ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 11, November 2017

Round Robin based Prioritized Earliest Deadline First Scheduling in Cloud Computing

[1] NeelimaPriyanka N, [2] Suresh Varma P, [3] R KrishnamRaju Indukuri, [4] B Sukumar Babu
[1] Research Scholar, Dept of CSE, AdikaviNannaya University ,[2] Dept of CSE, AdikaviNannaya University, [3] Department of CS,B.V.Raju College, [4] Department of CSE, VITW

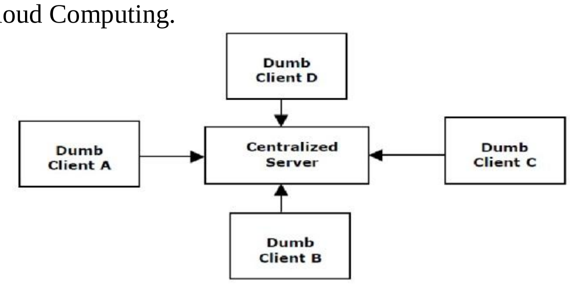
***Abstract* - "Cloud computing is a ultimate model for processing the user requests in a convient way by providing on-demand network accessing to a available shared computing resources like , networks ,servers , storage, high end applications ,and many other services etc which can be fastly provided and acquired back with minimal management effort or with service provider interaction ".Thus Cloud Computing is a new phenome`non in Information Technology where computing is delivered as service rather than product, through shared resources, software and information to consumers as an utility over networks. In cloud computing scheduling is an important activity to improve resource utilization. In this paper we presented Round Robin based Prioritized Earliest Deadline First (EDF) Scheduling model for resource allocation based on Time Quantum, priorities, deadlines and processing times while allocating resources to the required jobs. The performance metrics Average Turnaround Time, Average Waiting Time and Average Deadline Violation are reduced reasonably, when compare to traditional scheduling models like FCFS and SJF Scheduling Models.**

Keywords - Hidden Web Crawler, Query Optimization, Search engines, Metadata, document frequency, term weights, Time Quantum, Optimization, Efficient Round Robin EDF.

INTRODUCTION

In a general way the computation or computing main purpose is to solve a problem so it is goal-oriented activity for requiring, benefiting from, or for creating computers. Thus, computing includes developing and building hardware and software systems for a wide range of problems or purposes. They are processing a problem, structuring a model, and managing various kinds of information or data the example activities by using the computations are doing scientific studies using computers, making computer systems to act or behave as intelligent robots, creation of entertainment and for communications and media etc. This list goes on endlessly, and the possibilities are also very vast".

In Computing we have seen centralized computing, Distributed computing, Grid computing, Cluster Computing, Utility Computing and Finally now seeing Cloud Computing.



Centralized Network Computing Model

Centralized computing: Example mainframe computer where every resource is placed in a single centralized system if that system goes down everything goes down

Distributed computing: there are several autonomous computational bodies, each of which has its own local memory and the workstations communicate with each other by the concept of message passing examples for Distributed computations are Internet. Workgroups etc.

Grid Computing: Grid computing is a making a super computer infrastructure by including the infrastructure across the network to work on particular problem with less expensive power

Cluster Computing: Making a single unit, which is locally deployed to improve speed, reliability, and accuracy compared to a single computer with the same thing much cost effective.

Utility Computing: The practical implementation of cloud can be seen in utility computing .In this modelthe service provider makes resources and infrastructure available to the customer as and then needed, and charges him for usage of the resource rather than a common or fixed rate, which generally called flat rate.

Cloud Computing is a merging amalgam in which all the resources are permanently stored on the server and are utilized by the clients through internet [Barrie Sosinsky, 2011]. Internet is set of public and private networks, which are interconnected, with a large pool of devices. A cloud provides dynamic services to the end user's by

IFERP
connecting engineers... developing research

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 10, November 2017

using all scalable and virtualized resources over the internet. In cloud computing the services are utilized by all end users on rental bases of pay-per-use model, in such a way that the services are guaranteed and targeted to produces high quality and utility of network storage space and system resources. Many experts believe that this cloud technology will occupy a larger important space in IT Industry. Services are the basic principle behind the emergence of cloud computing. Cloud computing has now been considered as a good business and enterprise model for the future of computing sectors. Cloud Computing provides different service models: Software-as-Service(SaS): The concept of providing software application as a service on demand over the internet, which means it can be run anywhere and anytime on pay per use, and a single platform is used to utilize the software, Examples: salesforce.com, buying software on demand, Platform-as-Service(PaS): The concept of providing software application as a service level application development environment as a service over the internet right from requirements to the complete life cycle, it offers a common platform on which the software and data can be accessed, Examples: Azure Services, Amazon Web Services and Google AppEngine. Infrastructure-as-Service(IaS): The concept of providing the whole IT infrastructure such as storage, virtual environment, servers, platforms and application. It looks after the bare computing resources and backup services, Examples: Amazon EC2, VMWare. Cloud Computing is used in applications like: Scientific, Commercial and Educational etc. In entire cloud environment, Job scheduling is one of the most important activity – which takes care efficiency and work load distribution of the jobs with the virtual machines[Jagbeer Singh, 2011]. In Cloud environment the main goal of the scheduling algorithms, ensure an effective utilization of resources.

In cloud computing to accomplish the user task, the job requires the cloud resources. Usually in cloud computing the resource required for a job request is allocated in the form of Virtual Machine (VM). The scheduler in cloud computing schedules the given ‘n’ number of job requests and assigns the required cloud Virtual Machines for each job request. In Cloud Computing, ‘m’ number of Virtual Machine types may require to complete a job request. Priorities of job requests its deadlines are also play a major role in resource allocation in cloud computing [NeelimaPriynaka N, Suresh Varma P and R Krishnam Raju Indukuri, 2017]. Scheduling is the process in which the available resources are shared by the jobs in an order. The resource can be a machine instance, data storage device, an application or an environment. Allocation of these cloud resources to the clients is an important task in cloud computing. Hence, an efficient deadline aware scheduling model is required to schedule the given ‘n’ number of jobs.

In this paper, we designed and developed PEDF Scheduling model for resource allocation by considering the Average Turnaround Time, Average Waiting Time and Average Deadline Violation as performance metrics..

RELATED WORK

Greedy Based Job Scheduling Algorithm [Li, Ji, LonghuaFeng, 2014] focuses on QoS, as cloud computing is a business-oriented service. User fairness and efficiency are important issues for job scheduling in cloud environments. As cloud is a business-oriented service, it must concern about both shorter completion time as well as better QoS of cloud costumer. The goal of this algorithm is to decrease the completion time in order to provide a faster solution to the scheduling problem. Based on the QoS, the algorithm classifies the task category and calls the appropriate function. The algorithms produced the best results when compared with the other algorithms based on Berger model and existing CloudSim tool scheduling strategy.

Priority Based Earliest Deadline First Scheduling Algorithm [Gupta, Gaurav, et al, 2014]is defined by considering the two scheduling models, Priority Based Scheduling Algorithm and Earliest Deadline First algorithm. This algorithm mainly concentrates on utilization of memory and resource allocation. This algorithm optimizes the completion time of preempted jobs and improves the efficiency of scheduling. In this algorithms the authors finds the solution for problem of waiting time on preempted tasks, the waiting queue is used to processes the preempted tasks.

Earliest Feasible Deadline First [Jagbeer Singh, 2011] algorithm reduces the time complexity of Earliest Deadline First (EDF). Deadline is taken as the scheduling criteria. In this the process migration between the machines optimizes the time complexity. This work focuses on some modification to the global Earliest Deadline First algorithms to decrease the number of task migration and also to add predictability to its behavior.

IFERP
connecting engineers... developing research

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 10, November 2017

The experiment results of the algorithm reduced the time complexity

A Scheduling Algorithm based on Priority for VM Allocation [Xiao, Jing, and Zhiyuan Wang, 2012] focus to provide more advantages to the service vendors and providers. In the previous algorithm, as there are no sufficient resources to handle all the request, the new algorithms proposes a priority based algorithm to find the best fit. This strategy improves the effective utilization of resources when compared with FCFS strategy.

Improved Cost Based Algorithm [Selvarani, S., and G. SudhaSadhasivam, 2010] improves the efficiency of allocating resources to the jobs is much improved, when compared with the traditional cost-based scheduling algorithm. The objective of this paper is to schedule task groups in cloud computing platform, where resources have different resource costs and computation performance. Due to job grouping, communication of coarse-grained jobs and resources optimizes computation/communication ratio. For this purpose, an algorithm based on both costs with user task grouping is defined. This scheduling approach in cloud employs an improved cost-based scheduling algorithm for making efficient mapping of tasks to available resources in cloud. This scheduling algorithm measures both resource cost and computation performance, it also improves the computation / communication ratio by grouping the user tasks according to a particular cloud resource's processing capability and sends the grouped jobs to the resource.

In Priority Based Job Scheduling Algorithm [Ghanbari, Shamsollah, 2012]each job is assigned with some priority value, based on the job priority the resources are allocated to the jobs. This algorithm the author focuses on problems like complexity, consistency and makespan. As per author,the performance can be improved by reducing the makespan.

ArnavWadhonkar and DeeptiTheng[ArnavWadhonkar and DeeptiTheng, 2016] developped scheduling algorithm which schedules the tasks based on their length and deadline. Results are compared with traditional algorithms and comparative analysis shown that a reduction in makespan and average waiting time.

Priority Based Earliest Deadline Scheduling In Cloud Computing [NeelimaPriynaka N, Suresh Varma P and R KrishnamRajuIndukuri, 2017] developed a scheduling algorithm in which the jobs are scheduled in a multi stage virtual machine by considering the Priority and Deadline Violation time of the processes. The results shown that the PEDF is giving better Average time and Turnaround time when compared with the regular scheduling algorithms

In the review of literature no author reports cloud scheduling with round robin model with priorities and deadlines. Hence in this paper we designed and developed Round Robin based Prioritized Earliest Deadline First (EDF) Scheduling model for resource allocation based on priorities, deadlines and processing times while allocating resources to the required jobs.

ROUND ROBIN BASED PRIORITIZED EDF

In the cloud computing the scheduling concepts can be categorized into mainly two, one is clock-driven and second one is event-driven. Under clock driven the scheduling points are determined by the interrupts received from a clock. In Event-driven the scheduling are defined by certain events. The Clock-driven works well only when the number of requests are low. When the number of tasks increases, it is very complex to determine a suitable frame size as well as a feasible schedule.

To overcome this problem the Event-driven Schedulers are introduced. Event-driven Schedulers can handle sporadic and aperiodic tasks more efficiently. There are many types under the Event-driven model like "Simple Priority Based, Rate Monotonic Analysis, Earliest Deadline First etc."

EDF Working process: conventional EDF model works on uniprocessor .In Earliest Deadline First Scheduling at every scheduling point the task having the shortest deadline is taken up for scheduling. And this is proved that EDF is optimal for uniprocessor. In order to apply EDF model first point is, to identify the set of tasks, whether they are schedulable or not by applying Schedulability Test, with the assumption that the period of each task is the same as its deadline

The necessary and sufficient condition for the Schedulability Test when Pi>=dli

For a set of tasks (T1,T2,T3,………Tn) with (e1,e2,e3……..en) execution times

IFERP
connecting engineers... developing research

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 10, November 2017

$\sum ei/min(pi,dli) <= 1$ (i=1 to n). 3.1

The same equation becomes just the sufficient condition if pi<=dli

The deadline values should be in such a way that the task should be completing its execution with in the stipulated period of time The defining deadline should satisfy the necessary and sufficient condition of EDF schedulability Test.

Thus the deadlines are considered based on the following calculation.

$dl_i = (t_{i1} + t_{i2} + \dots + t_{i4}) / 4 + (Median(t_{i1}) * \alpha_1 + t_{i1} * \alpha_2) + (Median(t_{i2}) * \alpha_3 + t_{i2} * \alpha_4) + \dots\dots + (Median(t_{im}) * \alpha_{2m-1} + t_{i4} * \alpha_{2m})$

Where $0 < \alpha_i < 1$ 3.2

For example when m=4

$dl_i = (t_{i1} + t_{i2} + t_{i3} + t_{i4}) / 4 + (Median(t_{i1}) * \alpha_1 + t_{i1} * \alpha_2) + (Median(t_{i2}) * \alpha_3 + t_{i2} * \alpha_4) + (Median(t_{i3}) * \alpha_5 + t_{i3} * \alpha_6) + (Median(t_{i4}) * \alpha_7 + t_{i4} * \alpha_8)$ 3.3

The performance metrics can be computed by the following computations for a given scheduling sequence. Waiting time for requestr_i is the difference between completion time (c_i) and total processing time of request ($t_{i1}+ t_{i2}+ \dots + t_{im}$). Waiting time of a r_i is the time taken to start its work on VM of type-1, plus the time elapsed between completion of work on VM of type-1 and start of its work on VM of type-2, plus time elapsed between completion of work on VM type-2 and start of its work on VM of type -3, and so on, plus time elapsed between completion of work on VM of type-(M-1) and start of its work on VM of type-M. We denote s_i is the starting time of r_i on VM of type-1 and c_i is completion time of r_i on VM of type-M.

$w_i = c_i - (t_{i1}+ t_{i2}+ \dots + t_{im})$

Deadline violations of a job request r_i with respect to turn around time (dvt_i) is the difference between actual turnaround time (c_i) and deadline of job r_i (dl_i) i.e

$dvt_i = w_i - dl_i$ where $w_i > dl_i$

Average Waiting Time(AWT) and Average Turnaround Time(ATT) of all job requests will be computed as follows

$AWT = \left(\sum_{i=1}^{n} w_i \right) / n$

$ATT = \left(\sum_{i=1}^{n} c_i \right) / n$

Average Deadline Violation with respect to turn around time (ADVT) can be calculated as follows.

$ADVT = \left(\sum_{i=1}^{n} dvt_i \right) / n$

The obtained schedulable RR EDF is compared with FCFS, SJF and proved to be optimized

The resource allocation in cloud computing generally done in terms of allocating virtual machines as resources to the requested jobs. As the users are very specific in terms of the response time and waiting time, hence an Earliest Deadline Scheduling model based on RoundRobin is developed. In this Scheduling the scheduler receives 'n' jobs from various users and assigns the resources in the form of Virtual Machines by scheduling the job requests based on their time quantum. In this model a job requires 'm' different types of Virtual Machines in sequence to complete its task by considering the deadline with respect to waiting time and response time. A model is developed and analyzed for evaluation of average turnaround time, average waiting time and violation in deadlines when compared with other scheduling strategies First Come First Serve (FCFS) scheduling, Shortest Job First (SJF) scheduling and RoundRobin based Prioritized EDF Scheduling Models.

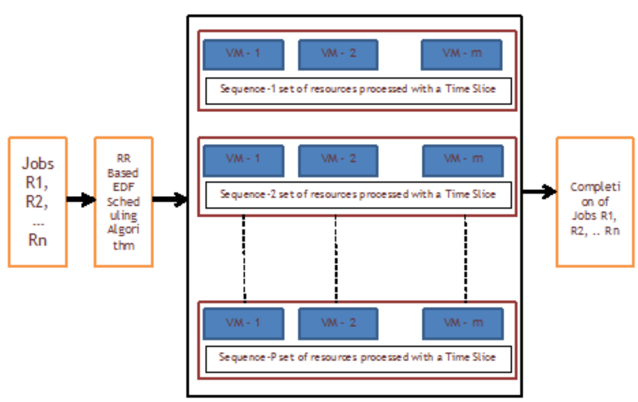


Figure 1: Round Robin Based Prioritized EDF Scheduling Model

Virtual Machine (VM) allocation for each job request in the proposed model has been described in Figure 1. Let r_1, r_2, …. , r_n be the set of job requests in the cloud computing, where 'n' is number of job requests and r_i denotes i^{th} job request. Each job request r_i requires t_{i1} units of time to process on an instance of VM of type-1 which will be allocated based on time slice and t_{i2} units of time to process on an instance of VM of type-2 which will be based on time slice and so on such that t_{im}units of time to process on an instance of VM of type-M to complete its task.

Let dl_i be the deadline of r_i to complete its task. Similarly q_i is the priority of each job request, and the low value of q_i is considered as immediate priority.

The deadlines are considered based on the following calculation.

$dl_i = t_{i1}+t_{i2}+ \dots +t_{i4}) /4 + (Median(t_{i1}) * \alpha_1 + t_{i1} * \alpha_2) + (Median(t_{i2}) * \alpha_3 + t_{i2} * \alpha_4) + \dots\dots + (Median(t_{im}) * \alpha_{2m-1} + t_{i4} * \alpha_{2m})$ Where $0 < \alpha_i < 1$

For example when m=4

$dl_i = t_{i1}+t_{i2}+t_{i3}+t_{i4}) /4 + (Median(t_{i1}) * \alpha_1 + t_{i1} * \alpha_2) + (Median(t_{i2}) * \alpha_3 + t_{i2} * \alpha_4) + (Median(t_{i3}) * \alpha_5 + t_{i3} * \alpha_6) + (Median(t_{i4}) * \alpha_7 + t_{i4} * \alpha_8)$

The performance metrics can be computed by the following computations for a given scheduling sequence. Waiting time for requestr_i is the difference between completion time (c_i) and total processing time of request ($t_{i1}+ t_{i2}+ \dots + t_{im}$**)**. Waiting time of a r_i is the time taken to start its work on VM of type-1, plus the time elapsed between completion of work on VM of type-1 and start of its work on VM of type-2, plus time elapsed between completion of work on VM type-2 and start of its work on VM of type -3, and so on, plus time elapsed between completion of work on VM of type-(M-1) and start of its work on VM of type-M. We denote s_i is the starting time of r_i on VM of type-1 and c_i is completion time of r_i on VM of type-M.

$w_i = c_i - (t_{i1}+ t_{i2}+ \dots + t_{im})$

Deadline violations of a job request r_i with respect to turn around time (dvt_i) is the difference between actual turnaround time (c_i) and deadline of job r_i (dl_i) i.e

$dvt_i = w_i - dl_i$ where $w_i > dl_i$

Average Waiting Time(AWT) and Average Turnaround Time(ATT) of all job requests will be computed as follows

$$AWT = \left(\sum_{i=1}^{n} w_i \right) / n$$

$$ATT = \left(\sum_{i=1}^{n} c_i \right) / n$$

Average Deadline Violation with respect to turn around time (ADVT) can be calculated as follows.

$$ADVT = \left(\sum_{i=1}^{n} dvt_i \right) / n$$

Algorithm : Round Robin based Prioritized EDF Scheduling Algorithm

Input : 'n' number of job requests with processing times $t_{i1}, t_{i2}, t_{i3} \dots, t_{im}$on

'm' types of Virtual Machines VM_1, VM_2, VM_3 and VM_m,

p number of instances are available for each Machine with a time slice ts

$d1_i$ is deadline of i^{th} job request

q_i is the priority of each job request

Output : Optimal Scheduling sub sequences Seq_1, Seq_2, $Seq_3, \dots Seq_p$

1. begin
2. i=0;
3. solution_vector = empty;
4. for k=1 to 2m do
5. α_k = choice(0.1,0.9);
6. end for;
7. for each possible values of β and γ to do
8. for each job request r_i with highest priority and minimum deadline and processing time, will be processed first for the given time slice, followed by the remaining sequence based on β and γ among all unprocessed jobs do
9. For sequence-i: Set (Ra, Rb, Rc, Rd..) repeat step-10 until all processing time equals to zero
10. If(Processing time(i) >ts)
    Processing time(i) = processing time(i) – ts
    Execution time(i) = Execution time(i) + ts
    Else
    Processing time = 0
    Execution time = Execution time + Processing time
11. add the job request r_i to the solution_vector at index i;

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 10, November 2017

```
            i=i+1;
        end for;
        for i=0 to n-1 do
            j = i % p;
            append solution vector[i] to the scheduling sub sequence Seq_j;
        end for;
        for i=1 to p do
            calculate performance metrics for each scheduling sub sequence Seq_i;
        end for;
        calculate aggregate performance metrics for the entire scheduling sequence;
    end for;
end;
```

The RoundRobin Based Prioritized EDF Scheduling Algorithm accepts 'n' job request and assign feasible values for α_i ($0 < \alpha_i <= 1$) and β and γ for better performance metrics. This scheduling algorithm schedules the given job request based on time slicing. If two jobs have same priorities then deadlines and processing times will be considered to break the tie. This scheduling algorithm finds best values for α_k, β and γ for optimal scheduling sequences.

PERFORMANCE EVALUATION OF THE MODEL

A customized simulator has been developed to analyze the First Come First Serve (FCFS) Scheduling, Shortest Job First (SJF) Scheduling and Round Robin based Prioritized Earliest Deadline First Scheduling with 'p' Virtual Machine instances for each resource type. Gaussian distribution is used to generate job requests and its processing times randomly.

Initially First Come First Serve (FCFS) scheduling algorithm is applied for the given instance by considering the jobs in order of arriving and priority, splitting the scheduling sequence into 'p' sub sequences. Next SJF Scheduling is applied to the given instance by sorting the jobs in ascending order of ($t_{i1} + t_{i2} + t_{i3} + \ldots + t_{im}$) and priorities, splitting the scheduling sequence into sub scheduling sequences.

Now Round Robin Based Prioritized Earliest Deadline Scheduling algorithm is applied at the end for the given instance to generate optimal scheduling sequence and splits the scheduling sequence into p number of scheduling sub sequences. This algorithm finds optimal solution by assigning reasonable values for β and γ. At end performance evaluation metrics are calculated.

In Table 1, we have considered n=32 jobs on m=4 types of Virtual Machines with p=8 instances are available, and considered α_1=0.2, α_2=0.3, α_3=0.2, α_4=0.3, α_5=0.2, α_6=0.3, α_7=0.2, α_8= 0.3 in calculating deadline of each job request (dl_i). Also 'P' is the priority of each job request. RID is the resource ID, T1, T2, T3 and T4 are the processing time of virtual machine 1, 2, 3 and 4, Dl is the Deadline Time.

Table 1: Scheduling Instance n=32, m=4 and p=8

RID	T1	T2	T3	T4	DL	P
0	22	191	277	39	915	2
1	238	368	112	17	1236	3
2	346	65	206	7	1001	5
3	135	85	77	5	588	5
4	348	78	14	75	863	1
5	126	136	74	11	662	2
6	445	226	265	60	1533	2
7	294	55	195	76	993	2
8	296	201	73	50	1037	2
9	53	71	249	74	773	4
10	373	48	228	31	1069	3
11	154	299	269	33	1241	3
12	187	326	36	28	1018	2
13	498	126	138	0	1199	2
14	167	273	255	19	1180	4
15	321	15	296	76	1095	2
16	484	65	290	2	1283	4
17	378	436	70	50	1515	5
18	191	295	85	90	1118	3
19	139	115	216	97	942	5
20	46	146	146	52	721	4
21	240	268	214	79	1292	2
22	259	200	202	85	1200	4
23	357	160	156	30	1132	1
24	130	163	48	37	711	5
25	11	394	241	5	1135	5
26	471	311	279	17	1665	5
27	197	182	89	91	952	1
28	320	176	255	1	1201	4
29	499	77	78	12	1059	2
30	355	197	173	52	1213	3
31	344	45	259	37	1074	1

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 10, November 2017

Table 2: FCFS Scheduling for Instance n=32, m=4 and p=8

TYPE	RID	ST	ET	WT	CT	DL	P	DV
FCFS	0	0	529	0	529	915	2	0
FCFS	1	100	735	806	1641	1236	3	503
FCFS	2	300	624	852	1776	1001	5	647
FCFS	3	300	302	297	899	588	5	187
FCFS	4	0	515	815	1330	863	1	602
FCFS	5	0	347	300	647	662	2	169
FCFS	6	0	996	753	1749	1533	2	418
FCFS	7	0	620	600	1220	993	2	426
FCFS	8	100	620	735	1455	1037	2	487
FCFS	9	200	447	0	647	773	4	0
FCFS	10	100	680	780	1560	1069	3	570
FCFS	11	200	755	300	1255	1241	3	74
FCFS	12	100	577	834	1511	1018	2	578
FCFS	13	100	762	813	1675	1199	2	501
FCFS	14	122	714	758	1594	1180	4	538
FCFS	15	100	708	865	1073	1095	2	697
FCFS	16	200	841	561	1602	1283	4	287
FCFS	17	253	934	536	1723	1515	5	129
FCFS	18	200	661	496	1357	1118	3	253
FCFS	19	222	567	628	1417	942	5	501
FCFS	20	200	390	100	690	721	4	4
FCFS	21	22	801	850	1673	1292	2	596
FCFS	22	200	259	594	1053	1200	4	364
FCFS	23	0	703	1066	1769	1132	1	808
FCFS	24	300	378	226	904	711	5	80
FCFS	25	246	651	335	1232	1135	5	133
FCFS	26	300	1078	626	2004	1665	5	235
FCFS	27	0	559	300	859	952	1	111
FCFS	28	300	752	612	1664	1201	4	364
FCFS	29	100	666	556	1322	1059	2	268
FCFS	30	200	757	732	1689	1213	3	466
FCFS	31	0	685	853	1538	1074	1	659

Table 3: SJF Scheduling for Instance n=32, m=4 and p=8

TYPE	RID	ST	ET	WT	CT	DL	P	DV
SJF	0	0	529	0	529	915	2	0
SJF	1	200	765	886	1821	1236	3	583
SJF	2	253	624	420	1297	1001	5	215
SJF	3	300	302	344	946	588	5	234
SJF	4	0	515	901	1416	863	1	688
SJF	5	0	347	262	609	662	2	131
SJF	6	22	996	730	1748	1533	2	395
SJF	7	100	620	600	1320	993	2	426
SJF	8	0	620	600	1220	1037	2	352
SJF	9	200	447	74	721	773	4	12
SJF	10	100	680	896	1676	1069	3	686
SJF	11	200	755	488	1443	1241	3	262
SJF	12	0	511	895	1472	1018	2	639
SJF	13	100	762	442	1304	1199	2	130
SJF	14	122	714	806	1642	1180	4	586
SJF	15	100	708	1042	1850	1095	2	874
SJF	16	300	841	755	1896	1283	4	481
SJF	17	300	934	630	1864	1515	5	223
SJF	18	100	661	596	1357	1118	3	353
SJF	19	246	657	200	1013	942	5	73
SJF	20	200	390	0	590	721	4	0
SJF	21	100	801	482	1383	1292	2	228
SJF	22	200	259	378	837	1200	4	149
SJF	23	0	703	485	1188	1132	1	227
SJF	24	300	378	300	978	711	5	154
SJF	25	222	651	700	1573	1135	5	498
SJF	26	300	1078	722	2100	1665	5	331
SJF	27	0	559	300	859	952	1	111
SJF	28	200	752	849	1801	1201	4	601
SJF	29	100	666	487	1253	1059	2	199
SJF	30	200	757	945	1902	1213	3	679
SJF	31	0	685	1030	1715	1074	1	836

Table 4: Round Robin based Prioritized EDF Scheduling for Instance n=32, m=4 and p=8 when β=0.3 and γ=0.7

TYPE	RID	ST	ET	WT	CT	DL	P	DV
RRPEDF	0	0	529	0	529	915	2	0
RRPEDF	1	200	735	1006	1941	1236	3	703
RRPEDF	2	253	624	420	1297	1001	5	215
RRPEDF	3	300	302	297	899	588	5	187
RRPEDF	4	0	515	848	1363	863	1	635
RRPEDF	5	0	347	262	609	662	2	131
RRPEDF	6	22	996	730	1748	1533	2	395
RRPEDF	7	0	620	622	1242	993	2	448
RRPEDF	8	100	620	1042	1762	1037	2	794
RRPEDF	9	200	447	74	721	773	4	12
RRPEDF	10	100	680	673	1453	1069	3	463
RRPEDF	11	200	755	974	1929	1241	3	748
RRPEDF	12	0	577	300	877	1018	2	44
RRPEDF	13	100	762	442	1304	1199	2	130
RRPEDF	14	122	714	806	1642	1180	4	586
RRPEDF	15	100	708	1088	1896	1095	2	920
RRPEDF	16	300	841	875	2016	1283	4	601
RRPEDF	22	200	259	187	946	1200	4	258
RRPEDF	17	300	934	519	1753	1515	5	112
RRPEDF	18	100	661	300	1061	1118	3	57
RRPEDF	19	246	567	200	1013	942	5	73
RRPEDF	20	200	390	0	590	721	4	0
RRPEDF	21	100	801	482	1383	1292	2	228
RRPEDF	23	0	703	485	1188	1132	1	227
RRPEDF	24	300	378	424	1102	711	5	278
RRPEDF	25	222	651	700	1573	1135	5	498
RRPEDF	26	300	1078	510	1888	1665	5	119
RRPEDF	27	0	559	300	859	952	1	111
RRPEDF	28	200	752	585	1537	1201	4	337

IFERP connecting engineers... developing research

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 10, November 2017

RRPEDF	29	100	666	568	1334	1059	2	280
RRPEDF	30	200	757	609	1566	1213	3	343
RRPEDF	31	0	685	1076	1761	1074	1	882

The following are the Gantt chart representation for a single sequence set of job request for Round Robin Based Prioritized EDF Scheduling algorithm

RRPEDF: Sequence1 : R4, R8, R1, R16 **Timeslice: 100**

0	100	200	300	400	500	600	700	800	900	996	1034	1134	1182	1282	1366
VM of Type-1	R4 384 (284)	R8 296 (196)	R1 238 (138)	R16 484 (384)	R4 248 (148)	R8 196 (96)	R1 138 (38)	R16 384 (284)	R4 148 (48)	R8 96 (0)	R1 38 (0)	R16 284 (184)	R4 48 (0)	R16 184 (84)	R16 84 (0)

RRPEDF: Sequence1 : R8, R1, R4, R16 **Timeslice: 100**

996	1096	1196	1274	1374	1474	1539	1540	1640	1708
VM of Type-2	R8 201 (101)	R1 368 (268)	R4 78 (0)	R8 101 (1)	R1 268 (168)	R16 65 (0)	R8 1 (0)	R1 168 (68)	R1 68 (0)

RRPEDF: Sequence1 : R4, R16, R8, R1 **Timeslice: 100**

1274	1288	1539	1639	1712	1812	1912	1924	2014
VM of Type-3	R4 14 (0)	IDEAL 251	R16 290 (190)	R8 73 (0)	R1 112 (12)	R16 190 (90)	R1 12 (0)	R16 90 (0)

RRPEDF: Sequence1 : R4, R8, R1, R16 **Timeslice: 100**

1288	1363	1712	1762	1924	1941	2016
VM of Type-4	R4 75 (0)	IDEAL 349	R8 50 (0)	IDEAL 162	R1 17 (0)	R16 2 (0)

The First Come First Serve (FCFS) Scheduling Metrics for the given problem instance are shown in Table 2. Similarly Table 3 shows the scheduling metrics for Shortest Job First (SJF) Scheduling. Priority based EDF Scheduling metrics are shown in Table 4 where β=0.2 and γ=0.8 in the selection order of jobs.

Table 6: Comparison of Scheduling Metrics when n=32, m=4 and p=8

S_TYPE	Average Turn Around Time (ATR)	Average Waiting Time (AWT)	Average Deadline Violation (ADVT)
FCFS	1394.0	581.59	360.18
SJF	1385.93	578.03	357.18
Round Robin Priority based EDF (β=0.3 and γ=0.7)	1362.15	554.25	333.40

In the table 6, the new model Round Robin Based Prioritized EDF is compared by taking the results obtained from the measured values, with the two scheduling models FCFS and SJF on the three performance metrics Average Turn Around Time (ATR), Average Waiting Time(AWT) and Deadline. All the three performance metrics are calculated on 32 jobs, 4 Virtual machines and 8 instances, and it is observed that the obtained values of ATR, AWT and Deadline of Priority based EDF is more optimized than the FCFS and SJF Scheduling algorithms.

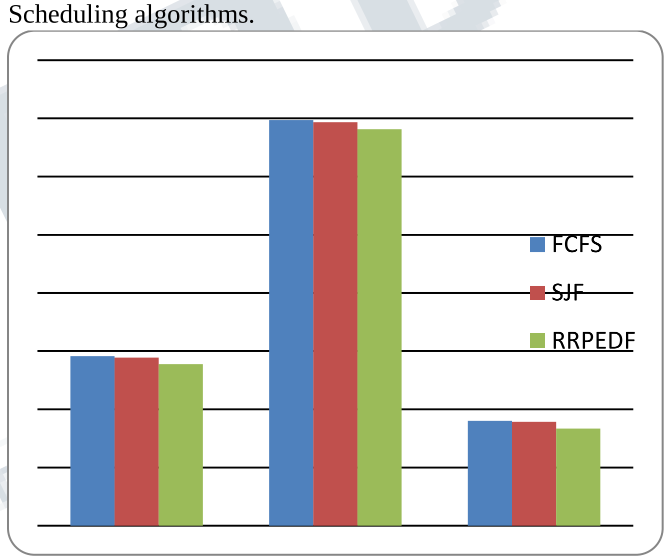


Figure 2: Comparison of Scheduling Metrics when n=32, m=4 and p=8

The graphical representation of performance metrics ATR, AWT and ADVT of FCFS, SJF and Round Robin Based Prioritized EDF Scheduling models are shown in Figure 2. The three performance metrics in Round Robin Based Prioritized EDF is optimal when compared with other scheduling models.

CONCLUSION

Round Robin Based Prioritized EDF Scheduling model finds an optimal scheduling sequence for resource allocation in cloud computing by considering priorities, deadlines and processing times of the jobs with suitable mix using βand γ. The experiment results shown that

Priority based EDF Scheduling Model have better performance metrics of Average Turn Around Time, Average WaitingTime(AWT) and Average Deadline Violation when compare to scheduling models like FCFS and SJF Scheduling models.

REFERENCES

[1] Barrie Sosinsky, "Defining Cloud Computing" in Cloud Computing Bible, Indianapolis, Indiana: Wiley Publishing, Inc., 2011

[2] Li, Ji, LonghuaFeng, and Shenglong Fang. "An greedy-based job scheduling algorithm in cloud computing", Journal of Software Vol. 9, Issue 4, 2014, pp: 921-925

[3] Gupta, Gaurav, et al. "A simulation of priority based earliest deadline first scheduling for cloud computing system", Proceedings of the IEEE First International Conference on Networks & Soft Computing (ICNSC), 2014..

[4] Jagbeer Singh, BichitranandaPatra, Satyendra Prasad Singh, "An Algorithm to Reduce the Time Complexity of Earliest Deadline First Scheduling Algorithm in Real-Time System", Proceedings of the International Journal of Advanced Computer Science and Applications (IJACSA), 2011.

[5] Xiao, Jing, and Zhiyuan Wang. "A PriorityBased Scheduling Strategy for Virtual Machine Allocations in Cloud Computing Environment." Proceedings of the IEEE International Conference on Cloud and Service Computing (CSC), 2012.

[6] Selvarani, S., and G. SudhaSadhasivam. "Improved Cost-Based Algorithm For Task Scheduling In Cloud Computing.", Proceedings of the IEEE International Conference on Computational Intelligence And Computing Research (ICCIC), 2010.

[7] Ghanbari, Shamsollah, and Mohamed Othman. "A priority based job scheduling algorithm in cloud computing." Journal of Procedia Engineering, Vol. 50, 2012, pp: 778-785.

[8] ArnavWadhonkar and DeeptiTheng (2016), " A Task Scheduling Algorithm Based on Task Length and Deadline in Cloud Computing ", International Journal of Scientific & Engineering Research, Vol. 7, Issue 4, pp: 1905-1909.

[9] NeelimaPriyanka N, Suresh Varma P, R KrishnamRajuInsukuri, "Priority Based Earliest Deadline Scheduling In Cloud Computing", International Journal of Computer Engineering and Application, Vol XI, Issue VIII, Aug 17.

