Lecture No 18
Open Closed and Queue Models

Open, Closed and Mixed Queue Models

- Certain systems can behave as open queue up to a certain queue size and then behave as closed queues.
- Such systems are called ***Mixed Queue*** systems

Open Queue (Flores) Memory Model

- Open queue model is not very suitable for processor memory interaction but its most simple model and can be used as initial guess to partition of memory modules.
- This model was originally proposed by flores using M/D/1 queue but M_B/D/1 queue is more appropriate.

Open Queue (Flores) Memory Model

- The total processor request rate λ_s is assumed to split uniformly over m modules.
- So request rate at module $\lambda = \lambda_s / m$
- Since $\mu = 1/T_c$ (T_c is memory cycle time)
- So $\rho = \lambda / \mu = (\lambda_s / m) . T_c$
- We can now use M_B /D/1 model to determine T_w and Q_0 (Per module buffer size)

Open Queue (Flores) Memory Model

- Design Steps:
  - Find peak processor instruction execution rate in MIPS.
  - MIPS * refrences / instruction = MAPS
  - Choose m so that $\rho = 0.5$ and $m=2^k$ (k an integer)
  - Calculate T_w and Q_0.
  - Total memory access time = $T_w + T_a$
  - Average open Q size = $m \,.Q_0$

Open Queue (Flores) Memory Model

- Example:
- Design a memory system for a processor with peak performance of 50 MIPS and one instruction decoded per cycle.

Assume memory module has Ta = 200 ns and Tc = 100 ns. And 1.5 references per instruction.

Open Queue (Flores) Memory Model

- Solution:
- MAPS = 1.5 * 50 = 75 MAPS
- Now $\rho = \lambda s / m * Tc$
- So $\rho = 75 \times 10^{6} \times 1/m \times 0.1 \times 10^{-6} = 7.5 /m$
- Now choose m so that $\rho = 0.5$
- If m =16 then $\rho = 0.47$
- For MB/D/1 model $Tw = 1/\lambda * (\rho^{2} - \rho p)/ 2(1-\rho)$

  $= Tc * (\rho - 1/m)/ 2 (1-\rho)$

  = 38 ns

Open Queue (Flores) Memory Model

- Total memory access time = Ta + Tw = 238 ns
- $Q_0 = \rho^2 – \rho p / 2 (1 – \rho) = 0.18$
- So total mean Q size = m x Q_0 = 16 x .18 =3

Closed Queues

- Closed queue model assumes that arrival rate is immediately affected by service contention.
- Let λ be the offered arrival rate and λa is the achieved arrival rate.
- Let ρ is the occupancy for λ and ρa for λa .
- Now $(\rho - \rho a)$ is the no of items in closed Qc.

Closed Queues

- Suppose we have an n, m system in overall stability.
- Average Q size (including items in service) denoted by N = n/m and

  closed Q size Qc = n/m – ρa = ρ – ρa where ρa is achieved occupancy.

From discussion on open queue we know that

Average Q size $N = Q_0 + \rho$

Closed Queues

- Since in closed Queue Achieved Occupancy is ρa, and for M/D/1, Q_0 is $\rho^2 /2(1- \rho)$, so we have

$N = n/m = \rho a^2 /2(1- \rho a) + \rho a$

Solving for ρa

we have $\rho a = (1+n/m) - \sqrt{(n/m)^2 +1}$

Bandwidth B (m,n) = m. ρa so

$$\mathbf{B (m,n) = m+n - \sqrt{n^2+m^2}}$$

This solution is called the Asymptotic Solution

Closed Queues

- Since $N = n/m$ is the same as open Queue occupancy ρ. We can say

$$\rho a = (1+\rho) - \sqrt{\rho^2 + 1}$$

Simple Binomial Model: While deriving asymptotic solution , we had assumed m and n to be very large and used M/D/1 model.

For small n or m the *binomial* rather than *poisson* is a better characterization of the request distribution .

Binomial Approximation

- Substituting queue size for $M_B/D/1$

$N = n/m = (\rho a^2 - p\rho a) / 2(1- \rho a) + \rho a$

Since Processor makes one request per Tc

$p = 1/m$ (prob of request to one module)

Substituting this and solving for ρa

$$\rho a = 1+n/m - 1/2m - \sqrt{(1+n/m-1/2m)^2 -2n/m)}$$

and $B(m,n) = m \cdot \rho a$

$$B(m,n) = m+n-1/2 - \sqrt{(m+n - 1/2)^2 - 2mn}$$

Binomial Approximation

- Binomial approximation is useful whenever we have
  - **Simple processor memory configuration (a binomial arrival distribution)**
  - **n >= 1 and m >= 1.**
  - **Request response behavior: where processor makes exactly n requests per Tc**

The (δ) Binomial Model

- If simple processor is replaced with a pipelined processor with buffer (I-buffer,register set , cache etc) the simple binomial model may fail.
- Simple binomial model can not distinguish between single simple processor making one request per Tc with probability =1, and two processors each making 0.5 requests per Tc.
- In second case there can be contention and both processors may make request with varying probability.

The (δ) Binomial Model

- To correct this δ binomial model is used.
- Here the probability of a processor access during Tc is not 1 but δ, so p = δ /m
- Substituting this we get a more general definition

$$B(m,n,\delta) = m + n - \delta/2 - \sqrt{(m + n - \delta/2)^2 - 2mn}$$

The (δ) Binomial Model

- This model is useful in many processor designs where the source is buffered or makes requests on a statistical basis
- If *n* is the mean request rate and *z* is the no. of sources, then $\delta = n/z$

The (δ) Binomial Model

- This model can be summarized as follows:
  - Processor makes n requests per Tc.
  - Each processor request source makes a request with probability δ.

Offered bandwidth per Tc $Bw = n/Tc = m\lambda$

Achieved Bandwidth = $B(m,n,\delta)$ per Tc.

Achieved bandwidth per second

$= B(m,n,\delta) / Tc = m\ \lambda a$.

Achieved Performance = $\lambda a / \lambda$ * (offered performance)

Using the δ- Binomial Performance Model

- Assume a processor with cycle time of 40ns. Memory request each cycle are made as per following
  - Prob (IF in any cycle) = 0.6
  - Prob (DF in any cycle) = 0.4
  - Prob (DS in any cycle) = 0.2
  - Execution rate is 1 CPI., Ta = 120ns, Tc =120 ns

Determine Achieved Bandwidth / Achieved Performance (Assuming Four way Interleaving)

Using the δ- Binomial Performance Model

- M=4, Compute n:(Mean no of requests per Tc)

so n = requests/per cycle x cycles per Tc

$= (0.6+0.4+0.2) \times 120/40$

= 3.6 requests / Tc

Compute δ: z = cp x Tc/ processor cycle time

Where cp is no of processor sources.

So $z = 3 \times 120/40 = 9$

So $\delta = n/z = 3.6 / 9 = 0.4$

Using the δ- Binomial Performance Model

Compute B(m,n,δ):

$$B(m,n,\delta) = m + n - \delta/2 - \sqrt{(m + n - \delta/2)^2 - 2mn}$$

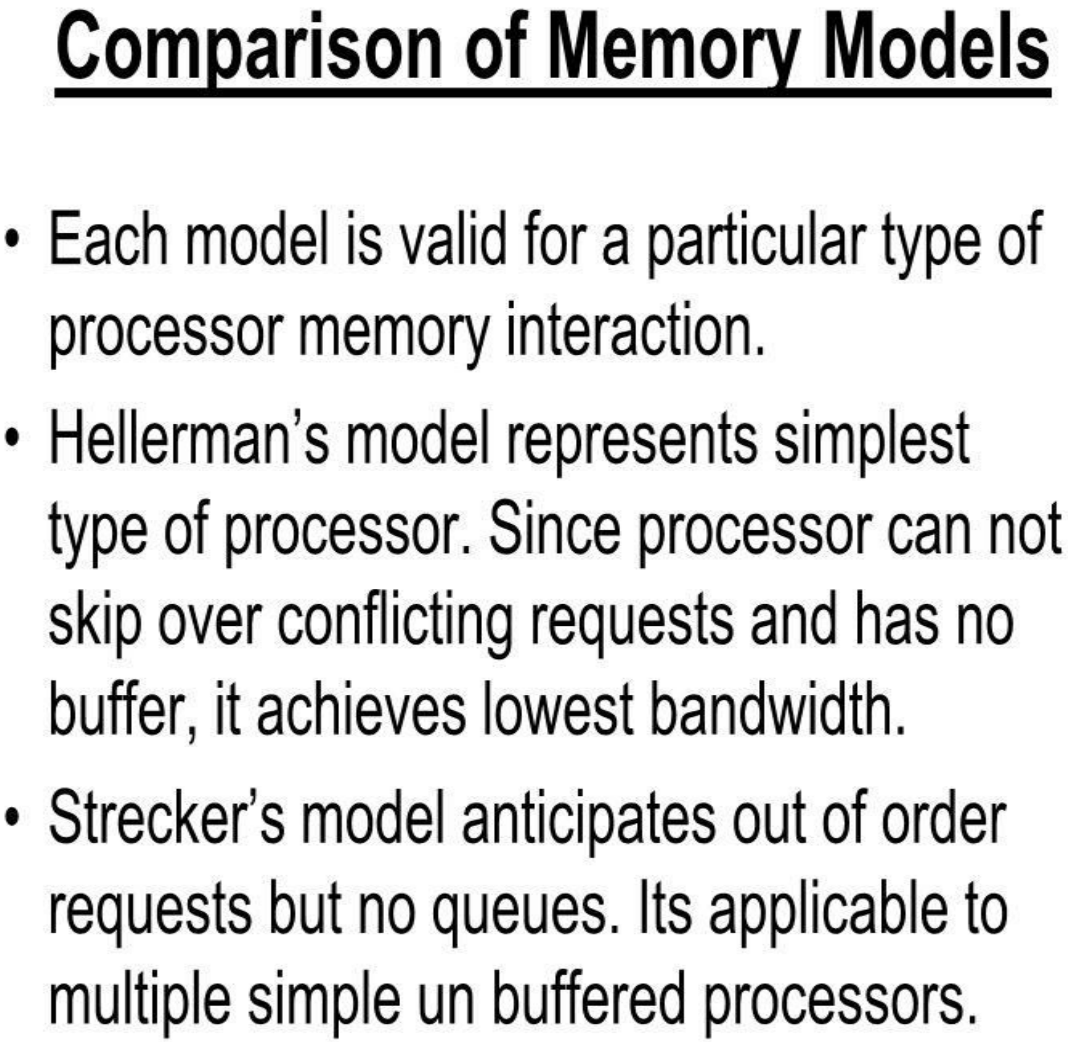
$$= 2.3 \text{ Requests/ Tc}$$

So processor offers 3.6 requests each Tc but memory system can deliver only 2.3. this has direct effect on processor performance.

Performance achieved = 2.3/3.6 (offered Perf.)

At 1cpi at 40 ns cycle offered perf = 25 MIPS.

Achieved Performance = 2.3/3.6 (25) = 16MIPS.



Comparison of Memory Models

- Each model is valid for a particular type of processor memory interaction.
- Hellerman's model represents simplest type of processor. Since processor can not skip over conflicting requests and has no buffer, it achieves lowest bandwidth.
- Strecker's model anticipates out of order requests but no queues. Its applicable to multiple simple un buffered processors.