ARISTOTLE
UNIVERSITY OF
THESSALONIKI

# Κατανεμημένα και Διαδικτυακά Συστήματα

## Apache Hadoop

Αστέριος Χουλιάρας ΑΕΜ:2428
Κωνσταντίνος Μπένος ΑΕΜ:2384

13 Ιουνίου 2016

`https://github.com/asterisch/hadoop_kds`

# 1 Περιγραφή

## 1.1 Γενικά

*Το θέμα αυτής της εργασίας είναι η μελέτη της τεχνολογίας hadoop και της λογικής του Map-Reduce που βρίσκεται πίσω από αυτό.Το σενάριο είναι ό,τι έχουμε ένα κοινωνικό δίκτυο που περιέχει κάποιους χρήστες που σχετίζονται μεταξύ τους, δηλαδή είναι φίλοι και άλλους που προφανώς δεν σχετίζονται.Σκοπός είναι να δημιουργήσουμε ένα σύστημα το οποίο θα προτείνει φίλους σε κάποιους χρήστες οι οποίοι δεν είναι ήδη φίλοι μέσα στο δίκτυο αλλά έχουν κάποια πιθανότητα να γνωρίζονται πέρα από αυτό.Το συμπέρασμα εξάγεται με βάση το πόσους κοινούς φίλους έχουν οι δύο χρήστες με τους φίλους τους, δηλαδή αν ο αριθμός των κοινών φίλων κάποιου χρήστης, με έναν φίλο του ξεπερνά κάποιο όριο τότε μπορούμε να με ασφάλεια να προτείνουμε και στους δύο τους μη-κοινούς φίλους τους.*

*Η βασική διαφορά αυτής της τεχνολογίας είναι ότι ο αλγόριθμος που βασίζεται σ' αυτήν, έχει τη δυνατότητα να τρέχει παράλληλα σε μια συστοιχία υπολογιστών (cluster) επιτυγχάνοντας υψηλή απόδοση στην επεξεργασία των δεδομένων.Κάτι τέτοιο είναι σχεδόν απαραίτητο, αφού μιλάμε για δεδομένα των οποίων το μέγεθος σχετίζεται με μία έκρηξη συνδυασμών.Για παράδειγμα, αν το δίκτυο έχει $n$ χρήστες τότε τα δεδομένα που πρέπει να επεξεργαστούν είναι $(n * (n-1))$ ,αφού κάθε χρήστης μπορεί να έχει $(n-1)$ φίλους.*

## 1.2 Generator

*Υπάρχει ένας απλός γεννήτορας ο οποίος δέχεται μια λίστα από τα ονόματα των χρηστών χωρισμένα με κενό και παράγει ένα θεωρητικό τυχαίο σε ένα αρχείο, όπου κάθε γραμμή αναπαριστά: [όνομα χρήστη] [κενό] [ονόματα φίλων χωρισμένα με κόμμα].*

# 2 Βασικά σημεία του κώδικα

## 2.1 Η κλάση Friends_recommendation

*Η κλάση αυτή αποτελεί την βασική κλάση (main) του προγράμματος, όπου αυτό ξεκινάει ορίζοντας τις παραμέτρους των λειτουργιών των συναρτήσεων του hadoop.Συγκεκριμένα ορίζονται:*

1. *Το όνομα της εργασίας στο περιβάλλον hadoop.*
2. *Οι διαδρομές των αρχείων εισόδου και ο φάκελος εξόδου στο σύστημα αρχείων του hadoop.*
3. *Το όνομα των κλάσεων των λειτουργιών **MAP-REDUCE**.*
4. *Ο τύπος της παραμέτρου κλειδιού της **REDUCE** (τύπος κλειδιού εξόδου της **MAP**) που είναι μια custom κλάση .*
5. *Ο τύπος της τιμής που σχετίζεται με το κλειδί, ως Text.*
6. *Ο τύπος των δεδομένων εξόδου της **REDUCE** κλειδιού-τιμής ως Text.*
7. *Η μορφή του αρχείου εισόδου ως* $user[space][friends]$ *για κάθε γραμμή.*

*Τέλος, ξεκινάει η διαδικασία επεξεργασίας των δεδομένων στέλνοντας τη δουλειά στον διαχειριστή εργασιών της υπηρεσίας hadoop.*

## 2.2 Η κλάση Friend_tuple

*Είναι η custom κλάση την οποία χρησιμοποιεί οι συνάρτηση Reduce ως κλειδί για τα δεδομένα.Αποτελείται από δύο πεδία τύπου String τα οποία είναι τα ονόματα δύο χρηστών και όλες τις μεθόδους που κληρονομεί από την κλάση του hadoop WritableComparable.Αυτές οι μέθοδοι αφορούν στο να γράφεται, να διαβάζεται και να συγκρίνεται σωστά ένα αντικείμενο αυτής της κλάσης κατά την διάρκεια της λειτουργίας των ενεργειών Map και Reduce.*

## 2.3 Map - Reduce

*Η βασική ιδέα είναι να χωρίσουμε τα δεδομένα εισόδου μας σύμφωνα κάποιο κριτήριο, όπως ορίζει το **MAP**, ώστε να μπορούν να επεξεργαστούν παράλληλα και ξεχωριστά για τους διαφορετικούς χρήστες, με το **REDUCE**.*

### 2.3.1 MAP

*Ο ρόλος της είναι να διαβάζει το αρχείο με τα δεδομένα εισόδου και να τα αντιστοιχεί σε μια δομή τύπου κλειδί-τιμές.Σ' αυτήν την περίπτωση έχουμε ότι κάθε γραμμή του αποτελεί μια σχέση ανάμεσα σε κάποιους χρήστες και σκοπός μας είναι να αντιστοιχίσουμε τις λίστες των φίλων-των φίλων ενός χρήστη στο όνομα του καθώς και τους δικούς του φίλους, ώστε να μπορούμε να μετρήσουμε τους κοινούς μεταξύ αυτού και των φίλων του.Εάν το επιτύχουμε αυτό μπορούμε να εξάγουμε προτεινόμενους φίλους για τον χρήστη, με βάση των αριθμό των κοινών γνωστών με τους φίλους του.*

*Ο τρόπος με τον οποίο υλοποιείται είναι σε κάθε γραμμή της εισόδου η **MAP** να αντιστοιχεί έναν συνδυασμό δύο χρηστών, ενός του χρήστη, με τους φίλους του χρήστη στη μορφή* $[user\ ,\ (friend, "null")]$ *και για κάθε φίλο του χρήστη αντιστοιχεί όλους του πιθανούς συνδυασμούς* $[friend1\ ,\ (friend2, user)]$*, που προκύπτει από το γεγονός ότι δύο φίλοι του χρήστη έχουν κοινό φίλο τον χρήστη.Στο τέλος κάθε χρήστης θα έχει κάθε πιθανό συνδυασμό με έναν φίλο και έναν κοινό φίλο και τους ίδιους τους φίλους του αντιστοιχισμένους σ αυτόν.*

### 2.3.2 REDUCE

*Η μέθοδος αυτήν λαμβάνει από την **MAP** ζευγάρια με κλειδί το όνομα του χρήστη της μορφής είτε* $[user\ ,\ (friend, "null")]$*, που σημαίνει ότι το πρώτο όνομα στο ζεύγος είναι κάποιος από τους φίλους του, είτε* $[user\ ,\ (user1, user2)]$*, που μ' αυτόν τον τρόπο συγκεντρώνει όλους τους κοινούς φίλους με τους υπόλοιπους χρήστες.Στη συνέχεια μετρούνται οι κοινοί φίλου με τους φίλους του και αν κάποιος έχει περισσότερους κοινούς από κάποιο όριο φίλους με τον χρήστη τότε μπορεί να προτείνει τους μη κοινούς*

*του στον χρήστη.Οπότε, αρχίζει μια διαδικασία όπου προτείνονται νέοι φίλοι στον χρήστη από αυτούς που μπορούν να προτείνουν.Από όλους τους συνδυασμούς που διαθέτει ο χρήστης, αν κάποιος δεύτερος χρήστης είναι φίλος με κάποιον φίλο του χρήστη που μπορεί να προτείνει στον χρήστη, τότε ο δεύτερος χρήστης προτείνεται στον χρήστη.* [1]

*Τελικά, στο/στα αρχεία εξόδου κάθε γραμμή είναι μοναδική και περιέχει για κάθε χρήστη που έρχεται σαν κλειδί στην reduce, το όνομα του μαζί με τους προτεινόμενους φίλους του στη μορφή* $[user\ ,\ [recommendations]\ ]$.

[1] Παρατήρηση: Το hadoop διαχειρίζεται δομές δεδομένων στη REDUCE που δηλώνονται στην μνήμη, ώστε αυτές να μην την επιβαρύνουν αποθηκεύοντας και στο δίσκο δεδομένα