

## Πρώτη ομαδική εργασία

### Ημερομηνία παράδοσης: Τρίτη 21 Δεκεμβρίου 2021

Σας δίνεται ένα σύνολο δεδομένων, το οποίο αφορά αρχεία καταγραφής ιστοτόπων. Το σύνολο δεδομένων μπορείτε να το κατεβάσετε από εδώ:

<https://www.sec.gov/dera/data/edgar-log-file-data-set.html>

Θα κατεβάσετε τα παρακάτω αρχεία:

<http://www.sec.gov/dera/data/Public-EDGAR-log-file-data/2017/Qtr2/log20170630.zip>

<http://www.sec.gov/dera/data/Public-EDGAR-log-file-data/2017/Qtr2/log20170629.zip>

<http://www.sec.gov/dera/data/Public-EDGAR-log-file-data/2017/Qtr2/log20170628.zip>

<http://www.sec.gov/dera/data/Public-EDGAR-log-file-data/2017/Qtr2/log20170627.zip>

<http://www.sec.gov/dera/data/Public-EDGAR-log-file-data/2017/Qtr2/log20170626.zip>

Ως εξής:

wget <http://www.sec.gov/dera/data/Public-EDGAR-log-file-data/2017/Qtr2/log20170630.zip>

Η περιγραφή του κάθε αρχείου καταγραφής βρίσκεται εδώ:

[https://www.sec.gov/files/EDGAR\\_variables\\_FINAL.pdf](https://www.sec.gov/files/EDGAR_variables_FINAL.pdf)

Στόχος σας είναι να κατασκευάσετε πρόγραμμα MapReduce το οποίο θα εντοπίζει τους χρήστες που προσπέρασαν το ίδιο αρχείο πάνω από μία φορά σε διαφορετικές ημερομηνίες. Αν κάποιος χρήστης προσπέρασε ένα συγκεκριμένο αρχείο περισσότερες από μία φορές στην ίδια ημερομηνία θεωρείται μία προσπέλαση. Θεωρίστε ότι ο κάθε χρήστης προσδιορίζεται μοναδικά από το πεδίο ip. Το όνομα του αρχείου βρίσκεται στο πεδίο extension. Αν στο πεδίο αυτό υπάρχει μόνο επέκταση και όχι και όνομα αρχείου, τότε το όνομα αρχείου προκύπτει από την παράθεση των πεδίων accession+extension. Το αποτέλεσμα του προγράμματός σας θα αποτελείται από ζεύγη <χρήστης, αρχείο> σύμφωνα με τα παραπάνω.

Αντιμετωπίστε το παραπάνω πρόβλημα με Hadoop & MapReduce σε κατανεμημένη διαμόρφωση, για α) έναν κόμβο, β) για δύο κόμβους, χρησιμοποιώντας 1, 2 και 4 διεργασίες reduce.

Αν η λύση σας αποτελείται από πλέον του ενός κύκλου map reduce, να χρησιμοποιήσετε το ίδιο πλήθος διεργασιών reduce για κάθε κύκλο.

Για κάθε μία από τις παραπάνω διαμορφώσεις, πραγματοποιήστε 5 εκτελέσεις και αφού αφαιρέσετε τη μεγαλύτερη και μικρότερη τιμή, καταγράψτε το μέσο όρο των 3 υπολοίπων εκτελέσεων για τον κάθε έναν από τους παρακάτω χρόνους:

Elapsed Time, Average Map Time, Average Reduce Time.

## Οδηγίες

- 1 Η εργασία πραγματοποιείται αποκλειστικά σε ομάδες των 2 φοιτητών.
- 2 Ως γλώσσα προγραμματισμού, μπορείτε να χρησιμοποιήσετε μία εκ των Java, Python.
- 3 Θα παραδώσετε:
  - 3.1 Τον κώδικά σας.
  - 3.2 Μία αναφορά που θα περιλαμβάνει:
    - 3.2.1 Τον αλγόριθμο που χρησιμοποιήσατε για να λύσετε το πρόβλημα.
    - 3.2.2 Τυχόν περιορισμούς και θεωρήσεις της λύσης σας.
    - 3.2.3 Επιπλέον πακέτα, αν χρησιμοποιήσατε.
    - 3.2.4 Τις παραμέτρους συστήματος, αν τροποποιήσατε.
    - 3.2.5 Γραφικές παραστάσεις και πίνακες με τους χρόνους που μετρήσατε.
    - 3.2.6 Σχολιασμό των χρόνων που παρατηρήσατε.
  - 3.3 Τα αρχεία εξόδου με το αποτέλεσμα του προγράμματός σας για κάθε αριθμό reducers που χρησιμοποιήσατε.
  - 3.4 Αν χρησιμοποιήσετε Java, το .jar που θα δημιουργήσετε.
- 4 Η εργασία θα παραδοθεί ηλεκτρονικά αποκλειστικά μέσω openeclass σε μορφή ενός αρχείου zip.
- 5 Για να μετρήσετε τους χρόνους, θα χρησιμοποιήσετε τον job history server. Ξεκινάει με την εντολή:

```
~/hadoop/sbin/mr-jobhistory-daemon.sh start historyserver
```

Και τερματίζει αν αντίστοιχα όπου start βάλουμε stop.

Μπορείτε να τον βρείτε στην εξής διεύθυνση για τη μηχανή σας:

<MASTER\_NAME>:19888

- 6 Για την εκτέλεση σε έναν κόμβο, μπορείτε να χρησιμοποιήσετε το παρακάτω script (χωρίς τις προαιρετικές παραμέτρους) για να τερματίσετε την κατάλληλη υπηρεσία:

```
~/hadoop/sbin/yarn-daemon.sh stop
```

## Ενδεικτικές Αναφορές:

- 1 Κεφάλαια 2.1-2.3 από το βιβλίο «Εξόρυξη από Μεγάλα Σύνολα Δεδομένων», διαθέσιμο online στο <http://www.mmds.org/> (αγγλικά)
- 2 «Data-Intensive Text Processing with MapReduce». Διαθέσιμο online στο <https://lintool.github.io/MapReduceAlgorithms/>
- 3 Σημειώσεις του μαθήματος