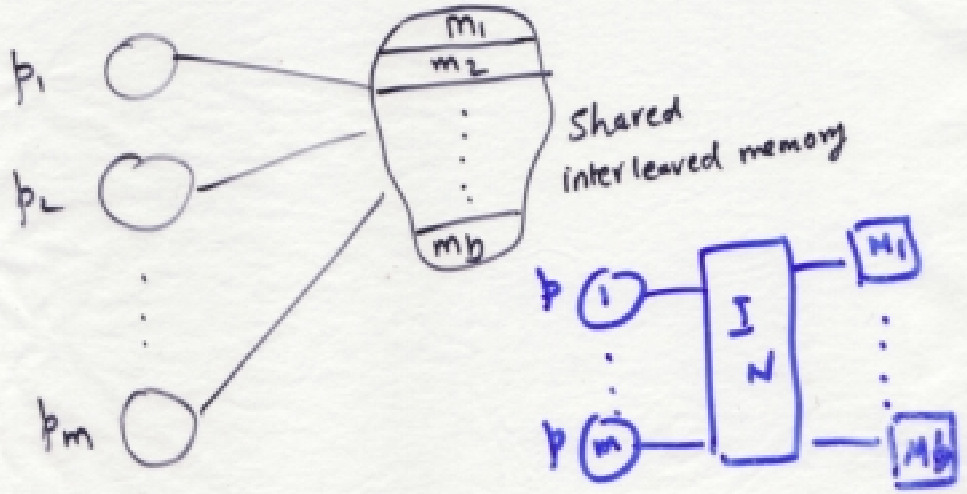
Multiprocessor Performance : Shared Memory Organisation

$\underline{\underline{m}}$ processors $\underline{\underline{b}}$ memory banks

- Processors are statistically independent (⫫)
- Mem. requests are ⫫ and randomly distributed in the range 1,…,b
- Each p_i (processor i) generates one memory reference per cycle with probability $\underline{r}$
- $\underline{\underline{x}}$: probability that a bank is busy
- Memory cycle time is constant
- Bank cycle time is $\underline{\underline{c}}$

⫫



⫫

To keep track of the state of a bank, a reservation of C cycles is placed on a bank when it is accessed.

- In each subsequent cycle the remaining reservation on the bank is decremented by one until it reaches 0, at which time the bank is free.

Assuming that processors are statistically $\perp\!\!\!\perp$ & identical, it is <u>sufficient to model a single processor</u>

Model tracks 2 quantities.

(a) # of reservation cycles remaining on the requested bank if it is busy

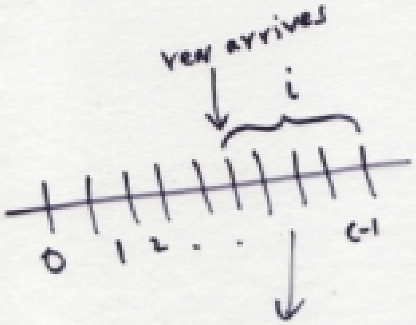
(b) # of pending requests in the queue of a busy bank.

So, each bank has a queue to hold the pending requests.

- We restrict our attention to the case where the max length of the queue attached to a bank is <u>2</u>.

Define a Markov chain as follows.

Let the random variable X_t = 0, if proc. is free
= i, if the proc is blocked on a busy bank that is reserved for i additional cycles



i cycles are ahead for the bank to become free.

Y_t = 0, if the req. bank is free
= j, if the bank is busy & there are j processors blocked on this bank.

Note: The proc that is currently serviced is not blocked & therefore not included in the above count j.

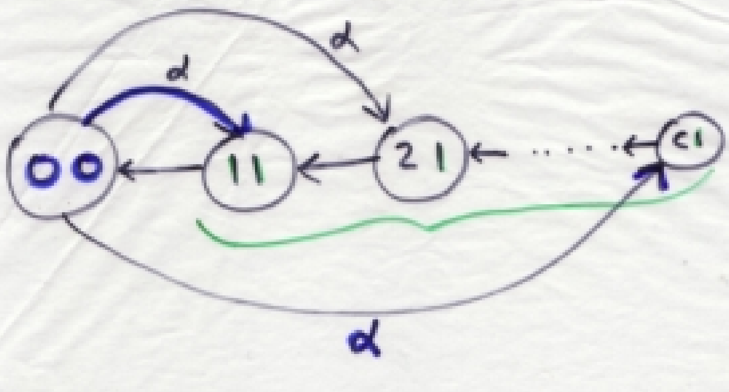
Then, $\{X_t, Y_t\}$ is a Markov chain with the transition probabilities,

$$P(\overset{X}{i}\overset{Y}{j}, \overset{X}{k}\overset{Y}{l}) = \text{Prob}\left[X_{t+1}=k,\ Y_{t+1}=l \,/\, X_t=i\ \ Y_t=j \right]$$

- Probability that a free proc makes a request to a busy bank is $\underline{\underline{rx}}$.

define: $\alpha = \left(\frac{rx}{c}\right)$: prob of a req is uniformly distributed in the range 1, ..., c.

(*) $P(00, i1) = \alpha, \quad \forall\ 1 \le i \le c$



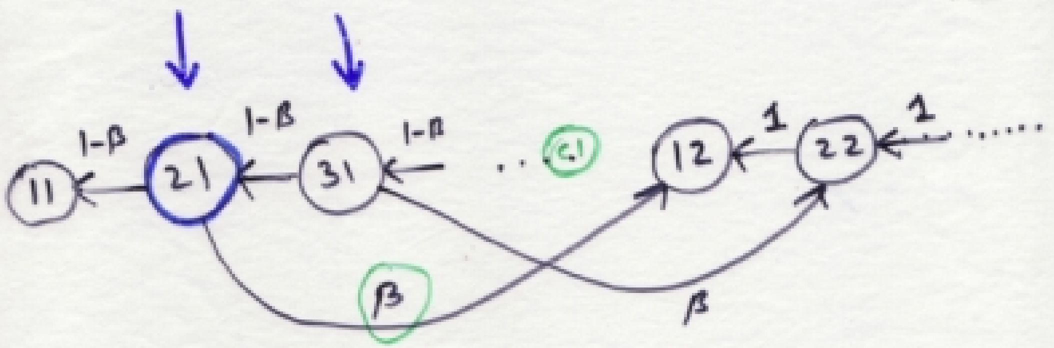
. When the system is in a blocked state $(i,1)$,

(*) a new request may arrive at the same bank.

we assume:

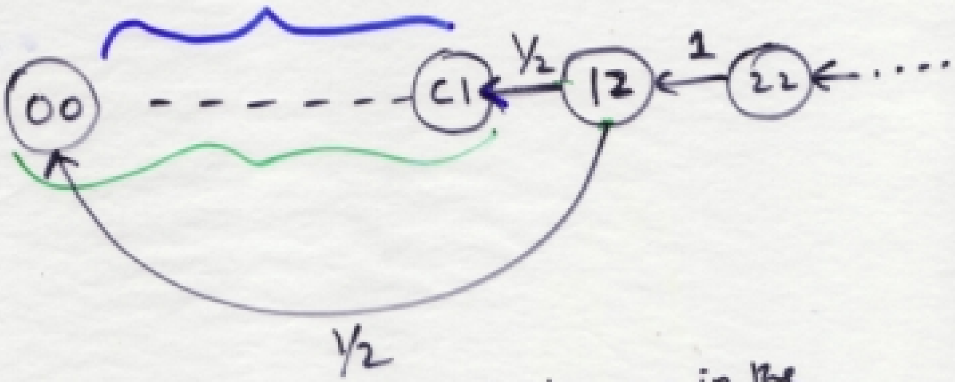
$$P(i1, (i-1)2) = \beta = \left(\frac{mr}{2b}\right),$$

$$2 \le i \le c$$



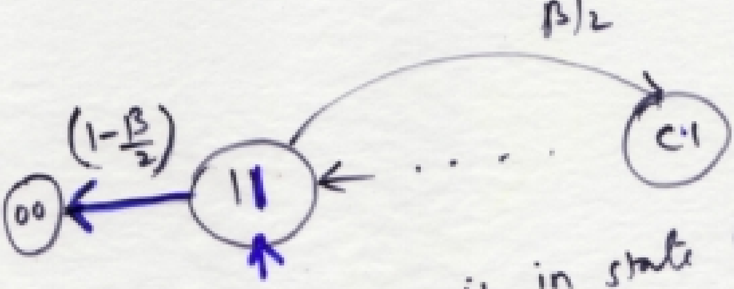
$$\left(\frac{mr}{2}\right)\frac{1}{b} = \beta$$

*

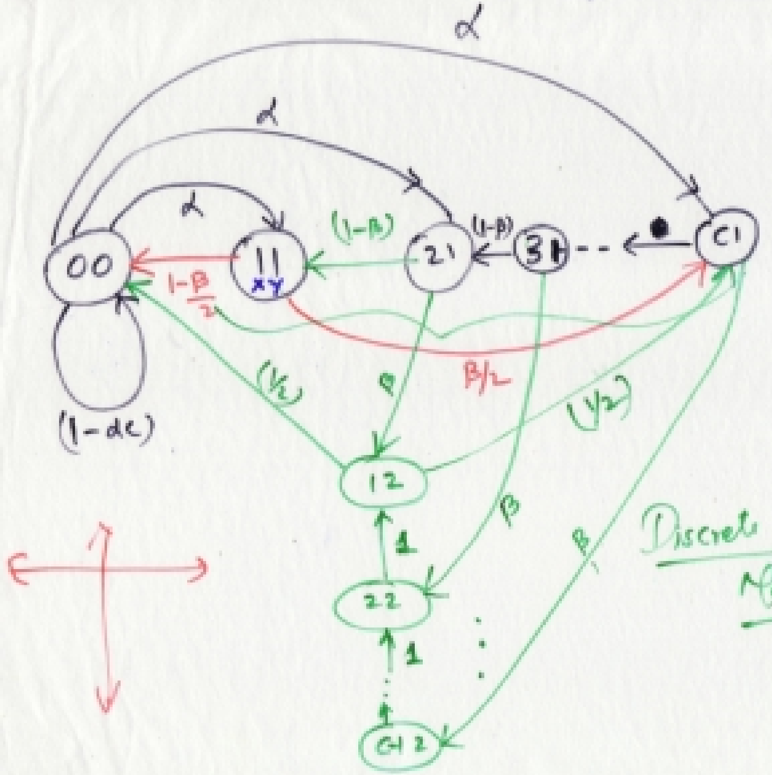


With 2 requests pending in the queue one is selected at random. if the accepted request belongs to the processor being modeled, the transition is to state (00), indicating that the processor is free

*



When the system is in state (11) - just one cycle before the bank is free & a new req. arrives (with prob β) there are again 2 requests & one is accepted at random (with prob 1/2)



Discrete Time Markov Chain

From the above diagram, following set of balance equations are obtained,

$$P(00) = (1-\alpha c)\,P(00) + \left(1-\frac{\beta}{2}\right)P(11) + \frac{1}{2}P(12)$$

$$\underline{P(i1)} = \alpha P(00) + (1-\beta)\,P(i{+}1\;1) \checkmark \qquad 1 \le i \le c-1$$

$$P(c1) = \alpha P(00) + \frac{\beta}{2}P(11) + \underbrace{\frac{1}{2}P(12)}$$

$$P(i2) = \beta P(i{+}1\;1) + P(i{+}1\;2), \qquad 1 \le i \le c-2$$

$$P(c{-}1\;2) = \beta P(c1)$$

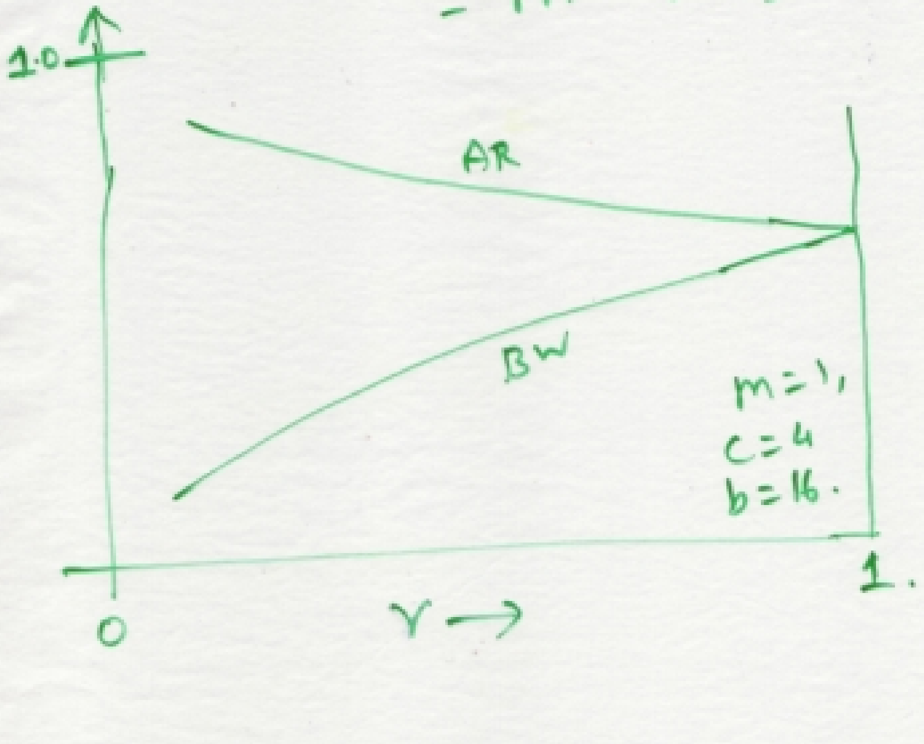
Solve these to determine $P(00)$.

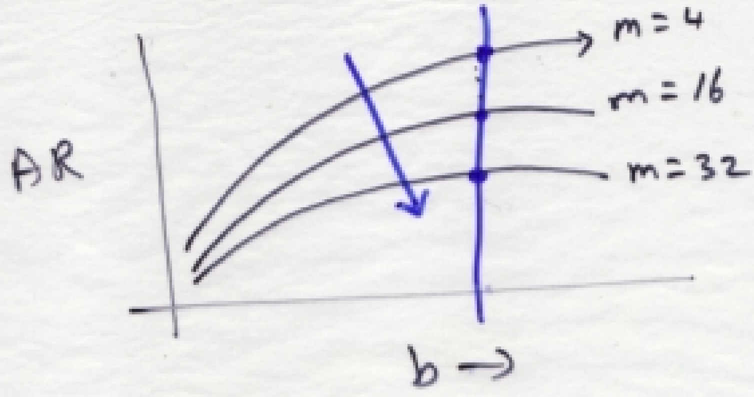
We define:

$$\text{Acceptance ratio (AR)} = \frac{r.\,P(00)}{r P(00) + (1 - P(00))}$$

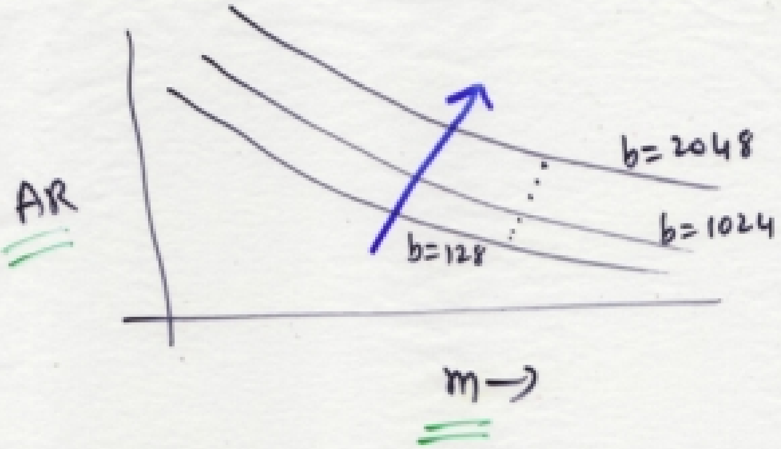
Bandwidth (total # of accepted requests for the m processors)

$$= m.\,r.\,P(00).$$





$r = 0.8$
$c = 16$ ✓



$r = 0.8$
$c = 16$

$P(0,0)$

$\underset{b \to \infty}{\mathrm{Lt}}\ AR = ?$ →