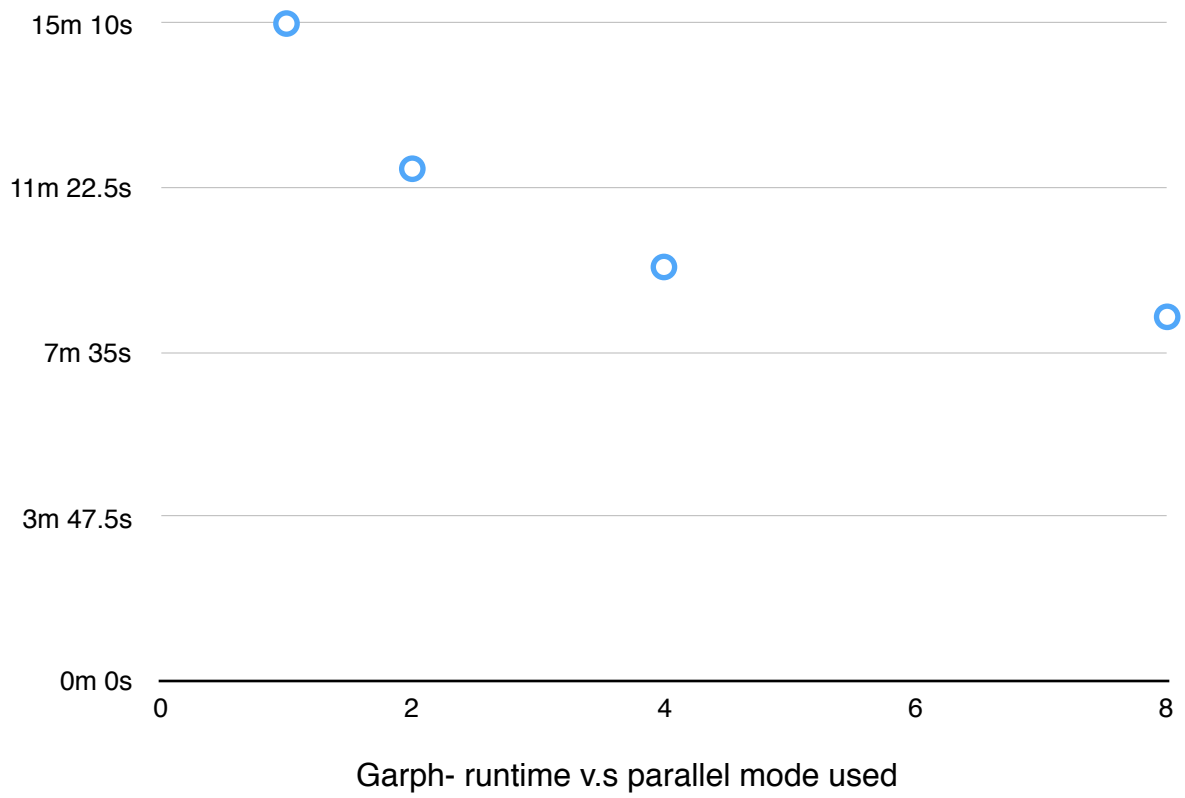
Monday, September 25, 2017

# Report

Yao Li id:yaol3

P2.

(a)



(b)

Reason1: There are many small map-reduce job that doesn't consume much resources, split the data on these job will not be efficient, but rather time-consuming since it need to shuffle and send the data through network.

Reason2: The initialization of the NameNode takes time, the efficiency will decrease, especially for those relative small map-reduce job, since the time for data processing is relatively short.

Reason3: The data processed will need to sent back to the master, it takes time.

P3.

Flatten_data = FlatMap(data, by=lambda(Id,pos_list): for (pos,company) yield (Id,company))

data = Augment(Flatten_data, sideview=query, loadedBy=lambda v:GPig.onlyRowOf(v))

F1 = Filter(data, by=lambda ((Id,company), (cmp1, cmp2)): company==cmp1) | ReplaceEach(by=lambda((Id,company), (cmp1, cmp2)):(Id,company)

F2 = Filter(data, by=lambda ((Id,company), (cmp1, cmp2)): company==cmp2)

output = Join(Jin(F1, by=lambda (Id,company):id), Jin(F2, by=lambda (Id,company):id)) | Replaceeach(by=lambda (Id,company):id) | Distinct()

P4.

The flat1 function simply flatmap the split input data into itself if the token isalpha

And the flat2 function flatmap those tokens is not alpha in the split input data and delete the alphanumeric inside the token.

Therefore, if we join these two views, we will get the tokens that isalpha() and is a subsequence of one of those not isalpha() in the data.