**Parallel Programming Skills**

**Part 1 (76p):**

* (56p) Read this paper “Introduction to Parallel Programming and MapReduce” and

answer the following questions:

* + (15p) What are the basic steps (show all steps) in building a parallel program? Show at least one example.

1. **Identify the set of tasks that can be executed concurrently and the partitions of data that can be processed concurrently**
   * **Fibonacci Function:**

**cannot be executed using parallelism because of dependency (each value is dependent on previous calculated value)**

1. **If the program and data can be executed using parallelism, develop a parallel solution**
   * **Master-Worker Solution**

**Calculating PI, where the master sends worker tasks, and workers return the results (how many points lies both in the circle and the square) to the master so the master can calculate PI**

* + **Fork-Join solution**
  + **Task Parallelism**
  + **Data Parallelism**
  + (5p) What is MapReduce?

**MapReduce is developed within Google as a tool for processing large amount of data (crawled documents or web request logs).**

**It is an abstraction that allows Google engineers to hid the details of parallelism, data distribution, load balancing, and fault tolerance while preforming calculations.**

* + (10p) What is map and what is reduce?

**The Map (developed by the user of the MapReduce library) takes an input pair and returns a set of key/value pairs. The library will then group all values associated with the same key and call the reduce function. The reduce (also developed by the user of the MapReduce library) function then merge the values (associated with the same key) together to form a smaller set of values.**

* + (5p) Why MapReduce?

**MapReduce helps reduce the amount of time needed to process large amount of data (by shrinking the data) in a reasonable time.**

* + (5p) Show an example for MapReduce.

**One of the most famous examples of MapReduce is counting how many times a set of words occurred in a number of documents. For example, say we are searching for how many times the words “map and “reduce” occurred in one hundred documents and we have 5 machines to do the job for us. We can have each machine take in 20 documents and look through the documents for how many times those words occur. The machines will perform calculations without communicating to each other in the map function. Furthermore, the words will be the key, and the number of times they appeared will be the value. Say machine A counted “map” 15 times and “reduce” 13 times in the first 20 documents; machine B counted “map” 11 times and “reduce” 9 times in the next 20 documents; machine C counted “map” 5 times and “reduce” 12 times in the next 20 documents; machine D counted “map” 1 times and “reduce” 3 times in the next 20 documents; machine E counted “map” 10 times and “reduce” 13 times in the last 20 documents. Next, these 5 machines will give their results to a machine (it can be one of the 5 machines or another machines). The machines receiving the results will add up all the values associated with the same key, and return the final result (42 count for “map and 50 count for “reduce”).**

* + (10p) Explain in your own words how MapReduce model is executed?

**There are four phases of MapReduce. First, we have Input Splits, where the input is being split up. Next, we have Mapping, where the data in each split is processed through a map function that produced key/value output. Then, we have Shuffling, which combines the results from the previous phase. Last, we have the Reducing phase, which merges the values from the previous phase to produce a single output.**

* + (6p) List and describe three examples that are expressed as MapReduce

computations.

**Count of URL Access Frequency:**

**Mapping function processes the web page request logs and returns <URL, 1>, and the reduce function merges all values of the same URL and returns <URL, total count> pair.**

**Reverse Web-Link Graph:**

**Mapping function return <target, source> pairs for each link to target URL found of “source” page, and the reduce function merges the list of all source URLs associated with given target URL and returns <target, list(source)>.**

**Term-Vector per Host:**

**Mapping function returns a <hostname, term vector> for each input document (hostname is the document’s URL), and the reduce function merges together the term values associated with the same hostname. It then discards infrequent terms and returns a final <hostname, term vector> pair.**

* (6p) When do we use OpenMP, MPI and, MapReduce (Hadoop), and why?
  + **OpenMP**

**We can use this model when we want to use shared memory parallelism in our code. It is an efficient directive-based library. Although it is neat and powerful, we must be careful when using it to avoid performance and correctness errors.**

* + **MPI**

**This is a distributed memory parallel model that is usually used in scientific programs, because it is tightly synchronous code and well balanced. This can also be used to develop pretty much any parallel code and can also be mixed with OpenMP (hybrid programming).**

* + **MapReduce (Hadoop)**

**We can use this model when we are dealing with large amount of data, and we want to apply some operation for each data element. The data is typically distributed using HDFS (Hadoop Distribution File System), a distributed file system designed to run on commodity hardware. The results can than eb reduced using the reduce function. When we have terabytes of data to ETL (Extract, transform, and load), it more efficient if we use this model than MPI, because of its fault tolerance feature (one of its key features).**

* (14p) In your own words, explain what a Drug Design and DNA problem is in no more than 150 words.

**After identifying a disease of interest’s protein, its three-dimensional structure can be found through experiments or molecular modeling computation. Then, a collection of ligands is tested against the protein, and computation is done to see how well the ligand binds with the protein in useful ways (tying up biologically active regions). The ligands will have a score based on their binding properties, and the ligands with high scores will be identified as good drug candidates.**