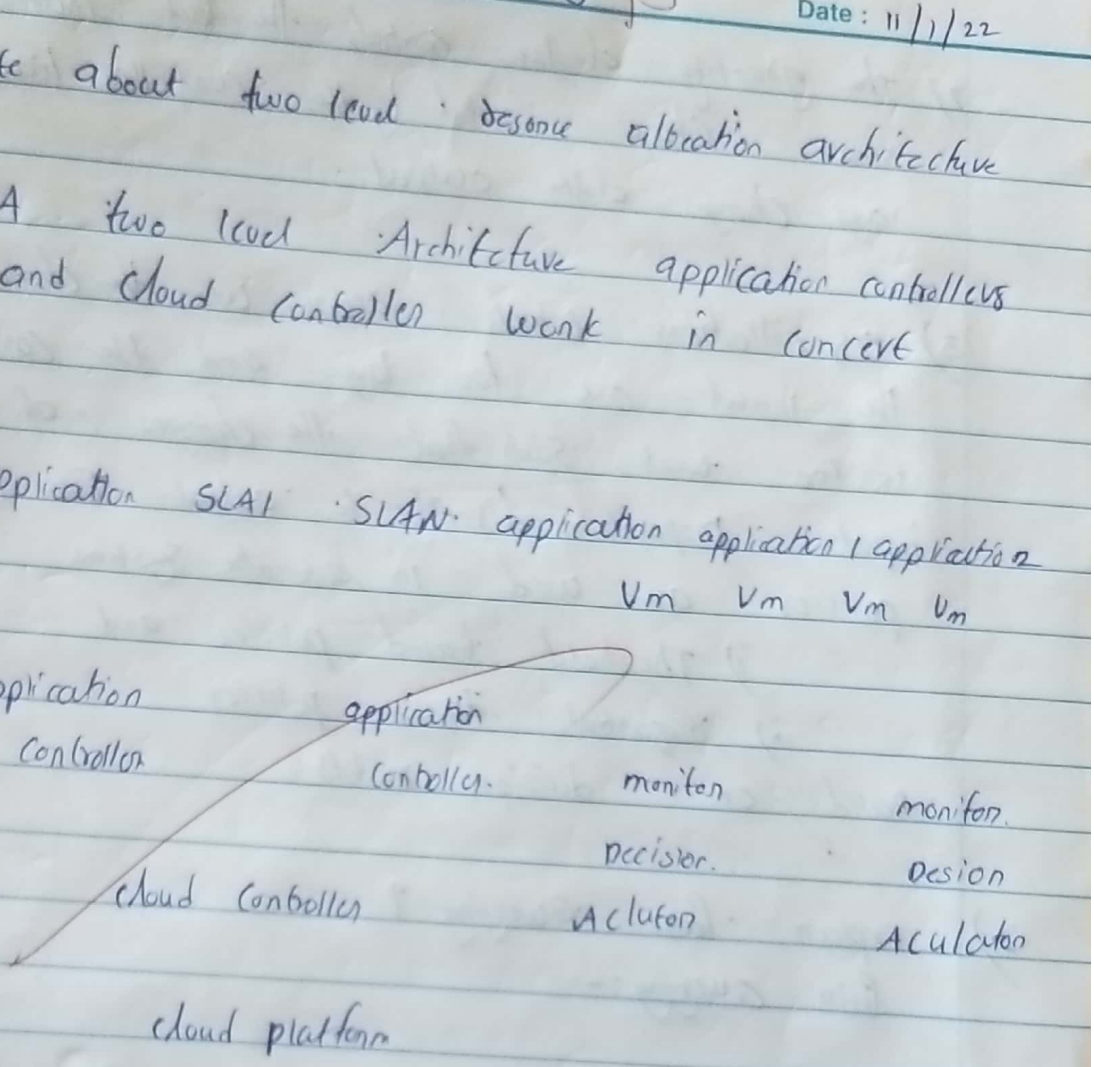
Assignment No. 05 Unit No. 05

G. Venkatesh.
19K05A0516
CSE-C

Date : 11/1/22

1. Write about two level resource allocation architecture

A two level Architecture application controllers and cloud controller work in concert



* The main compents of a contol system are inputs
* The Contol system compents and the outputs
* The inputs in such models are the offered workload and the policies for admission control, the capacity allocation, the load balancing the energy optimization and the QoS goals in the cloud.

* There are three main sources
1) The delay in getting the system reaction after a control action

2) The granularity of the control, the fact that a small change enacted by the controller leads to very large changes of the output.

3) Oscillations which occur when the change of the input are too large and the control is too weak, such that the changes of the input propagate directly to output. Two types of policies are used in automatic systems

i) Threshold based policies and

ii) Sequential decision policies

Based on decision models

2) Explain Fair Queuing ?

Fair queuing :-

Computing and communication on a cloud are intimately related

Therefor it should be no suprise that the first algorithm we discuss can be used for scheduling packet transmission as well as threads

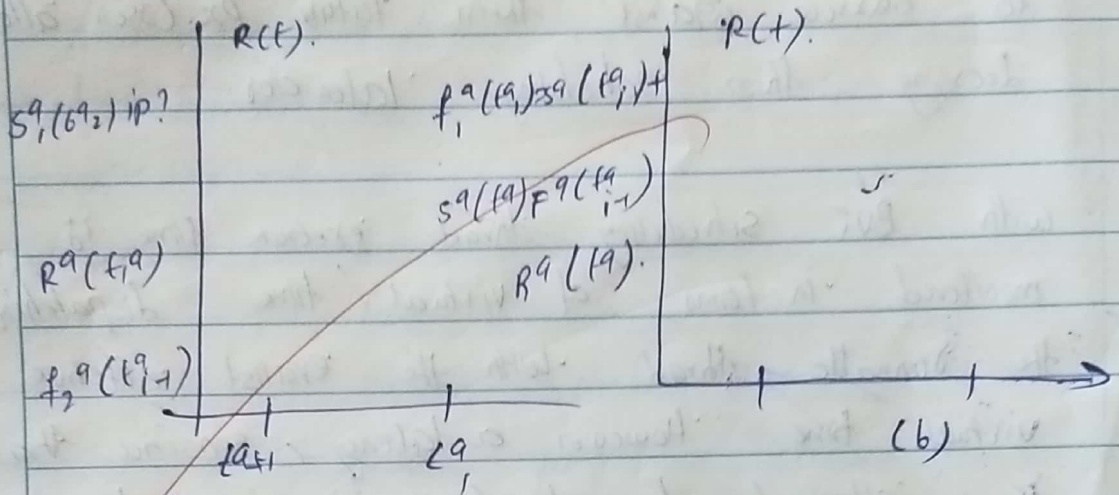
The fair queuing (FQ) algorithm proposs a solution to this problem.

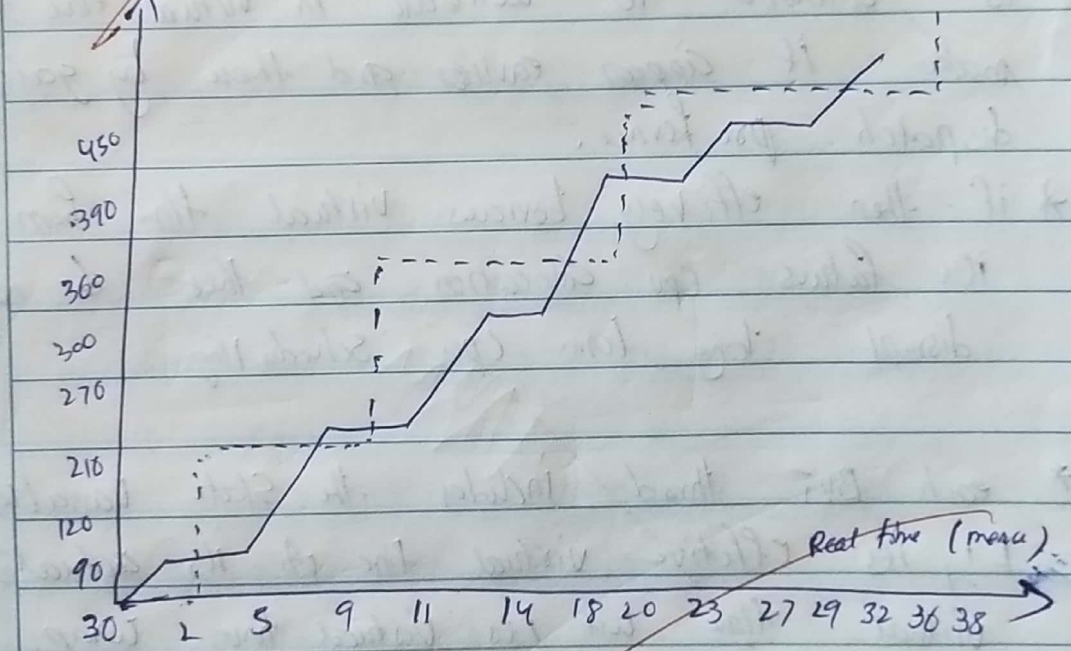
Call t_a, the time when the packet of flow of size p_a bits arrives and call p_a bits arrives. and call of S_f and P_a the value of $R(t)$ when

the first and the last bit respectively of the packet of flow a are transmitted then.

$f_i^q = s_i^q + p_i^q$ and $s_i^q = \max\{F_i^q, R(t), t_a)$

$R(t) \leq p_i^q$ when $i \in \max(V \mid t_i^q \leq t)$





* The effective virtual time and the real time of threads a and b with weights $w_a = 2w_b$ when the actual virtual time is incremented in steps ..
the the two threads are allowed to cpu is proportional to their weights

3) What is Borrowed virtual time explain ?

Borrowed virtual time :-
The Borrowed virtual time scheduler is a proportional share scheduler that allows scheduled threads to borrow against their future processor allotment during their scheduling latencies.

With BVT scheduling thread execution time is monitored in terms of virtual time dispatching the runnable thread with the earliest effective virtual time. However a latency sensitive thread is allowed to warp back in virtual time to make it appear earlier and thereby gain dispatch preference.

* it effectively borrows virtual time from its future CPU allocation and therefore does not disrupt long term CPU scheduling.

* each BVT thread includes the state variables E_i, its effective virtual time A_i its actual virtual time with its virtual time warp, warpBack & if warp is enabled

$$E_i \leftarrow A_i - (warp\ ?\ w_i : 0)$$

where warp is determined as described later the scheduler accounts for running time in units of minimum charging unit.

4. What is map reduce scheduling?

map reduce Scheduling:-

mapreduce application on the cloud subject to deed like several options for scheduling apache Hadoop, an open-source implation of the mapreduce. algorithm are:-

* The default FIFO schedule.
* The fairn schedular.
* The capacity schedulor.
* The dynamic proportional schedulor.

The System is homogenous:-

This means that in and pi the cost of processing a unit data by the map and the reduce task respectively are the same for all servers.

* under the conditions the duration of the job J with input of size a is

$$E(m_m, r_r, s) = s\left(\frac{p_m}{n_m} + \phi\left[\frac{p_r}{n_r} + v\right]\right)$$

Thus the condition that query $Q(A, \sigma, D)$ with arrival time A meets the deadline D can be expressed as

$$\left[m+1\left[\frac{p_m}{n_m} + \phi\left[\frac{p_r}{p_r} + v\right]\right]\right] \leq A + D.$$

$$t_0^{max} = A + n \div \sigma \phi \left(\frac{P_r}{P_r} + V \right)$$

* We now plug the expression of the maximum value for the Strat-up time of the reducer task into the condition to meet the deadline

$$t_{0m} + \sigma \frac{P_m}{n_m} \leq t_{j,max}$$

It follows immediately that n_m^{min}, the minimum no. of slots for map task

$$n_m^{min} > \frac{\sigma P_m}{t_r^{max} f_m}$$

Thus,

$$n_m^{min} = \left[\frac{\sigma P_m}{t_r max \div t_{jm}} \right]$$