# Εργασία 1 - Big Data

Φωτιάδου Βασιλική

Μανώλη Χριστίνα

# 

# Πίνακας Περιεχομένων:

[**Πίνακας Περιεχομένων:**](#_2oqoga7t3yf7) **1**

[**1.Τον αλγόριθμο που χρησιμοποιήσατε για να λύσετε το πρόβλημα.**](#_hxikmnow1pxf) **2**

[Η μέθοδος map:](#_d11cykadqzwp) 2

[Η μέθοδος reduce:](#_7jbnwadcd1n2) 2

[**2.Γραφικές παραστάσεις και πίνακες με τους χρόνους που μετρήσατε.**](#_kzyk06lk8zk7) **3**

[**3.Σχολιασμό των χρόνων που παρατηρήσατε.**](#_8c7jj4o89tp8) **5**

## 1.Τον αλγόριθμο που χρησιμοποιήσατε για να λύσετε το πρόβλημα.

Αρχικά, εργαστήκαμε με Java για να υλοποιήσουμε τον κώδικα και να παράξουμε το jar αρχείο. Μέσα στην κλάση που ονομάσαμε Project υπάρχουν 2 εσωτερικές κλάσεις:

1. TokenizerMapper
2. TheReducer

οι οποίες περιέχουν τις μεθόδους map και reduce αντίστοιχα. Οι παραπάνω εσωτερικές κλάσεις κληρονομούν τις κλάσεις Mapper<Object, Text, Text, Text> (καθώς το key και το value που παραδίδει ο mapper είναι τύπου Text) και Reducer<Text, Text, Text, Text> (καθώς το key και το value που βγάζει ως έξοδο ο reducer είναι τύπου Text) αντιστοιχα.

### Η μέθοδος map:

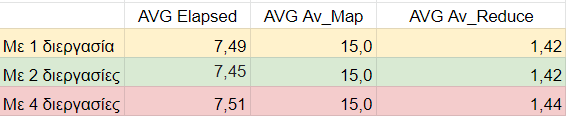
* Αρχικά, διαχωρίζει (tokenize) κάθε γραμμή που του έρχεται σε κάθε μεταβλητή (token) σύμφωνα με τον οριοθέτη (regex) "," και όριο (limit) 15 ωστε η γραμμή να κοπεί σε 15 κομμάτια και ο πίνακας που θα αποθηκεύει αυτά τα κομμάτια (contents) θα έχει μήκος 15 ώστε κάθε μεταβλητή να αποθηκευτεί σε κάθε κελί του πίνακα.
* Αφού ελέγξουμε ότι η γραμμή που χωρίσαμε δεν περιέχει τα ονόματα των μεταβλητών αλλά τις τιμές :
  + Ελέγχουμε εάν το όνομα του αρχείου λείπει και υπάρχει μόνο η επέκταση αρχείου. Αν συμβαίνει κάτι τέτοιο τότε προσθέτουμε ως όνομα του αρχείου το accession number του.
  + Στην συνέχεια, επιλέγουμε να ενώσουμε το ip (που βρίσκεται μέσα στο κελί 0 του contents) και το filename (που βρίσκεται μέσα στο κελί 6 του contents), οριοθετημένα με το “!”, και τα θέτουμε ως το key.
  + Επιπλέον, θέτουμε ως value το date (που βρίσκεται μέσα στο κελί 1 του contents).

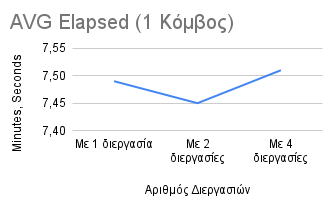
### Η μέθοδος reduce:

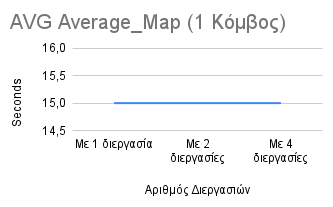
* Δέχεται τα keys (ip+filename) και μία λίστα από values (date) τα οποία έχουν ομαδοποιηθεί με βάση το key τους.
* Μέσα στην μέθοδο δημιουργούμε ένα Hashset τύπου String το οποίο θα μας βοηθήσει να εξαλείψουμε τις διπλότυπες ημερομηνίες.
* Καθώς διασχίζουμε την λίστα με τα values παίρνουμε το κάθε value και αφού το μετατρέψουμε σε string το βάζουμε μέσα στο set.
* Στο τέλος, θα πρέπει να έχουμε πάρει μια λίστα με όλες τις διαφορετικές ημερομηνίες που ο χρήστης x προσπέλασε ένα αρχείο y.
* Στην συνέχεια, ελέγχοντας αν το μέγεθος του set είναι μεγαλύτερο του 1 διασφαλίζουμε ότι ο χρήστης x που προσπέλασε ένα αρχείο y το έκανε πάνω από μία φορα άρα το set περιέχει παραπάνω από μία ημερομηνίες. Αν συμβαίνει αυτό τότε:
  + Χωρίζουμε το key στο αντίστοιχο ip και filename με βάση τον οριοθέτη που είχαμε θέσει νωρίτερα (“!”).
  + Επιστρέφουμε ως key τον χρήστη (ip) και ως value το όνομα αρχείου (filename) έτσι ώστε το αποτέλεσμα του προγράμματός μας να αποτελείται από ζεύγη <χρήστης, αρχείο>.

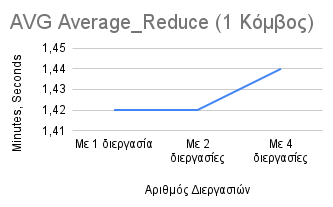
## 2.Γραφικές παραστάσεις και πίνακες με τους χρόνους που μετρήσατε.

**Με έναν κόμβο :**

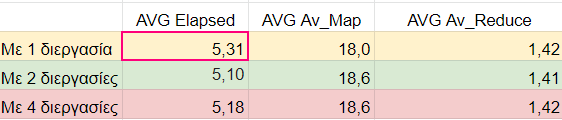


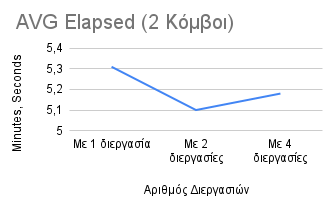


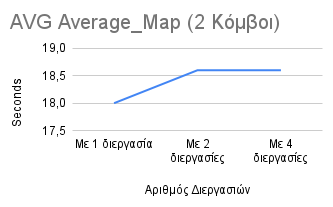


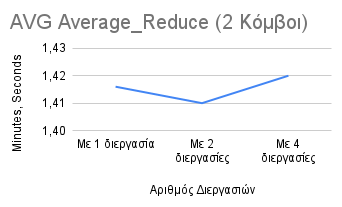


**Με 2 κόμβους :**









## 3.Σχολιασμό των χρόνων που παρατηρήσατε.

Όσον αφορά την αντιμετώπιση του προβλήματος με *έναν κόμβο* δεν παρατηρείται ιδιαίτερη διαφορά του **elapsed time** με την χρήση 1, 2 και 4 διεργασιών reduce καθώς και του **average reduce time** ενώ το **average map time** παραμένει *σταθερό*.Με τους *2 κόμβους* παρατηρείται παρόμοια συμπεριφορά όπου με την χρήση των διαφόρων διεργασιών δεν υπάρχει σημαντική αυξομείωση. Συγκρίνοντας τους χρόνους 1 και 2 κόμβων υπάρχει διαφορά περίπου 2 λεπτών στο **average elapsed time**, το **average map time** αυξάνεται από τον 1 στους 2 κόμβους ενώ το **average reduce time** παραμένει *σχετικά σταθερό*.