

Assignment 1

Arpit Prasad

COL718 - Architecture for High Performance Computers

August 15, 2025

1 Overview

In this assignment we perform analysis of different variants of computer hardware benchmarking on the matrix multiplication algorithm

2 Code for MM

For this assignment we assume the following code for MM. The fundamental principle for the given code is:

1. **Pattern of Memory Storage:** Main Memory stores information linearly however when we access (i, j) we might encounter misses due to two calls of (i, j) being in completely different blocks

Code for Matrix Multiplication

```
1 #define N 100
2 /**
3  * @brief multiplies in the block
4  */
5 void multiply_block(int n, double *A, double *B, double *C, int block_size
6 , int i, int j, int k) {
7     for (int ii = i; ii < i + block_size && ii < n; ii++) {
8         for (int jj = j; jj < j + block_size && jj < n; jj++) {
9             for (int kk = k; kk < k + block_size && kk < n; kk++) {
10                 C[ii * n + jj] += A[ii * n + kk] * B[kk * n + jj];
11             }
12         }
13     }
14 }
15 /**
16  * @brief for initializing a matrix
17  */
18 void initialize_matrix_to_zero(double* C, int n) {
19     for (int i = 0; i < n; i++) {
20         for (int j = 0; j < n; j++) {
```

```

21         C[i * n + j] = 0.0;
22     }
23 }
24 }
25
26 /**
27  * @brief multiplies across the matrices
28  */
29 void mat_mul_blocked(int n, double *A, double *B, double *C, int
    block_size) {
30
31     initialize_matrix_to_zero(C, n);
32
33     for (int i = 0; i < n; i += block_size) {
34         for (int j = 0; j < n; j += block_size) {
35             for (int k = 0; k < n; k += block_size) {
36                 multiply_block(n, A, B, C, block_size, i, j, k);
37             }
38         }
39     }
40 }
41
42 int main() {
43     double A[N][N], B[N][N], C[N][N];
44     mat_mul_blocked(N, (double *)A, (double *)B, (double *)C, 8);
45 }

```

3 Performance Metrics

The following is the overview of the performance metrics that are used to analyse the performance of a code with respect to a Model

1. Raw speed of the processor
2. Latency
3. Period

4 Performance vs CPU Models

Here we vary the CPU Models and check performance in the metrics specified above