Designing and building applications for extreme scale systems
CS598 SP2016
HW2: Memory of Performance

Objectives

- Gain and understanding of sustained memory bandwidth
- Become familiar with the effect of data size on sustained memory bandwidth
- Become familiar with the STREAM benchmark
- Use STREAM to predict performance for Sparse Matrix-Vector product

Tasks
Part 1:
Write a program that measures the performance of this operation (in C):

```
double c[n], a[n];
for (k=0; k<nk; k++) {
        for (i=0; i<n; i++)
                c[i] = a[i];
        //See note below
        }
```

This is a simple copy from an array a to an array c. Fill in the following table:

- Time for loop (s)
- Rate (MB/s)

(The note below) Depending on your compiler and the optimization settings, you may need to keep the compiler from eliminating code that it believes unnecessary. Add a call to a routine

```
void dummy(double *a, double *c) {}
```

after the copy loop but within the loop over k. Place this routine in a separate file (that keeps the compiler from finding and eliminating it as well).

Submit
Plot the results, using a semilog plot. Turn in both the table and the plot. Also report on which system you ran your test. Submit your code with your table.

n	Time for loop (s)	Rate (MB/s)
250		
1000		
5000		
10000		
50000		
100000		
500000		
1000000		

5000000		

Note
A MB is a Megabyte = 10^6 bytes (note: 2^{20} bytes is a Mebibyte, written MiB, not a Megabyte). To time the loop, you may use any accurate timer. You may also look at the code in Part 2 and use the same timer used for that code.

When timing the loop, pick nk such that the time measured for both loops is about one second. Report the total time divided by nk as the "Time for loop". When computing the rate, count each byte written or read as a separate byte. I.e., for n=10 and for 8 byte doubles, this copy loop moves n*8*2=10*8*2=160 bytes.

Part 2:
Download the STREAM benchmark from http://www.cs.virginia.edu/stream/ and run it. You may use either the C or Fortran version. Make sure that the array size is chosen so that it is at least 4 times the size of the cache on the processor that you are using. Report the results and indicate which version you ran (C or Fortran) in submission.

Part 3:
Using the COPY bandwidth from STREAM (Part 2) to compute a performance expectation for the sparse matrix-vector multiply operation. Assume that the read and write bandwidths are the same. Turn in the formula that you used, the values of the parameters, and the performance expectation in GigaFLOPS.

Think about the following (but you don't need to turn in):

1. How could you estimate a value of nk given an assumption that memory bandwidth is between 1 and 20 Gigabytes/sec?
2. How could you choose N to efficiently plot the different regions of memory performance?
3. How does the performance expectation in Part 3 compare to (a) the peak performance of your processor and (b) the performance of a sparse matrix-vector multiply (if you want to do this, either write a simple test program or use a library routine that performs a sparse matrix-vector multiply).