

Documentation

Περιεχόμενα

Ενότητα 1 : Θεματολογία Εφαρμογής Ανάλυσης Δεδομένων, σελ. 2

1. Περιγραφή του τομέα εφαρμογής Μεγάλων Δεδομένων που επιλέχθηκε, σελ. 2
2. Επεξήγηση του τύπου ανάλυσης Μεγάλων Δεδομένων που θα εφαρμοστεί, σελ. 2

Ενότητα 2 : Αρχιτεκτονική Εφαρμογής, σελ. 3

1. Ανάλυση της αρχιτεκτονικής που βασίστηκε η συνολική κατασκευή της εφαρμογής, σελ. 3

Ενότητα 3 : Εγχειρίδιο Χρήσης Εφαρμογής, σελ. 4

1. Αναλυτικές οδηγίες εγκατάστασης και εκτέλεσης της εφαρμογής, σελ. 4
2. Αναλυτικά screenshots των διεπαφών της εφαρμογής, σελ. 5

Ενότητα 4 : Αποτελέσματα Εφαρμογής, σελ. 12

1. Παρουσίαση τόσο των λέξεων κλειδιών που επιλέχθηκαν όσο και των συλλεχθέντων δεδομένων, σελ. 12
2. Παρουσίαση και επεξήγηση του configuration file που δημιουργήθηκε για το εργαλείο Flume, σελ. 12
3. Παρουσίαση και επεξήγηση του schema που δημιουργήθηκε για το εργαλείο Avro, σελ. 13
4. Παρουσίαση και επεξήγηση των αποτελεσμάτων που παρήχθησαν από το Spark, σελ. 13
5. Παρουσίαση των ζητημάτων που αντιμετωπίστηκαν κατά την προ-επεξεργασία των δεδομένων, και περιγραφή των διαδικασιών που ακολουθήθηκαν, σελ. 14
6. Παρουσίαση των ιδιοτήτων των συλλεχθέντων δεδομένων που χρησιμοποιήθηκαν για την ανάλυση των δεδομένων, σελ. 14
7. Αναλυτική περιγραφή της εισόδου/εξόδου κάθε πειράματος της ανάλυσης των συλλεχθέντων δεδομένων, σε συνδυασμό με τις αντίστοιχες παραμέτρους που αλλάχθηκαν σε κάθε πείραμα, σελ. 15
8. Σύγκριση όλων των εξαχθέντων αποτελεσμάτων, σελ. 15

Ενότητα 1 : Θεματολογία Εφαρμογής Ανάλυσης Δεδομένων

1. Περιγραφή του τομέα εφαρμογής Μεγάλων Δεδομένων που επιλέχθηκε

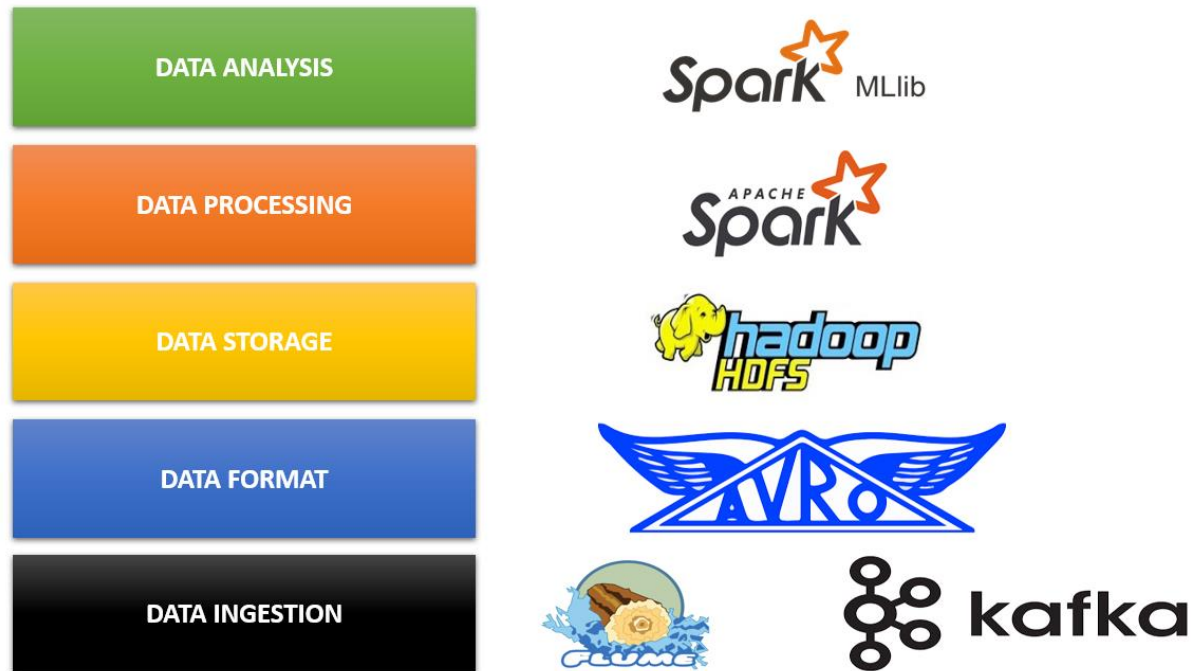
Ο τομέας εφαρμογής Μεγάλων Δεδομένων που επιλέχθηκε είναι η άθληση (sport). Το πρόβλημα που καλείται να επιλύσει η εφαρμογή στον εν λόγω τομέα είναι να βρεθεί ποιο είναι το πιο δημοφιλές άθλημα ανάμεσα στους χρήστες του Twitter. Πιο συγκεκριμένα, θα συγκριθούν ως προς την δημοφιλία τους τα εξής αθλήματα : football, basketball, volleyball, tennis, baseball, cricket, hockey. Η ανάγκη επίλυσης του συγκεκριμένου προβλήματος έγκειται στο γεγονός να βρεθεί για ποιο από τα παραπάνω αθλήματα υπάρχει περισσότερη ενασχόληση από την πλευρά των χρηστών στο Twitter.

2. Επεξήγηση του τύπου ανάλυσης Μεγάλων Δεδομένων που θα εφαρμοστεί

Ο τύπος ανάλυσης Μεγάλων Δεδομένων που θα εφαρμοστεί είναι η περιγραφική ανάλυση (descriptive analysis), η οποία απαντάει στο ερώτημα τι έχει συμβεί και μετατρέπει ακατέργαστα δεδομένα σε κατανοητά κομμάτια. Ο τρόπος που σχετίζεται η περιγραφική ανάλυση με το πρόβλημα που προσπαθεί να επιλύσει η εφαρμογή, το οποίο είναι να βρεθεί το πιο δημοφιλές άθλημα ανάμεσα στους χρήστες του Twitter, είναι ότι η εφαρμογή ψάχνει τις δραστηριότητες των χρηστών που έχουν ήδη συμβεί και καταλήγει σε ένα τελικό συμπέρασμα.

Ενότητα 2 : Αρχιτεκτονική Εφαρμογής

1. Ανάλυση της αρχιτεκτονικής που βασίστηκε η συνολική κατασκευή της εφαρμογής



Αρχικά, ξεκινώντας από κάτω προς τα πάνω, τα εργαλεία Flume και Kafka χρησιμοποιούνται για την συλλογή δεδομένων, το εργαλείο Avro είναι το format στο οποίο αποθηκεύονται τα δεδομένα και το εργαλείο HDFS είναι η τοποθεσία που μένουν αποθηκευμένα τα δεδομένα. Στην συνέχεια, με το εργαλείο Spark υπολογίζονται ορισμένα στατιστικά για τα συλλεχθέντα δεδομένα και τέλος, με το εργαλείο MLlib πραγματοποιείται η ανάλυση των δεδομένων και εξάγονται τα συμπεράσματα της εφαρμογής.

Ενότητα 3 : Εγχειρίδιο Χρήσης Εφαρμογής

1. Αναλυτικές οδηγίες εγκατάστασης και εκτέλεσης της εφαρμογής

Αρχικά, για να εγκαταστήσει κάποιος την εφαρμογή θα πρέπει να εγκαταστήσει την Java στην τοποθεσία C:\ και όχι σε κάποια τοποθεσία που περιέχει κενό στην ονομασία της.

JDK : <https://www.filehorse.com/download-java-development-kit-64/55825/>

JRE : https://download.cnet.com/Java-Runtime-Environment-JRE/3000-2213_4-10009607.html

Στην συνέχεια, αφού το source της εφαρμογής είναι το Twitter θα πρέπει να δημιουργήσει έναν λογαριασμό Twitter developer : <https://apps.twitter.com/> , για να μπορέσει να συμπληρώσει τα απαραίτητα στοιχεία στο αρχείο Page2.jsp που βρίσκεται στον πηγαίο κώδικα της εφαρμογής :

```
String consumerKey = "";  
String consumerSecret = "";  
String token = "";  
String secret = "";
```

Επίσης, θα πρέπει να εγκαταστήσει το εργαλείο Flume στην τοποθεσία C:\ και όχι σε κάποια τοποθεσία που περιέχει κενό στην ονομασία της :

<http://www.apache.org/dyn/closer.lua/flume/1.9.0/apache-flume-1.9.0-bin.tar.gz>

Επιπλέον, θα πρέπει να εγκαταστήσει το εργαλείο Kafka στην τοποθεσία C:\ και όχι σε κάποια τοποθεσία που περιέχει κενό στην ονομασία της :

https://www.apache.org/dyn/closer.cgi?path=/kafka/2.7.0/kafka_2.13-2.7.0.tgz

Επιπρόσθετα, θα πρέπει να εγκαταστήσει το εργαλείο Spark στην τοποθεσία C:\ και όχι σε κάποια τοποθεσία που περιέχει κενό στην ονομασία της :

<https://spark.apache.org/downloads.html>

1. Choose a Spark release:

2. Choose a package type:

3. Download Spark: [spark-3.1.2-bin-hadoop2.7.tgz](#)

Στην συνέχεια, θα πρέπει να εγκαταστήσει το IDE που είναι το Eclipse :

<https://www.eclipse.org/downloads/packages/release/2020-06/r/eclipse-ide-enterprise-java-developers> και να κάνει import τον φάκελο του project της εφαρμογής.

Και πρέπει να εγκαταστήσει το Apache Tomcat :

<https://tomcat.apache.org/download-90.cgi> και να το προσθέσει στο Eclipse.

Στην συνέχεια, θα πρέπει να έχει το αρχείο `twit.conf` στην τοποθεσία `C:\apache-flume-1.9.0-bin\conf` και το αρχείο `twitter.avsc` σε οποιοδήποτε path θέλει, αρκεί να γίνει η αντίστοιχη αλλαγή στο `Page2.jsp` που βρίσκεται στον πηγαίο κώδικα της εφαρμογής :

```
Schema schema = new Schema.Parser().parse(new File("C:/Users/USER/Desktop/My_Avro_Twitter/schema/twitter.avsc"));
```

Και τέλος, για να μπορέσει να εκτελέσει την εφαρμογή σωστά θα πρέπει από πριν πρώτα να εκτελέσει το cmd ως admin και να εισάγει τις εξής εντολές για να εκκινήσει το εργαλείο HDFS :

1) `cd C:\hadoop-3.1.0\sbin`

2) `start-all.cmd`

Μετά, θα πρέπει να ξανά εκτελέσει το cmd ως admin και να εισάγει τις εξής εντολές για να εκτελεστεί το Zookeeper :

1) `cd C:\kafka_2.13-2.7.0`

2) `.\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties`

Και πρέπει να ξανά εκτελέσει το cmd ως admin και να εισάγει τις εξής εντολές για να εκτελεστεί το Kafka :

1) `cd C:\kafka_2.13-2.7.0`

2) `.\bin\windows\kafka-server-start.bat .\config\server.properties`

Για να εκτελέσει την εφαρμογή θα πρέπει να εκκινήσει το αρχείο `Page1.jsp` μέσα από το Eclipse και να επιλέξει το Tomcat ως server για εκκινήσει η πρώτη σελίδα της εφαρμογής : <http://localhost:8080/BigDataProject/Page1.jsp>

2. Αναλυτικά screenshots των διεπαφών της εφαρμογής

Η πρώτη σελίδα της εφαρμογής :



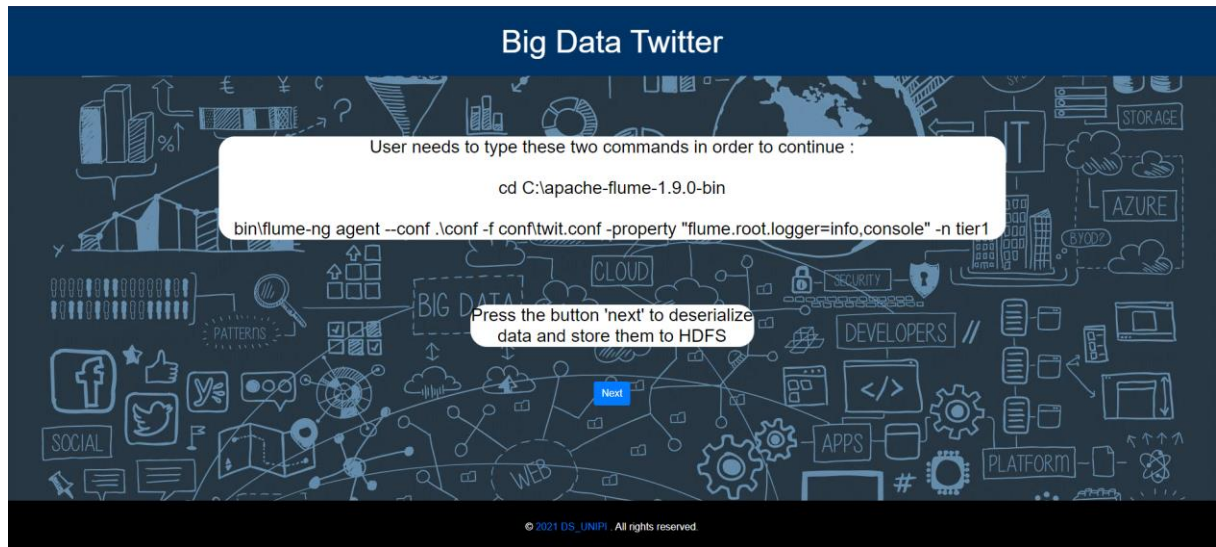
Εμφανίζει στον χρήστη τις 7 λέξεις κλειδιά με βάσει τις οποίες θα γίνει η αναζήτηση των δεδομένων.

Αν πατήσει το κουμπί search, τότε θα περάσει στην δεύτερη σελίδα της εφαρμογής (Page2.jsp), όπου με το εργαλείο Kafka συλλέγονται δεδομένα από το Twitter και αποθηκεύονται στο topic my_tweets_topic με partition 0 σε Avro format και εμφανίζονται και τα 500 Tweets στην διεπαφή του χρήστη. Πιο συγκεκριμένα, για κάθε Tweet τα χαρακτηριστικά που αποθηκεύονται και εμφανίζονται στον χρήστη είναι το text, που είναι το κείμενο του Tweet, το followers_count, που είναι ο συνολικός αριθμός των χρηστών που ακολουθούν τον λογαριασμό που δημοσίευσε το Tweet και το statuses_count, που είναι ο συνολικός αριθμός των Tweets που έχει δημοσιεύσει ο λογαριασμός :

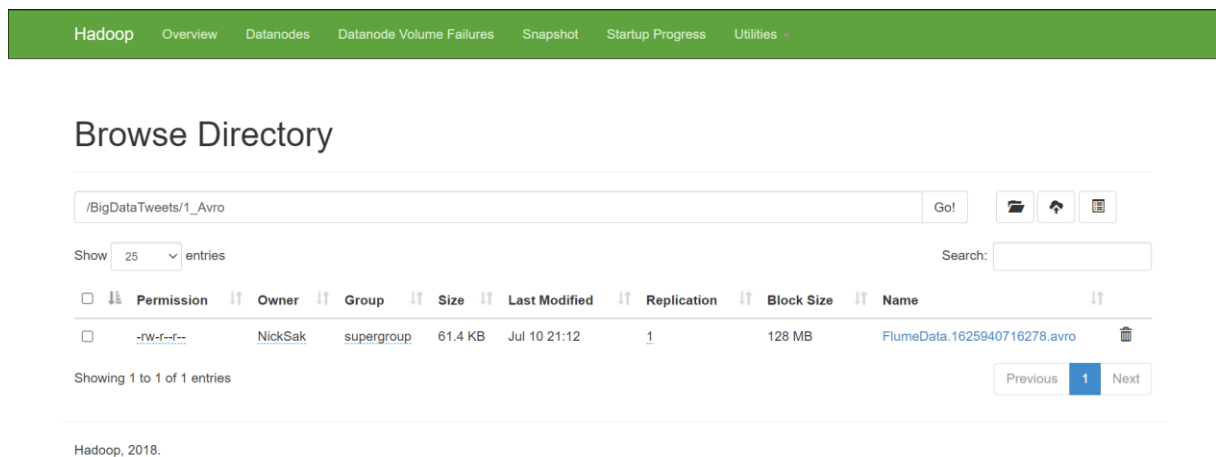


Στην συνέχεια, αν πατήσει το κουμπί next που βρίσκεται στην κορυφή της διεπαφής, θα περάσει στην τρίτη σελίδα της εφαρμογής (Page3.jsp) και με την χρήση του εργαλείου Flume, όλα τα δεδομένα που αποθηκεύτηκαν στο Kafka θα μεταφερθούν στο HDFS σε Avro format στην τοποθεσία :
hdfs://localhost:9000/BigDataTweets/1_Avro.

Για να γίνει αυτό, η εφαρμογή ανοίγει το cmd αυτόματα και ζητείται από τον χρήστη να εισάγει 2 εντολές :



HDFS :



Στην συνέχεια, αν πατήσει το κουμπί next, θα περάσει στην τέταρτη σελίδα της εφαρμογής (Page4.jsp). Αρχικά, γίνεται deserialize των δεδομένων που αποθηκεύτηκαν στο HDFS και εμφανίζονται στην διεπαφή του χρήστη. Στην συνέχεια τα deserialized δεδομένα αποθηκεύονται στο HDFS στην τοποθεσία :
hdfs://localhost:9000/BigDataTweets/2_Deserialized/Tweets.txt

Big Data Twitter

Press the button 'next' to see how many times each keyword appears

Next

All 500 tweets that have been deserialized

1)

```
{
  "text": "RT @Joe_Marcincuk: #WhenTheWifesAway it's baseball without guilt @HastroTags https://t.co/ivPjRxoFC0",
  "followers_count": 3691,
  "statuses_count": 33444
}
```

2)

HDFS :

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities

Browse Directory

Go!

Show 25 entries
Search:

| <input type="checkbox"/> | Permission | Owner | Group | Size | Last Modified | Replication | Block Size | Name |
|--------------------------|------------|---------|------------|----------|---------------|-------------|------------|----------------------------|
| <input type="checkbox"/> | -rw-r--r-- | NickSak | supergroup | 89.23 KB | Jul 10 21:13 | 3 | 128 MB | Tweets.txt |

Showing 1 to 1 of 1 entries

Previous
1
Next

Hadoop, 2018.


Στην συνέχεια, αν πατήσει το κουμπί next, θα περάσει στην πέμπτη σελίδα της εφαρμογής (Page5.jsp) όπου υπολογίζεται πόσες φορές κάθε keyword εμφανίζεται στα δεδομένα που συλλέχθηκαν με την χρήση του εργαλείου Spark, και τα αποτελέσματα αποθηκεύονται στο HDFS στην τοποθεσία :

hdfs://localhost:9000/BigDataTweets/3_SparkResults/Statistics.txt

Σημαντική σημείωση : Όταν εκτελείται το συγκεκριμένο αρχείο, βγάζει το εξής σφάλμα :


```
Stacktrace:] with root cause
com.fasterxml.jackson.databind.JsonMappingException: Scala module 2.10.0 requires Jackson Databind version >= 2.10.0 and < 2.11.0
    at com.fasterxml.jackson.module.scala.JacksonModule.setupModule(JacksonModule.scala:61)
    at com.fasterxml.jackson.module.scala.JacksonModule.setupModule$(JacksonModule.scala:46)
    at com.fasterxml.jackson.module.scala.DefaultScalaModule.setupModule(DefaultScalaModule.scala:17)
    at com.fasterxml.jackson.databind.ObjectMapper.registerModule(ObjectMapper.java:835)
    at org.apache.spark.rdd.RDDOperationScope$.<init>(RDDOperationScope.scala:82)
```

Αναφέρεται στο ότι το Scala module 2.10.0 απαιτεί η έκδοση του Jackson Databind να είναι $\geq 2.10.0$ and $< 2.11.0$. Όμως, στα αρχεία jar της εφαρμογής η έκδοση του Jackson Databind είναι 2.10.0 :

```
>  jackson-databind-2.10.0.jar - BigDataProject/WebContent/WEB-INF/lib
```

Για να προσπαθήσω να λύσω αυτό το σφάλμα, έκανα πολλούς διαφορετικούς συνδυασμούς των αρχείων jar, χωρίς όμως αποτέλεσμα. Ξόδεψα πάρα πολύ χρόνο, για να μπορέσω να το λύσω αλλά δεν τα κατάφερα. Όμως, αν εκτελεστεί ο ίδιος ακριβώς κώδικας σε ένα Java Project το οποίο έχει τα ίδια ακριβώς jar αρχεία με την εφαρμογή, τότε εκτελείται χωρίς κανένα πρόβλημα. Μόνο σε Dynamic Web Projects εμφανίζεται αυτό το σφάλμα χωρίς να καταλαβαίνω γιατί. Οπότε για τις ανάγκες της εργασίας αυτό που έκανα είναι να εκτελέσω τον κώδικα σε ένα Java Project. Αυτό που χρειάζεται είναι σαν input το αρχείο με τα deserialized δεδομένα : `hdfs://localhost:9000/flume10/2_Deserialized/Tweets.txt` και υπολογίζει τα στατιστικά και εξάγει τα αποτελέσματα στο HDFS : `hdfs://localhost:9000/flume10/3_SparkResults/Statistics.txt`

Ο κώδικας δηλαδή δουλεύει σωστά, απλά δεν υπάρχει η αλληλεπίδραση με τον χρήστη.

Οπότε, τα αποτελέσματα είναι :

```
The keyword football appears 283 times
The keyword basketall appears 61 times
The keyword volleyball appears 6 times
The keyword tennis appears 53 times
The keyword baseball appears 58 times
The keyword cricket appears 43 times
The keyword hockey appears 5 times
```

HDFS :

[Hadoop](#) [Overview](#) [Datanodes](#) [Datanode Volume Failures](#) [Snapshot](#) [Startup Progress](#) [Utilities](#)

Browse Directory

Show 25 entries

Search:

| <input type="checkbox"/> | Permission | Owner | Group | Size | Last Modified | Replication | Block Size | Name | |
|--------------------------|------------|---------|------------|-------|---------------|-------------|------------|--------------------------------|--|
| <input type="checkbox"/> | -rw-r--r-- | NickSak | supergroup | 262 B | Jul 10 21:16 | 3 | 128 MB | Statistics.txt | |

Showing 1 to 1 of 1 entries

[Previous](#) [1](#) [Next](#)

Στην συνέχεια, στην σελίδα Page6.jsp πραγματοποιείται η προ-επεξεργασία των δεδομένων, εμφανίζονται στην διεπαφή τα προ-επεξεργασμένα δεδομένα και αποθηκεύονται στο HDFS στην τοποθεσία :

hdfs://localhost:9000/BigDataTweets/4_Preprocessing/Data.txt

Big Data Twitter

Press the button 'next' to implement the clustering algorithm KMeans using MLlib on the preprocessed data

First we preprocess the data and store them to hdfs




1 means the text contained the word football
2 means the text contained the word basketball
3 means the text contained the word tennis
4 means the text contained the word volleyball
5 means the text contained the word baseball
6 means the text contained the word cricket
7 means the text contained the word hockey

1)
Sport : 5
Followers count : 3691
Statuses count : 33444











HDFS :

[Hadoop](#) [Overview](#) [Datanodes](#) [Datanode Volume Failures](#) [Snapshot](#) [Startup Progress](#) [Utilities](#)

Browse Directory

Show entries Search:

| <input type="checkbox"/> |  Permission |  Owner |  Group |  Size |  Last Modified |  Replication |  Block Size |  Name |  |
|--------------------------|--|---|---|--|---|---|--|--|---|
| <input type="checkbox"/> | -rw-r--r-- | NickSak | supergroup | 6.06 KB | Jul 10 21:18 | 3 | 128 MB | Data.txt |  |




Showing 1 to 1 of 1 entries Previous **1** Next

Hadoop, 2018.












Και στην σελίδα Page7.jsp υλοποιείται ο αλγόριθμος KMeans πάνω στα προεπεξεργασμένα δεδομένα και αποθηκεύονται τα εξαχθέντα αποτελέσματα στο HDFS στην τοποθεσία : `hdfs://localhost:9000/BigDataTweets/5_KMeansResults`

[Hadoop](#) [Overview](#) [Datanodes](#) [Datanode Volume Failures](#) [Snapshot](#) [Startup Progress](#) [Utilities](#)

Browse Directory

Show entries Search:

| <input type="checkbox"/> |  Permission |  Owner |  Group |  Size |  Last Modified |  Replication |  Block Size |  Name |  |
|--------------------------|--|---|---|--|---|---|--|--|---|
| <input type="checkbox"/> | drwxr-xr-x | NickSak | supergroup | 0 B | Jul 10 21:21 | 0 | 0 B | data |  |
| <input type="checkbox"/> | drwxr-xr-x | NickSak | supergroup | 0 B | Jul 10 21:21 | 0 | 0 B | metadata |  |

Showing 1 to 2 of 2 entries Previous **1** Next

Hadoop, 2018.

```
Cluster centers:
[2.2705410821643284,2985.49498997996,31911.414829659316]
[1.0,1328.0,2295918.0]
```

Σε αυτό το αρχείο υπάρχει το ίδιο σφάλμα που υπήρχε και στο Page5.jsp

```
Stacktrace:] with root cause
com.fasterxml.jackson.databind.JsonMappingException: Scala module 2.10.0 requires Jackson Databind version >= 2.10.0 and < 2.11.0
    at com.fasterxml.jackson.module.scala.JacksonModule.setupModule(JacksonModule.scala:61)
    at com.fasterxml.jackson.module.scala.JacksonModule.setupModule$(JacksonModule.scala:46)
    at com.fasterxml.jackson.module.scala.DefaultScalaModule.setupModule(DefaultScalaModule.scala:17)
    at com.fasterxml.jackson.databind.ObjectMapper.registerModule(ObjectMapper.java:835)
```

Οπότε εκτελέστηκε σε Java Project ο κώδικας για να αποθηκευτούν τα αποτελέσματα στο HDFS. Ο κώδικας δουλεύει σωστά, απλά δεν υπάρχει η αλληλεπίδραση με τον χρήστη.

Ενότητα 4 : Αποτελέσματα Εφαρμογής

1. Παρουσίαση τόσο των λέξεων κλειδιών που επιλέχθηκαν όσο και των συλλεχθέντων δεδομένων

Οι λέξεις κλειδιά που επιλέχθηκαν είναι συνολικά 7 : football, basketball, tennis, volleyball, baseball, cricket και hockey.

Για δεδομένο που συλλέχθηκε, εμφανίζονται στον χρήστη και αποθηκεύονται 3 χαρακτηριστικά του : το κείμενο του Tweet, πόσοι λογαριασμοί συνολικά ακολουθούν τον χρήστη που δημοσίευσε το Tweet και πόσα συνολικά Tweets έχει κάνει ο χρήστης που δημοσίευσε το Tweet.

```
1 {"text": "RT @Joe_Marcincuk: #WhenTheWivesAway it's baseball without guilt @HastroTags  
https://t.co/ivPjRxoFC0", "followers_count": 3691, "statuses_count": 33444}  
2 {"text": "@EricPMusselman check out this #Arkansas #XNA 16u basketball team in the Atlanta UA  
tourney. #ESPN Top 10 maybe??", "followers_count": 51, "statuses_count": 9}  
3 {"text": "RT @Rockies: The holy grail of baseball cards, a PSA gem mint 10 1952 Topps Mickey Mantle  
rookie card, has arrived in Denver 😊 https://t.co\u2026", "followers_count": 1, "statuses_count":  
642}
```

2. Παρουσίαση και επεξήγηση του configuration file που δημιουργήθηκε για το εργαλείο Flume

Το configuration file που δημιουργήθηκε για το εργαλείο flume, για την συλλογή και την αποθήκευση των συλλεχθέντων δεδομένων είναι :

```
1 tier1.sources = source1  
2 tier1.sinks = sink1  
3 tier1.channels = channel1  
4  
5 tier1.sources.source1.type = org.apache.flume.source.kafka.KafkaSource  
6 tier1.sources.source1.kafka.bootstrap.servers = localhost:9092  
7 tier1.sources.source1.kafka.consumer.auto.offset.reset = earliest  
8 tier1.sources.source1.kafka.topics = my_tweets_topic  
9 tier1.sources.source1.kafka.consumer.group.id = flume  
10 tier1.sources.source1.interceptors = i1  
11 tier1.sources.source1.interceptors.i1.type = timestamp  
12 tier1.sources.source1.kafka.consumer.timeout.ms = 100  
13 tier1.sources.source1.batchSize = 100  
14  
15 tier1.sinks.sink1.type = hdfs  
16 tier1.sinks.sink1.hdfs.path = hdfs://localhost:9000/BigDataTweets/1_Avro  
17 tier1.sinks.sink1.hdfs.fileType = DataStream  
18 tier1.sinks.sink1.hdfs.batchSize = 1000  
19 tier1.sinks.sink1.hdfs.rollSize = 0  
20 tier1.sinks.sink1.hdfs.rollCount = 10000  
21 tier1.sinks.sink1.hdfs.fileSuffix=.avro  
22  
23 tier1.channels.channel1.type = memory  
24 tier1.channels.channel1.capacity = 30000  
25 tier1.channels.channel1.transactionCapacity = 1000  
26  
27  
28 tier1.sources.source1.channels = channel1  
29 tier1.sinks.sink1.channel = channel1
```

Δέχεται σαν source το topic του Kafka που αποθηκεύει στην αρχή της εκτέλεσης της εφαρμογής τα συλλεχθέντα δεδομένα σε Avro format και δέχεται σαν sink το HDFS και αποθηκεύονται τα συλλεχθέντα δεδομένα στο path που έχει ορισθεί.

3. Παρουσίαση και επεξήγηση του schema που δημιουργήθηκε για το εργαλείο Avro

Το schema που δημιουργήθηκε για το εργαλείο Avro, για την κωδικοποίηση/αποκωδικοποίηση των συλλεχθέντων δεδομένων είναι :

```
1 {  
2   "type" : "record",  
3   "name" : "Tweet",  
4   "namespace" : "hadoop.ecosystem",  
5   "fields" : [ {  
6     "name" : "text",  
7     "type" : "string",  
8     "doc" : "The content of the user's Twitter message"  
9   }, {  
10    "name" : "followers_count",  
11    "type" : "int",  
12    "doc" : "The total followers of the account that posted the Tweet"  
13  }, {  
14    "name" : "statuses_count",  
15    "type" : "int",  
16    "doc" : "The total Tweets the account has posted"  
17  } ],  
18   "doc" : "A schema for storing Twitter messages"  
19 }
```

Κάθε δεδομένο είναι ένα record το οποίο περιέχει 3 fields : το text, το οποίο είναι το κείμενο του Tweet, το followers_count, το οποίο είναι πόσοι λογαριασμοί συνολικά ακολουθούν τον χρήστη που δημοσίευσε το Tweet και το statuses_count, το οποίο είναι πόσα συνολικά Tweets έχει κάνει ο χρήστης που δημοσίευσε το Tweet.

4. Παρουσίαση και επεξήγηση των αποτελεσμάτων που παρήχθησαν από το Spark

Τα αποτελέσματα που παρήχθησαν από το Spark είναι :

```
1 The keyword football appears 283 times  
2 The keyword basketall appears 61 times  
3 The keyword volleyball appears 6 times  
4 The keyword tennis appears 53 times  
5 The keyword baseball appears 58 times  
6 The keyword cricket appears 43 times  
7 The keyword hockey appears 5 times
```

Το οποίο ουσιαστικά αναφέρει πόσες φορές εμφανίζεται το κάθε keyword στα 500 δεδομένα που έχουν συλλεχθεί

5. Παρουσίαση των ζητημάτων που αντιμετωπίστηκαν κατά την προ-επεξεργασία των δεδομένων, και περιγραφή των διαδικασιών που ακολουθήθηκαν

Για την προ-επεξεργασία των δεδομένων, το πρώτο ζήτημα που έπρεπε να αντιμετωπιστεί είναι να αποδοθεί ένας αριθμός για κάθε keyword, επειδή ο clustering αλγόριθμος KMeans εφαρμόζεται μόνο πάνω σε numeric δεδομένα. Οπότε για το keyword football αποδόθηκε ο αριθμός 1, για το basketball αποδόθηκε ο αριθμός 2, για το tennis αποδόθηκε ο αριθμός 3, για το volleyball αποδόθηκε ο αριθμός 4, για το baseball αποδόθηκε ο αριθμός 5, για το cricket αποδόθηκε ο αριθμός 6 και για το hockey αποδόθηκε ο αριθμός 7.

Και το αρχείο που δημιουργήθηκε είναι :

| | | | |
|----|---|-------|--------|
| 1 | 5 | 3691 | 33444 |
| 2 | 2 | 51 | 9 |
| 3 | 5 | 1 | 642 |
| 4 | 5 | 259 | 72435 |
| 5 | 2 | 106 | 15812 |
| 6 | 1 | 1167 | 8764 |
| 7 | 1 | 8058 | 12627 |
| 8 | 1 | 610 | 4726 |
| 9 | 1 | 1493 | 53061 |
| 10 | 5 | 31 | 11308 |
| 11 | 1 | 28 | 3492 |
| 12 | 2 | 1067 | 13695 |
| 13 | 3 | 28 | 940 |
| 14 | 1 | 172 | 73978 |
| 15 | 1 | 86 | 1284 |
| 16 | 5 | 11161 | 199170 |
| 17 | 3 | 87 | 3413 |
| 18 | 1 | 520 | 22622 |

Όπου στην δεύτερη στήλη είναι πόσοι λογαριασμοί συνολικά ακολουθούν τον χρήστη που δημοσίευσε το Tweet και στην τρίτη στήλη είναι πόσα συνολικά Tweets έχει κάνει ο χρήστης που δημοσίευσε το Tweet.

6. Παρουσίαση των ιδιοτήτων των συλλεχθέντων δεδομένων που χρησιμοποιήθηκαν για την ανάλυση των δεδομένων

Οι ιδιότητες των συλλεχθέντων δεδομένων που χρησιμοποιήθηκαν για την ανάλυση των δεδομένων είναι δυο, το `followers_count`, το οποίο είναι πόσοι λογαριασμοί συνολικά ακολουθούν τον χρήστη που δημοσίευσε το Tweet και το `statuses_count`, το οποίο είναι πόσα συνολικά Tweets έχει κάνει ο χρήστης που δημοσίευσε το Tweet.

7. Αναλυτική περιγραφή της εισόδου/εξόδου κάθε πειράματος της ανάλυσης των συλλεχθέντων δεδομένων, σε συνδυασμό με τις αντίστοιχες παραμέτρους που αλλάχθηκαν σε κάθε πείραμα

Η είσοδος της ανάλυσης δεδομένων είναι το αρχείο που δημιουργήθηκε με τα numeric δεδομένα και σε κάθε πείραμα γινόντουσαν διάφοροι συνδυασμοί με τις τρεις στήλες που περιέχει. Δηλαδή, στο πρώτο πείραμα υπήρχαν και οι τρεις στήλες, στο δεύτερο πείραμα υπήρχαν η πρώτη και η τρίτη κ.ο.κ. Οι παράμετροι που αλλάχθηκαν σε κάθε πείραμα ήταν οι συνολικές επαναλήψεις, όπου δεν έφερε κάποια ουσιαστική αλλαγή στα εξαχθέντα αποτελέσματα και τα συνολικός αριθμός των clusters που δημιουργούνται, όπου παρατηρήθηκε ότι όσο πιο πολλά clusters δημιουργούνται τότε είναι και πιο πιθανό τα ίδια αθλήματα να βρεθούν και στο ίδιο cluster.

8. Σύγκριση όλων των εξαχθέντων αποτελεσμάτων

Το πιο κομβικό αποτέλεσμα της ανάλυσης των δεδομένων ήταν ότι το πιο δημοφιλές άθλημα με διαφορά ήταν το football και τα λιγότερα δημοφιλή το volleyball και το hockey. Τα υπόλοιπα αθλήματα που είναι το basketball, το tennis, το baseball και το cricket βρίσκονται κάπου στην μέση και έχουν περίπου το ίδιο ποσοστό δημοφιλίας με το basketball να βρίσκεται πάνω των υπολοίπων και το cricket να βρίσκεται κάτω των υπολοίπων. Από κει και πέρα, ένα αποτέλεσμα που εμφανίστηκε και δεν έχει σχέση με την άθληση, ήταν ότι όσο πιο πολλά Tweets κάνει ένας λογαριασμός στο Twitter, τότε τόσοι περισσότεροι θα είναι και οι χρήστες που τον ακολουθούν. Δεν συμβαίνει πάντα, αλλά παρατηρήθηκε αυτό το φαινόμενο.