CS 610 Semester 2022–2023-I: Assignment 1

8th August 2020

Due Your assignment is due by Aug 20, 2022, 11:59 PM IST.

General Policies

- You should do this assignment ALONE.
- Do not copy or turn in solutions from other sources. You will be PENALIZED if caught.

Submission

- Submission will be through mooKIT.
- Submit a PDF file with name "<roll-no>.pdf". We encourage you to use the LaTeX typesetting system for generating the PDF file.
- You will get up to TWO LATE days to submit your assignment, with a 25% penalty for each day.

Evaluation

- Show your computations where feasible and justify briefly.
- Write your answers that the EXACT output format (if any) is respected.

Problem 1 **[20 marks]**

Consider the following loop.

```
1  double s = 0.0, A[size];
2  int i, it, stride;
3  for (it = 0; it < 100 * stride; it++) {
4    for (i = 0; i < size; i += stride) {
5      s += A[i];
6    }
7  }
```

Assume an 8-way set-associative cache with a capacity of 256 KB, line size of 64 B, and word size of 8 B (for `double`). Furthermore, assume the cache is empty before execution and uses an LRU replacement policy. Given `size`=32K, determine the total number of cache misses on `A` for the following access strides: 1, 4, 16, 32, 2K, 8K, and 32K. Consider all the three kinds of misses: cold, capacity, and conflict.

Problem 2

[80 marks]

Consider a cache of size 32K words and lines of size 8 words. The array dimensions are 512×512. Perform cache miss analysis for the *ikj* and the *jik* forms of matrix multiplication (shown below) considering direct-mapped and 4-way set-associative caches. The arrays are stored in row-major order. To simplify the analysis, ignore misses from cross-interference between elements of different arrays (i.e., perform the analysis for each array, ignoring accesses to the other arrays).

Listing 1: ikj form

```
for (i = 0; i < N; i++)
  for (k = 0; k < N; k++)
    for (j = 0; j < N; j++)
      C[i][j] += A[i][k] * B[k][j];
```

Listing 2: jik form

```
for (j = 0; j < N; j++)
  for (i = 0; i < N; i++)
    for (k = 0; k < N; k++)
      C[i][j] += A[i][k] * B[k][j];
```

Your solution should have a table to summarize the total cache miss analysis for each loop nest variant and cache configuration, so there will be four tables in all. Justify your computations.

Problem 3

[30 marks]

Consider the following code.

```
double y[4096], X[4096][4096], A[4096][4096];
for (k = 0; k < 4096; k++)
  for (j = 0; j < 4096; j++)
    for (i = 0; i < 4096; i++)
      y[i] = y[i] + A[i][j] * X[k][j];
```

Assume a direct-mapped cache of capacity 16 MB, with 64 B cache lines and a word of 8 B. Assume that there is negligible interference between the arrays `A`, `X`, and `y` (i.e., each array has its 16 MB cache for this question), and arrays are laid out in the row-major form.

Estimate the total number of cache misses for `A` and `X`.