PERFORMANCE PROFILING WITH PERF

CS20B059

The objective of this assignment is to compare the cache effects of Integer Matrix multiplication using the common naive algorithm and a Blocked matrix multiplication.And find the cache block size using Blocked matrix multiplication.

Naive Algorithm:

```
for(int i=0;i<n;i++)
{
  for(int j=0;j<n;j++)
  {
    for(int k=0;k<n;k++)
    {
      c[i][j]+=(a[i][k]*b[k][j]);
    }
  }
}
```

Explanation:

We calculate each entry as the sum of products of ith row of matrix A and jth column of matrix B.That is

c(i,j)=sum of(a(i,k)*b(k,j)

Blocking Matrix Multiplication Algorithm:

```
for(int p=0;p<n;p=p+block_size)
    {
    for(int l=0;l<n;l=l+block_size)
      {
        for(int i=0;i<n;i++)
       {
            for(int j=p;j<min(p+block_size,n);j++)
      {
             int temp = 0;
             for(int k=l; k<min(l+block_size,n);k++)
        {
                temp += a[i][k]*b[k][j];
              }
              c[i][j] += temp;
            }
        }
      }
   }
```

Explanation:

The code divides the matrix into sub-matrix of size (block_sizeXblock_size). Firstly we multiply the elements in block matrix and then we will multiply the blocks.

First 2 loops specify the limits to which the number of rows and columns to be multiplied and next 3 loops specify the matrix multiplication to be done.(naive algorithm)

Commands Used:

Command:g++ gen_input.cpp
Complies the cpp file gen_input.

Command:./a.out 128 input
Takes the size of matrix as 128 and stores the random matrix formed using gen_input.cpp in file named as input.

Command:g++ template.cpp
Complies the cpp file template.

Command:./a.out 0 input output
Takes block_size=0 and input matrix as random matrix in input file and completes the matrix multiplication based on block_size giving and stores the output file in file named as output.

Command:sudo perf stat -r 10 -e instructions,cycles,cache-misses,cache-references,L1-dcache-load-misses,L1-icache-load-misses ./a.out 0 input128 output128
Using perf we will find the events we want and compare the values with two algorithms and decide the block_size by the values we get.

Naive Algorithm as block_size is '0' for random input matrix of size 128X128

```
 Performance counter stats for './a.out 0 input128 output128' (3 runs):

       128669091      instructions              #    3.73  insn per cycle           ( +-  0.01% )
        34517000      cycles                                                  ( +-  0.19% )
            7192      cache-misses              #   12.662 % of all cache refs      ( +- 41.22% )
           56797      cache-references                                        ( +-  1.39% )
         2322396      L1-dcache-load-misses                                   ( +-  0.30% )
          151072      L1-icache-load-misses                                   ( +-  0.65% )

       0.008950 +- 0.000411 seconds time elapsed  ( +-  4.60% )
```

Blocked Algorithm as block_size is '8' for random input matrix of size 128X128

```
 Performance counter stats for './a.out 8 input128 output128' (3 runs):

       161934957      instructions              #    3.56  insn per cycle           ( +-  0.00% )
        45494649      cycles                                                  ( +-  0.19% )
            7967      cache-misses              #   13.434 % of all cache refs      ( +- 36.41% )
           59309      cache-references                                        ( +-  1.49% )
          181731      L1-dcache-load-misses                                   ( +-  4.98% )
          151446      L1-icache-load-misses                                   ( +-  1.07% )

       0.011466 +- 0.000339 seconds time elapsed  ( +-  2.96% )
```

Blocked Algorithm as block_size is '16' for random input matrix of size 128X128

```
Performance counter stats for './a.out 16 input128 output128' (3 runs):

     150198250      instructions          #    3.70  insn per cycle         ( +-  0.01% )
      40588817      cycles                                                  ( +-  1.02% )
         13199      cache-misses          #   21.983 % of all cache refs    ( +- 23.13% )
         60041      cache-references                                        ( +-  2.98% )
        101446      L1-dcache-load-misses                                   ( +-  2.87% )
        153822      L1-icache-load-misses                                   ( +-  0.97% )

      0.010754 +- 0.000429 seconds time elapsed  ( +-  3.99% )
```

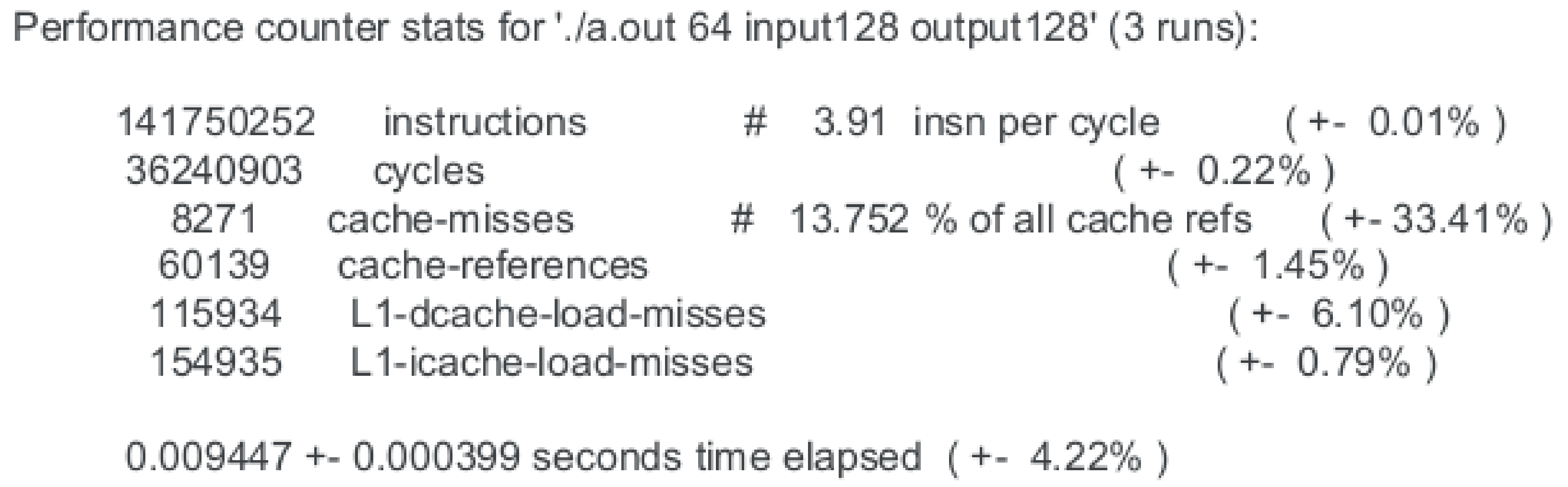
Blocked Algorithm as block_size is '32' for random input matrix of size 128X128

```
Performance counter stats for './a.out 32 input128 output128' (3 runs):

     144507467      instructions          #    3.80  insn per cycle         ( +-  0.00% )
      38038473      cycles                                                  ( +-  1.19% )
          7728      cache-misses          #   13.209 % of all cache refs    ( +- 37.60% )
         58510      cache-references                                        ( +-  1.78% )
         71961      L1-dcache-load-misses                                   ( +-  0.70% )
        152275      L1-icache-load-misses                                   ( +-  0.86% )

      0.010249 +- 0.000490 seconds time elapsed  ( +-  4.78% )
```

Blocked Algorithm as block_size is '64' for random input matrix of size 128X128



```
 Performance counter stats for './a.out 64 input128 output128' (3 runs):

       141750252      instructions              #    3.91  insn per cycle           ( +-  0.01% )
        36240903      cycles                                                        ( +-  0.22% )
            8271      cache-misses              #   13.752 % of all cache refs      ( +- 33.41% )
           60139      cache-references                                              ( +-  1.45% )
          115934      L1-dcache-load-misses                                         ( +-  6.10% )
          154935      L1-icache-load-misses                                         ( +-  0.79% )

        0.009447 +- 0.000399 seconds time elapsed  ( +-  4.22% )
```

Explanation:

As we load a[0][0] implies the data in the neighbourhood of a[0][0] will also be bought into the cache.So the access latency for the neighboring addresses of a[0][0] should be less as it is a cache hit as they are already present in the cache.So the dcache-load-misses will be less for the ideal block size.When we compare the load misses for different blocksizes(0,8,16,32,64) we can observe that when block size is 16B or 32B.Then Load misses are less and time elapsed is minimum.
By observing the values of L1-dcahce-load-misses we can conclude that block_size can be 16B or 32B.

Naive Algorithm as block_size is '0' for random input matrix of size 256X256

```
Performance counter stats for './a.out 0 input256 output256' (5 runs):

      895891342      instructions            #    3.80  insn per cycle          ( +-  0.00% )
      236056829      cycles                                                 ( +-  0.37% )
         46755      cache-misses             #   46.684 % of all cache refs      ( +-  4.87% )
        100154      cache-references                                        ( +-  2.12% )
      20386570      L1-dcache-load-misses                                      ( +-  0.25% )
        289748      L1-icache-load-misses                                     ( +-  1.13% )

      0.058038 +- 0.000595 seconds time elapsed  ( +-  1.03% )
```

Blocked Algorithm as block_size is '8' for random input matrix of size 256X256

```
Performance counter stats for './a.out 8 input256 output256' (5 runs):

      1162452092      instructions            #    3.64  insn per cycle          ( +-  0.00% )
       319318797      cycles                                                 ( +-  0.07% )
           43301      cache-misses             #   45.514 % of all cache refs      ( +-  4.81% )
           95139      cache-references                                        ( +-  1.67% )
         1108008      L1-dcache-load-misses                                      ( +-  5.93% )
          283632      L1-icache-load-misses                                     ( +-  1.26% )

      0.077447 +- 0.000329 seconds time elapsed  ( +-  0.42% )
```

Blocked Algorithm as block_size is '16' for random input matrix of size 256X256

```
Performance counter stats for './a.out 16 input256 output256' (5 runs):

    1068497802      instructions              #    3.80  insn per cycle           ( +-  0.00% )
     281449225      cycles                                                        ( +-  0.46% )
         43901      cache-misses              #   45.542 % of all cache refs      ( +-  6.73% )
         96396      cache-references                                              ( +-  1.70% )
        434851      L1-dcache-load-misses                                         ( +-  5.70% )
        288691      L1-icache-load-misses                                         ( +-  0.69% )

      0.068276 +- 0.000551 seconds time elapsed  ( +-  0.81% )
```

Blocked Algorithm as block_size is '32' for random input matrix of size 256X256

```
Performance counter stats for './a.out 32 input256 output256' (5 runs):

    1023329211      instructions              #    4.08  insn per cycle           ( +-  0.00% )
     250711591      cycles                                                        ( +-  0.87% )
         28875      cache-misses              #   30.915 % of all cache refs      ( +- 14.06% )
         93402      cache-references                                              ( +-  1.30% )
        427663      L1-dcache-load-misses                                         ( +-  1.92% )
        281623      L1-icache-load-misses                                         ( +-  0.92% )

      0.060656 +- 0.000602 seconds time elapsed  ( +-  0.99% )
```

Blocked Algorithm as block_size is '64' for random input matrix of size 256X256

```
 Performance counter stats for './a.out 64 input256 output256' (5 runs):

      1001115837      instructions              #    3.83  insn per cycle           ( +-  0.00% )
       261606631      cycles                                                  ( +-  0.38% )
           41058      cache-misses              #   41.224 % of all cache refs      ( +-  5.34% )
           99599      cache-references                                        ( +-  2.30% )
        20973822      L1-dcache-load-misses                                         ( +-  0.67% )
          286795      L1-icache-load-misses                                         ( +-  0.73% )

        0.063706 +- 0.000406 seconds time elapsed  ( +-  0.64% )
```

Blocked Algorithm as block_size is '128' for random input matrix of size 256X256

```
 Performance counter stats for './a.out 128 input256 output256' (5 runs):

       990125005      instructions              #    3.88  insn per cycle           ( +-  0.00% )
       255062096      cycles                                                  ( +-  0.52% )
           37793      cache-misses              #   42.207 % of all cache refs      ( +-  7.03% )
           89542      cache-references                                        ( +-  0.95% )
        20796875      L1-dcache-load-misses                                         ( +-  0.38% )
          282733      L1-icache-load-misses                                         ( +-  0.90% )

        0.062050 +- 0.000466 seconds time elapsed  ( +-  0.75% )
```

Explanation:

When we compare the load misses for different blocksizes(0,8,16,32,64,128).
We can observe that when block size is 16.Load misses are less and time elapsed is minimum.
By observing the values of L1-dcahce-load-misses and time elpased we can conclude ideal block_size is 16B for random input matrix 256X256.

Naive Algorithm as block_size is '0' for random input matrix of size 512X512

```
 Performance counter stats for './a.out 0 input512 output512' (10 runs):

        6725990818      instructions              #    3.32  insn per cycle           ( +-  0.00% )
        2023343135      cycles                                                        ( +-  3.44% )
          164232      cache-misses              #    2.888 % of all cache refs      ( +-  2.87% )
         5685985      cache-references                                              ( +- 28.41% )
        151754546      L1-dcache-load-misses                                         ( +-  0.30% )
          693350      L1-icache-load-misses                                         ( +-  0.84% )

       0.4853 +- 0.0167 seconds time elapsed  ( +-  3.45% )
```

Blocked Algorithm as block_size is '8' for random input matrix of size 512X512

```
Performance counter stats for './a.out 8 input512 output512' (10 runs):

      8860758779      instructions              #    3.61  insn per cycle           ( +-  0.00% )
      2453190307      cycles                                                        ( +-  0.15% )
        248341      cache-misses              #    8.273 % of all cache refs      ( +-  4.86% )
       3001828      cache-references                                              ( +-  5.55% )
       9669730      L1-dcache-load-misses                                         ( +-  4.63% )
        679551      L1-icache-load-misses                                         ( +-  1.14% )

     0.587862 +- 0.000903 seconds time elapsed  ( +-  0.15% )
```

Blocked Algorithm as block_size is '16' for random input matrix of size 512X512

```
Performance counter stats for './a.out 16 input512 output512' (10 runs):

      8109319018      instructions              #    3.74  insn per cycle           ( +-  0.00% )
      2170189526      cycles                                                        ( +-  0.28% )
        225761      cache-misses              #   15.469 % of all cache refs      ( +-  4.49% )
       1459432      cache-references                                              ( +-  5.38% )
       5620414      L1-dcache-load-misses                                         ( +-  2.17% )
        679801      L1-icache-load-misses                                         ( +-  1.40% )

     0.53120 +- 0.00377 seconds time elapsed  ( +-  0.71% )
```

Blocked Algorithm as block_size is '32' for random input matrix of size 512X512

```
Performance counter stats for './a.out 32 input512 output512' (10 runs):

      7747509388      instructions              #    3.71  insn per cycle           ( +-  0.00% )
      2090528264      cycles                                                        ( +-  0.65% )
        255408      cache-misses              #   14.375 % of all cache refs      ( +-  5.55% )
       1776738      cache-references                                              ( +-  5.35% )
       194067721      L1-dcache-load-misses                                         ( +-  0.65% )
        680050      L1-icache-load-misses                                         ( +-  0.79% )

     0.51683 +- 0.00379 seconds time elapsed  ( +-  0.73% )
```

Blocked Algorithm as block_size is '64' for random input matrix of size 512X512

```
 Performance counter stats for './a.out 64 input512 output512' (10 runs):

     7570024594      instructions              #    3.81  insn per cycle           ( +-  0.00% )
     1987031282      cycles                                                ( +-  0.13% )
         219302      cache-misses              #   19.648 % of all cache refs      ( +-  4.07% )
        1116172      cache-references                                          ( +-  2.23% )
      193375564      L1-dcache-load-misses                                           ( +-  0.54% )
         686755      L1-icache-load-misses                                         ( +-  0.87% )

       0.49118 +- 0.00206 seconds time elapsed  ( +-  0.42% )
```

Blocked Algorithm as block_size is '128' for random input matrix of size 512X512

```
 Performance counter stats for './a.out 128 input512 output512' (10 runs):

     7482101004      instructions              #    3.90  insn per cycle           ( +-  0.00% )
     1917606934      cycles                                                ( +-  0.04% )
         198022      cache-misses              #   30.034 % of all cache refs      ( +-  2.07% )
         659333      cache-references                                          ( +-  1.35% )
      188410034      L1-dcache-load-misses                                           ( +-  0.38% )
         701952      L1-icache-load-misses                                         ( +-  1.45% )

        0.4729 +- 0.0114 seconds time elapsed  ( +-  2.41% )
```

Blocked Algorithm as block_size is '256' for random input matrix of size 512X512

```
 Performance counter stats for './a.out 256 input512 output512' (10 runs):

     7438235153      instructions              #    3.83  insn per cycle           ( +-  0.00% )
     1940474075      cycles                                                ( +-  0.12% )
         169109      cache-misses              #   39.004 % of all cache refs      ( +-  2.21% )
         433568      cache-references                                          ( +-  1.61% )
      165004372      L1-dcache-load-misses                                           ( +-  0.05% )
         689654      L1-icache-load-misses                                         ( +-  1.49% )

       0.46687 +- 0.00134 seconds time elapsed  ( +-  0.29% )
```

By observing the values of cache-misses and time elpased we can conclude that blocksize is 16B as dcahce load misses is minimum for blocksize=16 and minimum time elapsed for random matrix size(512X512).