Parallel Progrmg: Sci & Engrg

Machine Problem 1

yhyuan2

A. Short-Answer Questions

1. Q: What happens if we change interchange the loops, would it impact performance?
   A: Interchanging the loops will impact performance because C stores array in row major. Thus, we can improve performance by moving loops that changes columns to outer layers.

2. Q: What are two compiler optimizations that could have improved our program's performance?
   A: First is -Ofast. It can optimize the code and enable aggressive loop transformations, and also give less precision of floating-point calculation to improve speed. Second is -qopt-prefetch. It can reduce cache miss to make processor don't wait much time on fetching data from memory to cache.

3. Q: Why is the tiled version of the program faster? For which matrix size did you notice a difference?
   A: Because multiplications of tiles can be calculated on different processors in multithread or multiprocessor environment. But the environment that we use now is only one processor. And there is much overhead by doing tiled matrix multiplication on one processor. So, the results show that tiled multiplication is slower than naïve one. It should be faster in multithread or multiprocessor scenario.

4. Q: How does changing the tile size affect performance? Which tile size was the best?
   A: When tile size ranges from 40 to 50, performance is the best. While tile size goes lower or higher than that range, performance will become worse.

B. LLC Miss Counts

1. Naïve

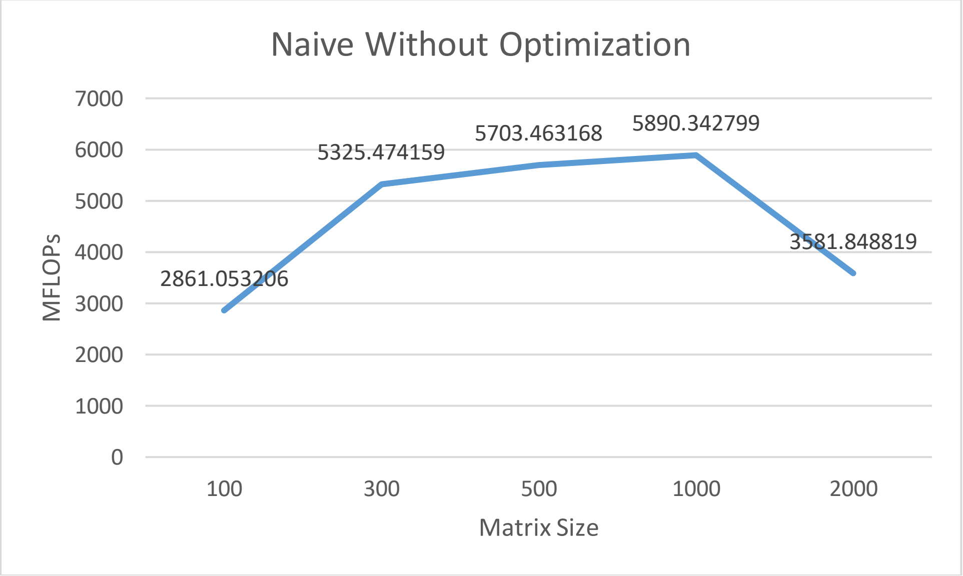
Matrix Size	LLC Miss Counts
100	0
300	0
500	0
1000	0
2000	80,002,400

2. Tiled

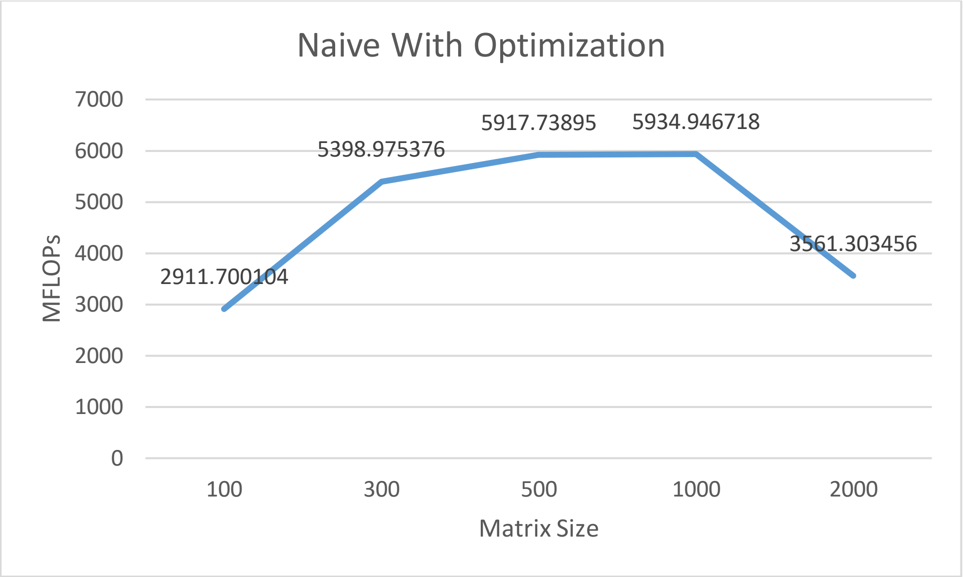
Matrix Size	Tile Size	LLC Miss Counts
100	50	0
300	150	0
500	250	0
1000	500	0
2000	1000	436,013,080

C. Plots

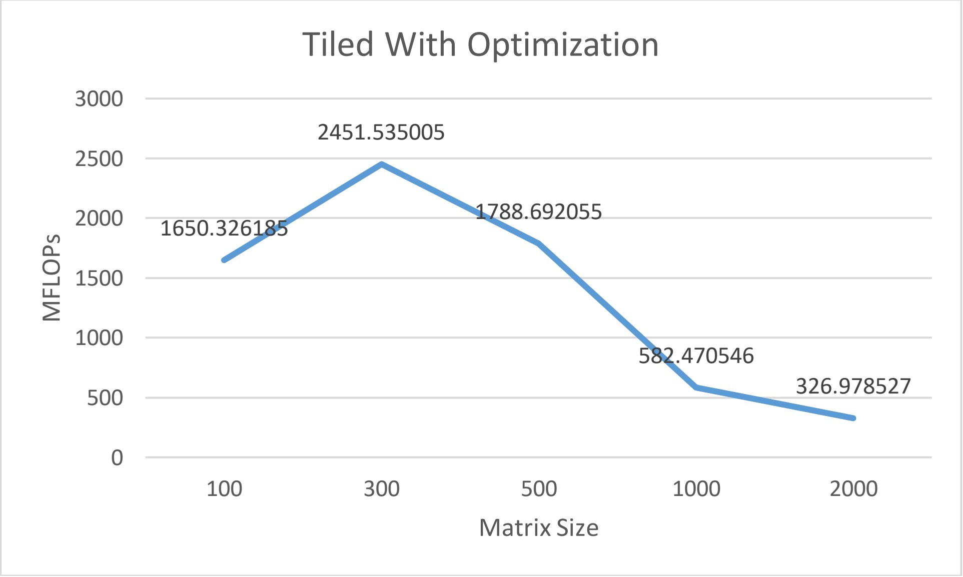
1. Naïve Implementation without Any Compiler Optimization



2. Naïve Implementation with Complier Optimization



3. Tiled Implementation with Compiler Optimization


Tiled With Optimization
MFLOPs
Matrix Size