Allocation of Cache Memory between Data & Instruction sets

• Model

- Interlaced I & D streams that exhibit different cache behaviors

Q: how to allocate cache in such cases?

Soln: An optimal allocation occurs at a point where the miss-rate derivatives of the competing processes are equal.

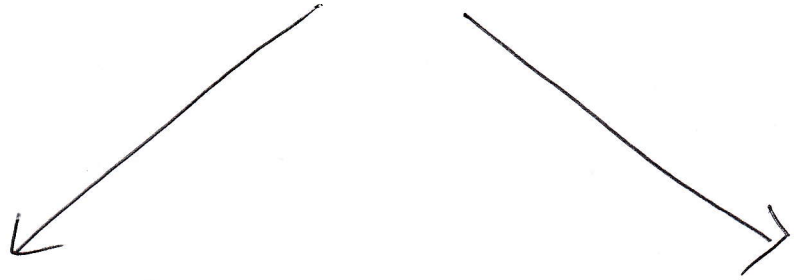
Practical Considerations

- Fully associative search is not used (speed consideration);
- use set-associative + LRU (overhead in realizing LRU is tolerable if # of items is relatively small)

(2)

Assumptions:

(A) Competing processes can be characterized as having a miss-rate as a function of allocation size



fully-associative

(ext. mem + cache interactions)

- Miss-rate for a given reference stream (I or D) as a function of allocation is indeed one parameter function & depends only on the # of blocks/lines allocated to that stream. (Why?)

(Because entire cache is seached in this implementation)

Set-associative

- Here, the miss-rate depends not only on # of blocks/lines allocated but where they are in the cache!!

(Since only a set of blocks are searched)

- Simplified model of set-associative cache — miss-rate is simply a one parameter function (blocks / lines allocated) regardless of the physical distributions of the blocks per set.

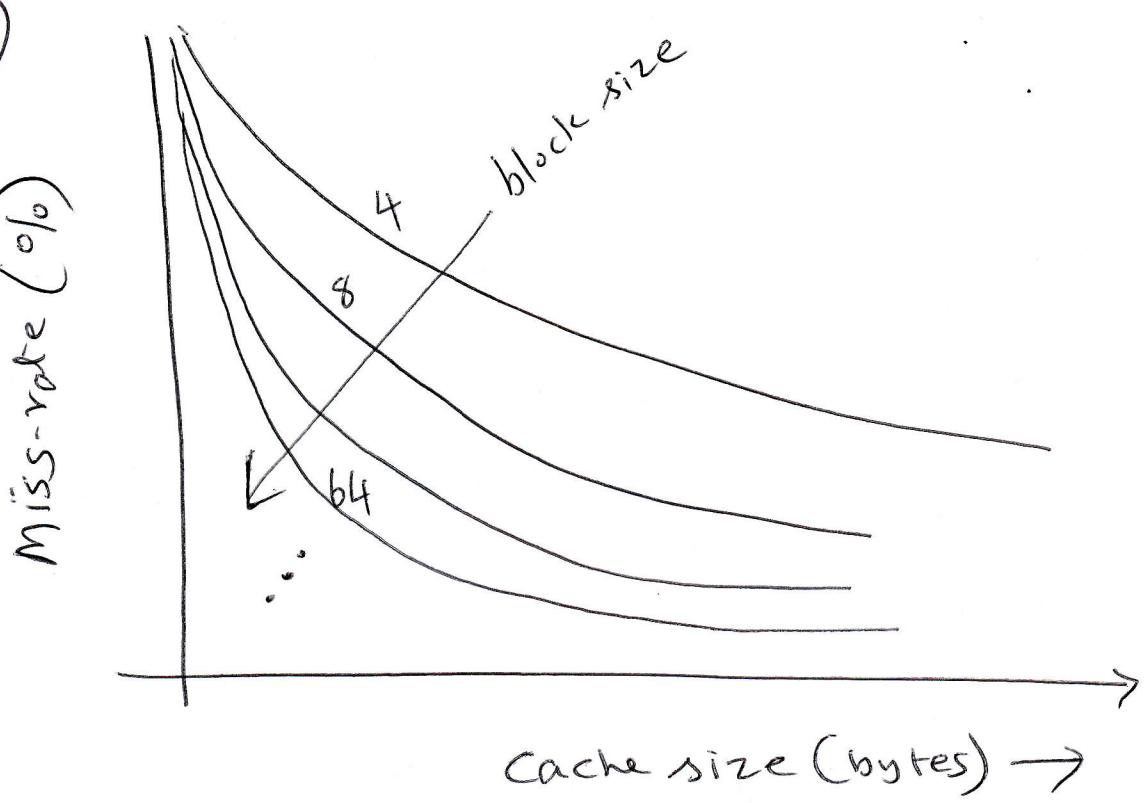
- Cache accessing processes — I & D streams (interlaced). ⇒ address references alternate between data & instructions; [I, I, D, I, D, D, I, D, ...]

- Each component stream has a known cache behavior given by a miss-rate function for that stream as a function of cache mem. allocated to that stream;

$$M_I(x), \quad M_D(x)$$

↳ stationary in time is a key assumption here ⇒ non-time varying functions;

(4)



To determine optimal (fixed) allocation for (I) & (D), we find an expression for the misses in a period of time that has exactly T references, & find an allocation at which the derivative of miss-rate function goes to zero.

$\Rightarrow$ $M_I(x)$ & $M_D(x)$ — continuous & differentiable (assumption)

In practice, we know that these are measureable only at discrete points.

Let us suppose we must allocate C bytes of cache memory (CM) between D & I references so that

$$\left.\begin{aligned} I &\leftarrow x \text{ bytes} \\ D &\leftarrow (C-x) \text{ bytes} \end{aligned}\right\}$$

- Total # of misses in a time period with T references is the composite miss-rate times the length of the period.

Assuming I & D references occur with equal frequency in the interval T, the total # of misses is given by,

$$\text{Total Misses} = \left[M_I(x) + M_D(C-x)\right] T/2$$

Obviously, to minimize the total # of misses, we set the derivative = 0 which occurs at a value of x^* that satisfies

$$\frac{dM_I(x^*)}{dx} = -\frac{dM_D(C-x^*)}{dx}$$

(6) Trace simulations & real-life benchmarking performances have shown that for a reasonable sizes of memory allocation, the miss-rate function exhibits convexity.

Thus, by convexity of the functions, such a point x^* is indeed a minimum.

Suppose, say if the D stream occurs τ times as frequently in the composite reference. Then for this case, at the minimum miss-rate the derivative of $M_D(x)$ is weighted by a factor of τ.

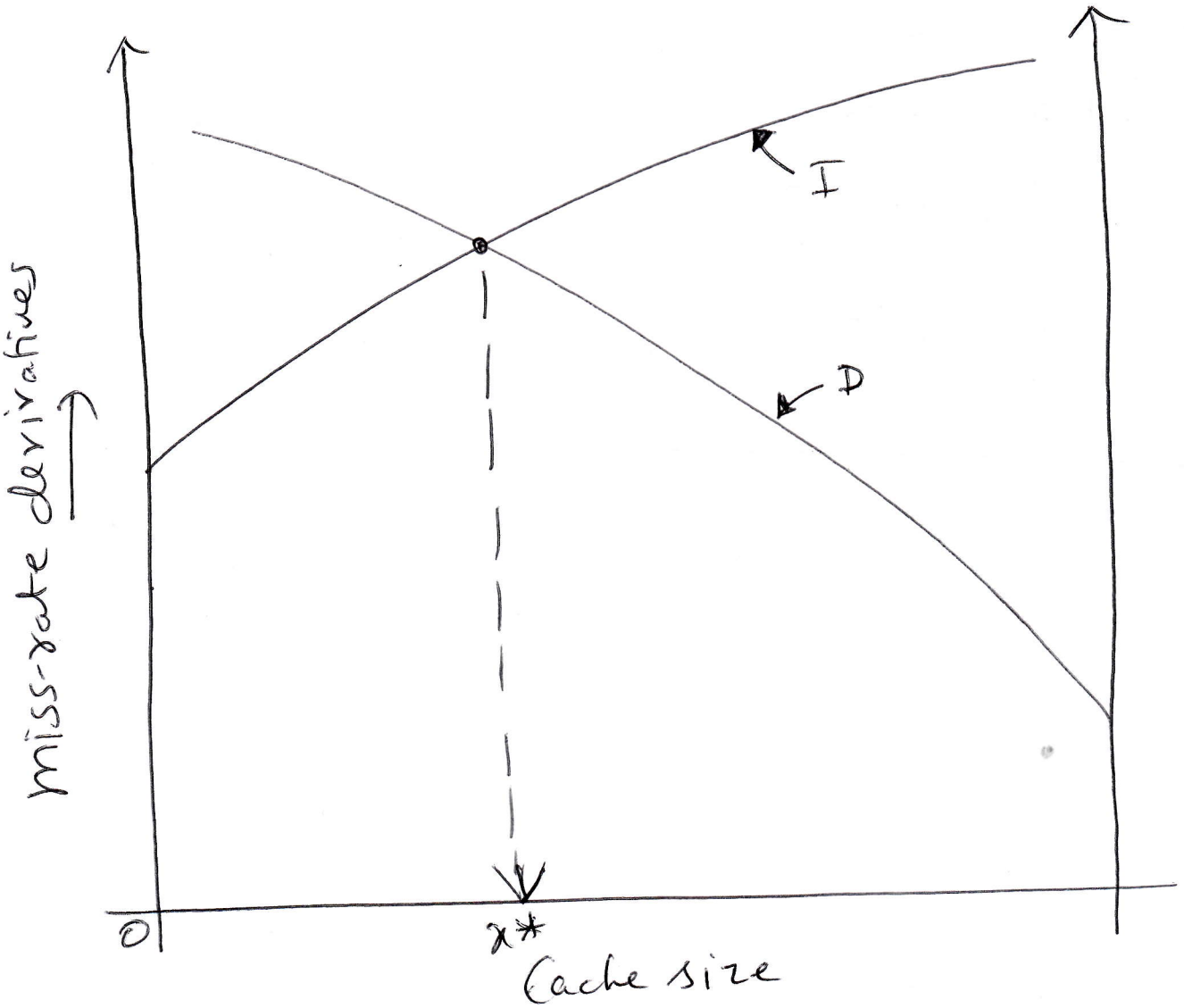
Using an emperical study & curve fitting,

$$\left.\begin{aligned} M_D(x) &= a\,x^{-b} \\ M_I(x) &= c\,x^{-d} \end{aligned}\right\} \quad a, b, d, c \in \mathbb{R}^+$$

$$M_I(x) = 1.311\, x^{-0.3815}$$

$$M_D(x) = 3.606\, x^{-0.4729}$$

fitted with empirical & trace data available for the respective functions;



Thus, once we have the above functions, we

(8)

Can take derivatives, which yields,

$$\frac{dM_I(x)}{dx} = 0.500\, x^{1.38151}$$

$$\frac{dM_D(x)}{dx} = -1.70\, x^{-1.47249}$$

} plot in the previous page.

---

(9) Systems with split caches for I & D

- advantages ?
- disadvantages ?
