Gurjus Singh

MSDS 432 Foundations of Data Engineering

August 9th, 2020

Module 7– Reading Comprehension

1. **Describe the MapReduce paradigm in terms of Mappers and Reducers, and illustrate with an example (2 pt)**

The MapReduce paradigm is a like a Unix tool but is distributed across thousands of machines. For example, one MapReduce job is equivalent to one Unix process. Map Reduce takes an input and produces a sequential output and does not produce any side effects besides the output. It is a programming framework which helps with large datasets.

The way MapReduce works is it obtains the large dataset and breaks up the data set into records. The Mapper function in MapReduce gets the key and value from each record that has been created by MapReduce. Then it sorts the key-value pairs the it has created. MapReduce then calls its reducer function which will merge duplicate keys.

An example of MapReduce is Hadoop's MapReduce. The Mapper and Reducer are written in Java while in MongoDBs MapReduce it is written in JavaScript. In Hadoop's implementation, there are separate MapReduce tasks for each partition or shard. The MapReduce framework has to copy the code as the code will not be on the machine in the beginning. The code if in Java is stores in a JAR file.

2. **Describe some ways (at least 3) in which batch jobs achieve good performance while being easy to maintain (Philosophy of batch process outputs) (2 pt)**

   Batch Jobs achieve good performance while being easy to maintain because the first thing you can do is if your code has a bug or is incorrect, you can easily go back to a previous version of the code and run the job. With this method of good performance, it is easier for feature development to be done more quickly. Another way that how batch jobs can achieve good performance is for example when using MapReduce if a task fails such as map task or reduce task, the process will automatically reschedule and run again.

   The third way batch jobs achieve good performance while being easy to maintain by allowing files to be used for different jobs. This can help monitor jobs to calculate metrics and help see if a job output has the expected characteristics. Batch jobs such as MapReduce also help separate logic from wiring. What this means is that concerns on what to implement can be divided and this can also allow reusing of code. In this context teams can divide the work. Batch jobs can also make inputs immutable while also preventing side effects in the output.

3. **Choose one workflow scheduler that is mentioned in the reading and provide a description of the tool (1 pt).**

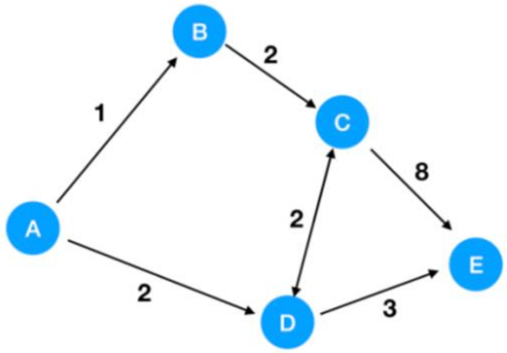
One workflow scheduler that is mentioned in the reading is called Oozie. This scheduler helps with Apache Hadoop jobs. This type of workflow involves a type of graph of actions to do such as Directed Acyclical Graphs meaning they do not have cycles. Oozie workflow jobs that are start because of time and data available are called Oozie Coordinator Jobs. Oozie also supports other Hadoop stacks. It is a scalable, extensible and integrated system.

4. **What are some challenges MapReduce has with Graph-Like Data Models and what are some solutions? (2 pt)**

   Some challenges MapReduce has with Graph-Like Models are firstly it MapReduce cannot use the repeating until done philosophy of graph algorithms since it only performs a single pass over the data. The alternative to this is to make the algorithm iterative style. The iterative style involves an external scheduler, which starts step one by running a batch process. One this completes, the scheduler checks if the iterative style has finished if not it repeats the batch process. The thing about this process is it is inefficient because MapReduce cannot account for the iterative style.

   A solution to MapReduce is to use the bulk synchronous parallel model of computation which is also known as the Pregel Model. The way the Pregel Model works is one vertex sends a message to another vertex and the messages are sent along the edges. A function is called for each vertex, and the all the messages are passed to the vertex function. The vertex remembers its state in memory from one iteration to the next. Some things about the Pregel Model that make it efficient is that messages can be batched and there is less waiting time with communication. Fault tolerance is achieved by checkpointing the state of the vertices at the end of each iteration of the model.

5. **Using the below graph, find the optimal path from A to E using Dijkstra's Algorithm by describing each step.** (3 pt)



Dijkstra’s algorithm is called the fastest path algorithm that accounts for edge weights. It differs from Breadth-First Search which searches for the shortest path. The algorithm works by first finding the cheapest node. Then you update the cost of the neighbor nodes from the cheapest node. You repeat this and add up the weights or path.

Dijkstra’s algorithm has terminology such as weighted which means that it has number associated with its edges. Edges are how nodes such as cities are connected. Dijkstra’s algorithm is used in maps such as Google Maps. For Dijkstra’s algorithm, you cannot have cycles as it will probably be the longest path.

In the graph you first start at Node A looking to get to Node E. You see you can get to Node B in the least amount of time which is 1. So you go there and then check its neighbor’s and update their weights which is C which has a weight 2. Next you check the second least node from A which is D and has weight of 2 update D’s neighbors and its neighbors are C and E. C has a weight of 2. Since you know that D is directly connected to E as well and the weight is 3, it will on take a total 5 to get from A to E going through D.