Hadoop BLAST

Sohil Jain Ashutosh Bhargave

[**sohjain@iu.edu**](mailto:sohjain@iu.edu)[**ashubhar@indiana.edu**](mailto:ashubhar@indiana.edu)

**Goal**

Now you are going to blend Hadoop programs applications by implementing a parallel version of [BLAST (Basic Local Alignment Search Tool)](http://blast.ncbi.nlm.nih.gov/Blast.cgi) using the programming interfaces of the Hadoop MapReduce framework. Note that this application is written in "Map-Only" fashion, which means no reduce code is necessary.

**Technical Report:**

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

**Answer 1]**

Distributed cache is a provision made by Mapreduce framework for caching. Distributed cache expects the the application to

Applications specify the file to be cached

Via urls ( hfds:// or http://

Framework copies necessary files on slave.

|  |  |
| --- | --- |
| Type of Files distributed by Framework | simple, read-only data/text files and/or more complex types such as archives, jars etc |

Distribute application-specific large, read-only files efficiently. DistributedCache tracks modification timestamps of the cache files. Clearly the cache files should not be modified by the application or externally while the job is executing.

In this program we take the prepackaged BlastProgramandDB.tar.gz from Hadoop’s Distributed Cache, and then execute BLAST binary as java external process with the assigned FASTA file

**Q 2].** Write the two lines that put and get values from Distributed cache. Also include the method and class information

Answer]

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();

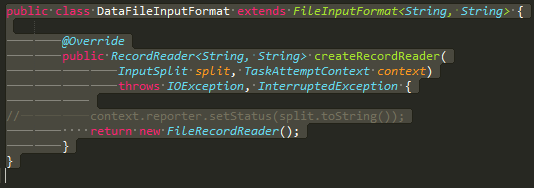
https://ssl.gstatic.com/ui/v1/icons/mail/images/cleardot.gif

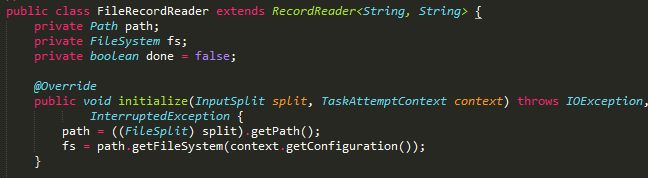
**Q 3].** In previous projects we used Hadoop’s TextInputFormat to feed in the file splits line by line to map tasks. In this program, however, we want to feed in a whole file to a single map task. What is the technique used to achieve this? Also, briefly explain what are the key and value pairs you receive as input to a map task and what methods are responsible for producing these pairs?

**Answer]**

In this program whole file is fed into single Mapper by first giving the file path on hdfs. Then the mapper copies the assigned Fasta files to local disk, by looking up the file from hdfs and generating an absolute file path. Input is of the type: a customized InputFormat DataFileInputFormat which emits key : FileName and value: FilePath in hdfs.

We have written our own input file format DataFileInputFormat to read the input. The technique we have used is to use RecordReader

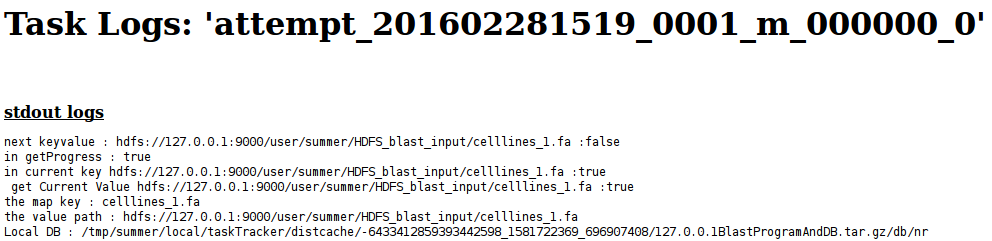




In this program whole file is fed into single Mapper by first giving the file path on hdfs. Then the mapper copies the assigned Fasta files to local disk, by looking up the file from hdfs and generating an absolute file path

The (key,value) pairs are the (filename, filepath).

We have implemented this in our custom methods as seen in above image of FileRecordReader class.



**Q 4].** 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]

**Answer 4]**

The default cache limit is 10gb, as we have not changed this explicitly in our mapred-site.xml we can upload files larger than block sizesource.

Yes, this implementation will work if the input files are larger than the default HDFS block size. Because FileInputFormat turn the larger files (larger than an HDFS block) into splits. The split size is normally the size of an HDFS block.

FileInputFormat turn the larger files (larger than an HDFS block) into splits generally equal to HDFS block.

**Q 5].** 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?

**Answer 5].**

Map :-   
Key : FileName  
Value : FilePath of the processed output of the map task  
   
Then Reducer will concatenate all the files produced by mapper to generate one concatenated file.

**Compile and run your code**

Use the same one-click script compileAndExecHadoopBlast.sh as in prior homework. Standard error messages such as “compile errors, execution errors, etc.” will be redirected on the screen. Follow the same debugging format.

$ cd /root/MoocHomeworks/HadoopBlast/

# usage: ./compileAndExecHadoopBlast.sh

$ ./compileAndExecHadoopBlast.sh

**Result**

The result is generated as 4 output FASTA files with .fa extension