Lab Course: distributed data analytics
Exercise Sheet 7

Nghia Duong-Trung, Mohsan Jameel
Information Systems and Machine Learning Lab
University of Hildesheim
Submission deadline: Friday 23:59PM (on LearnWeb, course code: 3117)

Instructions

Please following these instructions for solving and submitting the exercise sheet.

1. You should submit a zip or a tar file containing two things a) python scripts and b) a pdf document.
2. In the pdf document you will explain your approach (i.e. how you solved a given problem), and present your results in form of graphs and tables.
3. The submission should be made before the deadline, only through learnweb.
4. if you are M.Sc. Data Analytics winter 2016 intake student, you should submit to “Second term students” link on LearnWeb.
5. If you are not M.Sc. Data Analytics student, you can submit to anyone of the links above.

This lab is only for Winter 2016 intake students

Exercise 1: Apache Spark - Basics (10 points)

Exercise 1A: Basic Resilient Distributed Dataset (RDD) (3 points)

Let’s have two lists of words as follows:

- `a = ["spark", "rdd", "python", "context", "create", "class"]`
- `b = ["operation", "apache", "scala", "lambda","parallel","partition"]`

Create two RDD objects of `a`, `b` and do the following tasks. Words should be remained in the results of join operations.

1. Perform `rightOuterJoin` and `fullOuterJoin` operations between `a` and `b`. Briefly explain your solution. (1 point)
2. Using `map` and `reduce` functions to count how many times the character `"s"` appears in all `a` and `b`. (1 point)
3. Using `aggregate` function to count how many times the character `"s"` appears in all `a` and `b`. (1 point)

Exercise 1B: Spark Question (7 points)

In this exercise you will be using movilens1m or movilens10m dataset. Specifically you will be working with Tags Data File Structure (tags.dat), which contains data in the form “UserID::MovieID::Rating::Timestamp”. You have to solve following problem using Spark transformations and actions.

1. A tagging session for a user can be defined as the duration in which he/she generated tagging activity. Typically, an inactive duration of 30 mins is considered termination of the tagging session. Your task is to first separate out tagging sessions for each user.
2. In each tagging session, calculate the frequency of tagging for each user and report those users who have the distance of two standard deviation between their tagging frequency per session and mean.

Exercise 2: Running Apache Spark on HDFS using Yarn (10 points)

In this exercise you will enable Apache Spark to read or operate on data stored in HDFS and use Yarn to launch the jobs. You can reuse your data from last lab i.e. movielens1m or movilens10m. In this exercise, you are going to write a python code using **Hadoop Spark** to accomplish several requirements.

1. Find the user who has assign highest average rating among all the users who rated less than 60 times?
2. An interesting feature one wants to know is the time of the day when a particular user is likely to rate a movie. Please find time of the day when a given user will rate movies.

In this task you have to document how you configured your system and also perform performance analysis i.e. varying number of executors.

Bonus (Optional Question): Matrix Factorization with Coordinate Descent using MapReduce (10 points)

0.1 Instructions for Bonus Question

You will only get points if you have a complete solution. There are no points for partial solution. Specifically, you need to fulfill all the requirements for this questions and should have a report, which contains outputs from your solution i.e. performance analysis and convergence graphs by varying number of workers i.e. mappers and reducers. You can use either Apache Hadoop MapReduce or Apache Spark for solving this problem.

0.2 Problem statement

You are familiar with solving recommender systems problem using a matrix factorization. In this task you have to parallelize coordinate descent method for matrix factorization using mapreduce framework. Sample parallel coordinate descent algorithms are explain in resources mentioned in "Related reading material". The objective function optimizes a square loss.
You should be able to present your solution with the following information (not limited to).

1. Data division strategy?
2. How you parallelize your algorithm? Explain Map and Reduce functions
3. Any other optimization technique you used to optimized MapReduce i.e. caching.
4. Present performance analysis i.e. speedup graph, [hint: have a look at the "Related reading material" how to calculate speedup].
5. Measure the affect of using varying number of workers on the convergence].

0.3 Dataset

For this exercise you will use movielens10m or movielens20m dataset available at `http://files.grouplens.org/datasets/movielens`. The movielens dataset is a rating prediction dataset with five star ratings (on a scale of 1-5). [Hint: you can start with a smaller movielens dataset i.e. movielens1m or movielens100k. But your final solution should be based on 10m or 20m dataset]

0.4 Related reading material

1. Parallel Coordinate Descent `http://www.caam.rice.edu/~optimization/L1/optseminar/Parallel%20BCD.pdf`
2. Matrix factorization with coordinate descent `http://www.cs.utexas.edu/~cjhsieh/icdm-pmf.pdf`

3. Parallel Speedup analysis https://portal.tacc.utexas.edu/c/document_library/get_file?uuid=e05d457a-0fbf-424b-87ce-c96fc0077099&groupId=13601

4. Spark launching configuration https://spark.apache.org/docs/latest/submitting-applications.html