**Part 1**

1. What are the basic steps (show all steps) in building a parallel program? Show at least one example.

The basic steps in building a parallel program are decomposition; assignment; orchestrating; and mapping. This drug design is a great example of parallel programming since the computer does decomposition through generating ligands, the assigning and orchestrating through the value of the ligand and lastly, mapping through finding the highest value ligand.

1. What is MapReduce?

MapReduce is a type of programming model, which has associate implementations for processing and generating big data sets. It does this through an algorithm that is parallel and distributed on a cluster.

1. What is a map and what is reduce?

Map is afunction that takes in a key or a value pair, so that it can generate intermediate pairs. Reduce is a function that takes all the intermediate values that have the same immediate key and merge them.

1. Why MapReduce?

We choose MapReduce because of its ability to write the big data using the parallel programming techniques to process and to cluster the huge amounts of data, in a relatively small amount of time.

1. Show an example for MapReduce.

The Drug design program is once again a great example of MapReduce. Because for the computer to run it, it needs to generate protein ligands, which is a huge data and through parallel programming we can achieve it in less time.

1. Explain in your own words how MapReduce model is executed?

Every time MapReduce is used it breaks the tasks it has to do. During the executing the processor calls on the MapReduce library, which uses the parallel distribution to break the tasks and then they are assigned to clusters. After this the processor receives its task, then the master will assign the tasks to the worker which then either assigns it to map or to reduce. If the task is assigned to map, the worker program performs the task using the shred data set. After all the tasks have been completed, the worker program writes all the answers to the immediate files, then the generated output is reduced by yet another worker program. After reducing the result is given to the master.

1. List and describe three examples that are expressed as MapReduce computations.

Word counter is a great example. For word finder we will be finding the unique words and the number of occurrences of those unique words. First, we divide the input into splits. This will distribute the work among all the map nodes.

Then, we tokenize the words in each of the mappers and give a hardcoded value to each of the words. Now, a list of key-value pair will be created where the key is nothing but the individual words and value is one. The mapping process remains the same on all the nodes. After the mapper phase, a partition process takes place where sorting and shuffling happen so that all the tuples with the same key are sent to the corresponding reducer. So, after the sorting and shuffling phase, each reducer will have a unique key and a list of values corresponding to that very key. Now, each Reducer counts the values which are present in that list of values. Finally, all the output key/value pairs are then collected and written in the output file.

1. When do we use OpenMP, MPI, and MapReduce (Hadoop), and why?

OpenMP is highly effective and used directive-based libraries.

MapReduce (Hadoop) is used when there is a large data, it breaks it up and does it task by task. Because of this breaking up process this is highly effective over large datas.

MPI is used as a distributed memory parallel model implementation it is mainly used for scientific applications.

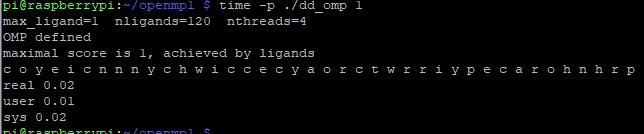
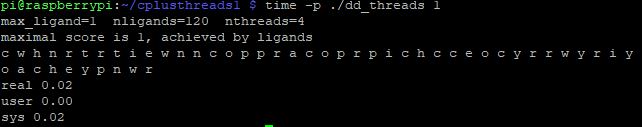
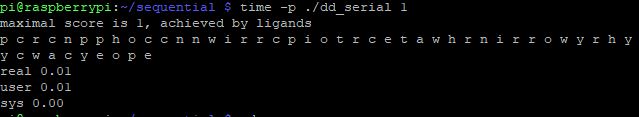
1. In your own words, explain what a Drug Design and DNA problem is in no more than 150 words.

DNA is the genetic code that contains the instructions for making proteins in all living things. The function of the protein is determined by its shape. To design a drug the programs needs to change the protein shape using the ligands it finds. In this program the computer is generating ligands to do the tasks of a particular protein. The programs computes a score/value for each ligand so that it can predict how well it will fit in the protein and change the shape accordingly. Then, the program identifies the highest value ligand and does the testing. In the drug design exemplar code, the line: “./drugdesign-static ***threads maxlen count,”*** shows the number of threads for the simultaneous and the maxlen is the maximum length of a ligand. Every ligand has a specific length and this is the score.

**Part 2**

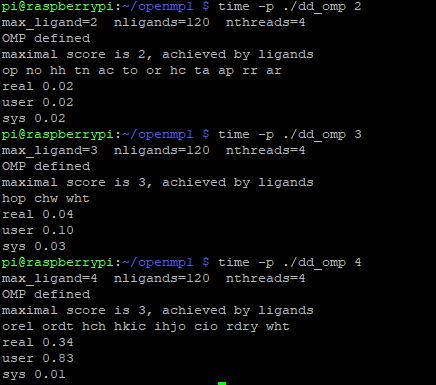
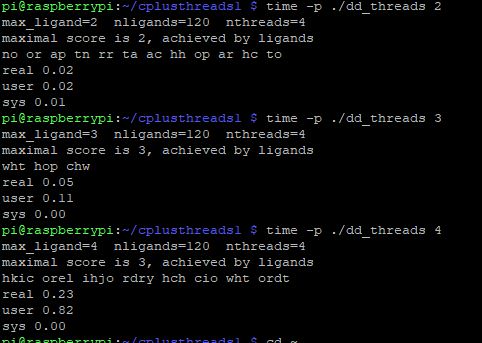
Run-time for 1:

|  |  |
| --- | --- |
| Implementation | Time 1 thread |
| dd\_serial | 0m0.01s |
| dd\_omp | 0m0.02s |
| dd\_threads | 0m0.02s |



Run-time for 2, 3, 4 on Omp and Threads

|  |  |  |  |
| --- | --- | --- | --- |
| Implementation | Time 2 threads | Time 3 threads | Time 4 threads |
| dd\_omp | 0.02s | 0.04s | 0.34s |
| dd\_thread | 0.02s | 0.05s | 0.23s |



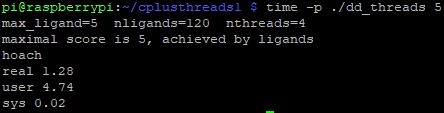
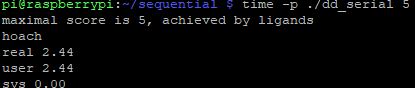
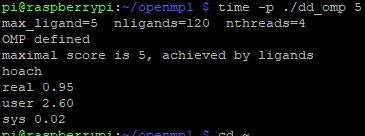
1. Which approach was the fastest?

For 1 thread the serial approach was the fastest. For 2 threads the runtime for omp and thread was the same; for 3 threads the omp approach was faster. For 4 threads the threads approach was the fastest.

1. Determine the number of lines in each file (use wc -l). How does the C++11 implementation compare to the OpenMP implementations?

Serial has 171 lines; Omp has 194 lines; and Threads has 208 lines. C++11 libraries are used for general purpose and only give the user certain amount of control. OpenMp on the other had is way more efficient than the C++ threads..

1. Increase the number of threads to 5 threads. What is the run time for each?



1. Increase the maximum ligand length to 7, and rerun each program. What is the run time for each?

