COMP9313 2018s2 Project 3

Set Similarity Join Using Spark on AWS EMR

Problem Definition:

Given two collections of records R and S, a similarity function **sim**(., .), and a threshold τ, the set similarity join between R and S, is to find all record pairs r (from R) and s (from S), such that **sim**(r, s) >= τ. We compute **sim**(., .) using the Jaccard similarity in this project.

id	set
r_1	$\{e_1, e_4, e_5, e_6\}$
r_2	$\{e_2, e_3, e_6\}$
r_3	$\{e_4, e_5, e_6\}$

(a) $\mathcal{R}$ sets

id	set
s_1	$\{e_1, e_4, e_6\}$
s_2	$\{e_2, e_5, e_6\}$
s_3	$\{e_3, e_5\}$

(b) $\mathcal{S}$ sets

Given the above example, and set τ=0.5, the results are (r_1, s_1) (similarity 0.75), (r_2, s_2) (similarity 0.5), (r_3, s_1) (similarity 0.5), (r_3, s_2) (similarity 0.5).

Input files:

Each set is stored in one text file, and each line is in format of: "RecordId list<ElementId>". Two example input files are as below (integers are separated by space):

File1	File2

0 1 4 5 6
1 2 3 6
2 4 5 6
	0 1 4 6
	1 2 5 6
	2 3 5

Another small test data set can be downloaded at (τ=0.1):
https://webcms3.cse.unsw.edu.au/COMP9313/18s2/resources/21255

Output:

The output file contains the similar pairs together with their similarity scores. Each line is in format of "(RecordId$_1$,RecordId$_2$)\tSimilarity" (RecordId$_1$ is from the first file and RecordId$_2$ is from the second file). **Round the similarities to six decimal places (you can use BigDecimal to do this).**

The pairs are sorted in ascending order by the first record and then the second. Given the example input data, the output file is like:

```
(0,0)\t0.75
(1,1)\t0.5
(2,0)\t0.5
(2,1)\t0.5
```

Code format:

Name the package as "comp9313.proj3" and your scala file as "SetSimJoin.scala". Your program should take four parameters: the input file 1, the input file 2, the output folder, and the similarity threshold τ (double precision).

Cluster configuration:

Create an S3 bucket with name "comp9313.<YOUR_STUDENTID>" in AWS. Create a folder "project3" in this bucket for holding the input files.

This project aims to let you see the power of distributed computation. Your code should scale well with the number of nodes used in a cluster. You are required to create three clusters in AWS to run the same job:

- Cluster1 - 2 node of instance type m3.xlarge;
- Cluster2 - 3 nodes of instance type m3.xlarge;
- Cluster3 - 4 nodes of instance type m3.xlarge.

Select release EMR-5.17.0 when creating each cluster. Unzip and upload the following data set to your S3 bucket, and set τ to 0.85 to run your program:

https://webcms3.cse.unsw.edu.au/COMP9313/18s2/resources/21256

Record the runtime on each cluster and draw a figure where the x-axis is the number of nodes you used, and the y-axis is the time of getting the result. Store this figure in a file "Runtime.jpg". Please also take a screenshot of running your program on AWS in each cluster as a proof of the runtime. Compress the three screenshots into a zip file "Screenshots.zip". Briefly describe your optimization techniques in a file "Optimization.pdf".

Notes

Create a project locally in Eclipse, test everything in your local computer, and finally do it in AWS EMR.

Documentation and code readability

Your source code will be inspected and marked based on readability and ease of understanding. The efficiency and scalability of this project is very important and will be evaluated as well. Below is an indicative marking scheme:

Result correctness: 25
Efficiency and Scalability: 10
Code structure, Readability, and Documentation: 5

Submission:

Deadline: Sun 28th October 11:59:59 PM

Log in any CSE server (williams or wagner), and use the give command below to submit your solutions:

```
$ give cs9313 project3 SetSimJoin.scala Runtime.jpg Screenshots.zip Optimization.pdf
```

Or you can submit through:
https://cgi.cse.unsw.edu.au/~give/Student/give.php

If you submit your assignment more than once, the last submission will replace the previous one. To prove successful submission, please take a screenshot as assignment submission instructions show and keep it by yourself.

Late submission penalty

10% reduction of your marks for the 1st day, 30% reduction/day for the following days.

Plagiarism:

The work you submit must be your own work. Submission of work partially or completely derived from any other person or jointly written with any other person is not permitted. The penalties for such an offence may include negative marks, automatic failure of the course and possibly other academic discipline. Assignment submissions will be examined manually.

Relevant scholarship authorities will be informed if students holding scholarships are involved in an incident of plagiarism or other misconduct.

Do not provide or show your assignment work to any other person - apart from the teaching staff of this subject. If you knowingly provide or show your assignment work to another person for any reason, and work derived from it is submitted you may be penalized, even if the work was submitted without your knowledge or consent.