DBMS Models and implementation (Section 001)
Instructor: Sharma Chakravarthy
Project 3: Map/Reduce implementation

Made available on: 11/5/2015
Due on: 12/9/2015 (11:55Pm) Last day of classes; cannot be extended!
Submit by: Blackboard (1 zipped folder containing all the files/sub-folders) https://elearn.uta.edu/
Weight: 15% of total
Total Points: 100

One of the advantages of cloud computing is its ability to deal with very large data sets and still have a reasonable response time. Typically, the map/reduce paradigm is used for these types of problems in contrast to the RDBMS approach for storing, managing, and manipulating data. Hadoop is a widely used open source map/reduce platform. Hadoop Map/Reduce is a software framework for writing applications which process vast amounts of data in parallel on large clusters. In this project, you will use the weather dataset given to you and develop a program to compute averages using Hadoop map/reduce paradigm to analyze the weather data further. Please use the following links for a better understanding of Hadoop and Map/Reduce.

http://hadoop.apache.org/docs/r0.18.3/mapred_tutorial.html

1. **Dataset: consists of weather data gathered on an hourly basis.**
   i. Dataset size: 1.3 GB (can be downloaded using the link given below)
   ii. Spatial coverage: Texas state only
   iii. Temporal coverage: 5 years from 2006 to 2010
   iv. Data attributes:
   Data format has been given in the following table. First record is the header record. All 9's in a field (e.g., 9999.99 for DewP) indicates no report or insufficient data.
   The fields are separated by one or more whitespaces.

Field	Description
STN	Station number (WMO/DATSAV3 number) for the location.
WBAN	WBAN number where applicable--this is the historical "Weather Bureau Air Force Navy" number - with WBAN being the acronym.
yearModa_hr	The year(first 4 digits), The month(next 2 digits) and day(the next 2 digits) and hour(0 to 23).
Temp	Mean Temperature of that hour in degrees Fahrenheit to tenths. Missing = 9999.9
DewP	Mean dew point for that hour in degrees Fahrenheit to tenths. Missing = 9999.9
Count	Number of observations used in calculating dew point
SLP	Sea level pressure for that hour in millibars to tenths. Missing =

	9999.9
Count	Number of observations used in calculating sea level pressure
STP	Mean Station pressure for that hour in millibars to tenths. Missing = 9999.9
Count	Number of observations used in calculating Station pressure
Visib	Mean Visibility for that hour in miles to tenths. Missing = 999.9
Count	Number of observations used in calculating Visibility
WDSP	Mean wind speed for the hour in knots to tenths. Missing = 999.9
Count	Number of observations used in calculating Mean wind speed
MXSDP	Maximum sustained wind speed reported for that hour in knots to tenths. Missing = 999.9
Gust	Maximum wind gust reported for that hour in knots to tenths. Missing = 999.9
PRCP	A = 1 report of 6-hour precipitation amount. B = Summation of 2 reports of 6-hour precipitation amount. C = Summation of 3 reports of 6-hour precipitation amount. D = Summation of 4 reports of 6-hour precipitation amount. E = 1 report of 12-hour precipitation amount. F = Summation of 2 reports of 12-hour precipitation amount. G = 1 report of 24-hour precipitation amount. H = Station reported '0' as the amount for the day (eg, from 6-hour reports), but also reported at least one occurrence of precipitation in hourly, observations--this could indicate a trace occurred, but should be considered as incomplete data for the day. I = Station did not report any precip data for the day and did not report any occurrences of precipitation in its hourly observations--it's still possible that precip occurred but was not reported.
SNDP	Snow depth in inches to tenths--last report for the day if reported more than once. Missing = 999.9
FRSHIFT	Indicators (1 = yes, 0 = no/not reported) for the occurrence during the day of: Fog ('F' - 1st digit). Rain or Drizzle ('R' - 2nd digit). Snow or Ice Pellets ('S' - 3rd digit). Hail ('H' - 4th digit). Thunder ('T' - 5th digit). Tornado or Funnel Cloud ('T' - 6th digit).

A few lines of data has been provided as an example of datasets.

690190 13910 20060201_0 51.75 33.0 24 1006.3 24 943.9 24 15.0 24 10.7 24 22.0 28.9 0.00I 999.9 000000

```
690190 13910  20060201_1  54.74    33.0 24  1006.3 24   943.9 24   15.0 24   10.7
24   22.0   28.9    0.00I 999.9  000000
690190 13910  20060201_2  50.59    33.0 24  1006.3 24   943.9 24   15.0 24   10.7
24   22.0   28.9    0.00I 999.9  000000
690190 13910  20060201_3  51.67    33.0 24  1006.3 24   943.9 24   15.0 24   10.7
24   22.0   28.9    0.00I 999.9  000000
```

The above data set can be downloaded from (right click on the link below). Contains 5 files each about 250MB, one for each year from 2006 to 2010.

http://itlab.uta.edu/downloads/cse5331_Project3_datasets.zip (1+GB)

2. Problem Specification

Problem Statement: Analyzing the weather data in order to find the stations with similar weather conditions for each month over the period of 1 year in Texas. In order to do similarity computation, one needs to convert the given weather data to a format which makes it easy to cluster similar temperature for each month over the period of 1 year in Texas. In this project, you will do the first step needed for computing similarity of weather. You will convert the given data into the form specified using the map/reduce paradigm. ***Finding similar weather condition is not included in the project as it requires a mining step.***

Weather patterns vary throughout the day. For example, afternoon temperatures are higher than midnight temperatures. Two stations' weather would be similar only if their weather patterns are similar for several corresponding section of the day. This is done by dividing the day into some number of sections where the difference is known to be not extreme. For this project, 4 sections of the day are considered: 1st section: 5:01 am – 11 am, 2nd section: 11:01 am – 5 pm, 3rd section: 5:01 pm – 11 pm, 4th section: 11:01 pm – 5 am. For this project,

i. You need to create a 3 attribute vector comprising of average temperature, average dew point and average wind speed calculated for each section of a day over a month for each station. Here are some suggestions. You are welcome to come up with alternate ways of doing this project.

First mapper can be used to organize the given input data to a <key, value> pair that helps to calculate the averages for the three attributes.

Corresponding reducer can take the input for the mapper and generate the <key, value> such that values are averages.

ii. Next, for each month a 12 attribute vector comprising of (four) section wise average values of temperature, dew point and wind speed is generated for each station. Second mapper should generate a <key, value> pair such that keys can be grouped so that reducer can create a 12 attribute vector for each month in all the years. The 12 attribute

vector consists of 3 attributes for each section for each month and station on a yearly basis.

The original data size is even larger and we have reduced it to a reasonable size and be still meaningful for this problem. If you want more data, we will be happy to provide.

For this problem, map/reduce can be beneficially used. It will be easy to adjust the number of mapper and reducer nodes based on the data size and the Hadoop framework will do most of the work in partitioning (sharding) the data and passing intermediate output from mappers to reducers. It will also take care of failures of nodes etc.

You need to design and develop a map program and a reduce program to solve the above problem. The most important aspects of this design will be to identify the key/value pairs to be output by the mapper and worked on by the reducer.

In order to understand and appreciate the power of this paradigm and the ease of scaling using this paradigm, you will run the same data set:

i. On multiple mappers and using equal temperature range
ii. Measure response time and number of records in each range for the above two approaches (and other measures as applicable) and compare and analyze them.

The input data is typically partitioned (sharded) to 64MB splits as default. The number of splits can also be configured by the user. Similarly the number of map and reduce nodes can also be configured. Remember that the number of map tasks are determined by the number of shards. And this is different from the number of mapper nodes used. The number of Reduce tasks can also be different from the number of reduce nodes. The number of mapper and reducer nodes need not be the same.

The purpose of this project is to understand the design and development of map/reduce programs as well as the speed up obtained by using different number of map and reduce nodes. You will compare at least two alternate configurations and draw your conclusions in the report.

3. **Installation:**

To complete the project you will need a Hadoop installation. This can be done in one of 2 ways:

Hadoop is available at http://hadoop.apache.org/. You may install Hadoop single node cluster on your own computer. Detailed guide (both text and video) for installing Hadoop on your Linux box can be downloaded from (right click on the link)

http://itlab.uta.edu/downloads/pdf_Hadoop_Installation_Guide.zip (1.4MB)

http://itlab.uta.edu/downloads/video_Hadoop_Installation_Guide.zip (208MB)

The second option is to use Amazon Elastic Map/Reduce. Amazon EMR (Elastic Map/Reduce) is a web service provided by Amazon that uses Hadoop and distributes large datasets and processes them into multiple EC2 instances. You have to sign up for AWS. Please make sure you read carefully your AWS agreements/contracts/fee for use. This project, if done on Amazon, may cost you around $5. For this, you need to monitor and understand your use. Don't leave things running when not necessary. Note that if leave processes running, you will get charged even after you exit. Also, a credit/debit card is necessary for signing up for this service. Detailed guide for executing a simple map/reduce program on Amazon EMR can be downloaded from (right click on the link). We will provide additional information as part of lecture.

http://itlab.uta.edu/downloads/pdf_AWS_Hadoop_EMR_Guide.zip (600KB)

Option 2 above allows you to access more mapper and reduce nodes than on your laptop. If you have a dual core laptop, you can run two configurations. If you use Amazon, make sure you kill your jobs properly to avoid any extra charges! If you have any difficulty in setting up an Amazon account, please talk to the TA.

4. **Project Report:**

Please include (at least) the following sections in a **REPORT.{txt, pdf, doc}** file that you will turn in with your code:

i. **Overall Status**
   Give a *brief* overview of how you implemented the major components. If you were unable to finish any portion of the project, please give details about what is completed and your understanding of what is not. (This information is useful when determining partial credit.)

ii. **Performance Measure:**
   Please include all the methods that you used to measure and compare performance of the results.

iii. **File Descriptions**
   List the files you have created and *briefly* explain their major functions and/or data structures.

iv. **Division of Labor**
   Describe how you divided the work, i.e. which group member did what. Please also include how much time each of you spent on this project. (This has no impact on your grade whatsoever; we will only use this as feedback in planning future projects -- so be honest!)

v. **Logical errors and how you handled them**:

List at least 3 logical errors you encountered during the implementation of the project. Pick those that challenged you. This will provide us some insights into how we can improve the description and forewarn students for future assignments.

5. **What to submit:**

- After you are satisfied that your code does exactly what the project requires, you may turn it in for grading. Please submit your project report with your project.
- You will turn in one zipped file containing you source code as well as the report
- All of the above files should be placed in a single zipped folder named as - 'proj3_firstname_lastname_Section_final'. **Only one zipped folder should be uploaded using blackboard.**
- You can submit your zip file at most 5 times. The latest one (based on timestamp) will be used for grading. So, be careful in what you turn in and when!
- **Only one person per group should turn in the zip file!**
- Three days after the due date, the submission will be closed
- Include as part your report or as separate file, your final output.

6. **Coding style:**

Be sure to observe the following standard Java naming conventions and style. These will be used across all projects for this course; hence it is necessary that you understand and follow them correctly. You can look this up on the web. Remember the following:

i. Class names begin with an upper-case letter, as do any subsequent words in the class name.
ii. Method names begin with a lower-case letter, and any subsequent words in the method name begin with an upper-case letter.
iii. Class, instance and local variables begin with a lower-case letter, and any subsequent words in the name of that variable begin with an upper-case letter.
iv. No hardwiring of constants. Constants should be declared using all upper case identifiers with _ as separators.
v. All user prompts (if any) must be clear and understandable
vi. Give meaningful names for classes, methods, and variables even if they seem to be long. The point is that the names should be easy to understand for a new person looking at your code
vii. Your program is properly indented to make it understandable. Proper matching of if … then … else and other control structures is important and should be easily understandable
viii. Do not put multiple statements in a single line

In addition, ensure that your code is properly documented in terms of comments and other forms of documentation for generating meaningful javadoc.

7. **Grading scheme:**

The project will be graded using the following scheme:

1.	Your approach	10
2.	Correctness of the Map and Reduce code:	50
3.	Correct results for 2 configurations	20
4.	Analysis of results	10
5.	Report	10

8. Class presentation on projects

Since this is the last project, each team (both members where applicable) will make a presentation of their project experience in the class on December 4 / December 8. You will answer any questions on the project experience. Download the template slides with questions from (right click on the link). Do not modify the format and layout! The order of presentation will be strictly according to team numbers.

http://itlab.uta.edu/downloads/4331_5331_Project_Presentation.zip

Please email the completed slides to the instructor by December 3rd. This presentation is a requirement and counts towards class participation. There will be a signup sheet for attendance.