

HPC Lab Week 5

Serial to Parallel (utilizing parallel-for loops)

Utilize the OpenMP library in Python to speed up the serial for-loop computation by converting them into parallel for loops.

Exploit serial version of the following implementation done previously.

1. Matrix Multiplication
2. Linear Search
3. Array Sum
4. Prime Numbers
5. Prime in a Power Set
6. Word Search in English Dictionary
7. Image Convolution

To do:

1. First take the serial implementation of the above tasks.
2. Install and use parallel for loop using OpenMP. You can use OpenMP's 'shared' list, dictionary, and array to keep the shared dictionary. <https://www.admin-magazine.com/mobile/HPC/Articles/Pymp-OpenMP-like-Python-Programming>
3. Parallelize for loop and analyze the running time of algorithm with different number of threads.
4. If there is nested for loops, parallelize 1. Only outer loop 2. Only inner loop and 3. All loops. Finally, analyze the time taken by different parallelization settings.
5. Note: Avoid parallelization in case of any dependencies.
6. (Optional) Explore and try to implement parallel for loops using 'joblib' Python library which also supports parallel/multiprocessing. <https://joblib.readthedocs.io/en/latest/parallel.html>
7. (Optional) Compare your previously done data parallel versions of the above topics with the parallel for loops from OpenMP (and with joblib if possible)