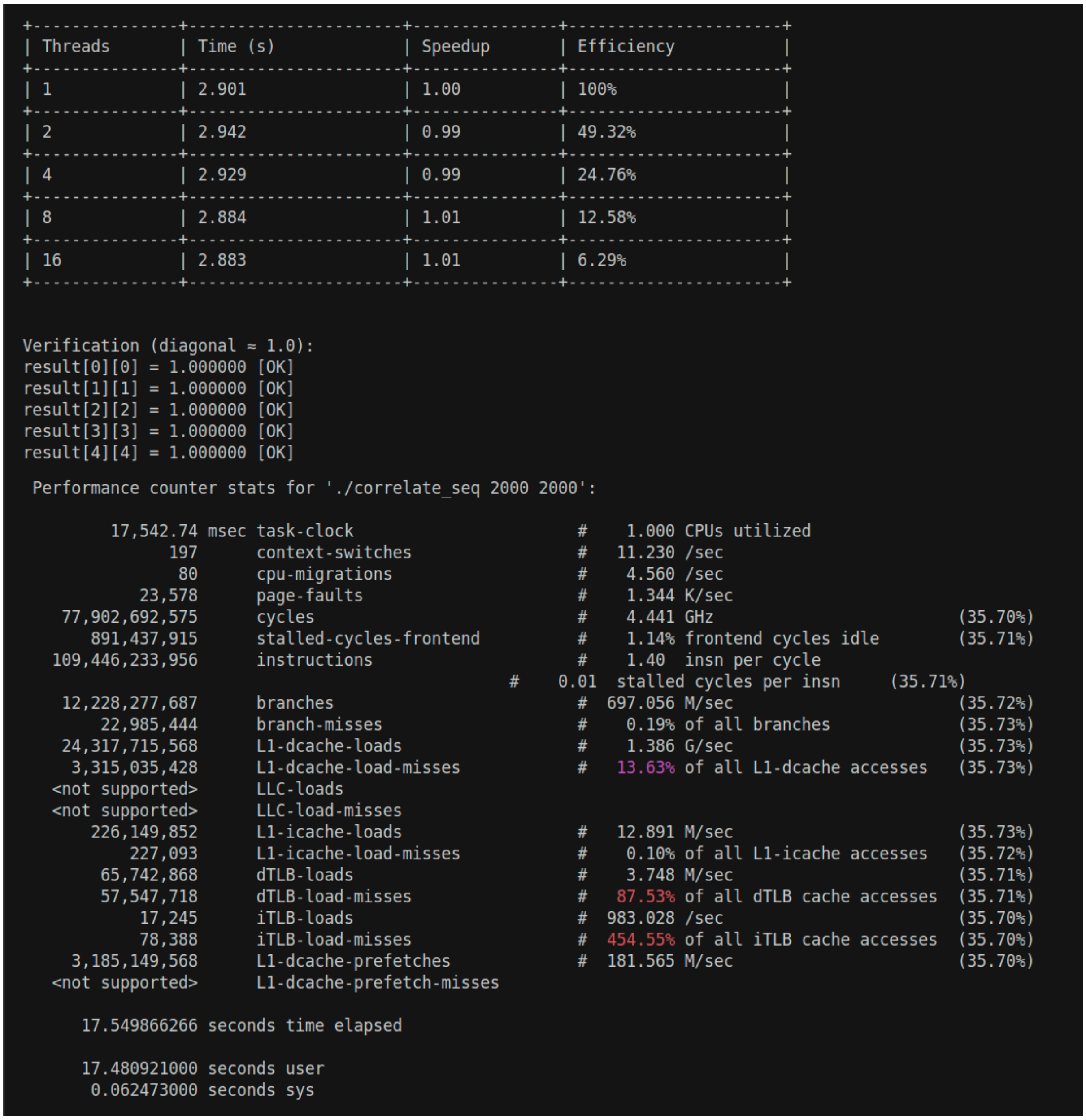
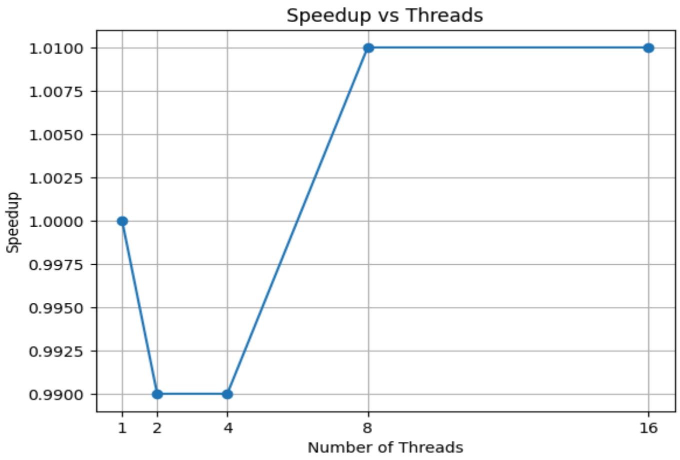
ASSIGNMENT-3
GAURANG GARG [102303134]

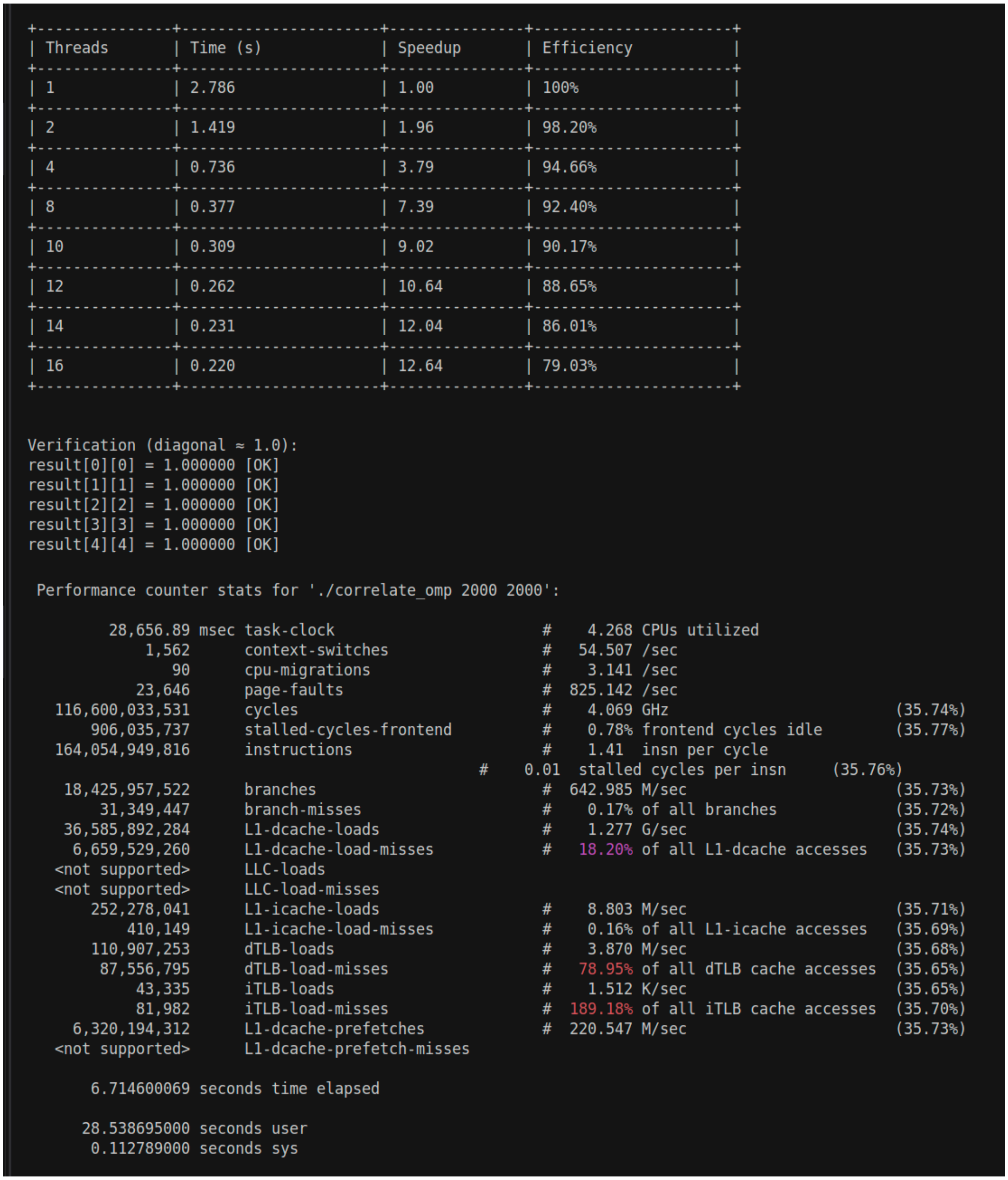
SEQUENTIAL :



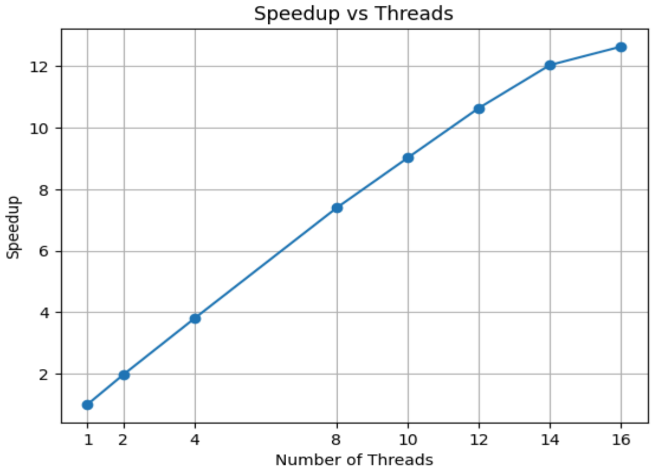
```
+--------------+---------------------+--------------+----------------------+
| Threads      | Time (s)            | Speedup      | Efficiency           |
+--------------+---------------------+--------------+----------------------+
| 1            | 2.901               | 1.00         | 100%                 |
+--------------+---------------------+--------------+----------------------+
| 2            | 2.942               | 0.99         | 49.32%               |
+--------------+---------------------+--------------+----------------------+
| 4            | 2.929               | 0.99         | 24.76%               |
+--------------+---------------------+--------------+----------------------+
| 8            | 2.884               | 1.01         | 12.58%               |
+--------------+---------------------+--------------+----------------------+
| 16           | 2.883               | 1.01         | 6.29%                |
+--------------+---------------------+--------------+----------------------+


Verification (diagonal ≈ 1.0):
result[0][0] = 1.000000 [OK]
result[1][1] = 1.000000 [OK]
result[2][2] = 1.000000 [OK]
result[3][3] = 1.000000 [OK]
result[4][4] = 1.000000 [OK]

 Performance counter stats for './correlate_seq 2000 2000':

        17,542.74 msec task-clock                       #    1.000 CPUs utilized
              197      context-switches                 #   11.230 /sec
               80      cpu-migrations                   #    4.560 /sec
           23,578      page-faults                      #    1.344 K/sec
   77,902,692,575      cycles                           #    4.441 GHz                         (35.70%)
      891,437,915      stalled-cycles-frontend          #    1.14% frontend cycles idle        (35.71%)
  109,446,233,956      instructions                     #    1.40  insn per cycle
                                                 #    0.01  stalled cycles per insn     (35.71%)
   12,228,277,687      branches                         #  697.056 M/sec                       (35.72%)
       22,985,444      branch-misses                    #    0.19% of all branches             (35.73%)
   24,317,715,568      L1-dcache-loads                  #    1.386 G/sec                       (35.73%)
    3,315,035,428      L1-dcache-load-misses            #   13.63% of all L1-dcache accesses   (35.73%)
  <not supported>      LLC-loads
  <not supported>      LLC-load-misses
      226,149,852      L1-icache-loads                  #   12.891 M/sec                       (35.73%)
          227,093      L1-icache-load-misses            #    0.10% of all L1-icache accesses   (35.72%)
       65,742,868      dTLB-loads                       #    3.748 M/sec                       (35.71%)
       57,547,718      dTLB-load-misses                 #   87.53% of all dTLB cache accesses  (35.71%)
           17,245      iTLB-loads                       #  983.028 /sec                        (35.70%)
           78,388      iTLB-load-misses                 #  454.55% of all iTLB cache accesses  (35.70%)
    3,185,149,568      L1-dcache-prefetches             #  181.565 M/sec                       (35.70%)
  <not supported>      L1-dcache-prefetch-misses

     17.549866266 seconds time elapsed

     17.480921000 seconds user
      0.062473000 seconds sys
```



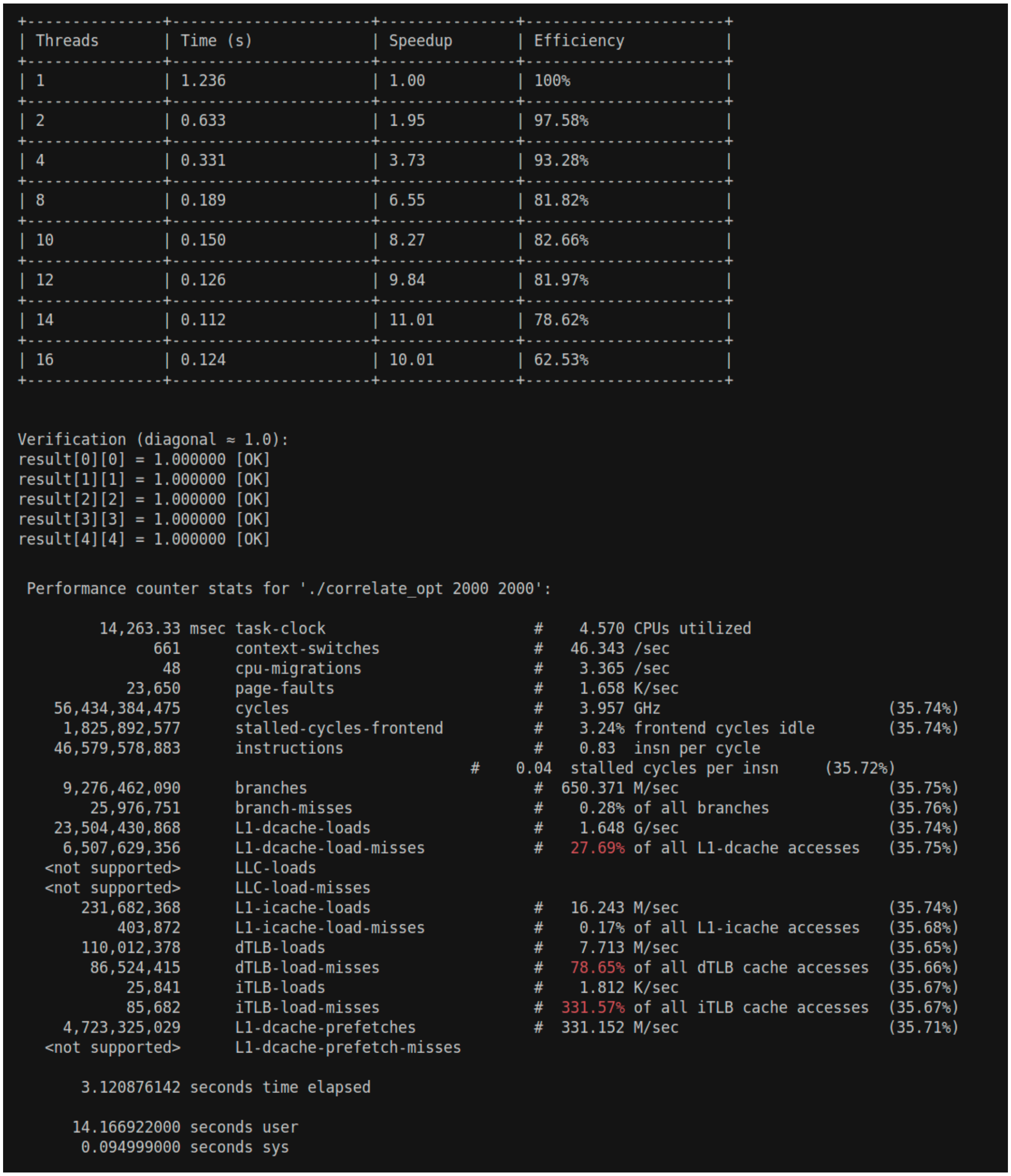
OPENMP :



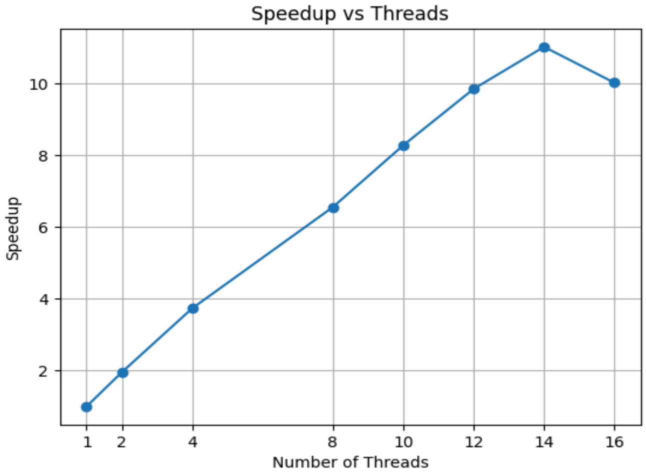
```
+--------------+--------------------+---------------+---------------------+
| Threads      | Time (s)           | Speedup       | Efficiency          |
+--------------+--------------------+---------------+---------------------+
| 1            | 2.786              | 1.00          | 100%                |
+--------------+--------------------+---------------+---------------------+
| 2            | 1.419              | 1.96          | 98.20%              |
+--------------+--------------------+---------------+---------------------+
| 4            | 0.736              | 3.79          | 94.66%              |
+--------------+--------------------+---------------+---------------------+
| 8            | 0.377              | 7.39          | 92.40%              |
+--------------+--------------------+---------------+---------------------+
| 10           | 0.309              | 9.02          | 90.17%              |
+--------------+--------------------+---------------+---------------------+
| 12           | 0.262              | 10.64         | 88.65%              |
+--------------+--------------------+---------------+---------------------+
| 14           | 0.231              | 12.04         | 86.01%              |
+--------------+--------------------+---------------+---------------------+
| 16           | 0.220              | 12.64         | 79.03%              |
+--------------+--------------------+---------------+---------------------+


Verification (diagonal ≈ 1.0):
result[0][0] = 1.000000 [OK]
result[1][1] = 1.000000 [OK]
result[2][2] = 1.000000 [OK]
result[3][3] = 1.000000 [OK]
result[4][4] = 1.000000 [OK]

 Performance counter stats for './correlate_omp 2000 2000':

        28,656.89 msec task-clock                       #    4.268 CPUs utilized
            1,562      context-switches                 #   54.507 /sec
               90      cpu-migrations                   #    3.141 /sec
           23,646      page-faults                      #  825.142 /sec
  116,600,033,531      cycles                           #    4.069 GHz                      (35.74%)
      906,035,737      stalled-cycles-frontend          #    0.78% frontend cycles idle     (35.77%)
  164,054,949,816      instructions                     #    1.41  insn per cycle
                                                 #    0.01  stalled cycles per insn     (35.76%)
   18,425,957,522      branches                         #  642.985 M/sec                    (35.73%)
       31,349,447      branch-misses                    #    0.17% of all branches          (35.72%)
   36,585,892,284      L1-dcache-loads                  #    1.277 G/sec                    (35.74%)
    6,659,529,260      L1-dcache-load-misses            #   18.20% of all L1-dcache accesses   (35.73%)
  <not supported>      LLC-loads
  <not supported>      LLC-load-misses
      252,278,041      L1-icache-loads                  #    8.803 M/sec                    (35.71%)
          410,149      L1-icache-load-misses            #    0.16% of all L1-icache accesses   (35.69%)
      110,907,253      dTLB-loads                       #    3.870 M/sec                    (35.68%)
       87,556,795      dTLB-load-misses                 #   78.95% of all dTLB cache accesses  (35.65%)
           43,335      iTLB-loads                       #    1.512 K/sec                    (35.65%)
           81,982      iTLB-load-misses                 #  189.18% of all iTLB cache accesses  (35.70%)
    6,320,194,312      L1-dcache-prefetches             #  220.547 M/sec                    (35.73%)
  <not supported>      L1-dcache-prefetch-misses

      6.714600069 seconds time elapsed

     28.538695000 seconds user
      0.112789000 seconds sys
```



OPTIMIZED :



```
+---------------+----------------------+---------------+-----------------------+
| Threads       | Time (s)             | Speedup       | Efficiency            |
+---------------+----------------------+---------------+-----------------------+
| 1             | 1.236                | 1.00          | 100%                  |
+---------------+----------------------+---------------+-----------------------+
| 2             | 0.633                | 1.95          | 97.58%                |
+---------------+----------------------+---------------+-----------------------+
| 4             | 0.331                | 3.73          | 93.28%                |
+---------------+----------------------+---------------+-----------------------+
| 8             | 0.189                | 6.55          | 81.82%                |
+---------------+----------------------+---------------+-----------------------+
| 10            | 0.150                | 8.27          | 82.66%                |
+---------------+----------------------+---------------+-----------------------+
| 12            | 0.126                | 9.84          | 81.97%                |
+---------------+----------------------+---------------+-----------------------+
| 14            | 0.112                | 11.01         | 78.62%                |
+---------------+----------------------+---------------+-----------------------+
| 16            | 0.124                | 10.01         | 62.53%                |
+---------------+----------------------+---------------+-----------------------+


Verification (diagonal ≈ 1.0):
result[0][0] = 1.000000 [OK]
result[1][1] = 1.000000 [OK]
result[2][2] = 1.000000 [OK]
result[3][3] = 1.000000 [OK]
result[4][4] = 1.000000 [OK]

 Performance counter stats for './correlate_opt 2000 2000':

         14,263.33 msec task-clock                       #    4.570 CPUs utilized
               661      context-switches                 #   46.343 /sec
                48      cpu-migrations                   #    3.365 /sec
            23,650      page-faults                      #    1.658 K/sec
    56,434,384,475      cycles                           #    3.957 GHz                         (35.74%)
     1,825,892,577      stalled-cycles-frontend          #    3.24% frontend cycles idle        (35.74%)
    46,579,578,883      instructions                     #    0.83  insn per cycle
                                                  #    0.04  stalled cycles per insn     (35.72%)
     9,276,462,090      branches                         #  650.371 M/sec                       (35.75%)
        25,976,751      branch-misses                    #    0.28% of all branches             (35.76%)
    23,504,430,868      L1-dcache-loads                  #    1.648 G/sec                       (35.74%)
     6,507,629,356      L1-dcache-load-misses            #   27.69% of all L1-dcache accesses   (35.75%)
   <not supported>      LLC-loads
   <not supported>      LLC-load-misses
       231,682,368      L1-icache-loads                  #   16.243 M/sec                       (35.74%)
           403,872      L1-icache-load-misses            #    0.17% of all L1-icache accesses   (35.68%)
       110,012,378      dTLB-loads                       #    7.713 M/sec                       (35.65%)
        86,524,415      dTLB-load-misses                 #   78.65% of all dTLB cache accesses  (35.66%)
            25,841      iTLB-loads                       #    1.812 K/sec                       (35.67%)
            85,682      iTLB-load-misses                 #  331.57% of all iTLB cache accesses  (35.67%)
     4,723,325,029      L1-dcache-prefetches             #  331.152 M/sec                       (35.71%)
   <not supported>      L1-dcache-prefetch-misses

       3.120876142 seconds time elapsed

      14.166922000 seconds user
       0.094999000 seconds sys
```



OBSERVATION :

Version 1 - Sequential Baseline :

The baseline took **~2.95 seconds** with 1 thread. When you gave it more threads (2, 4, 8... 16), the time barely changed it stayed around 2.8–3.1 seconds. That's because Version 1 has no parallel code. Adding more threads did nothing; the work still ran on one core. All "speedups" are basically 1x (no improvement at all).

Bottom line: More threads = no benefit. Works correctly, but uses only 1 core.

Version 2 - OpenMP Parallel :

Baseline here was **~2.79 seconds** with 1 thread. With more threads, it got noticeably faster:

Threads	Time	Speedup
1	2.79s	1×
2	1.42s	~2×
4	0.74s	~4×
8	0.38s	~7.4×
16	0.22s	~12.6×

This scales really well. Doubling the threads roughly halves the time. The efficiency (how well each thread is being used) stays above 85–90% until 14 threads, then drops slightly at 16. That's excellent parallel scaling.

Bottom line: Adding threads genuinely helps. Near perfect scaling up to ~12 threads.

Version 3 - Optimized (AVX2 SIMD + OpenMP) :

Baseline here was just **~1.24 seconds** that's already 2.25× faster than Version 2 with just 1 thread. This is because AVX2 does 4 floating-point operations in one CPU instruction instead of 1.

With threads:

Threads	Time	Speedup
1	1.24s	1×
8	0.19s	~6.5×
12	0.13s	~9.8×
16	0.12s	~10×

Notice at 16 threads the speedup actually drops a little compared to 14 threads (from 11× to 10×). This is called diminishing returns the overhead of managing 16 threads starts eating into the gains.

Bottom line: Fastest version overall. But above 12-14 threads, adding more threads stops helping.

The perf stat Numbers :

Metric	**Seq**	**OMP**	**Optimized**
Total CPU time	~26.6s	~28.7s	~14.3s
Instructions per cycle	1.39	1.41	0.83
L1 cache miss rate	13.6%	18.2%	27.7%

Instructions per cycle (IPC): Version 3 has a *lower* IPC (0.83 vs 1.39), which sounds bad, but it's misleading. Each AVX2 instruction does 4× the math. So fewer instructions doing more work is actually better.

Cache miss rate: Version 3 misses L1 cache more (27.7%) because AVX2 pulls in larger data chunks at a time, sometimes exceeding what fits in the fast L1 cache. This is a known tradeoff you gain from SIMD but pay a little in cache pressure.

dTLB misses (~87%): All three versions have very high TLB miss rates. This means the program accesses memory spread across many different pages, which is expected for a large 2000×2000 matrix. This is a minor bottleneck you can't easily avoid.