Designing and building applications for extreme scale systems
CS598 SP2016
HW7: Matrix-Matrix multiply with OpenMP

Goals
- Gain experience with OpenMP
- Use simple performance estimates to guide decisions

Tasks

In this exercise, you will add just enough OpenMP to parallelize the dense matrix-matrix multiply. Here is the Fortran version of the matrix-matrix multiply algorithm from Homework 4:

```
!! See below
do i=1,matSize
  do j=1,matSize
    sum = 0.0
    do k=1,matSize
       sum = sum + matA(i,k)*matB(k,j)
    enddo
    matC(i,j) = sum
  enddo
enddo
```

Add an OpenMP directive at "!! See below" to parallelize this computation.
Run with
- 1 thread (but no OpenMP)
- 1 thread with OpenMP
- t threads for t from 2 to the number of cores on your processor

Report the results for matrix sizes of at least 20, 100, and 1000. As in Homework 4, for the largest size, pick a matrix size (matSize) that is large enough that matB does not fit into cache.

Compare your results to a simple performance estimate. Is this computation memory bandwidth limited when several threads are used (refer to your results for Homework 6)? Comment on the behavior for the small matrix size (matSize = 20).

Submit
A PDF file containing table & plots for the results of with the three different cases above (i.e., 1 thread without OpenMP, 1 thread with OpenMP, and t threads from 2 to the number of cores) with respect to at least 3 values of matSize. A brief discussion is necessary.

To think about (but not turn in)
1. Why does this code not need an OpenMP atomic or critical section in the computation of sum and the assignment to matC(i,j) (note that requires the correct directive at the top of the loop)?
2. Do you expect adding blocking for cache (as in Homework 4) to the algorithm to change the result (e.g., the relative benefit of adding threads)? Why or why not?
3. If this was a matrix-vector multiply instead of a matrix-matrix multiply, do you expect the performance to scale as you add threads? How would you estimate the performance based on what you know from the results of Homework 6?
4. What happens if you interchange the loops over i and j? What did you think might happen?