Report

Q1: No and No

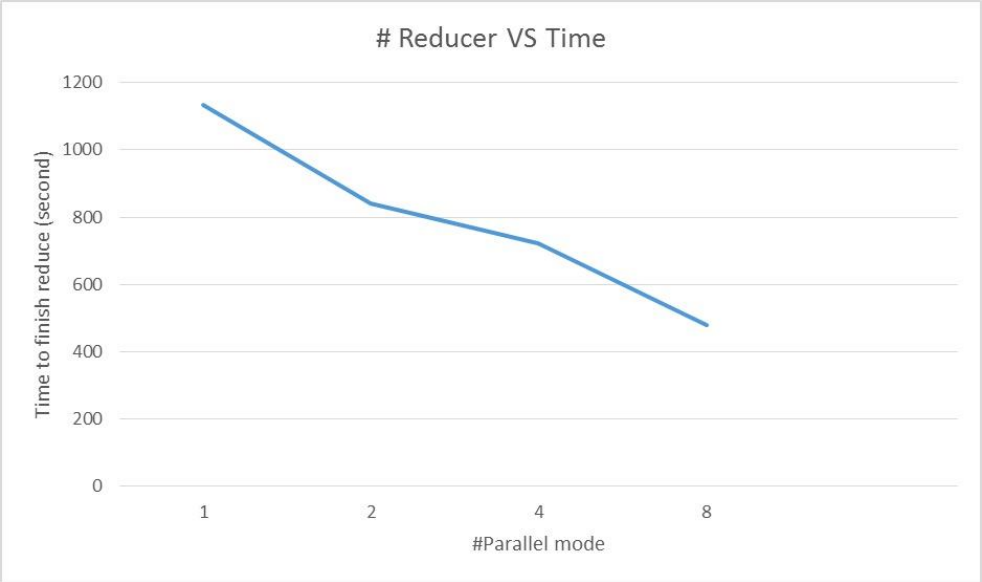
Q2:

(a). n = 1: 1136s

n= 2: 841s

n= 4: 722s

n= 8: 478s



(b).

IO cost is large, when mapper send data to all reducer, it take time

Keys are not evenly distrusted such that some reducer has more key value pair to process

Shuffle and sort takes time, when no combiner is used, some key may have a lot of value to sort, and it can cost a lot of times, and increase number of reduce does not help

Q3

r = Flatten(dataset, by = lambda x: [x[0],k[1] for k in x[1]]) | Group(by = lambda: (id, companyid):(id,companyid), reducingTo= ReduceToCount())

J = Join(Jin(r,by = lambda ((id,companyid),count):id), Jin(r,by = lambda ((id,companyid),count):id))

ReplaceEach(lambda(((id1,companyid1),count1),((id2,companyid2),count1)):id1,companyid1,companyid2) |Distinct()|Filter(by=lambda (id,compandid1,companyid2):companyid1 < companyid2)

k=Join(Jin(J,by=lambda(id,companyid1,companyid2):(companyid1,companyid2)),Jin(queryData,by=lambda (companyid1,companyid2):(companyid1,companyid2))) | ReplaceEach(by = lambda ((id,companyid1,companyid2),(companyid3,companyid4)):(companyid1,companyid2))

l=Join(Jin(J,by=lambda (companyid1,companyid2):(companyid1,companyid2)),Jin(queryData,by=lambda (companyid1,companyid2):(companyid2,companyid1)))|ReplaceEach(by=lambda ((id,companyid1,companyid2),(companyid3,companyid4)):(companyid1,companyid2))

output = Union(k,l) | Group(by = lambda x:1,reducingTo=ReduceToCount())

Q4:

Find the common word of two sets: one only contains words with only alphabetical characters and the other set contains words with at least one non alphabetical character, but those non-alphabetic

characters are removed