduling algorithm minimized the makespan of job and improved the utilization of the processing speed of the virtual machine. A hierarchical task model is proposed [18] that is associated with task scheduling for real applications in cloud computing. This algorithm considered parallel structure of sub-DAG and improved task execution concurrency and reduced the execution cost. A delay in task scheduling and delay-bound constraint is also discussed.

3 Scheduling Model

Cloud Scheduling Model (CSM) is represented with two parameters p and q, where p is the number of resources that are associated with scheduling model and q is the set of Cloud Applications (CA) that are associated with scheduling model. Cloud application is represented as DAG (Directed Acyclic Graph), i.e. CA = (T, E), where T is the set of nodes represents tasks of cloud application and E as edges represents the communication link between task. Each edge $e_{ij} = (t_i, t_j) \in E$ between task t_i and t_j represents the inter task communication and inter dependency of task.

Let consider CSM has m number of resources (R_1, R_2, …, R_m) and n number of cloud applications (CA_1, CA_2, …, CA_n) and each application with r number of tasks (T_1, T_2, …, T_r). Tasks of each cloud applications are submitted to resources for execution.

Execution Time matrix (ET_{ij}) for each task corresponding to each resource is generated.

Our proposed scheduling model is implemented with three phase. In the first phase, find the minimum execution time of each task corresponding to each resource, i.e. the task has different execution time on different resource (or machine) that has minimum execution time for tasks, for all tasks minimum execution time of resource are selected.

In second phase, makes the groups of tasks according to the minimum execution of tasks. The tasks with minimum execution time corresponding to that resource are placed in the same group. In this phase ranking is performed by mapping of the execution time of each task to a rank value. Ranking is based on the execution time of tasks. Rank value is assigned using Shortest Job First (SJF) strategy. The task with minimum execution time in a group is assigned the first rank, next minimum is assigned second rank and soon assign rank to each task.

Number of groups are formed as per the maximum number of resources, (let assume maximum value of resources is nn, then number of groups will be nn, Group 1, Group 2, …, Group nn). Group of tasks is represented with two parameters, Group [nn, a], where nn is the resource number and a is the number of tasks encountered in a group.

In third phase, these groups are executed one after another in the ascending numeric order of groups (Group[1, a], Group[2, a′], …, Group[nn, a″], where a, a′, a″ be the number of tasks in Group 1, Group 2, …, Group nn respectively) using the shortest job first (SJF) strategy, in the group which task has highest rank execute that task first, then second higher rank task and at last with lowest rank of task of that group and then execute second group and same process is repeated until all groups are executed.

3.1 Probabilistic Task Durations of Scheduling Model

In this scheduling model, execution time of tasks corresponding to resources is generated with two different approaches: one is the normally distributed and another one is exponentially distributed.

3.1.1 Normal Distribution

In this approach, execution time of tasks is to have normal distribution, following this distribution execution time is to be: ET_{ij} = sigma*randn(T,R) + mue;

Where, ET_{ij} is the execution time of task i on resource j, Sigma is standard deviation, Mue is mean value, T is Number of tasks, R is Number of Resources and randn() is function in MATLAB that generates normally distributed random numbers.

Here, value of sigma and mue is to be assumed for each application.

3.1.2 Exponential Distribution

In this approach, execution time of tasks is to have exponential distribution, following this distribution execution time is to be: ET_{ij}=exprnd(Mue,T,R);

Where, ET_{ij} is the execution time of task i on resource j, Mue is mean value, T is Number of tasks, R is Number of Resources and exprnd() is function in MATLAB that generates exponentially distributed random numbers. Here, value of Mue is to be assumed for each cloud application.

3.2 Notations Used in Algorithm

CSM(R,Q) – Cloud scheduling model with R resources and Q cloud applications
CA(T,E)- Cloud applications with T tasks and E communication links between tasks
R- Number of resources,
T- Number of tasks
ET_{ij} – Execution time of task i on resource j.
Rank- Rank Value of each task
CTT_{ij} – Completion Time of task i on resource j
CTG- Completion Time of a group on a resource
CT- Completion Time of all tasks or groups
WT-Waiting Time of task
AWT- Average Waiting Time
TAT- TurnAround Time
TTAT- Total TurnAround Time
ATAT- Average TurnAround Time
SJF-Shortest Job First
FCFS- First Come First Serve
LPTF- Largest Processing Time First

3.3 Scheduling Algorithm

Step1: Input T and R
Step 2: Generate Execution Time matrix ET_{ij}[T,R]
CASE 1: With task duration Normal Distribution
ET_{ij}=sigma*randn(T,R)+mue;
Sigma and mue are assumed for each cloud application.
CASE 2: With task duration Exponential Distribution
ET_{ij}=exprnd(Mue,T,R);
Mue are assumed for each cloud application.
Step 3: Find Minimum execution time of task for resources
Step 4: Make groups of tasks that execute on the same resource due to minimum execution time
Step 5: Assign Rank to each task in each group
Step 6: Execute all groups in ascending order of groups
Step 7: Calculate Total Waiting Time
TWT=WT+TWT;
Step 8: Compute Average Waiting Time
AWT=TWT/T;
Step 9: Calculate Total TurnAround Time
TTAT=WT+ET
Step 10: Compute Average TurnAround Time
ATAT=TTAT/T;
Step 11: Compute Completion Time of each Group
$CTG=CTT_1 +CTT_2+\ldots+CTT_n$; [n number of tasks in a group]
Step 12: Compute Completion time of all tasks
$CT=CTG_1+CTG_2+\ldots.+CTG_m$
Step 13: Compute Makespan [19]
Makespan=max $(CTG_1, CTG_2,\ldots.,CTG_m)$

4 Results and Discussion

In this paper, task scheduling model for cloud applications in multi cloud environment with Probabilistic Task Duration has proposed and its performance is evaluated. Simulation is performed using MATLAB. A comparison is made among three algorithms, in first one ranking is based on SJF strategy, in second one ranking is based on FCFS strategy, and third one is based on LPTF strategy. Here, we have showed two cases, in CASE 1 Execution Time is generated using Normal Distribution and in CASE 2 Execution Time is generated using Exponential Distribution. Performance is measured with average waiting time, average turnaround time, completion time of all groups and makespan = max (CTG_1, CTG_2, …, CTG_m) [19].

4.1 CASE 1: Execution Time of Tasks is Generated with Normal Distribution

In this execution time of tasks corresponding to resources is follows normal distribution and ten runs are made with different values of number of tasks and number of resources and different values of mue and sigma, average waiting time, average turnaround time, completion time and makespan is noted down as shown in Tables 1 and 2.

In Fig. 1, average waiting time is compared with different values of T, R, MUE and SIGMA as shown in Table 1, figure shows average waiting time of tasks with SJF ranking strategy is always less as compared to the other two ranking strategy.

In Fig. 2, average turnaround time is compared with different values of T, R, MUE and SIGMA as shown in Table 1, figure shows average turnaround time of tasks with SJF ranking strategy is always less as compared to the other two ranking strategy.

In Fig. 3, completion time of all tasks is compared with different values of T, R, MUE and SIGMA as shown in Table 2, figure shows completion time of all tasks with SJF ranking strategy is always less as compared to the other two ranking strategy.

In Fig. 4, makespan is compared with different values of T, R, MUE and SIGMA as shown in Table 2, figure shows makespan with SJF ranking strategy is always less as compared to the other two ranking strategy.

4.2 CASE 2: Execution Time of Tasks is Generated with Exponential Distribution

In this execution time of tasks corresponding to resources is follows exponential distribution and ten runs are made with different values of number of tasks and number of resources and different values of mue, average waiting time, average turnaround time, completion time and makespan is noted down as shown in Tables 3 and 4.

In Fig. 5, average waiting time is compared with different values of T, R and Mue as shown in Table 3, figure shows average waiting time of tasks with SJF ranking strategy is always less as compared to the other two ranking strategy.

Table 1. Table with normally distributed execution time of tasks in different cloud applications with different values of T, R, Mue and Sigma for average execution time and average turnaround time

Sr. no.	MUE	SIGMA	T	R	Average waiting time			Average turnaround time		
					SJF	FCFS	LPTF	SJF	FCFS	LPTF
Application 1	40.0342	29.9833	50	25	1.3801	3.1824	4.825	4.6622	6.4644	8.107
Application 2	29.7401	19.0356	350	100	0.75764	1.4661	2.1115	1.6193	2.3278	2.9731
Application 3	34.2915	24.9953	500	100	1.0577	2.1491	3.26	1.8808	2.9722	4.0831
Application 4	37.3542	29.3779	500	100	1.1419	2.2081	3.2859	1.9824	3.0486	4.1264
Application 5	42.2735	19.9938	600	150	1.5553	2.9551	4.3691	3.1268	4.5266	5.9406
Application 6	18.6186	6.5901	700	300	0.76688	1.4729	2.1857	2.0062	2.7122	3.425
Application 7	105.3209	48.4132	3000	500	1.85	3.6697	5.7004	3.1087	4.9284	6.9591
Application 8	93.2298	44.3965	3000	500	1.5153	3.0225	4.4857	2.5183	4.0255	5.4887
Application 9	86.4021	38.1572	5000	1000	0.77282	1.5234	2.3214	1.3876	2.1382	2.9362
Application 10	73.6956	34.2227	5000	1000	0.54622	1.0769	1.6587	0.98716	1.5178	2.0997

Table 2. Table with normally distributed execution time of tasks in different cloud applications with different values of T, R, Mue and Sigma for Completion time and Makespan

Sr. no.	MUE	SIGMA	T	R	Completion time			Makespan		
					SJF	FCFS	LPTF	SJF	FCFS	LPTF
Application 1	40.0342	29.9833	50	25	233.108	323.22	405.35	35.2635	45.1	62.306
Application 2	29.7401	19.0356	350	100	566.74	814.72	1040.6	26.6933	41.931	51.511
Application 3	34.2915	24.9953	500	100	940.412	1486.1	2041.5	62.0506	95.269	143.15
Application 4	37.3542	29.3779	500	100	991.188	1524.3	2063.2	73.064	116.75	131.02
Application 5	42.2735	19.9938	600	150	1876.08	2716	3564.3	63.0974	121.51	134.03
Application 6	18.6186	6.5901	700	300	1404.31	1898.5	2397.5	34.8354	54.512	62.551
Application 7	105.3209	48.4132	3000	500	9326.25	14785	20877	88.1595	166.93	210.9
Application 8	93.2298	44.3965	3000	500	7554.97	12077	16466	80.3987	150.99	216.96
Application 9	86.4021	38.1572	5000	1000	6937.98	10691	14681	35.7892	63.018	92.38
Application 10	73.6956	34.2227	5000	1000	4935.79	7589.1	10498	26.3422	48.746	65.765

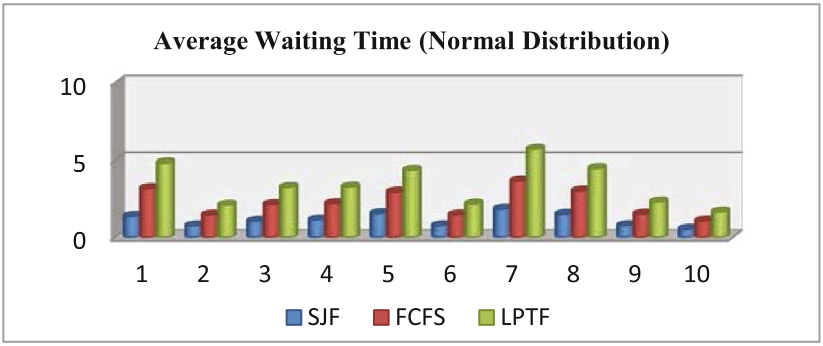

Fig. 1. Average waiting time for normally distributed execution time of cloud applications

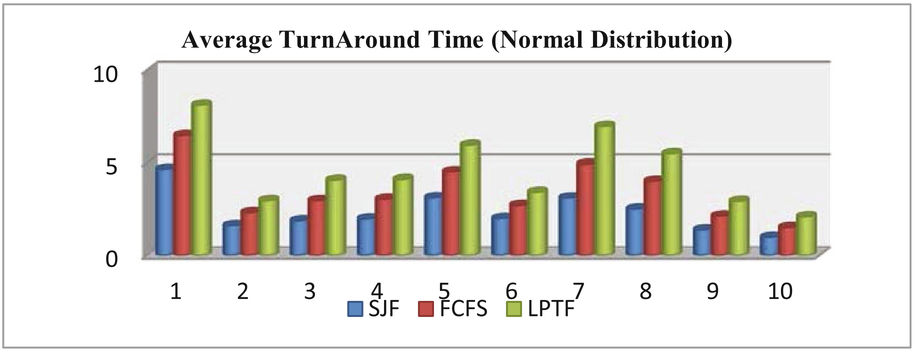

Fig. 2. Average turnaround time for normally distributed execution time of cloud applications

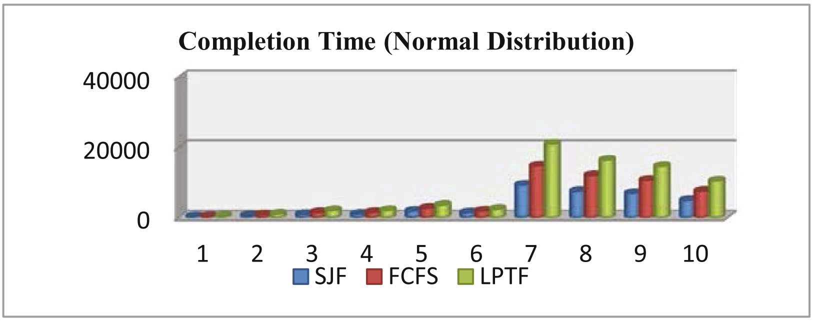

Fig. 3. Completion time for normally distributed execution time of cloud applications

In Fig. 6, average turnaround time is compared with different values of T, R and Mue as shown in Table 3, figure shows average turnaround time of tasks with SJF ranking strategy is always less as compared to the other two ranking strategy.

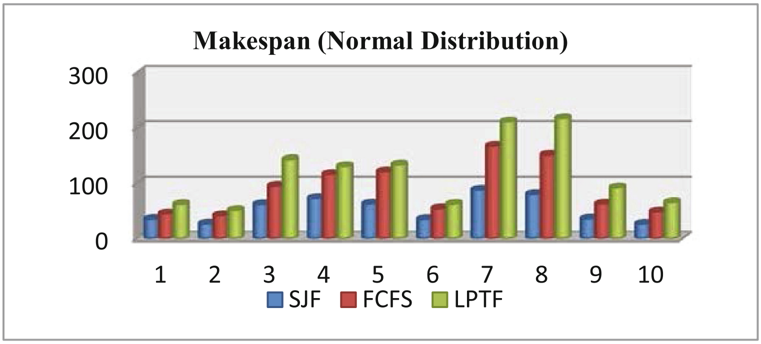

Fig. 4. Makespan for normally distributed execution time of cloud applications

Table 3. Table with exponential distributed execution time of tasks in different cloud applications with different values of T, R and Mue for average execution time and average turnaround time

Sr. no.	MUE	T	R	Average waiting time			Average turnaround time		
				SJF	FCFS	LPTF	SJF	FCFS	LPTF
Application 1	10.5638	100	30	0.382	0.71629	1.0394	0.76624	1.1005	1.4237
Application 2	14.1972	350	100	0.11869	0.24906	0.34515	0.25196	0.38232	0.47841
Application 3	21.89	350	100	0.18712	0.34369	0.58401	0.40219	0.55877	0.79909
Application 4	30.8007	500	100	0.35928	0.78456	1.1576	0.65786	1.0831	1.4562
Application 5	12.7863	500	100	0.16062	0.30157	0.46627	0.28914	0.43009	0.59479
Application 6	38.7043	600	150	0.28152	0.53863	0.83794	0.54884	0.80595	1.1053
Application 7	53.9315	800	250	0.16025	0.34642	0.51156	0.3862	0.57237	0.73751
Application 8	65.9462	1000	300	0.178	0.36543	0.54618	0.39964	0.58707	0.76783
Application 9	44.152	3000	500	0.12632	0.25779	0.38681	0.21218	0.34365	0.47268
Application 10	67.8355	3000	500	0.20992	0.42997	0.63597	0.35037	0.57042	0.77643

In Fig. 7, completion time of all tasks is compared with different values of T, R and Mue as shown in Table 4, figure shows completion time of all tasks with SJF ranking strategy is always less as compared to the other two ranking strategy.

In Fig. 8, makespan is compared with different values of T, R and Mue as shown in Table 4, figure shows makespan with SJF ranking strategy is always less as compared to the other two ranking strategy.

Table 4. Table with exponential distributed execution time of tasks in different cloud applications with different values of T, R and Mue for completion time and makespan

Sr. no.	MUE	T	R	Completion time			Makespan		
				SJF	FCFS	LPTF	SJF	FCFS	LPTF
Application 1	10.5638	100	30	76.6243	110.054	142.366	13.4921	21.6426	22.8266
Application 2	14.1972	350	100	88.1857	133.812	167.445	4.4786	8.3735	8.5562
Application 3	21.89	350	100	140.768	195.568	279.68	6.0347	7.3009	14.8408
Application 4	30.8007	500	100	328.93	541.572	728.092	19.1547	38.7021	45.6242
Application 5	12.7863	500	100	144.568	215.044	297.395	7.1397	11.8357	17.1671
Application 6	38.7043	600	150	329.302	483.569	663.152	11.832	15.2958	22.5624
Application 7	53.9315	800	250	308.963	457.897	590.01	8.8954	14.9909	19.0281
Application 8	65.9462	1000	300	399.642	587.071	767.829	9.5586	14.5679	23.3159
Application 9	44.152	3000	500	636.548	1030.96	1418.03	6.611	12.1458	17.682
Application 10	67.8355	3000	500	1051.12	1711.27	2329.29	11.9007	24.0619	27.4316

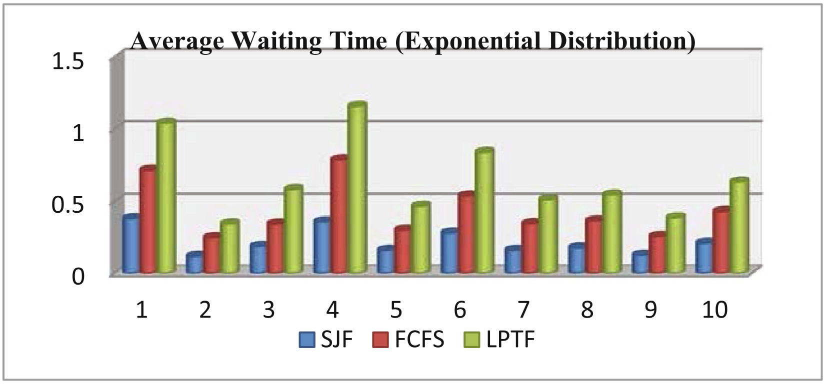

Fig. 5. Average waiting time for exponentially distributed execution time of cloud applications

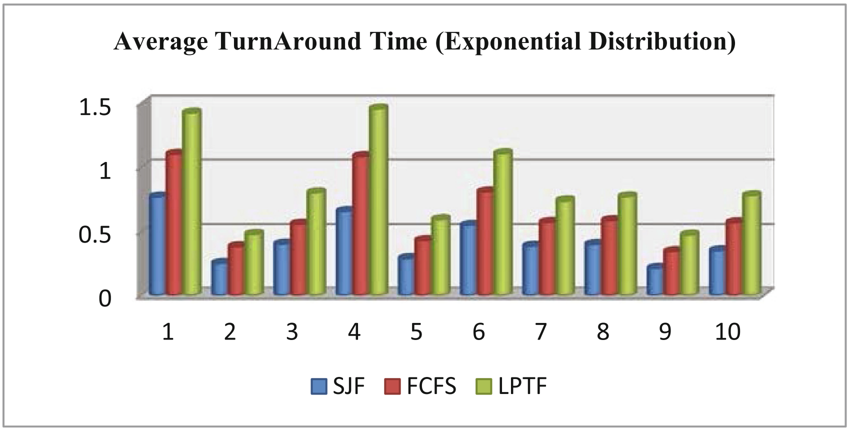

Fig. 6. Average turnaround time for exponentially distributed execution time of cloud applications

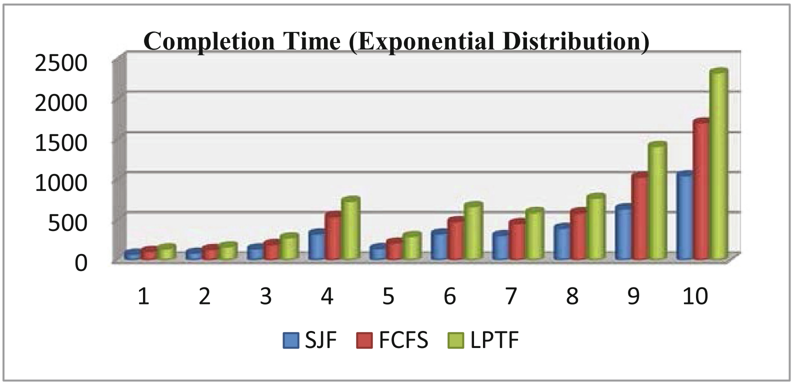

Fig. 7. Completion time for exponentially distributed execution time of cloud applications

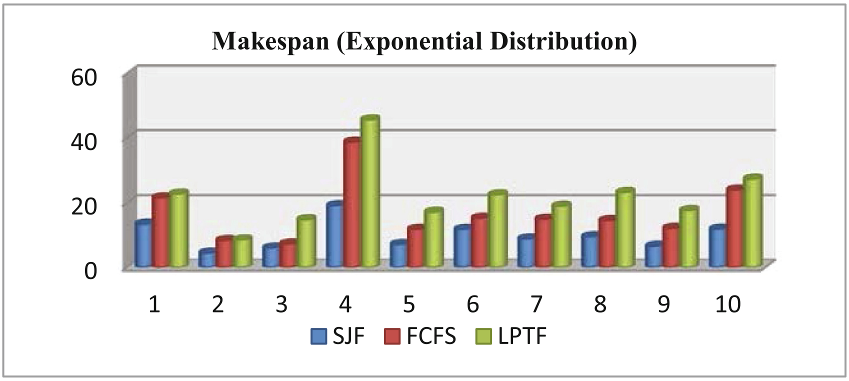

Fig. 8. Makespan for exponentially distributed execution time of cloud applications

5 Conclusion

The presented task scheduling model for different cloud applications in multi cloud environment is simulated in MATLAB and results show the presented algorithm give better performance i.e. minimized average waiting time and turnaround time and minimized completion time and makespan. The simulation results gives two cases, one case is where execution time of task in generated through normal distribution and another is where execution time of tasks is generated through exponential distribution. The simulation results show that presented scheduling model is compared with other two ranking strategy, i.e., first come first serve (FCFS) and largest processing time first (LPTF).

References

1. Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I.: Cloud computing and emerging IT platforms: vision, type, and reality for delivering computing as the 5th utility. Futur. Gener. Comput. Syst. **25**, 599–616 (2009)
2. Suri, P.K., Rani, S.: Simulator for priority based scheduling of resources in cloud computing. Int. J. Comput. Appl. **146**(14), 10–15 (2016)
3. Lee, Y.C., Subrata, R., Zomaya, A.Y.: On the performance of a dual-objective optimization model for workflow applications on grid platforms. IEEE Trans. Parallel Distrib. Syst. **20**(9), 1273–1283 (2009)
4. Muthuvelu, N., Liu, J., Soe, N.L., Venugopal, S., Sulistio, A., Buyya, R.: A dynamic job grouping-based scheduling for deploying applications with fine-grained tasks on global grids. In: Coddington, P., Wendelborn, A. (eds.) Australasian Workshop on Grid Computing and e-Research (AusGrid2005), Newcastle, Australia. Conferences in Research and Practice in Information Technology, vol. 44. Australian Computer Society, Adelaide (2005)
5. Muthuvelu, N., Chai, I., Eswaran, C.: An adaptive and parameterized job grouping algorithm for scheduling grid jobs. In: ICACT, pp. 975–980 (2008). ISBN 978-89-5519-136-3
6. Xhafa, F., Barolli, L., Durresi, A.: Immediate mode scheduling of independent jobs in computational grids. In: 21st International Conference on Advanced Networking and Applications (AINA 2007) (2007)

7. Liu, Q., Liao, Y.: Grouping-based fine-grained job scheduling in grid computing. In: First International Workshop on Education Technology and Computer Science, pp. 556–559 (2009)
8. Ang, T.F., Ng, W.K., Ling, T.C., Por, L.Y., Liew, C.S.: A bandwidth-aware job grouping-based scheduling on grid environment. Inf. Technol. J. Asian Netw. Sci. Inf. **8**, 372–377 (2009). ISSN 1812-5638
9. Mishra, M.K., Mohanty, P., Mund, G.B.: A modified grouping-based job scheduling in computational grid. In: International Conference on Current Trends in Technology, Nuicone, pp. 1–6 (2011)
10. Ru, J., Keung, J.: An empirical investigation on the simulation of priority and shortest-job-first scheduling for cloud-based software systems. In: 22nd Australian Conference on Software Engineering, pp. 78–87 (2013)
11. Panda, S.K., Nag, S., Jana, P.K.: A smoothing based task scheduling algorithm for heterogeneous multi-cloud environment. In: International Conference on Parallel, Distributed and Grid Computing, pp. 62–67. IEEE (2014)
12. Yadav, S., Agarwal, A., Rastogi, R.: Cost-based job grouping and scheduling algorithm for grid computing environments. Int. J. Comput. Appl. **91**(15), 21–27 (2014). ISSN 0975-8887
13. Gupta, G., Kumawat, V.K., Laxmi, P.R., Singh, D., Jain, V., Singh, R.: A simulation of priority based earliest deadline first scheduling for Cloud computing system. In: First International Conference on Networks & Soft Computing, pp. 35–39 (2014)
14. Panda, S.K., Jana, P.K.: An efficient task scheduling algorithm for heterogeneous multi-cloud environment. In: International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1204–1209 (2014)
15. Bittencourt, L.F., Madeira, E.R., da Nelson, L.S., Fonseca, N.L.S.: Resource management and scheduling. In: Fonseca, N., Boutaba, R. (eds.) Cloud Services, Networking, and Management, 1st edn, pp. 243–267. Wiley, New York (2014)
16. Chitra, S., Prashanth, C.S.R.: Probabilistic availability based task scheduling algorithm. IEEE (2015)
17. Komarasamy, D., Muthuswamy, V.: Adaptive deadline based dependent job scheduling algorithm in cloud computing. In: Seventh International Conference on Advanced Computing (ICoAC), pp. 1–5 (2015)
18. Mao, Y., Zhong, H., Li, X.: Hierarchical model-based associate tasks scheduling with the deadline constraints in the cloud. In: Proceeding of the 2015 IEEE International Conference on Information and Automation, Lijiang, China, pp. 268–273, August 2015
19. Oyetunji, E.O., Oluleye, A.E.: Minimizing makespan and total completion time criteria on a single machine with release dates. J. Emerg. Trends Eng. Appl. Sci. **1**(1), 100–108 (2010)

http://www.springer.com/978-981-10-6543-9

Information, Communication and Computing Technology
Second International Conference, ICICCT 2017, New Delhi, India, May 13, 2017, Revised Selected Papers
Kaushik, S.; Gupta, D.; Kharb, L.; Chahal, D. (Eds.)
2017, XIII, 388 p. 163 illus., Softcover
ISBN: 978-981-10-6543-9