Lecture No 20

Comparison of Memory Models

Comparison of Memory Models

- The δ binomial model is suitable for simple pipelined processors where n requests per Tc are each made with probability δ.

Review and Selection of Queuing Models

- There are basically three dimensions to simple (single) server queuing models.
- These three represent the statistical characterization of arrival Rate, Service rate and amount of buffering present before system saturates.
- For arrival rate, if the source always requests service during a service interval, Use M_B or simple binomial model.

Review and Selection of Queuing Models

- If the particular requestor has diminishingly small probability of making a request during a particular service interval, use poisson arrival.
- For service rate if service time is fixed , use constant (D) service distribution.
- If service time varies but variance is unknown, (choose $c^2=1$ for ease of analysis) use exponential (M) service distribution.

Review and Selection of Queuing Models

- If variance is known and C^2 can be calculated use M/G/1 model.
- The third parameter determining the simple queuing model is amount of buffering available to the requestor to hold pending requests.

Processors with Cache

- The addition of a cache to a memory system complicates the performance evaluation and design.
- For CBWA caches, the requests to memory consists of *line read* and *line write* requests.
- For WTNWA caches, its *line read* requests and *word write* requests.
- In order to develop models of memory systems with caches two basic parameters must be evaluated

Processors with Cache

1. $T_{\text{line access}}$,time it takes to access a line in memory.
2. T_{busy} , potential contention time (when memory is busy and processor/cache is able to make requests to memory)

Accessing a Line $T_{\text{line access}}$

- Consider a pipelined single processor system using interleaving to support fast line access.
- Assume cache has line size of L physical words(bus word size) and memory uses low order interleaving of degree m.
- Now if m >= L, the total time to move a line (for both read and write operations)

$$T_{\text{line access}} = Ta + (L-1)\, T_{\text{bus}}.$$

Where Ta is word access time & T_{bus} is bus cycle time.

Accessing a Line $T_{line\ access}$

- If L > m, a module has to be accessed more than once so module cycle time Tc plays a role.
- If Tc <= (m . T_{bus}), module first used will recover before it is to be used again so even for L > m

$$T_{line\ access} = Ta + (L-1)T_{bus}$$

- But for L > m and Tc >= (m. T_{bus}), memory cycle time dominates the bus transfer

Accessing a Line $T_{line\ access}$

- The line access time now depends on relationship between Ta and Tc and we can now use.

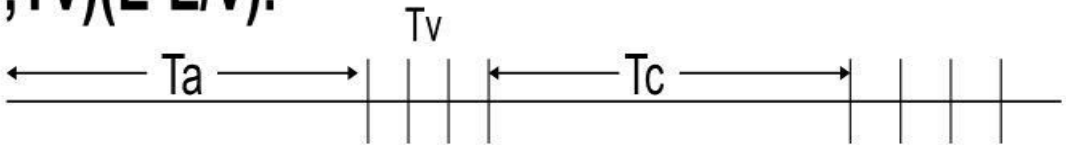
$T_{line\ access} = Ta + Tc \cdot ((L/m) - 1) + T_{bus} \cdot ((L-1) \bmod m)$.

- The first word in the line is available in Ta, but module is not available again until Tc. A total of L/m accesses must be made to first module with first access being accounted for in Ta. So additional (L/m -1) cycles are required.

Accessing a Line $T_{line\ access}$

- Finally ((L-1) mod m) bus cycles are required for other modules to complete the line transfer.
- If we have single module memory system (m=1), with nibble mode or FPM enabled module. Assume v is the no of fast sequential acceses and Tv is the time between each access

$$T_{line\ access} = Ta + Tc\ ((L/v) - 1) + (max\ (T_{bus}, Tv)(L - L/v).$$



Accessing a Line $T_{line\ access}$

- Now consider a mixed case ie m>1 and nibble mode or FPM mode.

T line access = Ta+ Tc((L/m.v)-1)+

Tbus (L-(L/m.v))

Computing $T_{\text{line access}}$

- Case 1: Ta = 300ns, Tc=200ns, m=2, Tbus=50 ns and L=8.

Here we have L>m and Tc > m.T bus

So T line acces = Ta +Tc((L/m) -1)+Tbus ((L-1) mod m).

=300+200(4-1)+50(1) =950ns

Computing $T_{\text{line access}}$

- Case 2: Ta=200ns, Tc=150ns, Tv=40ns,T bus =50 ns, L=8, v=4, m=1.

T line access = Ta + Tc((L/v)-1)+ max(Tbus, Tv)(L-L/v).

=200+ 150((8/4)-1)+ 50(8-(8/4))

=200+ 150 +300

=650 ns

Computing $T_{\text{line access}}$

- Case 3: Ta=200ns, Tc=150ns, Tv=50ns,T bus =25 ns, L=16, v=4, m=2.

T line access = Ta + Tc((L/m.v)-1)+ (Tbus)(L-L/m.v).

=200+ 150((16/2.4)-1)+ 25(16-(16/2.4))

=200+ 150 +350

=700 ns

Contention Time & Copy back Caches

- In a simple copy back cache processor ceases on cache miss and does not resume until dirty line (w =probability of dirty line) is written back to main memory and new line read into the cache.

The Miss time penalty thus is

$$T_{miss} = (1+w)\, T_{line\ access}$$

Contention Time & Copy back Caches

- Miss time may be different for cache and main memory.
  - **Tc.miss = Time processor is idle due to cache miss.**
  - **T m.miss= Total time main memory takes to process a miss.**
  - **T busy = T m.miss – T c.miss : Potential Contention time.**
  - **T busy is =0 for normal CBWA cache**

Contention Time & Copy back Caches

- Consider a case when dirty line is written to a write buffer when new line is read into cache. When processor resumes dirty line is written back to memory from buffer.

T m.miss = (1+w) T line access.

T c.mis = T line access

So T busy = w. T line access.

- In case of wrap around load.

T busy = (1+w) T line access - Ta

Contention Time & Copy back Caches

- If processor creates a miss during T busy we call additional delay as T interference.

T interference = Expected number of misses during T busy.

= No of requests during T busy x prob of miss.

= λp . T busy . F : where λp is processor request rate.

The delay factor given a miss during Tbusy is simply estimated as Tbusy /2

So T interference = λp .T busy. F. Tbusy/2

Contention Time & Copy back Caches

T interference = λp . f . $(Tbusy)^2$ / 2 and total miss time seen from processor

T miss = T c.miss + T interference. And Relative processor performance

$Perf_{rel}$ = 1/ 1+f λp T miss