COMS W4121: Compute Systems for Data Science

Assignment #2: MapReduce

Due March 9th, 2015 at 11:55PM

1. Overview

In this assignment, you will perform various data processing tasks using Hadoop Streaming. The dataset you will be dealing with is the latest English Wikipedia database dump in XML format, which is around 50GB in size.

2. Hadoop Streaming

Hadoop streaming is a utility that comes with the Hadoop distribution. The utility allows you to create and run Map/Reduce jobs with any executable or script as the mapper and/or the reducer. A typical job launching command looks like the following:

```
hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.6.0.j
ar \
    -input myInputDirectory \
    -output myOutputDirectory \
    -mapper myMapper.py \
    -reducer myReducer.py \
    -file myMapper.py \
    -file myReducer.py
```

Mapper and reducer scripts can be written in any language that supports standard streams. We will use Python in this assignment. Read this documentation for more details on how to use Hadoop Streaming.

2. Dataset

The dataset you will be dealing with in this assignment is the latest English Wikipedia database dump in XML format, which is around 50GB in size.

XML stands for EXtensible Markup Language, which is widely used for describing structured data. A sample Wikipedia page in XML format looks like the following:

```
  <page>
    <title>Page_Title</title>
    <ns>0</ns>
    <id>8378</id>
    <revision>
      <id>636341643</id>
      <parentid>636341378</parentid>
      <timestamp>2014-12-02T17:43:08Z</timestamp>
      <contributor>
        <username>KillerChihuahua</username>
        <id>82889</id>
      </contributor>
      <minor />
      <comment>Reverted edits by [[Special:Contribs/177.39.96.180|177.39.96.
180]] ([[User talk:177.39.96.180|talk]]) to last version by SciberDoc</comme
nt>
      <model>wikitext</model>
      <format>text/x-wiki</format>
      <text xml:space="preserve">
        Text contents of the page (Omitted)
      </text>
      <sha1>3z1vfprkns4tdxd7wzjgeu7nkal8yzp</sha1>
    </revision>
  </page>
```

By default, each Hadoop Streaming mapper gets one block of input data, which is 128MB of data based on the current settings of the cluster. This may cause serious problems in processing highly structured input such as XML.

You can see from the above example that each pair of tag encloses some data or some nested tags. For exmaple, all tags from `<title>` to `</revision>` are enclosed in the `<page>...</page>` tag pair in the above sample XML file. Given a large XML file, It would be nice if Hadoop knows how to cut the XML format into chunks while keeping tag pairs in the same chunk. Yet Hadoop processes text file on a line-by-line basis unless you customize the InputFormat module. One may also split the database into one page per XML file to avoid this problem, but there are close to 5 million pages in the database, and the smallest read/write size in HDFS is 128MB, which means that one needs at least half petabyte of storage space to store these files.

There are some existing solutions for processing large XML files, such as XMLInputFormat in Mahout, a Hadoop-based machine learning library. You certainly may use these solutions if you want to, but there is a simple solution to this problem.

Wikipedia are organized in individual pages, and the longest plaintext Wikipedia page is about

700KB in size. This means that we can preprocess the dataset by slicing it into smaller XML files that are very close to 128MB (1 hdfs block) in size while keeping the same page in the same block. This step is already done for you because we don't want everyone to add >50GB of file into the file system. You can take a look at the script in the assignment package if interested (**split_wiki_dump.py**). The XML chunks are located under **hdfs:/datasets/en_wikipedia_dump/xml_chunks**.

---

3. Programming: Count 3-grams in a text file (30 points)

*This task is a "Hello World" task to get you started with Hadoop Streaming.

In natural language processing, an n-gram is a contiguous sequence of n items from a given sequence of text. For example, the set of all 3-grams of the sentence `a fox jumps over the dog` are:

```
a fox jumps
fox jumps over
jumps over the
over the dog
```

Task:

Write a Hadoop Streaming program in Python that counts all 3-grams in the plaintext version of *Pride and Prejudice*. The input file is located under **hdfs:/datasets/test/pg1342.txt**

Requirements:

1. Count words that contains only [a-z] and [A-Z], i.e., English letters, and hyphens, e.g. `good-humored` .
2. All parsing should be case-insensitive. The output should be in lowercase.
3. Strip all punctuations and non-English letter characters that prefix or postfix a word. For example, `A fox jumps.` should be parsed into a 3-gram: `a, fox, jumps`

Ideally when processing English text, a contiguous n-gram sequence should stop at the end of a sentence. However, sentence segmentation is a surprisingly difficult task mainly because a '.' character does not always mean the end of a sentence. Therefore to simplify the task, a contiguous 3-gram sequence should stop only at a "newline" character in this assignment.

What to submit:

Write out the 10 most frequent 3-grams and the number of occurance. Save to a textfile named **text_3gram.txt** and submit with the mapper (**wc_mapper.py**) script, the reducer (**wc_reducer.py**)

script, and any other scripts that are needed. In addition, you may want to have a **Readme.txt** file that briefly describe how to use your program.

The output format should be:

```
word1 word2 word3 num_occurance
word1 word2 word3 num_occurance
word1 word2 word3 num_occurance
....
```

Hint:

1. You may want to take a look at the sample mapper (mapper.py) and reducer (reducer.py) in the assignment package.
2. You may need more than one set of mapper and reducer to get the top ten 3-grams. Think about how to solve the top-K problem using MapReduce.

---

4. Programming: Count 5-grams in Wikipedia (70 points)

Task:

Write a Hadoop Streaming program in Python that counts 5-grams in the English Wikipedia database dump using Hadoop Streaming.

Use the data chunks located under **hdfs:/datasets/en_wikipedia_dump/xml_chunks** as input.

DO NOT copy the entire database to your home directory. You may, however, copy one or two chunks to your home directory in hdfs or in the local file system for testing purposes.

Requirements:

Same requirements from section 3 and:

1. Count only the words that are enclosed by a `<text> ... <\text>` tag pair.
2. In this assignment, a page is defined as text that are enclosed by a `<page> ... <\page>` tag pair.
3. Wikipedia contains fair amount of unicode characters. They are not English letters so apply rule #1.
4. Use `unicode.encode('ascii', 'ignore')` to get rid of unicodes in titles.

What to submit:

Output the **5** most frequent 5-grams, the number of occurrance, and the titles and the page ids of all

associated pages. The output format should be:

```
word1 word2 word3 word4 word5 num_occurance
page_title_1 page_id_1
page_title_2 page_id_2
...
page_title_n page_id_n
word1 word2 word3 word4 word5 num_occurance
....
```

Save the result to a text file named **wiki_5gram.txt** and submit it with the mapper script (**wiki_mapper.py**), the reducer script (**wiki_reducer.py**), and any other scripts that are needed. Again, give instructions on how to use your program in a **Readme.txt** file

Hint:

1. You can use sample chunks located under **hdfs:/datasets/en_wikipedia_dump/sample_data** to debug your program. Alternatively, you can copy a chunk of data to your local file system (`hdfs dfs -get`) and test your program locally using UNIX Pipeline.
2. Again, you may need more than one set of mapper and reducer to get the desired output.
3. When running your program on a large dataset like Wikipedia dump, there might not be enough reducer to handle the output of mappers. Increase the number of reducer by adding `-D mapreduce.job.reduces=n` as part of the launch command. A recommended value for n is 8.

---

5. Hints and tips

Debug Hadoop Streaming Scripts

Debugging Hadoop program can be very painful because of the obscure error messages. A very convenient way of debugging the mapper and reducer scripts is to use the UNIX "Pipeline" utility. You can pipe a input file (abc.txt) directly to a mapper (mapper.py) by:

```
cat abc.txt | mapper.py
```

"|" tells the shell to connect the standard output of the left program to the standard input of the right program. **cat** is a Unix utility that turns a text file into a STDOUT standard stream. Therefore this command simply turns abc.txt into a stream and launch the reducer.py with the stream as STDIN.

Similarly, you can connect the output of the mapper script with the input of a reducer program by "|", i.e.,

```
cat abc.txt | mapper.py | reducer.py
```

This way you can debug the scripts without launching a Hadoop job.

Text processing with Python

Some useful functions and classes are:

```
strip()
isalpha()
lower()
translate()
string
string.punctuation
```

You may use The Element Tree for parsing XML files.

Extra Credit (25 points)

Perform the task described in section 4 with **Pig**. Same requirements apply.