PROJECT PRELIMINARY REPORT

Overview of the progress:

- I have build the normal matrix multiplication, of a given size and iterating it 1000 times to calculate the time.
- I have build the matrix multiplication of the inner ijk loos in 6 ways to determine which one has the best performance matrix multiplication loop
- I have prepared a estimation of the time for all the ijk loops and calculated the average
- I have created a basic cache blocking matrix program and compared to the time eastimate of the best ijk loop above.

Remaining items In the project:

- To perform the cache blocking more efficiently
- To perform the vectorization and perform the time analysis.
- To perform the hit or miss of the data through perf.

Preliminary result:

I have created an estimation_loop file, this contains the matrix multiplication with changing the ijk loops

I have created the estimation file, which contains the time lapsed between the matrix multiplication and cache blocked matrix multiplication(will try to find more apt solution for the cache blocking algorithm).

Milestones:

1. To take a correct subset for the cache memory, so that the frequently used data can be kept in cache

   I have taken the rows in the first matrix since that row would be constant and then multiplied with the remaining columns (the rows become constant) the columns would change to calculate the resultant matrix.
2. To determine a hit or a miss of the data
3. To calculate the matrix multiplication

   Created a program called matrix.c which does the matrix multiplication.
4. To determine the locality of the data.

   Spatial locality is address by the ijk loops and determining the estimated time.
5. To perform cache miss analysis of the matrix multiplication

   I will solve this trough perf.

6. To perform the blocked matrix multiplication

   Created a program called cacheblocking.c with number of block as 3 which does the matrix multiplication with cache blocking.

7. To perform the cache miss analysis of the blocked matrix multiplication .
8. To perform vectorization.