Big Data
2η Εργασία

Ομαδα 20

dai16054 Νικόλαος Στεφανίδης
dai16057 Δημήτριος Τουργαϊδης
dai16060 Κωνσταντίνος Τσιώλης
dai16067 Γεώργιος Μιχούλης

Αναφορές, τα προβλήματα και τρόπος επίλυσης τους

Αρχικά, εγκαταστήσαμε το spark σε ολες τις μηχανές, χρησιμοποιώντας τις εντολές που μας δόθηκαν, έπειτα κατεβάσαμε το αρχείο Spark_Resources και κάναμε τις κατάλληλες τροποποιήσεις ώστε να τρέξει σε συστάδα (cluster).

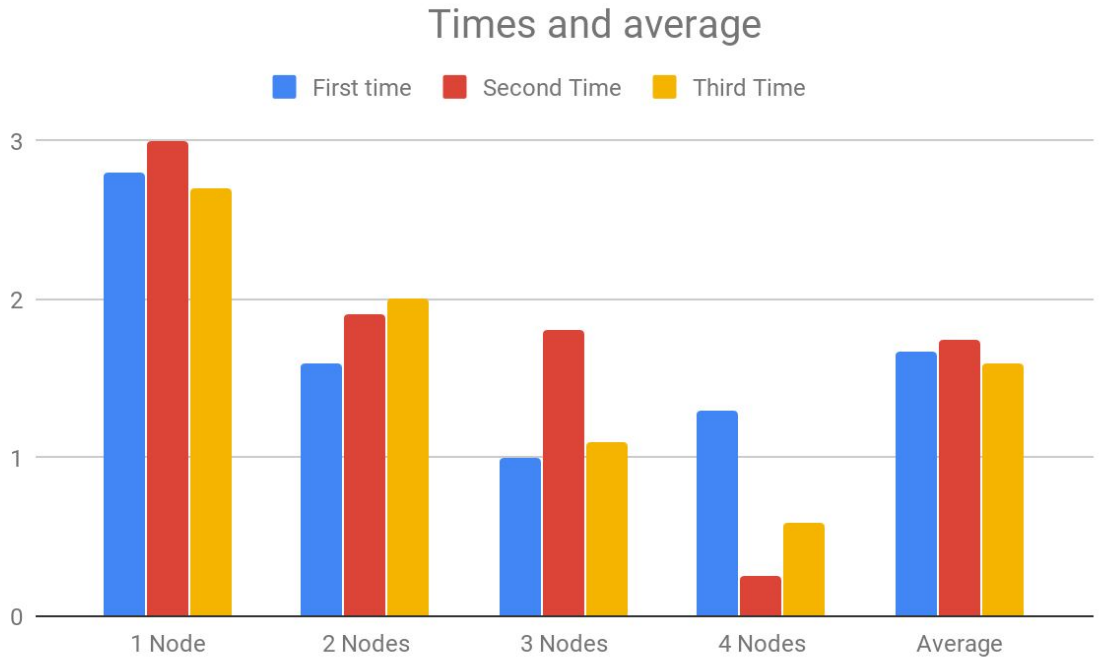
Στη συνέχεια, κατεβάσαμε τα δεδομένα μας, κάναμε κάποιες απαραίτητες αλλαγές στον κώδικα που δόθηκε και τρέξαμε των κώδικα μας για διαφορετικά K ώστε να βρούμε την τιμή του K η οποία θα ταίριαζε καλύτερα στα δεδομένα μας. Αφού βρήκαμε την τιμή που ψάχναμε, πραγματοποιήσαμε 3 εκτελέσεις με το συγκεκριμένο K σε τόσες διαμορφώσεις όσοι και οι slave μας.

Για κάθε διαφορετική διαμόρφωση έπρεπε να αλλάζουμε κατάλληλα το configuration ώστε να τρέχει στον αντίστοιχο αριθμό μηχανών κάθε φορά.

Σε όλες τις εκτελέσεις αρχικά ανοίξαμε το hdfs όπου, ήταν τοποθετημένα τα δεδομένα μας, ώστε να τα τραβάει από εκεί. Μετά, καταγράψαμε τα αποτελέσματα και τους χρόνους εκτέλεσης. Το μόνο πρόβλημα που αντιμετωπίσαμε ήταν πως τρέξαμε όλα τα παραδείγματα χωρίς να γίνεται τυχαία αρχικοποίηση των κέντρων οπότε, όταν το καταλάβαμε, κάναμε την κατάλληλη αλλαγή στον κώδικα και τα τρέξαμε απο την αρχή.

Χρόνοι εκτελέσεων και μέσος χρόνος

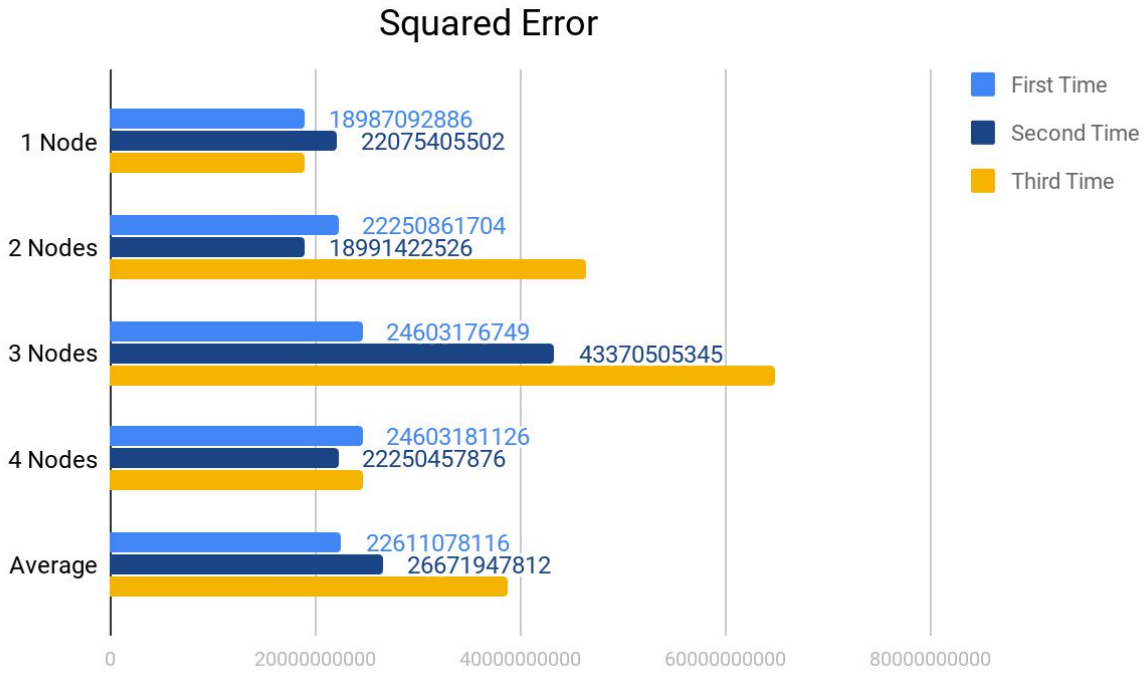
hours	First time	Second Time	Third Time
1 Node	2,8	3	2,7
2 Nodes	1,6	1,9	2
3 Nodes	1	1,8	1,1
4 Nodes	1,3	0,25	0,59
Average	1,675	1,7375	1,5975



Σχόλια στους χρόνους:

Παρατηρούμε ότι κάθε φορά αυξάνεται ο αριθμός των συστάδων (cluster), μειώνεται ο χρόνος εκτέλεσης. Όπως φαίνεται στο παραπάνω γράφημα, ο χρόνος που χρειάζεται ο μοναδικός κόμβος για να παράξει την συσταδοποίηση είναι πολύ περισσότερο ή ακόμα και διπλάσιο κάποιες φορές σε σχέση με τις υπόλοιπες εκτελέσεις. Παρατηρούμε ότι τον μεγαλύτερο χρόνο τον έκανε ο μοναδικός κόμβος την δεύτερη φορά που εκτελέστηκε ο κώδικας , ενώ τον μικρότερο χρόνο τον έκαναν οι τέσσερις κόμβοι μαζί την δεύτερη φορά που εκτελέστηκαν.

Τετραγωνικό σφάλμα



	First Time	Second Time	Third Time
1 Node	18987092886	22075405502	18987084688
2 Nodes	22250861704	18991422526	46459682864
3 Nodes	24603176749	43370505345	64756669953
4 Nodes	24603181126	22250457876	24603173344
Average	22611078116	26671947812	38701652712

Σχόλια για το τετραγωνικό σφάλμα

Το μέγιστο τετραγωνικό σφάλμα εντοπίζεται στην περίπτωση που έχουμε τους τρεις κόμβους την τρίτη φορά, ενω το μικρότερο εντοπίζεται στην περίπτωση που τρέχει μόνο ένας κόμβος την τρίτη φορά.

Κώδικας

```
from __future__ import print_function

import sys

import numpy as np
from pyspark import SparkContext
from pyspark.mllib.clustering import KMeans


def parseVector(line):
    return np.array([float(x) for x in line.split(',')])


if __name__ == "__main__":
    if len(sys.argv) != 3:
        print("Usage: kmeans <file> <k>", file=sys.stderr)
        exit(-1)
    sc = SparkContext(appName="KMeans")
    lines = sc.textFile(sys.argv[1])
    data = lines.map(parseVector)
    k = int(sys.argv[2])
    model = KMeans.train(data, k,initializationMode="random")
    print("Final centers: " + str(model.clusterCenters))
    print("Total Cost: " + str(model.computeCost(data)))
    sc.stop()
```

Οι αλλαγές που έγιναν στον κώδικα που μας δόθηκε είναι οι εξής:

- Από split(‘ ‘) σε split(‘,’)
- Στην εντολή train προστέθηκε η ιδιότητα initializationMode=”random”