**PROJECT 3: REPORT**

**[Maaz Syed (0003544185) & Venkat Sambandhan (0003544188)]**

1. DistributedCache is a facility provided by the Map-Reduce framework to cache files (text, archives, jars etc.) needed by applications. Applications specify the files, via urls (hdfs:// or http://) to be cached via the [JobConf](https://hadoop.apache.org/docs/r1.2.1/api/org/apache/hadoop/mapred/JobConf.html" \o "class in org.apache.hadoop.mapred). The DistributedCache assumes that the files specified via hdfs:// urls are already present on the [FileSystem](https://hadoop.apache.org/docs/r1.2.1/api/org/apache/hadoop/fs/FileSystem.html" \o "class in org.apache.hadoop.fs) at the path specified by the url.

The framework will copy the necessary files on to the slave node before any tasks for the job are executed on that node. Its efficiency stems from the fact that the files are only copied once per job and the ability to cache archives which are un-archived on the slaves. DistributedCache can be used to distribute simple, read-only data/text files and/or more complex types such as archives, jars etc.

Here the blast binary program is sent to the distributed cache for running the blast algorithm against the input **Fasta** files. Also, the DistributedCache is used for sharing files across multiple nodes and inputting multiple files.

1. The two lines that put and get values from Distributed cache are:

**this.localDB=local[0].toUri().getPath() + File.separator + conf.get(DataAnalysis.DB\_ARCHIVE) + File.separator + conf.get(DataAnalysis.DB\_NAME);**

**this.localBlastProgram=local[0].toUri().getPath();**

**Put file on to distributed cache:**  
The class used is DistributedCache and the method is addCacheArchive (URI uri, Configuration conf), it adds archives to be localized to the conf.  
DistributedCache.addCacheArchive(new URI(programDir), jc);

**Get file from distributed cache:**  
The class used is DistributedCache and the method is getLocalCacheArchive (Configuration conf), return the path array of the localized files.  
Path[] local = DistributedCache.getLocalCacheArchives(conf);

1. By the use of **DataFileInputFormat**, Hadoop reads the data from input file, to generate <key, value> pairs for the map function. The **DataFileInputFormat.java** class generates key-value pair in the form

**<Filename, File Path on HDFS>**

**Key:** Filename

**Value:** File Path

**Method:** getCurrentKey(), getCurrentValue()

1. **Error Message:**

Add Distributed Cache in 0.0010 seconds

16/02/28 16:00:57 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool for the same.

16/02/28 16:00:58 INFO input.FileInputFormat: Total input paths to process : 4

16/02/28 16:00:58 INFO mapred.JobClient: Running job: job\_201602281333\_0006

16/02/28 16:00:59 INFO mapred.JobClient:  map 0% reduce 0%

16/02/28 16:01:36 INFO mapred.JobClient:  map 25% reduce 0%

16/02/28 16:11:36 INFO mapred.JobClient:  map 0% reduce 0%

16/02/28 16:11:37 INFO mapred.JobClient: Task Id : **attempt\_201602281333\_0006\_m\_000000\_0, Status : FAILED**

**Task attempt\_201602281333\_0006\_m\_000000\_0 failed to report status for 600 seconds. Killing!**

attempt\_201602281333\_0006\_m\_000000\_0: next keyvalue : hdfs://[127.0.0.1:9000/user/summer/HDFS\_blast\_input/celllines\_1.fa](http://127.0.0.1:9000/user/summer/HDFS_blast_input/celllines_1.fa) :false

attempt\_201602281333\_0006\_m\_000000\_0: in getProgress : true

attempt\_201602281333\_0006\_m\_000000\_0: in current key hdfs://[127.0.0.1:9000/user/summer/HDFS\_blast\_input/celllines\_1.fa](http://127.0.0.1:9000/user/summer/HDFS_blast_input/celllines_1.fa) :true

attempt\_201602281333\_0006\_m\_000000\_0:  get Current Value hdfs://[127.0.0.1:9000/user/summer/HDFS\_blast\_input/celllines\_1.fa](http://127.0.0.1:9000/user/summer/HDFS_blast_input/celllines_1.fa) :true

**attempt\_201602281333\_0006\_m\_000000\_0: the map key : celllines\_1.fa**

**attempt\_201602281333\_0006\_m\_000000\_0: the value path : hdfs://**[**127.0.0.1:9000/user/summer/HDFS\_blast\_input/celllines\_1.fa**](http://127.0.0.1:9000/user/summer/HDFS_blast_input/celllines_1.fa)

attempt\_201602281333\_0006\_m\_000000\_0: Local DB : /tmp/summer/local/taskTracker/distcache/1505533767642493217\_1581722369\_699338907/127.0.0.1BlastProgramAndDB.tar.gz/db/nr

attempt\_201602281333\_0006\_m\_000000\_0: Before running the executable Finished in 2.421 seconds

attempt\_201602281333\_0006\_m\_000000\_0: Program Finished in 603.663 seconds

16/02/28 16:11:37 INFO mapred.JobClient: Task Id : **attempt\_201602281333\_0006\_m\_000001\_0, Status : FAILED**

**Task attempt\_201602281333\_0006\_m\_000001\_0 failed to report status for 600 seconds. Killing!**

From the console messages we can see that the FileRecordReader sends the whole 80MB filenames as the key and the file paths as the value to the mapping functions. (Example: <key, value> 🡺 <**celllines\_1.fa, hdfs://**[**127.0.0.1:9000/user/summer/HDFS\_blast\_input/celllines\_1.fa**](http://127.0.0.1:9000/user/summer/HDFS_blast_input/celllines_1.fa)**>** )

The FileRecordReader does not take care of splitting files greater than the HDFS block size(64MB) correctly which leads to the map task being terminated as the map task fails to read input splits and the time to report status exceeds the time limit (600 seconds in our configuration).

1. If you wanted to extend this program such that all output files will be concatenated into a single file:

* We can use any common key and the value will be the file paths.
* At the mapper we can emit **key/value** pairs with a common**key**which we can use to aggregate all the output file paths (**value**) at the reducer.
* At the reducer, we concatenate the values with similar keys. This way we will be able to combine all the output files into a single file.

**References:**

* [http://blog.cloudera.com/blog/2009/05/10-mapreduce-tips/](http://www.google.com/url?q=http%3A%2F%2Fblog.cloudera.com%2Fblog%2F2009%2F05%2F10-mapreduce-tips%2F&sa=D&sntz=1&usg=AFQjCNE37hL1H6Ix2ahG6YVDeEwNYtPYvg)
* [http://stackoverflow.com/questions/5864589/how-to-fix-task-attempt-201104251139-0295-r-000006-0-failed-to-report-status-fo](http://www.google.com/url?q=http%3A%2F%2Fstackoverflow.com%2Fquestions%2F5864589%2Fhow-to-fix-task-attempt-201104251139-0295-r-000006-0-failed-to-report-status-fo&sa=D&sntz=1&usg=AFQjCNHDy4_g-WdQpZGtzALUwjjCimboVQ)