

# ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

## ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

### ΚΑΤΑΝΕΜΗΜΕΝΑ ΣΥΣΤΗΜΑΤΑ

Εαρινό Εξάμηνο 2022-2023  
Υποχρεωτική εργασία

Τα τελευταία χρόνια έχει παρατηρηθεί μεγάλη αύξηση των εφαρμογών καταγραφής άσκησης (δραστηριότητας), activity tracker. Αυτά τα συστήματα συνήθως αποτελούνται από μια mobile frontend εφαρμογή για την καταγραφή της δραστηριότητας και ένα backend σύστημα το οποίο αναλαμβάνει την ανάλυση των δεδομένων. Παράλληλα πολλά από αυτά τα συστήματα δίνουν τη δυνατότητα κοινωνικής δικτύωσης μεταξύ των χρηστών τους με δυνατότητες όπως προβολή leaderboards με τους καλύτερους δρομείς ανά διαδρομή, σύγκριση με τον μέσο όρο και άλλα. Γνωστές υπηρεσίες που υπάρχουν διαθέσιμες online είναι π.χ. το Strava και το MapMyRun. Στα πλαίσια της εργασίας του μαθήματος καλείστε να δημιουργήσετε ένα απλό τέτοιο σύστημα ανάλυσης δεδομένων από activity tracking.

Τα συστήματα αυτά, έχουν την ικανότητα να εξυπηρετούν ένα μεγάλο αριθμό χρηστών. Αυτό σημαίνει ότι κάθε χρήστης διατηρεί ένα προσωπικό “προφίλ”, δηλαδή, ένα χώρο όπου μπορεί να προσθέτει τα activities του, τις οποίες στην συνέχεια μπορεί να αναλύσει μέσω της εφαρμογής και να συγκριθεί με άλλους χρήστες σε παρόμοιες δραστηριότητες. Επίσης μπορεί να δει τα συνολικά στατιστικά του, όπως πλήθος δραστηριοτήτων, συνολική απόσταση, συνολικός χρόνος άσκησης κτλ.. **Κάθε χρήστης μπορεί είτε να καταγράψει και να ανεβάσει εκείνη την στιγμή την δραστηριότητά του μέσω του mobile application, είτε να ανεβάσει μια δραστηριότητα που είχε καταγράψει προγενέστερα.** Μια δραστηριότητα είναι μια ακολουθία από GPS waypoints. Ένα waypoint χαρακτηρίζεται από τις συντεταγμένες του, (latitude, longitude), το υψόμετρο (elevation) και την ώρα καταγραφής του. Η ακολουθία αυτών των waypoints αποθηκεύεται σε μια ειδική μορφή αρχείου XML που ονομάζεται GPX.

```
<?xml version="1.0" encoding="UTF-8"?>
<gpx version="1.1" creator="user1">
  <wpt lat="52.2423614748556" lon="5.281985213730702">
    <ele>-0.45</ele>
    <time>2023-03-15T10:41:51Z</time>
  </wpt>
  <wpt lat="52.24078476451067" lon="5.294344832871327">
    <ele>-0.06</ele>
    <time>2023-03-15T10:43:59Z</time>
  </wpt>
  <wpt lat="52.2360542973452" lon="5.291083266709218">
    <ele>-0.18</ele>
    <time>2023-03-15T10:45:24Z</time>
  </wpt>
</gpx>
```

Οι χρήστες μέσω του mobile app ανεβάζουν αυτό το GPX αρχείο στο σύστημα και το σύστημα εκτελεί κάποιο processing πάνω σε αυτό το αρχείο. Ένα GPX αρχείο περιέχει μόνο μια δραστηριότητα / διαδρομή.

Συνήθως τα waypoints δημιουργούνται ανά  $x$  μέτρα και αυτό εξαρτάται από την ακρίβεια του GPS της συσκευής. Συνεπώς το μέγεθος του αρχείου μεγαλώνει ανάλογα με τη συνολική απόσταση που καλύφθηκε κατά τη διάρκεια της άσκησης. Η επεξεργασία του αρχείου μπορεί να γίνει παράλληλα από πολλά μηχανήματα με μορφή MapReduce ώστε να επιταχυνθεί η επεξεργασία μεγάλων αρχείων.

Το MapReduce framework είναι ένα προγραμματιστικό μοντέλο που επιτρέπει την παράλληλη επεξεργασία μεγάλων όγκων δεδομένων.

Το MapReduce στηρίζεται στη χρήση δύο συναρτήσεων:

`map(key,value) -> [(key2, value2)]`

`reduce(key2,[value2]) -> [final_value]`

- "Map" συνάρτηση: επεξεργάζεται ένα ζεύγος key/value και παράγει ένα ενδιάμεσο ζεύγος key/value. Η είσοδος στη συνάρτηση map μπορεί να είναι γραμμές ενός αρχείου κλπ., και έχουν τη μορφή (κλειδί, τιμή). Η συνάρτηση map μετατρέπει κάθε τέτοιο ζευγάρι σε ένα άλλο ζευγάρι (κλειδί2, τιμή2). Η map συνάρτηση μπορεί να εκτελείται παράλληλα, πάνω σε διαφορετική είσοδο δεδομένων και σε διαφορετικούς κόμβους. Ο βαθμός παραλληλίας εξαρτάται από την εφαρμογή και μπορεί να την ορίσει ο χρήστης.
- "Reduce" συνάρτηση: συγχωνεύει όλα τα ενδιάμεσα values που σχετίζονται με το ίδιο κλειδί και παράγει τα τελικά αποτελέσματα. Για κάθε ξεχωριστό κλειδί δημιουργείται μια λίστα από τις τιμές που αντιστοιχούν σε αυτό το κλειδί. Η συνάρτηση αυτή υπολογίζει μια τελική τιμή για το κλειδί, επεξεργάζοντας τη λίστα των τιμών που αντιστοιχούν σε αυτό το κλειδί. Η επεξεργασία της συνάρτησης reduce γίνεται αφού έχει τελειώσει η επεξεργασία όλων των map συναρτήσεων.

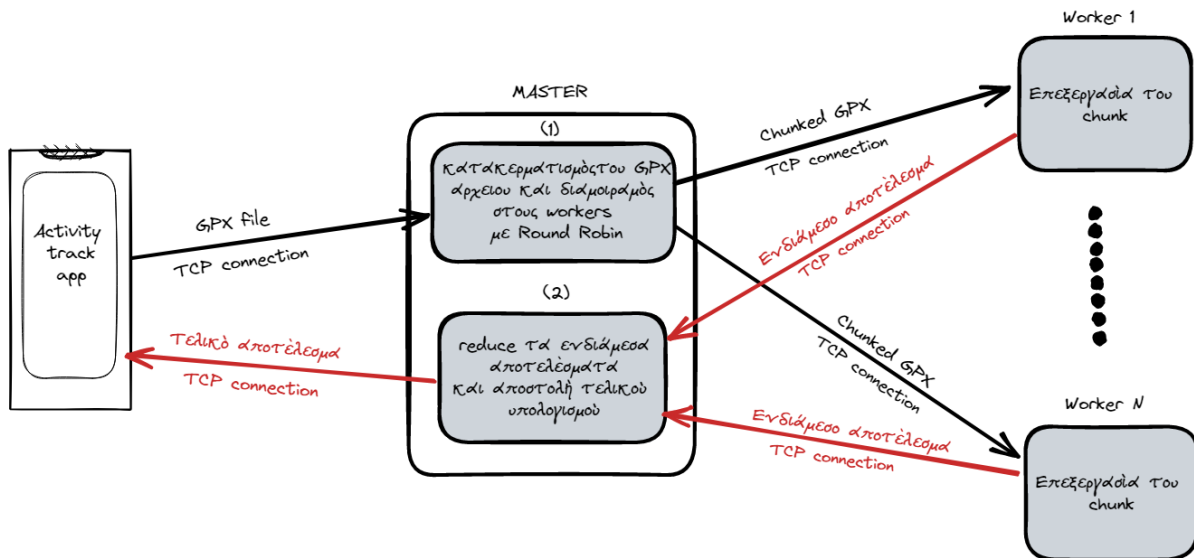
Το mobile application αρχικά στέλνει το GPX σε έναν Master Node. Στη συνέχεια ο Master Node δημιουργεί chunks από  $n$  waypoints και τα στέλνει στους Worker Nodes με σειρά Round Robin. Ο κάθε Worker υπολογίζει για το chunk που λαμβάνει την συνολική απόσταση, την μέση ταχύτητα, τη συνολική ανάβαση και το συνολικό χρόνο. Στη συνέχεια γυρίζει αυτά τα intermediate results πίσω στον Master ώστε να γίνουν reduce στο τελικό αποτέλεσμα. Συνεπώς στη συγκεκριμένη έκδοχή του συστήματος ο Master αποτελεί και τον Reducer.

→ Όταν ο Master λάβει όλα τα intermediate results και ολοκληρώσει το reduce, βγάζει τα τελικά αποτελέσματα για το activity. Τέλος, ασύγχρονα προωθεί τα αποτελέσματα πίσω στο mobile application ώστε να τα δει ο χρήστης.

→ Παράλληλα ο Master πρέπει να διατηρεί στατιστικά από όλους τους χρήστες. Συγκεκριμένα διατηρεί τον **Μέσο Χρόνο Άσκησης**, τη **Μέση Απόσταση** και τη **Μέση Ανάβαση** τόσο για κάθε χρήστη ξεχωριστά, όσο και για το σύνολο των χρηστών.

### Απαιτήσεις υλοποίησης Backend:

- Ο **Master** πρέπει να υλοποιηθεί σε **Java** και να υλοποιεί **TCP Server**. **Δεν επιτρέπεται** η χρήση έτοιμων libraries πέρα των default ServerSocket της Java ή HTTP πρωτοκόλλου με τη χρήση έτοιμου server, όπως της Java ή Apache.
- Ο **Master** πρέπει να είναι **πολυνηματικός** και να μπορεί να εξυπηρετεί πολλούς χρήστες ταυτόχρονα και να επικοινωνεί ταυτόχρονα με τους workers.
- Οι **Workers** πρέπει να υλοποιηθούν σε **Java** και να είναι **πολυνηματικοί** για να εκτελούν παράλληλα πολλά requests από τον Master.
- Οι **Workers** θα πρέπει να ορίζονται δυναμικά κατά το initialization του Master (από τα arguments ή config file) και ο αριθμός τους θα μπορεί να είναι αυθαίρετος.
- Η επικοινωνία Master / Worker να υλοποιείται και αυτή **αποκλειστικά μέσω TCP sockets**. Συγκεκριμένα οι Workers πρέπει να ανοίξουν Socket με τον Master. Συνεπώς **TCP Server θα υλοποιεί μόνο ο Master**. Μπορείτε να υλοποιήσετε δύο διαφορετικά TCP Servers στον Master, που να ακούν σε διαφορετικά ports. Ένα για την επικοινωνία με τους διαφορετικούς χρήστες που χρησιμοποιούν το Application και ένα για την εσωτερική επικοινωνία με τους Workers.
- Πρέπει να υπάρχει συγχρονισμός στα σημεία που κρίνετε απαραίτητο. Ο συγχρονισμός πρέπει να γίνει **αποκλειστικά** χρησιμοποιώντας τεχνικές **synchronized, wait - notify** και όχι με τη χρήση έτοιμων εργαλείων της βιβλιοθήκης *java.util.concurrent* ή άλλων έτοιμων εργαλείων.
- Δεν απαιτείται η χρήση αρχείων για την αποθήκευση των στατιστικών και προφίλ των χρηστών. Όλες οι δομές δεδομένων μπορούν να είναι αποθηκευμένες στην μνήμη του Master. **Απαγορεύεται η χρήση βάσης δεδομένων**.



### Απαιτήσεις υλοποίησης Frontend:

Θα αναπτύξετε μια εφαρμογή που θα εκτελείται σε συσκευές με λειτουργικό Android και θα αποτελεί interface για το σύστημα. Μέσω αυτής ο χρήστης:

- Θα μπορεί να επιλέξει ένα GPX αρχείο, αποθηκευμένο στη συσκευή του και να το στείλει στο backend για επεξεργασία **ασύγχρονα**.
- Θα πρέπει να μπορεί να λάβει ειδοποίηση από τον Master ότι η επεξεργασία τελείωσε και να μπορεί να δείξει τα αποτελέσματα της επεξεργασίας του GPX (συνολική απόσταση, μέση ταχύτητα, συνολική ανάβαση και το συνολικό χρόνο).
- Επίσης θα μπορεί να δει προσωπικά του στατιστικά (**Συνολικό Χρόνο Άσκησης, τη Συνολική Απόσταση και τη Συνολική Ανάβαση**) που έχει κάνει και να προβάλει με γραφικό τρόπο τη διαφορά τους από τον γενικό μέσο όρο (π.χ. έχει τρέξει συνολικά 24% περισσότερο από τον μέσο όρο).
- Η επικοινωνία του Application με τον Master θα πρέπει να γίνεται αποκλειστικά με τη χρήση **TCP Sockets**. Το Application θα πρέπει να **συνδέεται με TCP Socket στον Master**. Μέσω αυτού του Socket αποστέλλεται το GPX. Ο Master αποστέλλει πίσω τα αποτελέσματα της επεξεργασίας **μέσω του ίδιου Socket** το οποίο παραμένει ανοιχτό έως ότου αυτά ληφθούν. **Αυτή η διαδικασία θα πρέπει να υλοποιηθεί με τη χρήση Threads, ώστε η εφαρμογή να παραμένει διαδραστική μέχρι να ληφθούν τα αποτελέσματα.**

### Bonus (+20%)

Σε πολλές εφαρμογές, όπως το Strava, οι χρήστες μπορούν να ορίσουν μια ακολουθία από **Waypoints ως Segment**, π.χ. τη ένα μικρό κομμάτι από τη διαδρομή του Κλασικού Μαραθωνίου Αθήνας. Στη συνέχεια κάθε φορά που το σύστημα εντοπίζει μια υπο-ακολουθία

από Waypoints που ταυτίζεται με κάποιο Segment **μπορεί και κρατάει τα προηγούμενα στατιστικά των χρηστών για αυτό το κομμάτι**. Παράλληλα διατηρεί ένα leaderboard με τις επιδόσεις όλων των χρηστών σε φθίνουσα σειρά. Καλείστε να εντάξετε αυτή τη επιπλέον λειτουργία στο σύστημα τροποποιώντας κατάλληλα το MapReduce.

Τέλος το android application θα πρέπει να βγάζει και με γραφικό τρόπο, πχ. πίνακα, το **leaderboard για το επιλεγμένο segment**.

**Παρατήρηση:** Δύο συσκευές στο ίδιο σημείο μπορεί να διαβάζουν ελαφρώς διαφορετικές συντεταγμένες. πχ να δείχνουν διαφορά 5m. Αυτό ονομάζεται GPS Drift.

## Παραδοτέα εργασίας

Το project θα παραδοθεί σε **δύο** φάσεις:

### Παραδοτέο Α: (Ημερομηνία παράδοσης: 30/04/2023)

Στο παραδοτέο αυτό, θα πρέπει να έχετε ολοκληρώσει εντελώς το backend σύστημα, όπως ακριβώς σας έχει ζητηθεί, έτσι ώστε να μπορεί να χρησιμοποιηθεί στην επόμενη φάση της εργασίας του μαθήματος. Αντί για android application θα έχετε μια dummy εφαρμογή όπου θα στέλνει το GPX στον Master και θα λαμβάνει τα αποτελέσματα τα οποία θα αποθηκεύει σε αρχείο.

### Παραδοτέο Β: (Ημερομηνία παράδοσης: 11/06/2023)

Το παραδοτέο αυτό αποτελεί το Android application, που περιγράφηκε παραπάνω. Στη φάση αυτή το σύστημα θα πρέπει να είναι πλήρως λειτουργικό και ολοκληρωμένο, με όλα τα components του να λειτουργούν σωστά.

**Ομάδες:** Όλοι οι φοιτητές θα πρέπει να σχηματίσουν ομάδες των τριών (3) ατόμων προκειμένου να εκπονήσουν την προγραμματιστική τους εργασία. Γλώσσα προγραμματισμού θα είναι η Java, στην οποία και θα παρέχεται υποστήριξη από τους βοηθούς του μαθήματος.

**Για το πλαίσιο της εργασίας θα σας δώσουμε GPX αρχεία εμείς αλλά μπορείτε να φτιάξετε και τα δικά σας ή να φτιάξετε segments για το bonus με το [5].**

Αναφορές – Χρήσιμοι Σύνδεσμοι:

[1] <https://docs.oracle.com/javase/8/docs/api/java/util/stream/package-summary.html>

[2] Android. URL : <http://code.google.com/android/>

[3] Android SDK: <http://developer.android.com/sdk/index.html>

[4] Android Studio <http://developer.android.com/sdk/index.html>

[5] GPX Generator <https://www.gpxgenerator.com/>