

INDIAN INSTITUTE OF TECHNOLOGY GANDHINAGAR

CS 616 - Parallel and Distributed Systems

Assignment-0

Please find the source code at: <https://github.com/PechimuthuMithil/PDS-Assignment0>

Key optimization strategies used:

- 1) Divided the matrix into smaller subproblems that can be solved parallelly. This was motivated by Strassen's algorithm.
- 2) Each sub problem was computed by 7 threads running in parallel with no race dependencies.
- 3) Used SIMD instructions through numpy dot operation for base cases. Also found the optimal base case size for quickest performance on the host processor.
- 4) Reduced the use of temporary variables to increase cache hits. Also chose specific shapes to store the data in. The following example will make it clear.

```
    result = np.dot(A, B)
else:
    mid = n // 2
    P = np.zeros((7*mid, mid))

    strassen_matrix_multiply(P[:mid,:], A[:mid, :mid] + A[mid:, mid:], B[:mid, :mid] + B[mid:, mid:])
    strassen_matrix_multiply(P[mid:2*mid:], A[mid:, :mid] + A[mid:, mid:], B[:mid, :mid])
```

Fig 1: Final submission where P has more rows than columns

```
else:
    mid = n // 2
    P = np.zeros((mid, 7*mid))

    strassen_matrix_multiply(P[:, :mid], A[:mid, :mid] + A[mid:, mid:], B[:mid, :mid] + B[mid:, mid:])
```

Fig 2: Another way of initializing P where it has more columns than rows

The configuration in Fig 1 has an average calculation time of 1.27s. Whereas the configuration shown in Fig 2 has 1.59s.

References:

- [1] Sindhuja Gudala, "Strassen's Matrix Multiplication", scaler.com, [Online], Available: <https://www.scaler.com/topics/strassens-matrix-multiplication/>
- [2] Jim Anderson, "An Intro to Threading in Python", realpython.com, [Online], Available: <https://realpython.com/intro-to-python-threading/>

Side Note:

- 1) I tried reducing the computations even further by making the threads do more computation and add some locks to prevent race conditions. However it failed.

- 2) I also tried using Cython to compile the program, however, I was not able to get much benefit from doing so.
- 3) I strongly believe if this program is written in a language like C or C++, we would have seen faster run times with the help of various optimization flags during compilation.