

CS598 Homework 1
Due September 17, 2008

Performance Prediction

For this assignment, assume that you have a single core processor with the following characteristics:

Clock Speed: 2 GHz
Peak Floating Point Rate: 4 GFLOPs
Sustained Memory Bandwidth: 2 GB/s
Sustained L1 Cache Bandwidth: 16 GB/s

GFLOPs = 10^9 floating point operations per second
GB/s = GigaByte per second = 10^9 bytes per second

1. A key operation in many algorithms is an inner product. In Fortran-like pseudo code, this can be written as

```
sum = 0
Do i=1,n
    sum = sum + a(i) * b(i)
enddo
```

- (a) What is the maximum rate, in GFLOPs, at which this can perform if the data fits in L1 cache?
 - (b) What is the maximum rate, in GLOPSs, at which this can perform if the data does not fit in cache?
 - (c) What is the fraction of peak performance that can be obtained with this code in these two cases?
2. A common operation in some numerical algorithms is matrix-vector multiply. For a dense matrix $a(n,n)$, the pseudo code to compute $c = A b$ is

```
do i=1,n
    sum = 0
    do j=1,n
        sum = sum + a(i,j) * b(j)
    enddo
    c(i) = sum
enddo
```

Assume that the elements of the 2-dimensional matrix a are stored in memory in this order: $a(1,1)$, $a(2,1)$, $a(3,1)$, ..., $a(n,1)$, $a(1,2)$, $a(2,2)$, $a(3,2)$, ..., $a(n-1,n)$, $a(n,n)$. This is called **column major** ordering (if a is considered a **matrix**, the matrix is stored as successive columns).

- (a) What is the maximum rate, in GFLOPs, at which this can perform if the data fits in L1 cache?
 - (b) What is the maximum rate, in GLOPSs, at which this can perform if the data does not fit in cache? Assume perfect cache behavior (no data is read more than once).
 - (c) What is the fraction of peak performance that can be obtained with this code in these two cases?
3. The matrix-vector multiply may also be written in the following form:

```
do i=1,n
  c(i) = 0
enddo
do j=1,n
  do i=1,n
    c(i) = c(i) + a(i,j) * b(j)
  enddo
enddo
```

- (a) Based just on counts of loads, stores, and floating point operations, what is the performance of this form of matrix-vector multiply compared to that in question 2?
 - (b) Do you expect the measured performance on a real system to be nearly the same as the form in question 2? Why or why not?
4. Matrix-matrix multiply is another common operation in computational science codes. A pseudo-code form for this that computes the product of the matrices A and B into C is

```
do i=1,n
  do j=1,n
    sum = 0
    do k=1,n
      sum = sum + a(i,k) * b(k,j)
    enddo
    c(i,j) = sum
  enddo
enddo
```

- (a) Simply by changing the order of the loops (for example, loop over j, then I, then k), this operation can be rewritten in 6 different ways. Why are there 6 ways, and what are they (show the pseudo code for each of the 6).
- (b) Based on just the count of loads, stores, and floating point operations, what is the performance of each of these forms?
- (c) Do you expect the measured performance on a real system to be nearly the same for the six forms? Why or why not?