

Homework 1

Abhishek Johri



February 10, 2017

CS 484

University of Illinois at Urbana-Champaign

1. a. (L+1) \* N < C < 2 \* N2

For this scenario A will miss 1x and then hit (L-1) times each iteration. This happens before becoming part of the LRU line and getting kicked out of the cache. For B it will miss L number of times consecutively, but due to the access pattern it will only be the LRU line after L-1 access. This will add up to a total of 2N/L of misses

b. This cache will hold N/2 integers from A and then L\*N / 2 integers will be from B

A will miss N/L times and B will always miss N number of times in the inner loop iteration. This will end up having a total number of N2/L + N2 misses.

1. for(int i = 0; i < N; i++){

for(int j = 0; j < N; j += L){ //L = linesize

A[ i ][ j ] += B[ i ][ j ] + C[ i ][ j ]

A[ i ][ j+1 ] += B[ i ][ j+1 ] + C[ i ][ j+1 ]

…

A[ i ][ j + L - 1 ] += B[ i ][ j + L - 1 ] + C[ i ][ j + L - 1 ]

}

}

1. Answers below
   1. Offset: 6, Index: 13, Physical Tag: 13, Virtual Tag: 21
   2. Offset: 6, Index: 14, Physical Tag: 12, Virtual Tag: 20
   3. Offset: 6, Index: 0, Physical Tag: 26, Virtual Tag: 34
   4. 240 / 232 = 28 = 256
   5. The function of a TLB is to keep a cache of a program. What this does is that it stores recently visited locations in memory in a structure. So what the program will do is check to see if the location was looked at before and grab the information from there rather than having to traverse a page table, this ends up being very time consuming.
2. Code Balance: 1 store + 2 reads / 2 flops = 3/2

Peak Performance: Bm/Bc = 0.3 / 1.5 = 0.2

1. A[0] = 0

A[1] = 1 / c

A[2] = 4

A[3] = 3