



COMP9313 Project 4 : Optimization Report

Li Yu z3492782

1. Method for the project :

- Prefix filtering: minimise the number of prefix items emitted from the mappers

$$p = |record| - \text{ceil}(|record| * t) + 1$$

- Positional filtering: Apply this length constrain to minimise the number of (one String of prefix, (rid1+record1,rid2+record2)) candidate pairs

$$|record2| \geq |record1| * t$$

2. Code Steps :

2.1 Stage 1: Ordering Input

1. User tokens frequency to make input raw RDD ordered
2. When tokens have same frequency value, token will be ordered by each Int value from small to large.

2.2 Stage 2: Processing prefix

1. Find prefix value for each record following the formula : $P = |r| - \text{ceil}(|r| * t) + 1$
2. Yield (CommonString, rid1+record1) pairs for each prefix string
3. All these pairs are stored in prefixRecordMap RDD.

2.3 Stage 3: Processing prefix

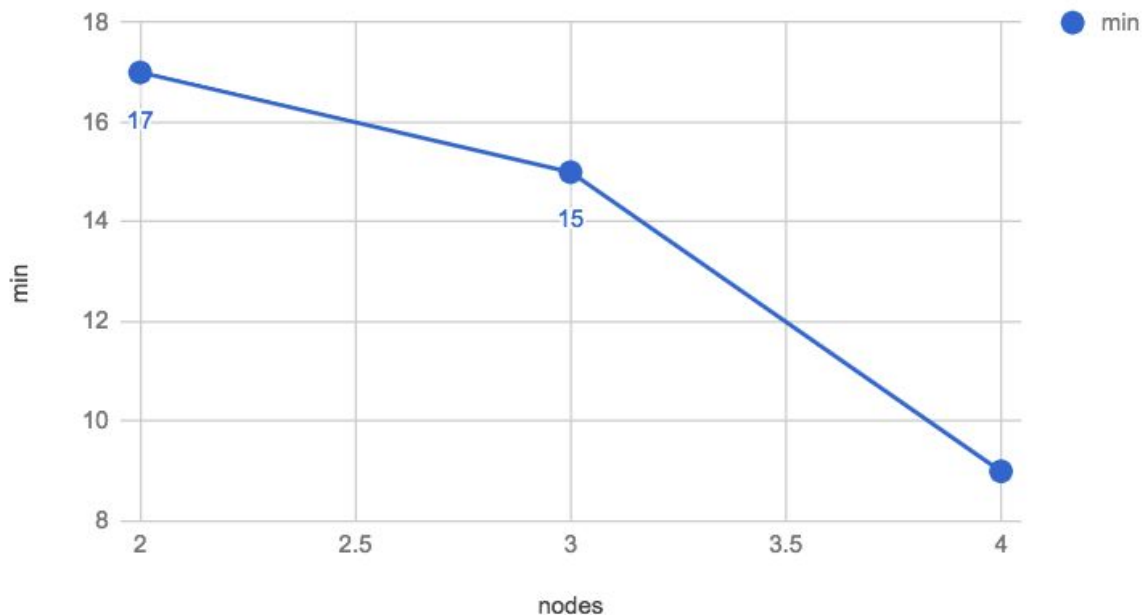
1. Self join with rdds
2. Filter the all candidates to meet the id constrain whose rid1 should less than rid2. For one common prefix string, rid1 only join with rid2 who is larger. This step can also remove half of candidate pairs which are redundant. For example, for id pairs like (1,2) and (2,1). Only id pair (1,2) is kept.
3. Filter with the length constraints of the record2 to meet the requirement that:
 $|record2| \geq |record1| * t$
4. After remove impossible candidates pairs, Jaccard similarity computation start.
5. Filter out results which meet the threshold requirement.
6. Reduce the duplicated result to one.
7. Sort the result.

3. RDD operation selection :

1. Join: Join all possible combinations of one prefix string as the structure below.
(*One prefix string, (rid1 + record1, rid2 + record2)*)
2. Filter : It is a map side operation. Filter is more efficiency than shuffle side operations like reduceByKey, groupByKey.
3. Reduce shuffle times in the middle of the RDD process. Just using reduceByKey once at the end to reduce duplicated results.
4. Only using “collect” for ordering once to keep away from “ out of memory” problem.
5. Use rdd.persist() to keep the original RDD in memory in order to process the data fast.

4. Outcome on AWS:

Running speed on AWS (with ordering code)



	Name	ID	Status	Creation time (UTC+10) ▾	Elapsed time	Normalized instance hours
<input type="checkbox"/> ▾	comp9313.ass4.JOIN152.int.p.noc.o	j-2RPR2T9OXYZTG	Terminated All steps completed	2018-05-27 10:55 (UTC+10)	24 minutes	16
Summary			Steps			Bootstrap actions
Master ec2-54-153-157-44.ap-southeast-public DNS: 2.compute.amazonaws.com			Name			Name
Termination protection: Off			Status			
Tags: --			Start time (UTC+10) ▾			
Hardware			Elapsed time			
Master: Terminated 1 m3.xlarge			Spark application Completed 2018-05-27 10:59 (UTC+10) 17 minutes			No bootstrap actions available
Core: Terminated 1 m3.xlarge			Setup hadoop debugging Completed 2018-05-27 10:59 (UTC+10) 2 seconds			
Task: --						
View cluster details			View monitoring details			

▼

comp9313.ass4.JOIN153.int.p.noc.o

j-3PF65PAAV7J1H

Terminated

All steps completed

2018-05-27 10:58 (UTC+10)

21 minutes

24

Summary

Master ec2-13-211-254-234.ap-southeast-public DNS: 2.compute.amazonaws.com

Termination protection: Off

Tags: --

Hardware

Master: Terminated 1 m3.xlarge

Core: Terminated 2 m3.xlarge

Task: --

View cluster details

View monitoring details

Steps

Name

Status

Start time (UTC+10) ▼

Elapsed time

Spark application

Completed

2018-05-27 11:02 (UTC+10)

15 minutes

Setup hadoop debugging

Completed

2018-05-27 11:02 (UTC+10)

2 seconds

View all interactive jobs

Bootstrap actions

Name

No bootstrap actions available

▼

comp9313.ass4 JOIN154.int.p.noc.o

j-3N2YBURR92KWD

Terminated

All steps completed

2018-05-27 11:00 (UTC+10)

15 minutes

32

Summary

Master ec2-13-236-177-218.ap-southeast-public DNS: 2.compute.amazonaws.com

Termination protection: Off

Tags: --

Hardware

Master: Terminated 1 m3.xlarge

Core: Terminated 3 m3.xlarge

Task: --

View cluster details

View monitoring details

Steps

Name

Status

Start time (UTC+10) ▼

Elapsed time

Spark application

Completed

2018-05-27 11:04 (UTC+10)

9 minutes

Setup hadoop debugging

Completed

2018-05-27 11:04 (UTC+10)

2 seconds

View all interactive jobs

Bootstrap actions

Name

No bootstrap actions available

Cluster	Nodes	Running time on AWS (with ordering process)
Cluster1	2 nodes	17min
Cluster2	3 nodes	15min
Cluster3	4 nodes	9min