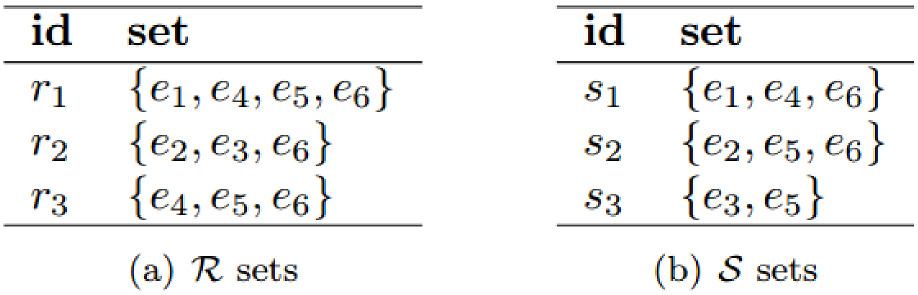
COMP9313 2021T3 Project 3 (20 marks)

Set Similarity Join Using Spark on Google Dataproc

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.

In this project, you are required to use the Jaccard similarity function to compute **sim**(r, s). Given the following example, and set τ=0.5,



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

the result pairs 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:

You are required to do the "self-join", that is, a single input file is given, in which each line is in format of:

"RecordId list<ElementId>",

and this file serves as both R and S.

An example input file is as below (integers are separated by space):

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

This sample file "tiny-data.txt" can be downloaded at:

https://webcms3.cse.unsw.edu.au/COMP9313/21T3/resources/69118

Another sample input file "flickr_small.txt" can be downloaded at:

https://webcms3.cse.unsw.edu.au/COMP9313/21T3/resources/69119

Output:

The output file contains the similar pairs together with their similarities. Each line is in format of "(RecordId$_1$,RecordId$_2$)\tSimilarity" (RecordId$_1$<RecordId$_2$, and there are no duplicate pairs in the result). The similarities are of double precision. 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:

```
(0,2)\t0.75
(0,3)\t0.75
(1,4)\t0.5
(2,3)\t0.5
(2,4)\t0.5
```

Code format:

Name your java file as "SetSimJoin.scala" and put it in the package "comp9313.proj3". Your program should take three parameters: the input file, the output folder, and the similarity threshold τ (double precision).

Cluster configuration:

Create a bucket with name "comp9313-<YOUR_STUDENTID>" in Dataproc. 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 Dataproc to run the same job:

- Cluster1 - 1 master node and 2 worker nodes;
- Cluster2 - 1 master node and 4 worker nodes;
- Cluster3 - 1 master node and 6 worker nodes.

For both master and worker nodes, select n1-standard-2 (2 vCPU, 7.5GB memory).

Unzip and upload the following data set to your bucket, and set τ to 0.85 to run your program: https://webcms3.cse.unsw.edu.au/COMP9313/21T3/resources/69125.

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, and store this figure in a file "Runtime.jpg". Please also take a screenshot of running your program on Dataproc 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".

Create a project locally in the IDE, test everything in your local computer, and finally do it in Google Dataproc.

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: 12
Efficiency and Scalability: 6
Code structure, Readability, and Documentation: 2

Submission:

Deadline: Sun 28th Nov 09:59:59 PM

You can submit through Moodle. You need to submit four files: SetSimJoin.scala, Runtime.jpg, Screenshots.zip, and Optimization.pdf.

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. If you have any problems in submissions, please email to yu.hao@unsw.edu.au.

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.