Justin Bahr
NUID: 001592707

High Performance Computing
Homework 3 - Question 3

I ran my program on the COE Vector Linux platform (Intel ® Xeon ® CPU E5-2698 v4 @ 2.20GHz, 20 cores per socket, 2 sockets, with 791957684 KB of memory, and Linux 4.18.0).

Dense Optimizations: I transposed matrix B to improve cache spatial locality. Then I implemented loop tiling with a block size of 32 (optimal size for doubles with this node's memory hierarchy). Finally, I used OpenMP to parallelize the outer two loops, so each block is multiplied in parallel.
Sparse Optimizations: First, I put matrices A and B into compressed row format (CSR).Then I used the CSR matrix multiplication algorithm to multiply and store the nonzero values in the proper locations in matrix C. Finally, I used OpenMP to parallelize the outer for loop, so each row of C is calculated in parallel.

Runtime results shown below:

Dense	Runtime (ms)	Runtime (ns)		**Sparse**	Runtime (ms)	Runtime (ns)
1	953.528097	953528097		1	38.131709	38131709
2	919.242845	919242845		2	38.406807	38406807
3	973.05479	973054790		3	36.164973	36164973
4	932.699781	932699781		4	38.406807	37316825
5	883.150913	883150913		5	38.906534	38906534
Average	932.3352852	932335285.2		Average	37.7853696	37785369.6
Fastest	883.150913	883150913		**Fastest**	36.164973	36164973

Figure 1. Runtime Trials for Dense and Sparse Matrices

```
[jbahr@pi Question3]$ ./Q3
Starting dense matrix multiply
A dense result: 4.44488e+07
Verified result: 4.44488e+07
The total time for matrix multiplication with dense matrices = 932699781 nanoseconds

Starting sparse matrix multiply
A sparse result: -749428
Verified result: -749428
The total time for matrix multiplication with sparse matrices = 37316825 nanoseconds
The sparsity of the a and b matrices = 0.750977
```

Figure 2. Screenshot of a Trial