ETH login ID:

(Please print in capital letters)

Full name: ____________________

263-2300: How to Write Fast Numerical Code

ETH Computer Science, Spring 2015

Midterm Exam

Wednesday, April 15, 2015

Instructions

- Make sure that your exam is not missing any sheets, then write your full name and login ID on the front.
- No extra sheets are allowed.
- The exam has a maximum score of 100 points.
- No books, notes, calculators, laptops, cell phones, or other electronic devices are allowed.

Problem 1 (16 = 1+1+5+4+5)	
Problem 2 (18 = 3+15)	
Problem 3 (18 = 2+6+10)	
Problem 4 (12 = 2 + 2 + 2 + 2 + 2 + 2)	
Problem 5 (12)	
Problem 6 (12 = 6 + 6)	
Problem 7 (12 = 2 + 5 + 5)	
Total (100)	

Problem 1 $(16 = 1 + 1 + 5 + 4 + 5)$

We consider a 128 byte data cache that is 2-way associative and can hold 4 doubles in every cache line. A double is assumed to require 8 bytes.

For the below `C` code we assume a cold cache. Further, we consider an array `A` of 32 doubles that is cache aligned (that is, `A[0]` is loaded into the first slot of a cache line in the first set). All other variables are held in registers. The code is parameterized by positive integers `m` and `n` that satisfy `m * n = 32` (i.e., if you know one you know the other).

```
int i, j;
double A[32], t = 0;
for(i = 0; i < m; i++)
  for(j = 0; j < n; j++)
    t += A[j * m + i];
```

Answer the following:

1. How many doubles can the cache hold?

2. How many sets does the cache have?

3. For `m = 1`:

   (a) Determine the miss rate.

   (b) What kind of misses occur?

   (c) What kind of locality does the code have with respect to accesses of `A` and this cache?

4. For `m = 2`:

   (a) Determine the miss rate.

   (b) What kind of misses occur?

5. For `m = 16`:

   (a) Determine the miss rate.

   (b) What kind of misses occur?

   (c) What kind of locality does the code have with respect to accesses of `A` and this cache?

Problem 2 ($18 = 3 + 15$ points)

Consider the following code, which computes the Cholesky decomposition of a hermitian positive definite matrix A ($N \times N$).

```
void cholesky(float **A, float **L, int N){
  int i,j,k;
  float temp;

  for(j = 0; j < N; j++){
    temp = A[j][j];
    for(k = 0; k < j; k++){
      temp = temp - L[j][k]*L[j][k];
    }
    L[j][j] = sqrt(temp);
    for(i = j+1; i < N; i++){
      temp = A[i][j];
      for(k = 0; k < j; k++){
        temp = temp - L[i][k]*L[j][k];
      }
      L[i][j] = temp/L[j][j];
    }
  }
}
```

1. Define a detailed floating point cost measure $C(N)$ for the function `cholesky`. Ignore integer operations.

2. Compute the cost $C(N)$ as just defined.

Notes: Lower-order terms (and only those) may be expressed using big-O notation. This means: as the final result something like $3n + O(\log(n))$ would be ok but $O(n)$ is not.

The following formulas may be helpful:

- $\sum_{i=0}^{n-1} i = \frac{n(n-1)}{2} = \frac{n^2}{2} + O(n)$
- $\sum_{i=0}^{n-1} i^2 = \frac{(n-1)n(2n-1)}{6} = \frac{n^3}{3} + O(n^2)$

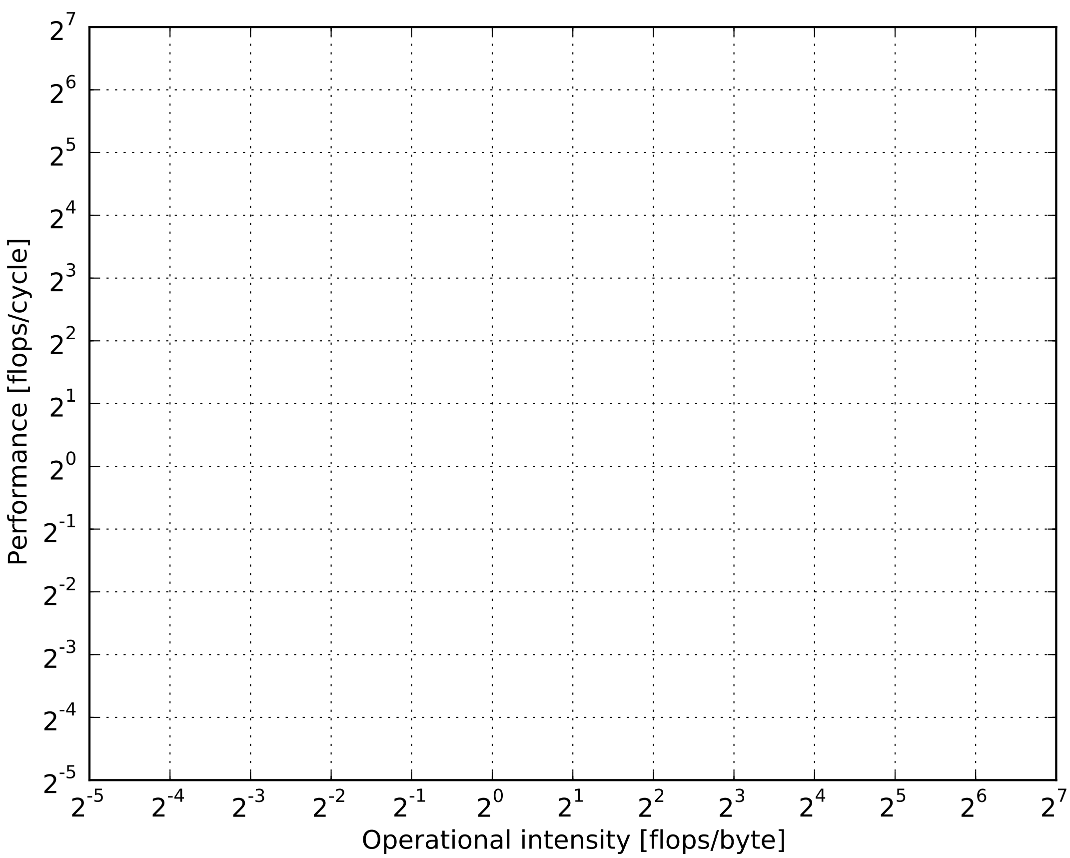
Problem 3 (18 = 2 + 6 + 10 points)

Assume you are using a system with the following features:

- A CPU that can issue 2 single precision multiplications and 2 single precision additions/subtractions per cycle.
- The interconnection between CPU and main memory (size 16 GB) has a maximal bandwidth of 8 bytes/cycle.
- The last level cache is write-allocate/write-back, direct mapped, has size 8 MB and block size of 64 bytes.

Answer the following two questions:

1. Draw the roofline plot for this system:



2. Consider the following code where all the entries in matrix m are initialized between 0 and 1:

```
void compute(float m[64]) {
    int i;

    for(i = 1; i < 64; i++) {
        m[i-1] = (1 - m[i-1]) * m[i];
        m[i] = (1 - m[i]) * m[i-1];
    }
}
```

Assume a cold cache, that the operators are left associative (expressions are evaluated from left to right), and that a float takes 4 bytes. Now compute,

(a) The operational intensity of this code (ignore write-backs).

(b) An upper bound (as tight as possible) for performance on the specified system.

You are allowed to make minor approximations. Show your work.

3. Consider the following code where *alpha* is initialized between 0 and 1:

```
void  compute(float A[4096][4096], float alpha) {
    int i,j;

    for(i = 0; i < 4096; i++)
        for(j=0; j < 4096; j++)
            A[i][j] = alpha*A[i][i] + (1 - alpha)*A[j][j];
}
```

Assume a cold cache, that the operators are left associative (expressions are evaluated from left to right), and that a float takes 4 bytes. Now compute,

(a) The operational intensity of this code (ignore write-backs).

(b) An upper bound (as tight as possible) for performance on the specified system.

You are allowed to make minor approximations. Show your work.

Problem 4 ($12 = 2 + 2 + 2 + 2 + 2 + 2$)

Mark the following statements as true (T) or false (F). Explanations are not needed. Wrong answers give negative points but you cannot get less than 0 points for this problem. You can leave questions unanswered.

- ☐ Assume a program runs N many floating point adds and N many floating point mults and that the gaps for the two instructions are respectively g_1 and g_2 cycles/issue. Then assuming a warm cache scenario where the data set fits in cache and that accesses to the cache have a negligible cost the achievable peak performance can always be estimated as $\pi = \frac{1}{g_1} + \frac{1}{g_2}$ flops/cycle.
- ☐ A direct mapped cache with parameters $(\text{number of sets}, \text{associativity}, \text{block size}) = (S, 1, B)$ always produces twice as many conflict misses as a 2-way set associative cache with parameters $(S/2, 2, B)$.
- ☐ Data prefetching can increase operational intensity.
- ☐ Every TLB miss will also cause a cache miss.
- ☐ Every cache miss will also cause a TLB miss.
- ☐ If two algorithms solve the same problem in the same time, they have the same performance.

Problem 5 (12)

Associative caches were designed to reduce conflict misses. However, increasing associativity (while maintaining the cache size) does not guarantee to achieve this in all cases. Consider a cache C_1 with (number of sets, associativity, block size) $= (S, 1, 8)$, i.e., the block size is one double. A second cache C_2 has the same size with parameters $(S/2, 2, 8)$. Both have LRU replacement and are empty.

Consider an array `a` of $2S$ doubles that is cache-aligned (i.e., `a[0]` is mapped to the first block of either cache). Provide an access sequence (of a length that you can choose) to this array such that on C_1 fewer misses occur than on C_2.

Hint: It helps to draw the caches.

Problem 6 ($12 = 6 + 6$)

In this problem we consider a computer with a fully associative cache of size γ (measured in doubles; one double is 8 bytes) and three algorithms for which the flop count W and lower bounds for the minimal memory traffic Q (in doubles) are known:

MMM: Matrix multiplication of $N \times N$ matrices with $W(N) = 2N^3$, $Q(N) \geq \frac{N^3}{2\sqrt{2\gamma}}$ doubles.

FFT: A variant of an N-point fast Fourier transform (N a power of 2) with $W(N) = 2N \log_2 N$, $Q(N) \geq \frac{2N \log_2 N}{\log \gamma}$ doubles.

CG: A conjugate gradient method that solves a system of linear equations over a two-dimensional grid of size $N \times N$ in T iterations with $W(N,T) = 20N^2T$, $Q(N,T) \geq 6N^2T$ doubles.

1. Compute for all three algorithms upper bounds on the operational intensity $I(N)$ or $I(N,T)$ (unit: flops/byte).

   MMM:

   FFT:

   CG:

2. We continue with assume a system that the computer has a peak performance of $\pi = 4$ flops/cycle and a memory bandwidth of $\beta = 8$ bytes per cycle. Determine, separately for all three cases, the cache sizes γ (again measured in doubles, and a power of 2) for which the computation is memory bound:

   MMM:

   FFT:

   CG:

Problem 7 $(12 = 2 + 5 + 5)$

Assume a CPU with the following parameters:

- Frequency $f = 5$ GHz
- One cache (L1) with instant access (i.e., no latency, infinite bandwidth)
- Main memory with access bandwidth of β doubles/cycle and a latency of $\ell_{\text{RAM}} = 100$ ns (time needed to have a double available for computation)
- Peak performance of 2 flops/cycle

1. Determine β (make it low enough) such that every L1 miss contributes exactly ℓ_{RAM} to the total execution time.

2. Now we execute a program P on this CPU with $W(N) = 20N^2$ flops and accesses $A(N) = N^2$ doubles. If all accesses did hit the cache, P would run at the CPU's peak. However, the hit rate is 96%. What is the runtime of P (using β from the previous part)? Assume that the computation and memory accesses do not overlap.

3. Assume the introduction of a second cache (L2) with access bandwidth of β (same as main memory) and latency 10 ns. The miss rate for this cache for program P is 0.5%. What is the speed-up obtained for P by introducing this cache?