Budget Based Scheduling

Cloud computing allows better utilization of hardware and software resources at lower price than traditional IT infrastructure. Cloud provider account the usage and charge user per use. Sometimes user is not sure how much resources a process might require and thus assigns abundance of resources, but they also have restrictions on amount of money spend. In Budget Based Scheduling resources are utilized with budget awareness, when the bill amount exceeds a certain limit then the resources are released. Tasks can be scheduled on various criteria like, maximum number of task must be completed or high priority task must be completed etc.

To accomplish this task user inputs list of task and budget, the cloud provider allocate some resources to analyze and predict the hardware utilization of those task, schedule it accordingly. For example, task can be scheduled on the basis of deadline, priority or minimum cost. Although execution time, Input/Output burst, resource utilization etc. of the task can’t be accurately determined beforehand, many methods are evolved to achieve this For example, Average and Exponential Average of all previous programs. Cloud Provider can use a self-evolving AI powered evaluation technique which works on the basis of past experiences from previous task.

This scheduling can be used with other scheduling algorithms, like first come first serve, priority algorithm etc. Although task are scheduled, but resources usage and cost of usage is continuously monitored and calculated. This requires cloud provider to allocate more resources for logging purposes. Cloud consumer gets comfort of automatic de-allocation of resources as budget exceeds, rather than manual checking expenditure.

ALGORITHM

Input (TaskList, Budget)

Variable: CurrentCost=0

SchedulingAlgorithm

For every task in the TaskList (From i=0 TO lengthOf(TaskList))

CalculateCostOfTask[i]

End For Loop

Arrange Task in ascending order of Cost

For i=0 TO lengthOf(TaskList)

If(currentCost+CostOfTask[i]<=budget)

ScheduledTask🡨 Task[i]

currentCost=currentCost+CostOfTask[i]

End For

Return(ListOfScheduledTask)

EXPLAINATION:

User input TaskList and Budget associated with it.

For every task in TaskList estimated cost of the task is calculated, with the help of loop.

Task are scheduled in increasing order of their cost.

ScheduledTaskList is created. In this all the scheduled task are stored.

Task Cost is added to the Current Cost and checked if it exceeds budget or not, if it doesn’t exceed budget then it is added to the ScheduledTaskList and Current Cost is update. Else task is not added and ScheduledTaskList is displayed.