1. What is Hadoop Distributed Cache and how is it used in this program?

Hadoop Distributed Cache is a facility that caches the files onto HDFS and distribute the file across the compute nodes. In this program the fa files will be uploaded to HDFS and when the nodes start the computation, they generate <filename, filepath> pairs for each file.

1. Write the two lines that put and get values from Distributed cache. Also include the method and class information.

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.filecache.DistributedCache;

Configuration conf = context.getConfiguration();

Path[] local = DistributedCache.getLocalCacheArchives(conf);

1. In previous projects we used Hadoop’s TextInputFormat to feed in the file splits line by line to map tasks. In this program, however, we want to feed in a whole file to a single map task. What is the technique used to achieve this? Also, briefly explain what are the key and value pairs you receive as input to a map task and what methods are responsible for producing these pairs?

We can still use <file\_name, file\_path> pair to feed a whole file to a single map task. DataInputFormat will split the data and FileRecordReader will read the split data and generate <key, value> pair.

1. Do you think this particular implementation will work if the input files are larger than the default HDFS block size? Briefly explain why. [Hint: you can test what will happen by concatenating the same input file multiple times to create a larger input file in the resources/blast\_input folder]

Yes. The implementation will still work. Since in hadoop it is easy to split the input data. So if the input files are larger than the default block size, we can use DataInputFormat class to split the files into multiple chunks, and creates a <key, value> pair for each chunk.

1. If you wanted to extend this program such that all output files will be concatenated into a single file, what key and value pairs would you need to emit from the map task? Also, how would you use these in the reduce that you would need to add?

To accomplish this, the map function should have the same key for all the pairs. And the absolute path on HDFS for each pair is still different. So when we call the reduce function, for the pairs that have the same key, we can create a new file that aggregates all the content of the files.