Programming Assignment 3

Staging and Analytics of Wikipedia Edit History Using Hadoop, HBase, and the Hive Query Language

Due: April 18th, 2017 By 5:00PM
Submission: via Canvas, individual submission

Objectives
The objectives of this programming assignment are to enable you to gain experience with:

- Automated data staging using Hadoop MapReduce
- Data Import from HDFS into HBase Table
- Performing interactive analytics using the Hive Query Language

1. Introduction

The goal of programming assignment 3 is to stage data in HBase, and perform interactive analytics using Hive Query Language(HQL).

To achieve the above goal, you should demonstrate:

- Installing HBase and Hive (use machines in CSB-120)   (section 2)
- Writing a MapReduce program to create data ready to be staged into HBase (section 3)
- Building a data structure in HBase   (section 4)
- Performing analytics using command line queries over HBase tables using Hive (section 5)

2. Installation of HBase in the CSB-120 cluster

You will first need to install and configure both HBase and Hive on top of your Hadoop cluster. Apache HBase is an open-source implementation of Google's BigTable that allows you efficient access to portions of large-scale structured data. The installation guide for HBase and Hive is available at
http://www.cs.colostate.edu/~cs435/Assignments.html .

Please note that you should have an up-and-running Hadoop cluster in order to start the above two services.

3. Writing a MapReduce program to Stage Data into HBase

3.1 Download Dataset

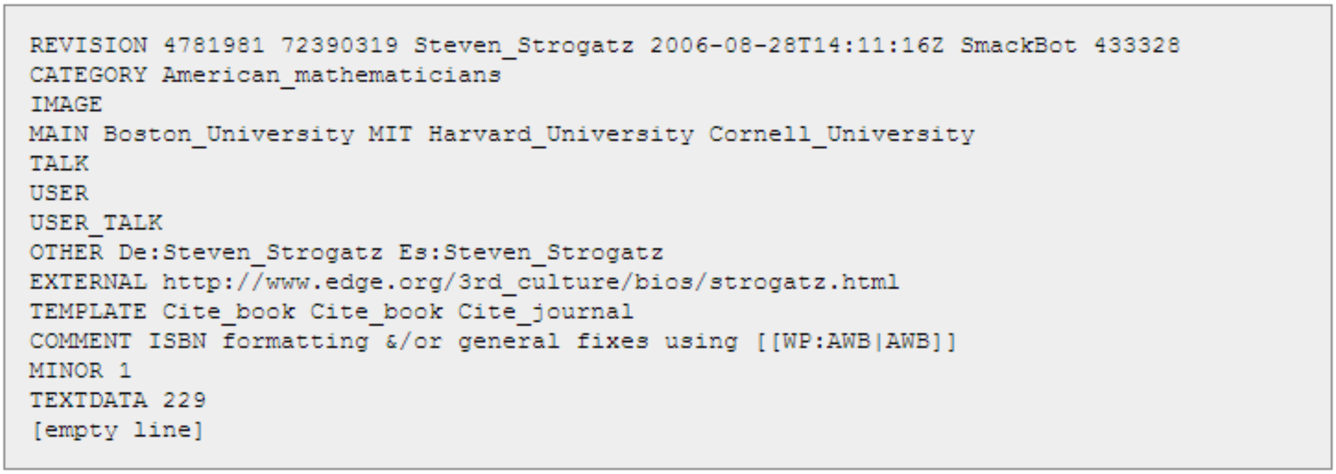
Before you download the dataset, please go to the directory where you have enough space. The data file will be about 1.2 GB after extraction. You should stage this dataset into your HDFS.

```
wget http://www.cs.colostate.edu/~cs435/datafiles/PA3/wikiData.tar

tar –xvf wikiData.tar
```

3.2 Components of the Dataset

The dataset you will be working on is a subset of the Wikipedia edit dump history from the first day of that service up until 2008-01-03. The dataset contains a collection of records where each record looks like the sample record listed below:



```
REVISION 4781981 72390319 Steven_Strogatz 2006-08-28T14:11:16Z SmackBot 433328
CATEGORY American_mathematicians
IMAGE
MAIN Boston_University MIT Harvard_University Cornell_University
TALK
USER
USER_TALK
OTHER De:Steven_Strogatz Es:Steven_Strogatz
EXTERNAL http://www.edge.org/3rd_culture/bios/strogatz.html
TEMPLATE Cite_book Cite_book Cite_journal
COMMENT ISBN formatting &/or general fixes using [[WP:AWB|AWB]]
MINOR 1
TEXTDATA 229
[empty line]
```

The elements included in the record are:

- REVISION article_id rev_id article_title timestamp [ip:]username user_id
- CATEGORY list of categories
- IMAGE list of images (each listed as many times as it occurs)

- MAIN through OTHER cross-references to pages in other namespaces
- EXTERNAL hyperlinks to pages outside Wikipedia
- TEMPLATE list of all templates (each listed as many times as it occurs)
- COMMENT contains the comments as entered by the author
- MINOR whether the edit was marked as minor by the author
- TEXTDATA word count of revision's plain text

3.3 Parsing Dataset

Once you have loaded the dataset into your HDFS, you should run a Map-Reduce job to convert the format of this dataset into the format that is suitable for HBase bulk loading. HBase supports the CSV file format for bulk loading. As a part of this process, you should write your `WikiHBaseInputFormat` that provides suitable `InputSplit` and `RecordReader` for this dataset. (See section 3.4).

3.4 Creating a customized Input Format

As described in the section 3b, each record contains 13 lines of text data separated by newline characters. Therefore, the object created by Hadoop's RecordReader does not encompass complete contents of a record. You are required to write your own WikiHBaseInputFormat to generate correct inputs for your mapper. Your mapper should parse the record and generate a string that is compatible with the CSV format.

3.5 Bulk Import

Now, you should load your data (formatted as CSV) into your HBase table. HBase provides a bulk loading feature that allows you to import in your data directly from HDFS into an HBase table. You may use the `importtsv` utility in HBase to do so.

4. Building a Data Structure in HBase

Before you import data into an HBase table, you are required to create a table (structure of the table is up to you) by using an HBase Shell or through client-side APIs.

You will then import this dataset into a table in HBase.

5. Performing analytics over data stored in HBase

To perform interactive analytics over the data stored in your HBase, use the Hibernate Query Language that Hive provides, whose syntax is very similar to SQL.

Your command line queries (using HQL) should provide accurate results. You may combine outputs from multiple queries to get your results:

Task 1: What is the top-10 list of most frequently edited pages (consider only edit counts)?
Task 2: Who are the top-10 users that are performing edits (consider only edit counts)?
Task 3: What is the percentage of pages that have had no edits since they were created?
Task 4: What percentage of pages have had more than 30 edits since they were created?
Task 5: What is the top-10 list of most frequently edited pages for the year 2008 (consider only edit counts)?

6. Submission

This assignment requires an individual submission.

For this assignment, you are required to submit the following:

1. Code as well as jar for your Map-Reduce program
2. A text file containing the following:
   - The create command for your HBase table.
   - The sequence of HQL queries you ran for each of the analytic tasks along with the outputs printed out to console.

6. Grading

Each submission will be graded based on a demonstration of your software. During the demonstration, you should present:

Part 1: A functional HBase and Hive installed on top of your Hadoop Cluster (use machines in CSB-120) [1 point]
Part 2: Building a data structure in HBase [1 point]
Part 3: Writing a MapReduce program to stage data into HBase [4 points]
Part 4: Perform the specified analytic tasks [4 points]

Demos will include a short interview about your software design and implementation details. Each question and answer will count toward your score. This assignment will account for 10% of your final course grade.