VOL. 12, NO. 12, JUNE 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

CRASA: CLOUD RESOURCE AWARE SCHEDULING ALGORITHM A HYBRID TASK SCHEDULING ALGORITHM USING RESOURCE AWARENESS

P. Akhilandeswari, Kona Kruthi Nymisha, Jahnavi Gandavaram and H. Srimathi
Computer Science and Engineering, SRM University, Chennai, Tamil Nadu, India
E-Mail: akilandeswari.p@ktr.srmuniv.ac.in

ABSTRACT

Cloud computing provides numerous resources to end users. Users can access these resources through internet. Load balancing is very important technique. Max-Min and Min-Min algorithms are proposed for load balancing in cloud computing. These algorithms are implemented based on study of Cloud Resource Aware Scheduling Algorithm (CRASA). Advantages of both the algorithms are used by CRASA to overcome their drawbacks. The drawbacks of these algorithms are in Min-Min the larger tasks have to starve for VM and in Max-Min smaller tasks have to starve for VM allocation. CRASA is achieved by estimating the execution time and finish time of all the tasks on each resource that is available and then alternatively applies Max-Min and Min-Min algorithms. In this paper, the discussion is about CRASA on scheduling tasks that are independent for demonstrating the applicability of CRASA so as to achieve schedules that have lower makespan.

Keywords: cloud computing, task scheduling, load balancing, CRASA, min-min, max-min, makespan.

INTRODUCTION

Cloud Computing is a trending technology which has many end users to which unlimited resources and services will be provided. The required resources that are pooled for computing can be accessed from any browser. It is a computing technology that uses pay per use methodology. In real time there are several examples like Amazon, Microsoft and Google as service providers [2]. Virtualisation is the key component in cloud computing which provides access to the resources which are geographical and remote. Cloud comprises of many virtual machines (VM). So according to the requirement and preference of the Cloud service provider and the user there will be a need of number of virtual machines in cloud. The most exigent task in cloud is resource allocation and task scheduling. For this there is a need for scheduling algorithm so that within the given specifications we can schedule the jobs [1]. There are various stages in cloud computing scheduling and these stages of the cloud scheduling in virtual platform are discussed in Figure-1. In today's world there are several traditional task scheduling algorithms available in cloud [3]. Few traditional task scheduling algorithms are First Come First Serve (FCFS), Round Robin, Shortest job first (SJF), Priority Scheduling, Honey Bee Scheduling, Ant Colony Optimization Scheduling, Min-Min, Max-Min, etc.

Many of the task scheduling algorithms will focus on reducing the total finishing time of the task. But it does not mean that the algorithm ends up with minimizing the execution time of the task. In case of Max-Min and Min-Min algorithms it is possible to estimate the execution and finishing period of all the tasks [4].

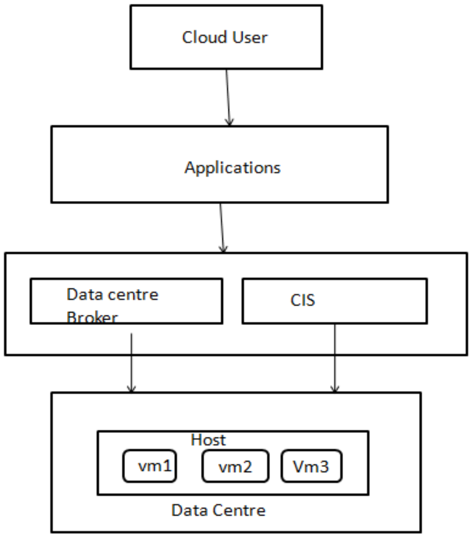


Figure-1. Stages of scheduling in cloud.

Task with minimum finish time will be chosen by Min-Min algorithm when the execution time of the task on different resources is estimated. With this the task can be assigned to the resource which will end up with minimum execution time. The same algorithm is applied to other tasks also. The main problem with Min-Min algorithm is that it gives its preference to the smaller tasks where vast tasks must wait until the smaller tasks are executed. And the same goes with the Max-Min algorithm also where Max-Min gives its preference to the vast tasks first. Now smaller tasks must wait until the vast tasks are executed. So to overcome this problem Cloud Resource Aware Scheduling Algorithm (CRASA) is used where Min-Min algorithm and Max-Min algorithm will be executed alternatively so that the tasks will be assigned to suitable resource [4]. In this paper Cloud Resource Aware

Scheduling Algorithm (CRASA) is implemented in CloudSim environment. CloudSim is a framework for modelling and simulation of cloud computing infrastructures and services. CloudSim functionalities include: (i) support for simulation and modelling of large scale computing data centres. (ii) it is a platform that is complete, independent and separate. CloudSim contributes in faster growth and progress by hastening advancements in methods and algorithms [6].

SCHEDULING ALGORITHMS

Min-min algorithm

The main idea behind the Min-Min algorithm is to end the task in minimum possible time by assigning or mapping the task to appropriate resource. So that we will be able to estimate the execution time and finish time for each task on each resource. There are two parts in this Min-Min algorithm. In the first part the minimum execution time of all the tasks will be calculated individually. In the second part, amid all the tasks the task that has minimum execution time will be selected. The same procedure will be repeated till all the tasks are scheduled [3]. Finish time is also considered which the time is taken to completely allocate the resources to the task after finishing the previously assigned task.

Consider resources R_j (j= *1*,…..,m) that has to be processed by tasks T_i (i= *1*,…..,n). Schedule for individual task will be an allotment of one or more time interims to one or more resources. E_{ij} is the estimated execution time for task T_i on virtual machine (VM) V_j , this is time taken to execute T_i by V_j . b_i be the start of execution of task T_i which is the start time of the task . Expected finish time of a task is calculated by F_{ij} which is a summation of start time and execution time that is estimated of task T_i.

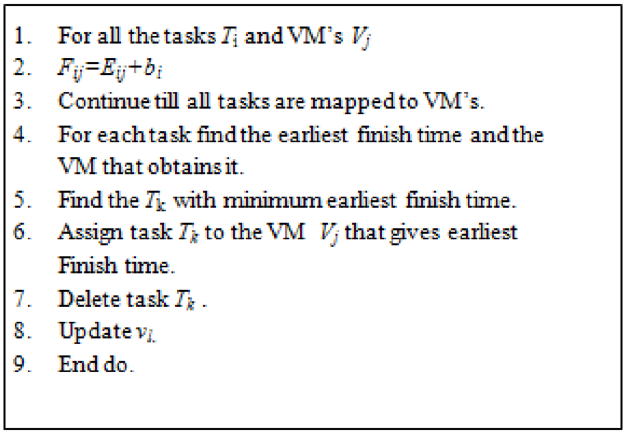
$F_{ij} = b_i + E_{ij}$

v_j is used to represent the expected time at which the VM V_j is in ready state to execute further task assigned to it.
T_k is the task which holds the value of minimum earliest execution time which is further allocated to the VM [5].
Figure-2. shows the above stated Min-Min algorithm.

This algorithm has complexity of time $O(mn^2)$,
Where, n = *no. of tasks*
m= *no. of resources*

Here, all the tasks T_i are sent into vm and now the estimated finish time is calculated by the formulae stated above. Now, the task with earliest finish time with its resource is sent into tasks T_k and now VM is allocated to the task T_k and update the v_1 for all i and perform the scheduling.

Figure-3 is used to depict the sequential execution in Min-Min algorithm. The algorithm is used to compute each resource's allocation to a task finish time with estimated execution time [6].



```
1. For all the tasks $T_i$ and VM's $V_j$
2. $F_{ij}=E_{ij}+b_i$
3. Continue till all tasks are mapped to VM's.
4. For each task find the earliest finish time and the
   VM that obtains it.
5. Find the $T_k$ with minimum earliest finish time.
6. Assign task $T_k$ to the VM $V_j$ that gives earliest
   Finish time.
7. Delete task $T_k$ .
8. Update $v_1$
9. End do.
```

Figure-2. Min-min algorithm.

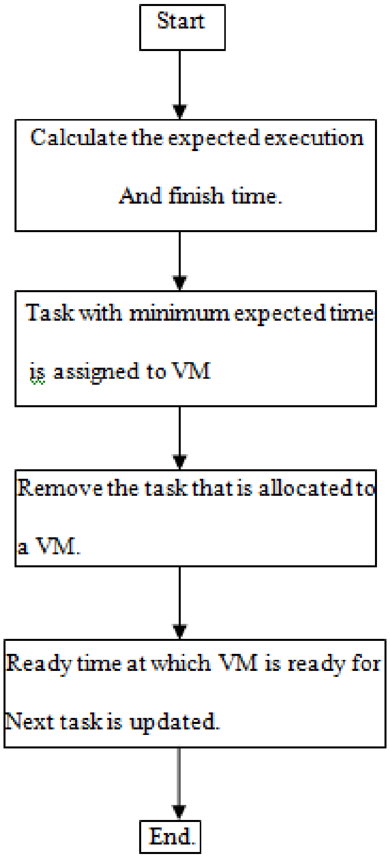

Start
Calculate the expected execution And finish time.
Task with minimum expected time is assigned to VM
Remove the task that is allocated to a VM.
Ready time at which VM is ready for Next task is updated.
End.

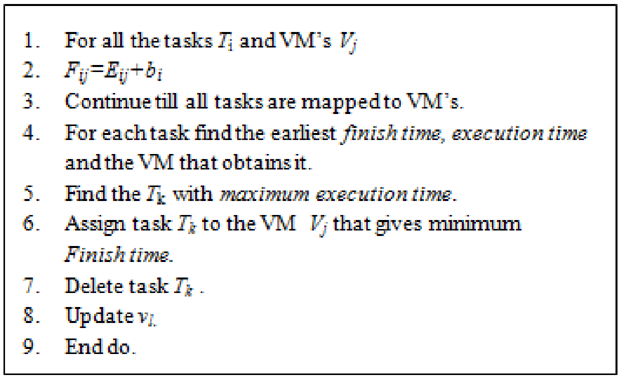
Figure- 3. Sequential execution of min-min.

Max-min algorithm

The Max-Min algorithm is similar to the scheduling procedure of Min-Min algorithm. Here the task T_i having the maximum execution time (largest task) is allotted to the virtual machine VM V_j with minimum finish time [5].

Figure-4 illustrates the Max-Min algorithm where all the tasks T_i for all resources V_j estimated finish time F_{ij} is calculated. Now the task T_k with maximum execution time is found and assigned to the resource with minimum completion time. This algorithm have the time complexity similar to Min-Min algorithm which is denoted by O (mn^2).

Figure-5 is used to depict the sequential execution of Max-Min algorithm. [5].



```
1. For all the tasks $T_i$ and VM's $V_j$
2. $F_{ij}=E_{ij}+b_i$
3. Continue till all tasks are mapped to VM's.
4. For each task find the earliest finish time, execution time
   and the VM that obtains it.
5. Find the $T_k$ with maximum execution time.
6. Assign task $T_k$ to the VM $V_j$ that gives minimum
   Finish time.
7. Delete task $T_k$ .
8. Update $v_l$.
9. End do.
```

Figure-4. Max-min algorithm.

Gittins index

The concept of Gittins Index is used to measure the reward that can be achieved by processes being handled randomly where a process might be in a state of termination and being evolved from its present state where it is a true scalar value affiliated to the state of stochastic process with a reward function and possibility of termination. Here there are two scheduling strategies used in Cloud Resource Aware Scheduling Algorithm (CRASA) which are Min-Min and Max-Min. Among the two strategies it is it is difficult to analyse a better strategy in the long run. By collating the development of each strategy at a set time interim a judgement can be made. Here in this method it firsts takes the scheduling problem and reduces it to the machine which has to perform the jobs within a set of period. A reward value is given to the machine based on the finishing of task within the time period and probability for which completion and non-completion of job is calculated. It has a decision maker where it maximises the total reward by distributing a limited amount of effort to a number of completing projects. Here we calculate average completion time and deduct average completion time from individual task completion time.

temp = finishTime[minCloudlet][minVm] – avg finishTime;

temp2 = temp * temp;
temp3 = temp2/reqTasks;
sd = Math.sqrt(temp3);

By doing the above mathematical calculation the probability of each task based is found based on which a suitable algorithm is chosen.

Cloud resource aware scheduling algorithm (CRASA)

It consists of two techniques called Min-Min and Max-Min that makes it a hybrid algorithm. These two techniques or methodologies will be used alternatively for the task which performs successive execution of smaller and vast tasks on different resources as a result.. So basically in calculation of Min-Min we gloss over the waiting time of large tasks and in

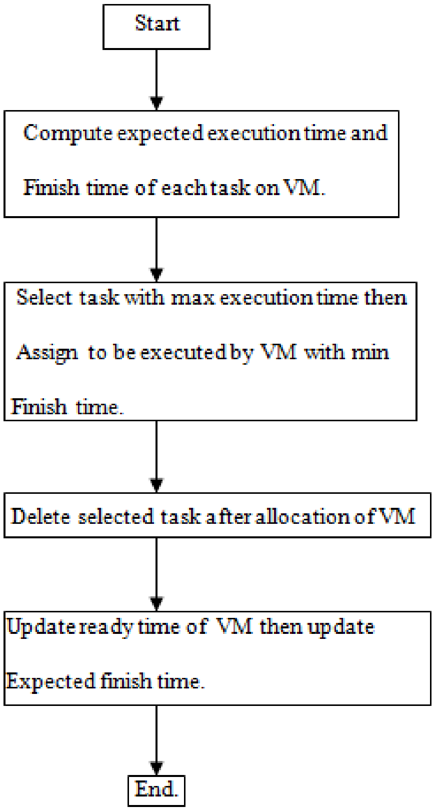
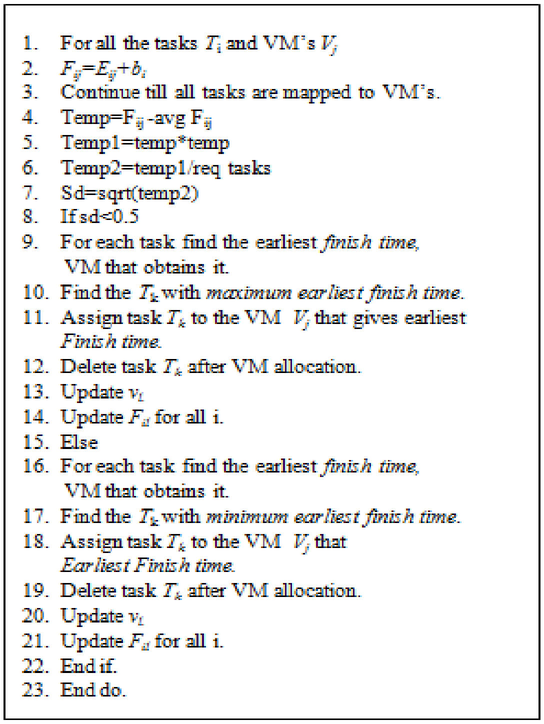

Start
Compute expected execution time and
Finish time of each task on VM.
Select task with max execution time then
Assign to be executed by VM with min
Finish time.
Delete selected task after allocation of VM
Update ready time of VM then update
Expected finish time.
End.

Figure-5. Sequential representation of max-min algorithm.

Calculation of Max-Min we gloss over the waiting time of smaller tasks [3].

Figure 6 represents our proposed algorithm Cloud Resource Aware Scheduling Algorithm (CRASA).

T_i is the task on *virtual machine* VM V_j where the finish time of task T_i is represented by C_{ij}. If the probability sd found using Gittins index is less than 0.5 them Min-Min algorithm is chosen and if probability sd is greater than 0.5 then Max-Min algorithm is chosen. This phenomenon keeps repeating till the tasks are scheduled. From a research it is preferable to apply Min-Min strategy when resources are odd for the first step. The time complexity of Cloud Resource Aware Scheduling Algorithm (CRASA) is denoted by $O(mn^2)$ where this factor is similar to Max-Min and Min-Min [5].Fig 7 gives a sequential representation of CRASA.



```
For all the tasks $T_i$ and VM's $V_j$
$F_{ij}=E_{ij}+b_i$
Continue till all tasks are mapped to VM's.
Temp=$F_{ij}$ -avg $F_{ij}$
Temp1=temp*temp
Temp2=temp1/req tasks
Sd=sqrt(temp2)
If sd<0.5
For each task find the earliest *finish time*,
 VM that obtains it.
Find the $T_k$ with *maximum earliest finish time*.
Assign task $T_k$ to the VM $V_j$ that gives earliest
*Finish time*.
Delete task $T_k$ after VM allocation.
Update $v_L$
Update $F_{il}$ for all i.
Else
For each task find the earliest *finish time*,
 VM that obtains it.
Find the $T_k$ with *minimum earliest finish time*.
Assign task $T_k$ to the VM $V_j$ that
*Earliest Finish time*.
Delete task $T_k$ after VM allocation.
Update $v_L$
Update $F_{il}$ for all i.
End if.
End do.
```

Figure-6. Resource aware scheduling algorithm (RASA).

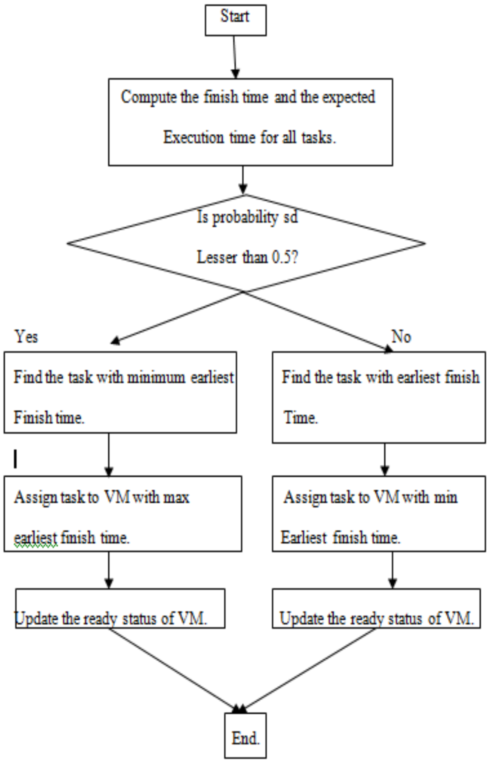

Start
Compute the finish time and the expected Execution time for all tasks.
Is probability sd Lesser than 0.5?
Yes
No
Find the task with minimum earliest Finish time.
Find the task with earliest finish Time.
Assign task to VM with max earliest finish time.
Assign task to VM with min Earliest finish time.
Update the ready status of VM.
Update the ready status of VM.
End.

Figure-7. Sequential representation of CRASA.

SIMULATION AND IMPLEMENTATION

To examine and compare Cloud Resource Aware Scheduling Algorithm (CRASA) with other existing algorithms like

Min-Min and Max-Min, a simulation tool kit known as CloudSim simulator version 3.0 is used to demonstrate the comparisons [4].

Now let us consider an illustration where certain tasks are considered which are to be allocated to certain virtual machines.

Virtual machine characteristics:

	VM
Mips	*200*
Band Width	*1000*
RAM	*512mb*

Task length of each task considered for all three algorithms based on which execution time for each task in a given algorithm is calculated by CloudSim are 9000, 8000, 7800, 2000, 3000, 1800, 2500, 3500, 2800, 3200.

For the above stated task lengths the execution time and finish time of the tasks are tabulated below.

Task configuration min-min

Tasks	Execution time	Start time	Finish time
T1	77.19	0.1	77.29
T2	35.6	0.1	35.7
T3	72.9	0.1	73
T4	8	0.1	8.1
T5	12	0.1	12.1
T6	10	0.1	10.1
T7	29.9	0.1	29.9
T8	14	0.1	14.1
T9	11.2	0.1	11.3
T10	35.6	0.1	35.7

Task configuration of max-min

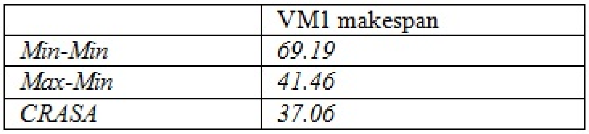
Tasks	Execution time	Start time	Finish time
T1	49.46	0.1	49.56
T2	32	0.1	32.1
T3	31.2	0.1	31.3
T4	8	0.1	8.1
T5	25.46	0.1	25.56
T6	16.79	0.1	16.89
T7	22.39	0.1	22.49
T8	27.46	0.1	17.56
T9	24.39	0.1	24.49
T10	26.26	0.1	26.36

Task configuration of CRASA

Tasks	Execution time	Start time	Finish time	Algo
T1	61.06	0.1	61.16	*Max-min*
T2	57.06	0.1	57.16	*Max-min*
T3	56.26	0.1	56.36	*Max-min*
T4	26.4	0.1	26.5	*Max-min*
T5	36.13	0.1	36.23	*Max-min*
T6	24	0.1	24.1	*Min-min*
T7	31.73	0.1	31.83	*Min-min*
T8	39.06	0.1	39.16	*Max-min*
T9	34.53	0.1	34.63	*Max-min*
T10	37.46	0.1	37.56	*Max-min*

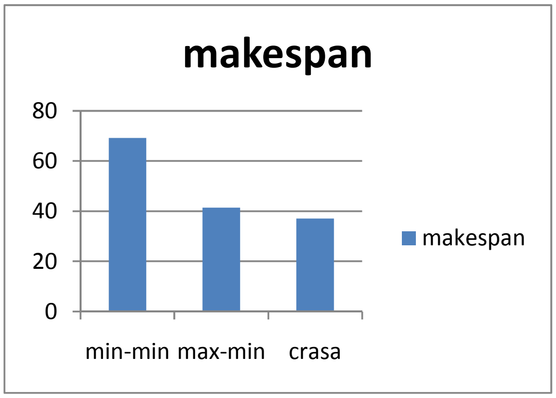
Now for all the above values Min-Min, Max-Min and CRASA scheduling is performed and total makespan is calculated for the stated data.

Makespan:



	VM1 makespan
Min-Min	*69.19*
Max-Min	*41.46*
CRASA	*37.06*

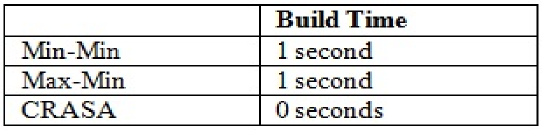
A graph is plotted for the above makespan



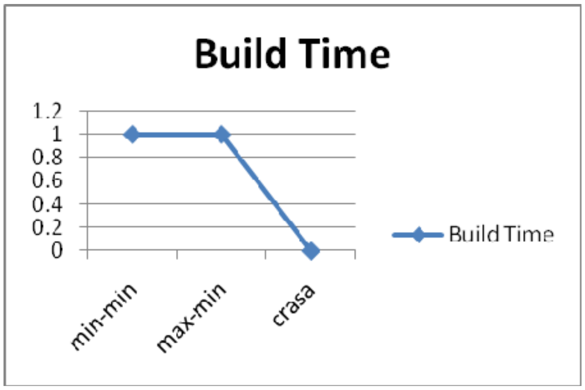
Here X-axis is used to denote the type of algorithm and Y-axis is used to plot the time.

From the above graph, among all the algorithm CRASA stands more efficient than the other two algorithm.

Build Time of all three algorithms is stated below:



	Build Time
Min-Min	1 second
Max-Min	1 second
CRASA	0 seconds



CONCLUSIONS

In this paper, a task scheduling algorithm called Cloud Resource Aware Scheduling Algorithm (CRASA) is proposed. Traditional algorithms like Max-Min and Min-Min are used in

CRASA. The advantages of Max-Min and Min-Min algorithms are used to overcome the disadvantages. By implementing CRASA using CloudSim environment the result shows that CRASA will gloss over existing task scheduling algorithms in large scale. We will examine advantages and disadvantages of Max-Min and Min-Min and relate the study with odd number or even number of resources. The empirical assessment can be open complication in this area when we apply this algorithm in actual cloud environment.

REFERENCES

[1] Kavyasri M N, Dr.B.Ramesh, “Comparative Study of Scheduling Algorithms to Enhance the Performance of Virtual Machines in Cloud Computing”.

[2] Arnav Wadhonkar, Depti Theng, “A Survey on Different Scheduling Algorithms in Cloud Computing”.

[3] Abhishek Gupta, H S Bhadauria, Annapurna Singh, Jagadish Chandra Patni, “A Theoretical Comparison Of Job Scheduling Algorithms In Cloud Computing”.

[4] Saeed Parsa, Reza Entezari-Maleki, “RASA: A New Task Scheduling Algorithm in Grid Environment”.

[5] O.M.Elzeki, M.Z. Reshad, M. A. Elsoud, “Improved Max-MinAlgorithm in Cloud Computing”.

[6] R.Jemina Mina Priyadarsini, L. Arockiam, “Performance Evaluation of Min-Min and Max-Min Algorithms for Job Scheduling in Federated Cloud”.