



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάλυση Δεδομένων από Κοινωνικά Δίκτυα για τη Μελέτη των Συναισθημάτων των Χρηστών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αντώνιος Α. Απειρανθίτης

και

Βασίλειος Α. Τσαχανσάχης

Επιβλέπων : Μιχαήλ Θεολόγου
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2016



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάλυση Δεδομένων από Κοινωνικά Δίκτυα για τη Μελέτη των Συναισθημάτων των Χρηστών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αντώνιος Α. Απειρανθίτης

και

Βασίλειος Α. Τσαχανσάχης

Επιβλέπων : Μιχαήλ Θεολόγου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 11^η Ιουλίου 2016

.....
Μ. Θεολόγου
Καθηγητής Ε.Μ.Π.

.....
Ε. Συκάς
Καθηγητής Ε.Μ.Π.

.....
Γ. Στασινόπουλος
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2016

.....
Αντώνιος Α. Απειρανθίτης

.....
Βασίλειος Α. Τσαχανσάχης

Διπλωματούχοι Ηλεκτρολόγοι Μηχανικοί και Μηχανικοί Υπολογιστών Ε.Μ.Π.

Copyright © Αντώνιος Α. Απειρανθίτης, 2016
Copyright © Βασίλειος Α. Τσαχανσάχης, 2016
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τους συγγραφείς.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τους συγγραφείς και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Αντικείμενο της παρούσας διπλωματικής αποτελεί η ανάλυση δεδομένων του κοινωνικού δικτύου Twitter για την μελέτη των συναισθημάτων των χρηστών. Το σύνολο των δεδομένων που χρησιμοποιείται αποτελείται από πραγματικά tweets, τα οποία δημοσιεύθηκαν στη βάση του Twitter, έχουν στο περιεχόμενό τους τη λέξη κλειδί refugees και συγκεντρώθηκαν σε διάστημα 4 εβδομάδων σε συγκεκριμένες ώρες της ημέρας. Η καταγραφή αυτών των δεδομένων έγινε με την βοήθεια του Apache NiFi.

Αρχικά, η εργασία έχει σα στόχο την υλοποίηση ενός συστήματος το οποίο ανά πάσα στιγμή είναι σε θέση να λαμβάνει μια λέξη κλειδί και να συγκεντρώνει τα tweets τα οποία πραγματοποιήθηκαν με την λέξη αυτή στο περιεχόμενό τους. Εν συνεχεία, να τα μεταφέρει και να τα αποθηκεύει σε μορφή πίνακα στον αποθηκευτικό του χώρο. Επίσης, να μπορεί να συνδυάζει αυτές τις αρχικές πληροφορίες με ένα σύνολο δευτερευόντων πληροφοριών, ώστε με την χρήση αυτών και περαιτέρω ανάλυση, να είναι σε θέση να παράγει χρήσιμα στατιστικά στοιχεία. Για τον σκοπό αυτό χρησιμοποιήθηκαν εργαλεία ανοιχτού κώδικα όπως Apache Hadoop, Apache NiFi και Apache Hive. Σε πρώτη φάση, γίνεται ενασχόληση με ζητήματα όπως η αποθήκευση και ο καθαρισμός των δεδομένων.

Σε δεύτερη φάση, η εργασία καταπιάνεται με το πεδίο της εξόρυξης γνώσης από δεδομένα (data mining), με την εφαρμογή ενός αλγορίθμου κατηγοριοποίησης. Συγκεκριμένα, γίνεται προσπάθεια για κατηγοριοποίηση σε τρεις ευρύτερες ομάδες του συναισθήματος που εκφράζει κάθε tweet με βάση τα στοιχεία που εξήχθησαν. Στόχος είναι η αναπαράσταση και απεικόνιση των αποτελεσμάτων σε έναν παγκόσμιο χάρτη με την βοήθεια του google maps.

Λέξεις κλειδιά

Apache, Hadoop, HDFS, Hive, NiFi, Twitter, Εξόρυξη Δεδομένων, Ανάλυση Συναισθημάτων, Απεικόνιση

Abstract

Subject of this thesis is the data analysis of the social network Twitter with the aim to study user emotions. The data used in this study consist of real tweets, which were posted on Twitter, include the keyword *refugees* in their text content, and were collected in a time span of 4 weeks in specific daytimes. The data collection was conducted using the software Apache NiFi.

Initially, the dissertation has as its goal the implementation of a system, which in any given time will be able to receive a keyword and collect tweets including that same keyword. Subsequently, the tweets will get transferred and stored in the system's internal storage file system unit in a matrix format. Furthermore, the system will be able to combine the initial information with a set of secondary data and additional analysis, in order to produce valuable statistical insights. For this purpose, open sourced software like Apache Hadoop, Apache NiFi, and Apache Hive were used. During the first stage, this dissertation focuses on matters such as data storing and cleansing.

During the second stage, this paper focuses on the field of data mining by use of a classification algorithm. More specifically, an effort is being made to categorize the tweets in three broad groups based on the emotion described in those tweets. Aim of this study is the representation and visualization of the results in a global map with the help of Google Maps.

Keywords

Apache, Hadoop, HDFS, Hive, NiFi, Twitter, Sentiment Analysis, Data Mining, Visualization

Ευχαριστίες

Αρχικά, θα θέλαμε να ευχαριστήσουμε θερμά τον Καθηγητή Ε.Μ.Π. κύριο Μιχαήλ Θεολόγου για την στήριξη και την καθοδήγηση που μας παρείχε όλα αυτά τα χρόνια εντός και εκτός αμφιθεάτρων, καθώς και για την εμπιστοσύνη που μας έδειξε δίνοντας μας την ευκαιρία να ασχοληθούμε με ένα τόσο ενδιαφέρον και σύγχρονο θέμα στα πλαίσια της διπλωματικής εργασίας.

Επίσης, θα θέλαμε να ευχαριστήσουμε τον Παύλο Κοσμίδη, ο οποίος έπαιξε καθοριστικό ρόλο στα πρώτα στάδια αυτής της διπλωματικής και τη διδακτορική ερευνητρια, Χαρά Ρεμούνδου, για τη βοήθεια και την καθοδήγησή της κατά την εκπόνηση της παρούσας εργασίας, χωρίς τη συμβολή της οποίας δε θα υπήρχε αυτό το αποτέλεσμα.

Τέλος, θα θέλαμε να ευχαριστήσουμε, ο καθένας ξεχωριστά, τους γονείς μας για την υπομονή και την αμέριστη στήριξη που μας παρείχαν όλα αυτά τα χρόνια. Και φυσικά όλους εκείνους, φίλους και γνωστούς, που ήταν συνοδοιπόροι κατά τη διάρκεια των σπουδών μας, στο μονοπάτι του Πολυτεχνείου.

Απειρανθίτης Αντώνιος και Τσαχανσάχης Βασίλειος,

Αθήνα, 11^η Ιουλίου 2016

Περιεχόμενα

Κεφάλαιο 1 : Εισαγωγή.....	19
1.1 Κίνητρο.....	19
1.2 Σκοπός	21
1.3 Οργάνωση κειμένου	23
Κεφάλαιο 2 : Twitter.....	25
2.1 Γενικές πληροφορίες	25
2.2 Χρήστες	26
2.3 Tweets και Twitter Feed	28
2.4 API.....	29
2.4.1 REST API.....	30
2.4.2 Streaming API.....	31
2.5 OAuth	32
Κεφάλαιο 3 : Real-Time Processing	34
3.1 Πραγματικού Χρόνου Συστήματα Επεξεργασίας	34
3.2 Apache NiFi.....	35
3.2.1 Εισαγωγή.....	35
3.2.2 Βασικές έννοιες.....	37
3.2.3 Αρχιτεκτονική	38
3.2.4 Προσδοκώμενη επίδοση και χαρακτηριστικά του NiFi.....	41
3.2.5 Πλεονεκτήματα από τη χρήση του NiFi	42
3.2.6 Περιβάλλον Χρήστη.....	45
Κεφάλαιο 4 : Hadoop	47

4.1	Γενικές πληροφορίες	47
4.2	Ιστορική αναδρομή.....	48
4.3	Ορισμός	48
4.4	Σχεδιασμός.....	49
4.5	Λειτουργία	49
4.6	Βασικό πλαίσιο	50
4.7	Αρχιτεκτονική.....	51
4.7.1	Hadoop Cluster.....	52
4.7.2	HDFS.....	53
4.7.3	MapReduce.....	55
4.7.4	YARN.....	58
4.8	Το Οικοσύστημα.....	60
Κεφάλαιο 5 : Εξόρυξη Δεδομένων (Data Mining)		64
5.1	Εισαγωγή	64
5.2	Ορισμός	64
5.3	Στόχος.....	64
5.4	Ιστορική Αναδρομή	65
5.5	Ανακάλυψη γνώσης.....	66
5.5.1	Προ-επεξεργασία.....	68
5.5.2	Τεχνικές.....	68
5.5.3	Επικύρωση Αποτελέσματος	69
5.6	Εφαρμογές	70
5.6.1	Ιατρική.....	70
5.6.2	Οικονομία.....	71
5.6.3	Τηλεπικοινωνία	72
5.7	Σημασία	73

Κεφάλαιο 6 : Υλοποίηση Συστήματος Ανάλυσης Δεδομένων	75
6.1 Εισαγωγή – Παρουσίαση του συστήματος.....	75
6.2 Αρχιτεκτονική.....	76
6.2.1 Γενική αρχιτεκτονική του συστήματος	76
6.2.2 Αρχιτεκτονική φυσικού επιπέδου	77
6.2.3 Δομές Δεδομένων.....	78
6.3 Περιγραφή δομικών συστατικών.....	81
6.3.1 Εξαγωγή Δεδομένων (Data Extraction)	83
6.3.2 Καθαρισμός Δεδομένων (Data Cleansing).....	88
6.3.3 Επεξεργασία Δεδομένων (Data Processing)	90
6.3.4 Ανάλυση Δεδομένων (Data Analysis)	93
6.3.5 Απεικόνιση δεδομένων (Data Visualization).....	95
Κεφάλαιο 7 : Επίλογος.....	99
7.1 Σύνοψη – Συμπεράσματα	99
7.2 Μελλοντικές Εργασίες.....	101
Υπόμνημα : HDFS	103
1. Εισαγωγή	103
2. Παραδοχές και Στόχοι	103
2.1 Αποτυχίες Υλικού (Hardware Failure)	103
2.2 Πρόσβαση Ροής Δεδομένων (Streaming Data Access).....	103
2.3 Μεγάλα Σύνολα Δεδομένων (Large Data Sets).....	104
2.4 Απλό Μοντέλο Συνοχής (Simple Coherency Model).....	104
2.5 Μετακίνηση υπολογισμών αντί μετακίνησης δεδομένων	104
2.6 Φορητότητα.....	104
3. NameNode και DataNodes	105
4. Ονομασία και οργάνωση αρχείων (File System Namespace).....	106

5.	Αντιγραφή Δεδομένων (Data Replication).....	106
5.1	Τοποθέτηση Αντιγράφου (Replica Placement)	106
5.2	Επιλογή Αντιγράφου (Replica Selection).....	108
5.3	Ασφαλής Λειτουργία (Safemode).....	108
6.	Η ανθεκτικότητα του συστήματος αρχείων μεταδεδομένων.....	108
7.	Πρωτόκολλα Επικοινωνίας	109
8.	Ευρωστία	110
8.1	Αποτυχία Δίσκου, Heartbeats και Επαναντιγραφή.....	110
8.2	Επαναπροσαρμογή Συμπλέγματος (Cluster Rebalancing)	110
8.3	Ακεραιότητα Δεδομένων (Data Integrity)	111
8.4	Αποτυχία Δίσκου Μεταδεδομένων (Metadata Disk Failure)	111
8.5	Στιγμιότυπα (Snapshots)	111
9.	Οργάνωση Δεδομένων.....	112
9.1	Τεμάχια Δεδομένων (Data Blocks).....	112
9.2	Staging	112
9.3	Διασωλήνωση Αντιγραφής (Replication Pipelining).....	113
10.	Προσβασιμότητα	113
	Βιβλιογραφία.....	115

Κατάλογος Εικόνων

Εικόνα 1 : Σύστημα Εξόρυξης Δεδομένων	22
Εικόνα 2 : Twitter Logo	25
Εικόνα 3 : API.....	29
Εικόνα 4 : Σύνδεση REST API	30
Εικόνα 5 : Σύνδεση Streaming API	31
Εικόνα 6 : OAuth	32
Εικόνα 7 : NiFi	35
Εικόνα 8 : Αρχιτεκτονική Κόμβου NiFi	39
Εικόνα 9 : Αρχιτεκτονική Συμπλέγματος NiFi	40
Εικόνα 10 : Γραφικό Περιβάλλον Χρήστη NiFi	45
Εικόνα 11 : Big Data και Hadoop	47
Εικόνα 12 : Ιστορική Αναδρομή	48
Εικόνα 13 : Hadoop Logo	48
Εικόνα 14 : Αλληλεπίδραση του Hadoop	49
Εικόνα 15 : Βασικό Πλαίσιο του Hadoop.....	50
Εικόνα 16 : Αρχιτεκτονική Hadoop	51
Εικόνα 17 : Αρχιτεκτονική Συμπλέγματος Hadoop.....	53
Εικόνα 18 : Αρχιτεκτονική HDFS	54
Εικόνα 19 : Αρχιτεκτονική MapReduce	55
Εικόνα 20 : Τμήματα MapReduce	57
Εικόνα 21 : Αρχιτεκτονική YARN	58
Εικόνα 22 : Οικοσύστημα Hadoop	62
Εικόνα 23 : Εξόρυξη Δεδομένων,	67
Εικόνα 24 : Γραφική Παρουσίαση του Συστήματος Εξόρυξης Δεδομένων.....	76
Εικόνα 25 : Αρχιτεκτονική Συστήματος Εξόρυξης Δεδομένων	77
Εικόνα 26 : Δημιουργία Twitter Application.....	81
Εικόνα 27 : Twitter Application.....	82
Εικόνα 28 : Δημιουργία Access Token	82

Εικόνα 29 : Γραφικό Περιβάλλον NiFi.....	83
Εικόνα 30 : Επεξεργαστής GetTwitter.....	84
Εικόνα 31 : Επεξεργαστής RouteOnAttribute	85
Εικόνα 32 : Επεξεργαστής EvaluatesJsonPath.....	85
Εικόνα 33 : Επεξεργαστής ReplaceText	86
Εικόνα 34 : Επεξεργαστής MergeContent	86
Εικόνα 35 : Επεξεργαστής PutFile.....	87
Εικόνα 36 : Επεξεργαστής PutHDFS.....	87
Εικόνα 37 : Απεικόνιση Ανάλυσης Συναισθήματος σε Χάρτη.....	93
Εικόνα 38 : Αριθμός των tweets ανά χώρα.....	94
Εικόνα 39 : Sentiment Analysis	97

Κατάλογος Πινάκων

Πίνακας 1 : Διαφορές NiFi – FBP	37
Πίνακας 2 : Χαρακτηριστικά Κόμβου Συστήματος.....	75

Κεφάλαιο 1 : Εισαγωγή

1.1 Κίνητρο

Μία από τις βασικές ανθρώπινες ανάγκες είναι να αλληλεπιδρούν και να συναναστρέφονται μεταξύ τους. Η χρήση της Τεχνολογίας των Επικοινωνιών έχει αλλάξει ραγδαία τον τρόπο που κοινωνικοποιούνται οι άνθρωποι μεταξύ τους.

Έχει παρατηρηθεί ότι τα δεδομένα που παράγονται τα τελευταία χρόνια έχουν αυξηθεί δραματικά [1]. Δεδομένα π.χ. από κοινωνικά δίκτυα, χρηματιστηριακές κινήσεις, αναζητήσεις χρηστών στο διαδίκτυο, μετρήσεις επιστημονικών οργάνων όπως μετεωρολογικοί σταθμοί ή ακολουθίες DNA παράγονται με πολύ γρήγορους ρυθμούς. Η γρήγορη και αποδοτική επεξεργασία τέτοιων δεδομένων είναι απαραίτητη για την εξαγωγή χρήσιμων συμπερασμάτων, ανάλογα το εκάστοτε πεδίο: η ανάλυση π.χ. δεδομένων από κοινωνικά δίκτυα ή αναζητήσεις χρηστών μπορεί να εμφανίσει τις προτιμήσεις τους σχετικά με προϊόντα ή υπηρεσίες. Πρόσφατες έρευνες δείχνουν ότι το μεγαλύτερο μέρος του χρόνου που δαπανάται στο διαδίκτυο είναι στην κοινωνική δικτύωση, ακολουθεί το online gaming και το ηλεκτρονικό ταχυδρομείο. Επιπλέον, περίπου 906 ώρες μηνιαίως δαπανώνται για κοινωνική δικτύωση σε ιστότοπους όπως το Twitter.

Ως εκ τούτου, οι τεχνικές ανάλυσης μεγάλου όγκου δεδομένων (big data analytics) έχουν γνωρίσει ιδιαίτερη άνθηση [2]. Οι τεχνικές αυτές βασίζονται σε καταναμημένες προσεγγίσεις, όπου συστοιχίες υπολογιστών χρησιμοποιούνται συνεργατικά για την επεξεργασία του όγκου των δεδομένων, με σκοπό την εύκολη κλιμάκωση της υποδομής καθώς αυξάνονται τα δεδομένα. Τεχνικές όπως το μοντέλο MapReduce της Google χρησιμοποιούνται κατά κόρον για τον σκοπό αυτό. Η υλοποίηση του MapReduce στο πρόγραμμα Apache Hadoop είχε σαν αποτέλεσμα την δημιουργία ενός οικοσυστήματος προγραμμάτων για την διεξαγωγή big-data analytics.

Τα προγράμματα αυτά συνήθως εκτελούνται σε εικονικές συστοιχίες υπολογιστών χρησιμοποιώντας τεχνολογίες cloud computing (π.χ. Hortonworks Data Platform) που επιτρέπουν το αυτόματο “στήσιμο” της υποδομής για την εκτέλεση των εργασιών και τον αυτόματο τερματισμό της υποδομής μετά το πέρας της επεξεργασίας.

Μια ακόμα νεότερη εκδοχή των παραπάνω τεχνικών είναι η ανάλυση των δεδομένων εισόδου σε πραγματικό χρόνο και η διεξαγωγή έγκαιρων συμπερασμάτων. Αυτή η δυνατότητα μπορεί να φανεί χρήσιμη σε ένα μεγάλο πλήθος εφαρμογών, όπως ο έγκαιρος εντοπισμός και αντιμετώπιση απρόοπτων ή ανεπιθύμητων συμβάντων, η κατηγοριοποίηση των δεδομένων και το sentiment analysis (ανάλυση συναισθημάτων).

Η έννοια της εξόρυξης γνώσης από δεδομένα (data mining) αναφέρεται στην εξεύρεση (ενδιαφέρουσας, χρήσιμης, αυτονόητης, ή μη προφανής) πληροφορίας ή προτύπων, από μεγάλες βάσεις δεδομένων, με χρήση αλγορίθμων ομαδοποίησης ή κατηγοριοποίησης (classification) και των αρχών της στατιστικής, της τεχνητής νοημοσύνης, της μηχανικής μάθησης (machine learning) και των συστημάτων βάσεων δεδομένων. Στόχος της εξόρυξης γνώσης είναι η πληροφορία που θα εξαχθεί και τα πρότυπα που θα προκύψουν να έχουν δομή κατανοητή προς τον άνθρωπο έτσι ώστε να τον βοηθήσουν να πάρει τις κατάλληλες αποφάσεις.

Το πεδίο της εξόρυξης γνώσης από δεδομένα γνωρίζει τρομερή άνθιση στη σύγχρονη εποχή, τόσο σε ερευνητικό επίπεδο, όσο και σε επίπεδο πρακτικής εφαρμογής. Ο συνδυασμός του με καταναμημένες τεχνικές επεξεργασίας δεδομένων, επιτρέπει την παράλληλη επεξεργασία μεγάλου όγκου δεδομένων (εκατοντάδων εκατομμυρίων δειγμάτων) σε μερικά μόλις λεπτά. Στην περίπτωση δε των καταναμημένων συστημάτων πραγματικού χρόνου, αυτή πληροφορία μπορεί να ανανεώνεται (και ανάλογα να ισχυροποιείται ή να καταρρίπτεται) με βάση τα δείγματα τα οποία εισέρχονται συνεχώς στο σύστημα.

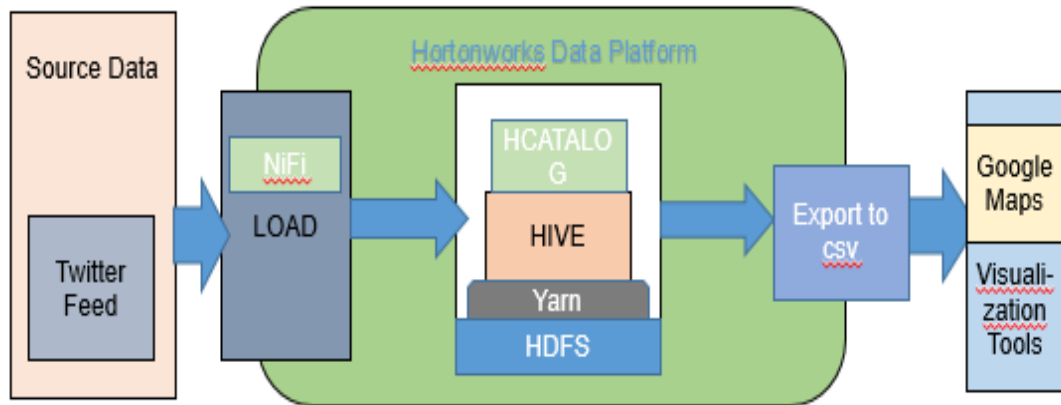
1.2 Σκοπός

Σκοπός της παρούσας διπλωματικής είναι η σχεδίαση και υλοποίηση ενός συστήματος εξόρυξης γνώσης από δεδομένα μεγάλου όγκου σε αδόμητη μορφή, το οποίο έχει τη μορφή ενός κλασικού συστήματος με μία είσοδο και μία έξοδο. Μία ειδική περίπτωση εφαρμογής αυτού του μοντέλου δεδομένων είναι η ανάλυση συναισθημάτων που μπορεί να εμπεριέχεται στο περιεχόμενο των δεδομένων αυτών. Στην περίπτωση αυτή τα αδόμητα δεδομένα (raw data) που προέρχονται από τη βάση δεδομένων του Twitter και καλύπτουν ένα χρονικό διάστημα μερικών εβδομάδων, είναι η λεγόμενη πηγή δεδομένων (source data), η οποία βρίσκεται στην αρχή του συστήματος. Όλα αυτά τα δεδομένα εισέρχονται στο σύστημα μέσω ενός εργαλείου εξαγωγής δεδομένων, το Apache NiFi, και αποθηκεύονται στο τρίτο κατά ιεραρχία εργαλείο του συστήματος, το κατανεμημένο σύστημα αρχείων HDFS.

Αρχικός στόχος είναι η δημιουργία ενός υποσυστήματος το οποίο θα επεξεργάζεται αυτά τα δεδομένα, χρησιμοποιώντας τεχνικές κατανεμημένης επεξεργασίας δεδομένων σε πραγματικό χρόνο, με κύριο εργαλείο το ανοιχτό λογισμικό του Apache Hive. Συγκεκριμένα, πραγματοποιείται η επεξεργασία και η ανάλυση των δεδομένων σε λέξεις, προσδίδοντας σε κάθε μία από αυτές μία πολικότητα ανάλογα με το συναίσθημα που εκφράζουν.

Επιπλέον, γίνεται προσπάθεια για ανάλυση των συναισθημάτων (sentiment analysis) των συγγραφέων των δεδομένων που αναλύθηκαν με βάση αλγορίθμων κατηγοριοποίησης και η αναπαράσταση των αποτελεσμάτων της ανάλυσης στην έξοδο του συστήματος, το εργαλείο απεικόνισης του Google Maps.

Στην Εικόνα 1 φαίνεται εποπτικά το σύστημα που υλοποιείται στην παρούσα διπλωματική εργασία.



Εικόνα 1 : Σύστημα Εξόρυξης Δεδομένων

1.3 Οργάνωση κειμένου

Στο Κεφάλαιο 2 παρουσιάζεται ο ιστότοπος κοινωνικής δικτύωσης του Twitter. Γίνεται αναφορά στην αρχιτεκτονική και τις λειτουργίες του, στους χρήστες και τα δεδομένα, καθώς επίσης και στη σημασία που έχει ή προκύπτει από την ανάλυση των δεδομένων του.

Στο Κεφάλαιο 3 παρουσιάζεται λεπτομερώς ένα εργαλείο εξαγωγής δεδομένων για συστήματα πραγματικού χρόνου, το Apache NiFi. Εξετάζεται η αρχιτεκτονική και οι δυνατότητές του και παρουσιάζονται τα πλεονεκτήματα που αποκομίζονται από τη χρήση του.

Στο κεφάλαιο 4 γίνεται ιδιαίτερη αναφορά στην πλατφόρμα του Apache Hadoop. Το Hadoop είναι ένα προγραμματιστικό πλαίσιο ανοιχτού κώδικα, για την κατανομημένη αποθήκευση και επεξεργασία πολύ μεγάλων συνόλων δεδομένων. Στο οικοσύστημά του ανήκουν πολλά ανοιχτού κώδικα εργαλεία, τα οποία κατείχαν επικουρικό ρόλο στην παρούσα διπλωματική εργασία, όπως το HDFS, το Yarn και το Apache Hive.

Στο Κεφάλαιο 5 παρουσιάζεται η διαδικασία της εξόρυξης γνώσης από δεδομένα, σαν ένα ενιαίο σύνολο επιμέρους λειτουργιών, όπως η εξαγωγή, ο καθαρισμός, η επεξεργασία, η ανάλυση και η απεικόνιση των δεδομένων. Αναφέρεται η σημασία της και η έως τώρα ευρεία χρήση της.

Στο Κεφάλαιο 6 παρουσιάζεται εκτενώς η υλοποίηση που έγινε στην παρούσα διπλωματικής, προκειμένου να ολοκληρωθεί το πρακτικό κομμάτι αυτής και να μπορούν να παρουσιαστούν τα αποτελέσματά της. Αναλύονται λεπτομερώς οι επιμέρους λειτουργίες που αναφέρθηκαν στο Κεφάλαιο 6 και εκτελέστηκαν με απώτερο σκοπό την ανάλυση των συναισθημάτων των χρηστών. Όπως επίσης παρουσιάζεται και το προγραμματιστικό κομμάτι που αναπτύχθηκε ώστε να γίνει εφικτή η εν λόγω υλοποίηση.

Τέλος, στο Κεφάλαιο 7 γίνεται μία ανασκόπηση των γνώσεων που αποκομίστηκαν στα πλαίσια της παρούσας διπλωματικής και συνοψίζονται τα τελικά αποτελέσματα και σχόλια επί του πρακτικού μέρους. Ακόμα, παρουσιάζονται τυχόν επεκτάσεις της παρούσας εργασίας σε διαφορετικές κατευθύνσεις.

Κεφάλαιο 2 : Twitter

2.1 Γενικές πληροφορίες

Υπάρχουν πολλοί και διαφορετικοί ορισμοί για το τι είναι ακριβώς το Twitter, παρ' όλ' αυτά στην παρούσα διπλωματική εργασία γίνεται μία προσπάθεια να εξεταστεί από μία οπτική γωνία που αντιλαμβάνεται τη θεμελιώδη σχέση της τεχνολογίας με την ανθρωπότητα. Οποιαδήποτε τεχνολογία για να είναι χρήσιμη και επιτυχής πρέπει να λαμβάνει υπόψη της αυτή τη σχέση, άλλωστε, σκοπός της είναι να ενισχυθεί η ανθρώπινη εμπειρία μέσα από τη χρήση της.



Εικόνα 2 : Twitter Logo

Οι άνθρωποι, μέσω της τεχνολογίας, επιζητούν ως επί το πλείστον :

- να ακουστούν
- να ικανοποιήσουν την περιέργειά τους
- να το κάνουν εύκολα
- όλα αυτά να γίνουν γρήγορα.

Υπάρχει μία βαθιά ριζωμένη ανάγκη στον άνθρωπο για να μοιραστεί τις ιδέες και τις εμπειρίες του, η οποία του προσφέρει τη δυνατότητα να συνδεθεί με άλλους ανθρώπους, να ακουστεί και να νιώσει ένα αίσθημα αξίας και σημασίας. Ακόμη, ο άνθρωπος είναι περίεργος για τον κόσμο γύρω του, για το πώς να τον οργανώσει και να τον χειραγωγήσει, για αυτό χρησιμοποιεί την επικοινωνία για να μοιραστεί τις παρατηρήσεις του, να κάνει ερωτήσεις και να συμμετάσχει με άλλους ανθρώπους σε σημαντικούς διαλόγους για τα διλήμματα και τις απορίες του. Επίσης, δε θέλει να εργαστεί σκληρότερα από ό,τι είναι απολύτως απαραίτητο για να ικανοποιήσει αυτή του την περιέργεια και θέλει να κάνει όλα αυτά τα πράγματα άμεσα, τείνοντας να γίνεται ανυπόμονος όταν δεν υπάρχει πραγματική πρόοδος με την ταχύτητα της σκέψης του.

2.2 Χρήστες

Ένας άλλος τρόπος για να περιγράψει κανείς το Twitter είναι ως μία υπηρεσία microblogging, η οποία επιτρέπει στους χρήστες ([users](#)) του να επικοινωνούν με μικρά μηνύματα 140 χαρακτήρων, τα λεγόμενα tweets, που μετά βίας αντιστοιχούν σε σκέψεις ή ιδέες [3]. Συναφώς, το Twitter θα μπορούσε να παρομοιαστεί με μία δωρεάν, υψηλής ταχύτητας, παγκόσμια υπηρεσία ανταλλαγής μηνυμάτων κειμένου. Ουσιαστικά, πρόκειται για μία πολύτιμη υπηρεσία που επιτρέπει την ταχεία και εύκολη επικοινωνία. Ωστόσο, αυτό δε μπορεί να περιγράψει όλη τη δυναμική του Twitter, καθώς δεν αντικατοπτρίζει την αξία που αναδύεται όταν υπάρχουν πάνω από 500 εκατομμύρια άνθρωποι εγγεγραμμένοι σε αυτό, με περισσότερους από 100 εκατομμύρια από αυτούς να ασκούν ενεργά την περιέργειά τους σε τακτική μηνιαία βάση.

Εκτός από τις δυνατότητες που προέκυψαν για μάρκετινγκ και διαφήμιση σε μακρο-επίπεδο, οι οποίες είναι πάντα προσοδοφόρες με μία βάση χρηστών τέτοιου μεγέθους, αυτό που δημιούργησε τη βαρύτητα προκειμένου να προκύψει μία τέτοια βάση χρηστών είναι οι βαθύτερες δυναμικές δικτύου κάτω από το Twitter, οι οποίες και το καθιστούν τόσο σημαντικό. Ο διάυλος επικοινωνίας που επιτρέπει στους χρήστες να μοιράζονται σύντομα αποφθέγματα με ταχύτητα σκέψης μπορεί να είναι απαραίτητη προϋπόθεση για την ιογενή έγκριση και τη διαρκή εμπλοκή με την πλατφόρμα του Twitter, ωστόσο δεν είναι επαρκής. Η επαρκής προϋπόθεση είναι το ασύμμετρο ακολουθητικό (following) μοντέλο του Twitter, που ικανοποιεί την ανθρώπινη περιέργεια και το καθιστά περισσότερο ως ένα γράφημα ενδιαφέροντος (interest graph) παρά ως κοινωνικό δίκτυο, και τα APIs (Application Programming Interface) που παρέχουν ένα επαρκές υπολογιστικό πλαίσιο για δομή και αυτό-οργανωτική συμπεριφορά.

Έτσι, ενώ κάποιες ιστοσελίδες κοινωνικής δικτύωσης, όπως το Facebook και το LinkedIn, απαιτούν την αμοιβαία αποδοχή της σύνδεσης μεταξύ των χρηστών (υπονοώντας μία σύνδεση στον πραγματικό κόσμο), το σχεσιακό μοντέλο του Twitter επιτρέπει σε ένα χρήστη να παρακολουθεί τις τελευταίες ενέργειες οποιουδήποτε άλλου χρήστη, ακόμα και αν αυτός ο άλλος χρήστης δεν επιλέξει να ακολουθεί τον πρώτο (follow back). Το ακολουθητικό μοντέλο του Twitter μπορεί να είναι απλό αλλά

εκμεταλλεύεται μία βασική πτυχή της ανθρωπότητας, την περιέργειά της. Είτε πρόκειται για ενδιαφέρον σε ένα συγκεκριμένο πολιτικό ή κοινωνικό θέμα, για ενημέρωση πάνω στις τελευταίες τεχνολογικές εξελίξεις ή σε κάποια αθλητική ομάδα, για κουτσομπολιό ή επιθυμία για επικοινωνία με κάποιον άλλο, το Twitter προσφέρει απεριόριστες ευκαιρίες για να ικανοποιηθεί η ανθρώπινη περιέργεια.

Ένα γράφημα ενδιαφέροντος είναι ένας τρόπος μοντελοποίησης συνδέσεων μεταξύ των ανθρώπων και των αυθαίρετων ενδιαφερόντων τους. Τα γραφήματα ενδιαφέροντος παρέχουν ένα σημαντικό αριθμό δυνατοτήτων στο πεδίο της εξόρυξης δεδομένων, που κατά κύριο λόγο αφορούν τη μέτρηση συσχετίσεων μεταξύ των αντικειμένων, με στόχο να καταστούν ευφυείς συστάσεις και άλλες εφαρμογές για μηχανική μάθηση (machine learning). Για παράδειγμα, θα μπορούσε να γίνει χρήση ενός γραφήματος ενδιαφέροντος για τη μέτρηση συσχετισμών και την υποβολή συστάσεων για το ποιον να ακολουθήσει κανείς στο Twitter, τι να αγοράσει on-line και με ποιους θα πρέπει να βγει έξω σήμερα. Για να φανεί η έννοια του Twitter ως γράφημα ενδιαφέροντος, αρκεί να σκεφτεί κανείς ότι ένας χρήστης του Twitter δεν χρειάζεται να είναι ένα πραγματικό πρόσωπο, αλλά μπορεί πολύ εύκολα να είναι ένα άψυχο αντικείμενο, μια εταιρεία, ένα μουσικό συγκρότημα, μια φανταστική προσωπικότητα, μια πλαστοπροσωπία κάποιου άλλου (ζωντανού ή νεκρού) προσώπου, ή οτιδήποτε άλλο. Αυτό δείχνει τη σημασία του Twitter στην εποχή μας και το κέρδος που μπορεί να προκύψει από την εξόρυξη δεδομένων από πληροφορίες που προέρχονται από αυτό.

Υπάρχει πολύ λίγος έλεγχος στο τι μπορεί να είναι ένας λογαριασμός Twitter, πέρα από το διακριτικό σήμα σε κάποιους λογαριασμούς που προσδιορίζουν διασημότητες και δημόσια πρόσωπα ως «επαλήθευση λογαριασμών» και τους βασικούς περιορισμούς στους [Όρους Χρήσης](#) του Twitter, που απαιτούνται για τη χρήση της υπηρεσίας. Μπορεί να φαίνεται πολύ λεπτή, αλλά είναι σημαντική η διάκριση για κάποιες ιστοσελίδες κοινωνικής δικτύωσης, στις οποίες οι λογαριασμοί πρέπει να ανταποκρίνονται σε πραγματικούς, ζωντανούς ανθρώπους, επιχειρήσεις ή οντότητες παρόμοιας φύσης. Το Twitter δεν έχει κανένα συγκεκριμένο περιορισμό σχετικά με το πρόσωπο ενός λογαριασμού και στηρίζεται στην αυτο-οργανωτική συμπεριφορά, όπως ακολουθητικές σχέσεις και λαϊκούς νόμους (folksonomies) που προκύπτουν από τη χρήση hashtag (#), για να δημιουργήσει ένα συγκεκριμένο είδος τάξης εντός του συστήματος.

2.3 Tweets και Twitter Feed

Το Twitter θα μπορούσε επίσης να περιγραφεί ως μία πραγματικού χρόνου, ιδιαίτερα κοινωνική microblogging υπηρεσία, που επιτρέπει στους χρήστες να δημοσιεύουν σύντομες ενημερώσεις κατάστασης (status update), *tweets* (τιτίβισμα), που εμφανίζονται στα *timelines* (χρονοδιαγράμματα). Τα tweets μπορούν να περιλαμβάνουν μία ή περισσότερες οντότητες (*entities*) στους 140 χαρακτήρες περιεχομένου τους και να αναφέρουν μία ή περισσότερες θέσεις (*places*) που αντιστοιχούν σε τοποθεσίες στον πραγματικό κόσμο [4].

Τα tweets είναι η ουσία του Twitter και ενώ θεωρητικά είναι οι 140 χαρακτήρες περιεχομένου που σχετίζονται με την ενημέρωση κατάστασης του χρήστη, στην πραγματικότητα υπάρχουν πολλά περισσότερα μεταδεδομένα (metadata) εντός τους. Εκτός από το περιεχόμενο κειμένου του ίδιου του tweet, τα tweets συνοδεύονται από δύο επιπλέον κομμάτια μεταδεδομένων ιδιαίτερης σημασίας: οντότητες και θέσεις. Οι οντότητες ενός tweet είναι ουσιαστικά αναφορές χρηστών, hashtags, διευθύνσεις URL, και δεδομένα (media) που μπορεί να σχετίζονται με ένα tweet. Ενώ, οι θέσεις είναι τοποθεσίες στον πραγματικό κόσμο που μπορεί να συνδέονται με ένα tweet, είτε η πραγματική θέση στην οποία είχε συγγραφεί, είτε μία αναφορά σε κάποια θέση που περιγράφεται στο περιεχόμενό του.

Παρακάτω εξετάζεται ένα tweet με το εξής περιεχόμενο :

For the first time in history, a team of [#refugees](#) will compete at the [@olympics](#) in Rio de Janeiro. <http://buff.ly/25FxZCe>

Το tweet αυτό αποτελείται από 124 χαρακτήρες και περιέχει τις εξής τρεις οντότητες:

- αναφορά στο χρήστη @olympics
- το hashtag #refugees
- και το http URL: <http://buff.ly/25FxZCe>.

Παρά το γεγονός ότι στο tweet εμπεριέχεται μία θέση, Rio de Janeiro, που αναφέρεται ρητά στο περιεχόμενό του, τα μεταδεδομένα θέσης που σχετίζονται με το tweet θα μπορούσαν να περιλαμβάνουν την τοποθεσία στην οποία συγγράφηκε το tweet, η οποία μπορεί να είναι ή και να μην είναι το Rio de Janeiro. Όλα αυτά είναι

αρκετά μεταδεδομένα που εμπεριέχονται σε κάτι λιγότερο από 140 χαρακτήρες και δείχνουν ακριβώς πόσο ισχυρό μπορεί να είναι ένα μικρό tweet. Το οποίο όπως φάνηκε μπορεί να αναφέρεται σε πολλούς άλλους χρήστες του Twitter, συνδέσεις σε ιστοσελίδες και παραπομπές σε διάφορα θέματα μέσω hashtags που δρουν ως σημεία συγκέντρωσης των tweets που αναφέρονται σε αυτό το θέμα σε όλο το Twitter, για εύκολη αναζήτηση.

Τέλος, τα timelines είναι χρονολογικά ταξινομημένες συλλογές από tweets. Με την αφηρημένη έννοια, ένα timeline είναι οποιαδήποτε συλλογή από tweets που εμφανίζονται με χρονολογική σειρά, ωστόσο, δύο μόνο timelines είναι σημαντικά. Το *home timeline*, το οποίο εμφανίζεται μόλις ένας χρήστης συνδέεται στο λογαριασμό του και περιέχει όλα τα tweets από τους χρήστες που ακολουθεί, και το *user timeline*, που είναι μία συλλογή από tweets ενός ορισμένου χρήστη.

Ενώ τα timelines είναι συλλογές από tweets με σχετικά χαμηλή ταχύτητα, τα *streams* είναι δείγματα δημόσιων tweets που ρέουν μέσω του Twitter σε πραγματικό χρόνο και μπορεί συγκεντρώσει εκατοντάδες χιλιάδες tweets ανά λεπτό κατά τη διάρκεια εκδηλώσεων με ιδιαίτερα μεγάλο ενδιαφέρον, όπως προεδρικές συζητήσεις, εκλογές, μεγάλα αθλητικά γεγονότα κ.α.. Όλος αυτός ο όγκος δεδομένων παρουσιάζει ενδιαφέρουσες τεχνολογικές (engineering) προκλήσεις και είναι ένας σημαντικός λόγος που διάφορες εταιρίες έχουν συνεργαστεί με το Twitter για να μετατρέψουν αυτόν τον όγκο σε μία πιο καταναλωτική μορφή.

2.4 API

Η Διεπαφή Προγραμματισμού Εφαρμογών, εν συντομία διεπαφή, διασύνδεση ή API (Application Programming Interface) είναι ένα εργαλείο το οποίο διευκολύνει την αλληλεπίδραση με προγράμματα ηλεκτρονικών υπολογιστών και υπηρεσίες web [5]. Πολλές υπηρεσίες web παρέχουν APIs στους προγραμματιστές προκειμένου να αλληλεπιδρούν με τις υπηρεσίες τους και να έχουν πρόσβαση σε δεδομένα με ένα προγραμματιστικό τρόπο.



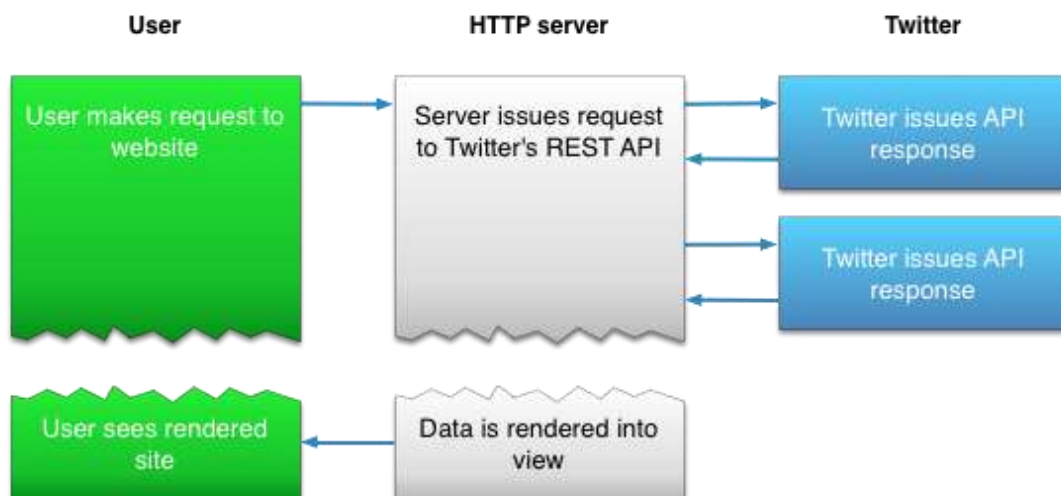
Εικόνα 3 : API

2.4.1 REST API

Το Twitter έχει δώσει ιδιαίτερη σημασία στη δημιουργία ενός κομψού και απλού [RESTful API](#), το οποίο είναι διαισθητικό και εύκολο στη χρήση. Τα REST (Representational State Transfer) APIs παρέχουν προγραμματιστική πρόσβαση για ανάγνωση και εγγραφή δεδομένων του Twitter και προσδιορίζουν εφαρμογές και χρήστες του Twitter χρησιμοποιώντας το πρωτόκολλο ασφαλείας OAuth. Το REST API επιτρέπει την αναζήτηση και εξαγωγή ενός υπάρχοντος συνόλου δεδομένων που έχει δημιουργηθεί από τα tweets που έχουν ήδη δημοσιευθεί. Μέσα από τη χρήση του μπορεί κανείς να ζητήσει tweets που ταιριάζουν σε κάποια συγκεκριμένα κριτήρια αναζήτησης, όπως hashtags, ονόματα χρηστών, τοποθεσίες, θέσεις κλπ.

Με το REST API του Twitter, οι προγραμματιστές αναζητούν ή τραβούν μόνο ένα συγκεκριμένο αριθμό από tweets που έχουν ήδη δημοσιευθεί, ο οποίος περιορίζεται από τα όρια ταχύτητας ([Rate Limits](#)) του Twitter. Για έναν μεμονωμένο χρήστη, ο μέγιστος αριθμός των tweets που μπορεί να λάβει είναι τα τελευταία 3.200 tweets, ανεξάρτητα από τα κριτήρια αναζήτησης. Επιπλέον, υπάρχει περαιτέρω περιορισμός στον αριθμό των αιτήσεων που μπορούν να γίνουν σε ένα ορισμένο χρονικό διάστημα (180 αιτήσεις σε διάστημα 15 λεπτών).

Μία web εφαρμογή, που δέχεται αιτήματα χρηστών, κάνει μία ή περισσότερες αιτήσεις στο Twitter API και στη συνέχεια διαμορφώνει και επιστρέφει το αποτέλεσμα στο χρήστη, ως απάντηση στην αρχική του αίτηση, όπως φαίνεται στην Εικόνα 4.



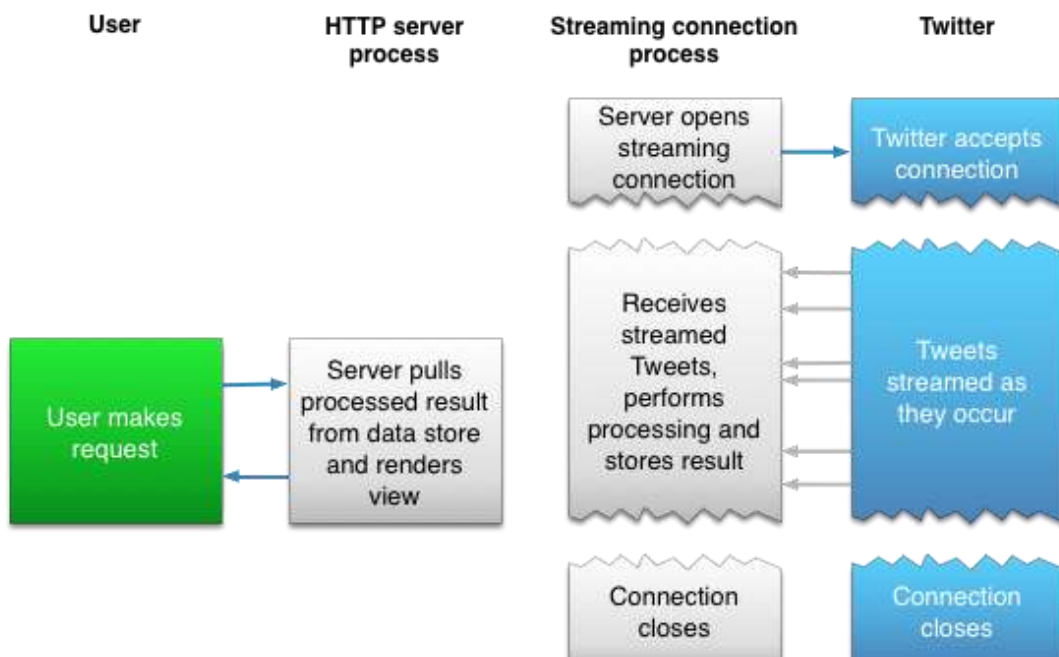
Εικόνα 4 : Σύνδεση REST API

2.4.2 Streaming API

Το [Streaming API](#) του Twitter επιτρέπει την αναζήτηση δεδομένων, σε σχεδόν πραγματικό χρόνο, από tweets τα οποία μόλις έχουν δημοσιευθεί, χρησιμοποιώντας είτε βασικά είτε OAuth πρωτόκολλα ασφαλείας.. Με το Streaming API, οι χρήστες καταχωρούν συγκεκριμένα κριτήρια (hashtags, ονόματα χρηστών, τοποθεσίες, θέσεις κλπ.) και όσο δημοσιεύονται tweets που ταιριάζουν με τα κριτήρια αυτά, ωθούνται απευθείας στο χρήστη μαζί με πληροφορίες για το συγγραφέα του εκάστοτε tweet. Είναι περισσότερο ώθηση δεδομένων από το Twitter, παρά τράβηγμα των δεδομένων από τον τελικό χρήστη.

Το Streaming API έχει το μειονέκτημα πως το πραγματικό ποσοστό του συνόλου των tweets, που λαμβάνουν οι χρήστες, ποικίλλει σε μεγάλο βαθμό με βάση τα κριτήρια αναζήτησης και τις τρέχουσες κυκλοφοριακές συνθήκες. Παρ' όλα αυτά τα οφέλη της ύπαρξης μίας πραγματικού χρόνου ροής δεδομένων Twitter, καθιστούν ιδιαίτερα σημαντική την ενσωμάτωση του Streaming API σε διάφορους τύπους εφαρμογών.

Μία σύνδεση με το Streaming API απαιτεί τη διατήρηση μίας μόνιμης ανοιχτής HTTP σύνδεσης. Όπως φαίνεται στην Εικόνα 5, η διαδικασία Streaming λαμβάνει τα tweets, εκτελεί ανάλυση, φιλτράρισμα ή συνάθροιση – ανάλογα με τις απαιτήσεις – και αποθηκεύει το αποτέλεσμα σε μία αποθήκη δεδομένων, απ' όπου η διαδικασία χειρισμού HTTP αναζητεί αποτελέσματα σε απάντηση των αιτημάτων του χρήστη.



Εικόνα 5 : Σύνδεση Streaming API

2.5 OAuth

Το OAuth είναι ένα πρωτόκολλο εξουσιοδότησης που επιτρέπει την αποστολή ασφαλών εξουσιοδοτημένων αιτήσεων προς το Twitter API. Οι χρήστες δεν χρειάζεται να μοιράζονται τους κωδικούς τους με άλλες εφαρμογές, αυξάνοντας έτσι την ασφάλεια του λογαριασμού τους. Επιπλέον, υπάρχουν πολλές συμβατές βιβλιοθήκες με την υλοποίηση του OAuth στο Twitter, που το καθιστούν πρότυπο πρωτόκολλο ασφαλείας [6].



*Εικόνα 6 :
OAuth*

Η πιο κοινή μορφή εξουσιοδότησης πόρων στο Twitter OAuth μέχρι σήμερα είναι αυτή μεταξύ εφαρμογής και χρήστη (application-user authentication). Μία πιστοποιημένη αίτηση αναγνωρίζει τόσο την ταυτότητα της εφαρμογής, όσο και την ταυτότητα που συνοδεύει τα δικαιώματα του τελικού χρήστη ο οποίος εκτελεί API κλήσεις, εκπροσωπούμενη από το διακριτικό πρόσβασης του χρήστη (access token).

Η εξουσιοδότηση μόνο της εφαρμογής (application-only authentication) είναι μια μορφή ταυτοποίησης που επιτρέπει σε μία εφαρμογή να εκτελέσει αιτήσεις API για δικό της λογαριασμό, χωρίς κάποιο πλαίσιο χρήστη. Οι API κλήσεις εξακολουθούν να τελούν υπό περιορισμούς, οι οποίοι όμως υπόκεινται σε ολόκληρη την εφαρμογή σα σύνολο και όχι ανά χρήστη. Οι API μέθοδοι που υποστηρίζουν αυτή τη μορφή εξουσιοδότησης περιέχουν δύο όρια ταχύτητας στην τεκμηρίωση τους, ένα ανά χρήστη (application-user authentication) και ένα ανά εφαρμογή (application-only authentication).

Κεφάλαιο 3 : Real-Time Processing

3.1 Πραγματικού Χρόνου Συστήματα Επεξεργασίας

Με τον όρο πραγματικού χρόνου συστήματα επεξεργασίας, αναφερόμαστε στα συστήματα με συνεχή είσοδο, έγκαιρη επεξεργασία και συστηματική έξοδο των δεδομένων. Τέτοιου είδους συστήματα γίνονται όλο και περισσότερο αναγκαία για την κάλυψη συγκεκριμένων αναγκών, όπως επίβλεψη συστήματος, ηλεκτρονικό εμπόριο, ανίχνευση απάτης και λήψη αποφάσεων, τόσο για επιστημονικά όσο και εταιρικά περιβάλλοντα. Είναι απαραίτητο να επεξεργάζονται υψηλό ποσοστό δεδομένων σε μικρό χρονικό διάστημα και κατά συνέπεια πρέπει να εγγυώνται απόκριση μέσα σε αυστηρούς χρονικούς περιορισμούς.

Σύμφωνα με την κρισιμότητα της άμεσης απάντησης, τα συστήματα πραγματικού χρόνου χωρίζονται σε 3 κατηγορίες:

- **Αυστηρά:** Η απώλεια του χρονικού ορίου για την επεξεργασία μιας μονάδας δεδομένων ισοδυναμεί με αποτυχία του συστήματος.
- **Σταθερά:** Η σπάνια απώλεια χρονικού ορίου για την επεξεργασία μιας μονάδας δεδομένων είναι ανεκτή αλλά μπορεί να υποβαθμίσει την ποιότητα υπηρεσιών του συστήματος.
- **Χαλαρά:** Η χρησιμότητα ενός αποτελέσματος μειώνεται με το πέρας του χρονικού ορίου, υποβαθμίζοντας την ποιότητα υπηρεσιών του συστήματος.

Ένα επιτυχημένο σύστημα πραγματικού χρόνου πρέπει να είναι ικανό να επεξεργαστεί πολύ μεγάλο αριθμό δεδομένων συνεχόμενα. Επίσης, επιβάλλεται να παρέχει ένα μηχανισμό ερωτήσεων προς τη ροή, για τα τρέχοντα ή τα προγενέστερα δεδομένα. Η ακεραιότητα αυτών των δεδομένων, η αποθήκευση και η ανάκτησή τους πρέπει να εξασφαλίζεται ανά πάσα στιγμή, ανεξάρτητα από οποιαδήποτε αποτυχία του συστήματος.

Συνεπώς το σύστημα είναι απαραίτητο να διαχειρίζεται ατέλειες, όπως καθυστέρηση στην άφιξη, άφιξη εκτός σειράς, ή ακόμη και απώλεια των δεδομένων.

Τέλος, η παραγόμενη έξοδος συνηθίζεται να είναι προβλέψιμη και διαθέσιμη για επίβλεψη, έτσι ώστε να επιβεβαιώνεται ο ντετερμινισμός και η επαναληπτικότητα του συστήματος.

3.2 Apache NiFi

3.2.1 Εισαγωγή

Το NiFi σχεδιάστηκε για να αυτοματοποιήσει τη ροή των δεδομένων (dataflow) μεταξύ των συστημάτων. Ενώ, ο όρος ροή δεδομένων χρησιμοποιείται με ποικίλες έννοιες, στην παρούσα εργασία θα αναφέρεται ως η αυτόματη και διαχειριζόμενη ροή πληροφοριών μεταξύ των συστημάτων.



Εικόνα 7 : NiFi

Το πρόβλημα χώρου υπάρχει περίπου από τότε που οι επιχειρήσεις είχαν περισσότερα από ένα σύστημα, όπου μερικά από τα συστήματα δημιουργούσαν δεδομένα και άλλα καταλάωναν δεδομένα. Τα προβλήματα και οι τρόποι επίλυσης που προέκυψαν, συζητήθηκαν και διατυπώθηκαν εκτενώς. Μία περιεκτική και άμεσα υλοποιήσιμη μορφή βρίσκεται στο Enterprise Integration Patterns [\[EIP\]](#).

Μερικές από τις υψηλού επιπέδου προκλήσεις στη ροή δεδομένων περιλαμβάνουν:

- **Αποτυχίες συστήματος** : Αποτυχίες δικτύου, δίσκου, λογισμικού, ανθρώπινα λάθη.
- **Η πρόσβαση δεδομένα υπερβαίνει την ικανότητα κατανάλωσής τους** : Μερικές φορές μια δεδομένη πηγή δεδομένων μπορεί να ξεπεράσει κάποιο μέρος της αλυσίδας επεξεργασίας ή διανομής, το οποίο μπορεί να δημιουργήσει πρόβλημα σε μία αδύναμη σύνδεση (weak-link).
- **Οι οριακές συνθήκες είναι απλές προτάσεις** : Κατά κανόνα εξάγονται δεδομένα που μπορεί να είναι πάρα πολύ μεγάλα ή πάρα πολύ μικρά, πάρα πολύ γρήγορα ή πολύ αργή, διεφθαρμένα, λάθος ή σε λάθος μορφή.
- **Ταχύτατη εξέλιξη** : Οι προτεραιότητες των οργανισμών αλλάζουν γρήγορα, έτσι η ενεργοποίηση νέων ροών και η αλλαγή των υφισταμένων πρέπει να γίνεται το ίδιο γρήγορα.

- **Τα συστήματα εξελίσσονται με διαφορετικούς ρυθμούς :** Τα πρωτόκολλα και οι μορφοποιήσεις που χρησιμοποιούνται από ένα δεδομένο σύστημα μπορούν ανά πάσα στιγμή να αλλάξουν και συχνά ανεξάρτητα από τα συστήματα γύρω τους. Η ροή δεδομένων υπάρχει για να συνδέεται αυτό που ουσιαστικά είναι ένα μαζικά καταναμημένο σύστημα από συστατικά τα οποία είναι χαλαρά ή καθόλου σχεδιασμένα για να εργάζονται από κοινού.
- **Συμμόρφωση και ασφάλεια :** Νόμοι, κανονισμοί, πολιτικές και συμφωνίες ανάμεσα σε επιχειρήσεις επίσης αλλάζουν, έτσι χρειάζεται οι αλληλεπιδράσεις μεταξύ συστημάτων και ανάμεσα σε σύστημα και χρήστη να είναι ασφαλείς, αξιόπιστες και υπόλογες.
- **Συνεχής βελτίωση παρατηρείται στην παραγωγή :** Είναι συχνά αδύνατο να αντιγραφεί το περιβάλλον παραγωγής στο εργαστήριο.

Με τα χρόνια η ροή δεδομένων έγινε ένα από εκείνα τα αναγκαία κακά μίας αρχιτεκτονικής. Πλέον, όμως, υπάρχει ένας αριθμός ενεργών και ταχέως εξελισσόμενων κινήσεων που κάνουν τη ροή δεδομένων πολύ πιο ενδιαφέρον και ζωτικής σημασίας παράγοντα για την επιτυχία μιας δεδομένης επιχείρησης, όπως: Service Oriented Architecture [\[SOA\]](#), άνοδος του API [\[API\]](#), Internet of Things [\[IoT\]](#) και Big Data [\[bigdata\]](#). Επιπλέον, το επίπεδο αυστηρότητας που απαιτείται για συμμόρφωση, ιδιωτικότητα και ασφάλεια είναι συνεχώς σε άνοδο. Ακόμα και με όλες αυτές τις νέες έννοιες, τα πρότυπα και οι ανάγκες της ροής δεδομένων παραμένουν σε μεγάλο βαθμό τα ίδια. Οι βασικές διαφορές είναι η έκταση της πολυπλοκότητας, η τάξη της αλλαγής που είναι απαραίτητη να προσαρμοστεί και ότι κατά την επεκτασιμότητα η ακραία περίπτωση γίνεται σύνηθες φαινόμενο. Το NiFi είναι φτιαγμένο για να βοηθήσει στην αντιμετώπιση αυτών των σύγχρονων προκλήσεων ροής δεδομένων [\[7\]](#).

3.2.2 Βασικές έννοιες

Οι θεμελιώδεις ιδέες σχεδιασμού του NiFi σχετίζονται στενά με τη βασική φιλοσοφία γύρω από το Flow Based Programming [FBP], όπως φαίνεται στον παρακάτω πίνακα:

NiFi	FBP	Περιγραφή
FlowFile	Information Packet	Το FlowFile αντιπροσωπεύει κάθε αντικείμενο που μετακινείται μέσω του συστήματος, για κάθε ένα από το οποίο το NiFi παρακολουθεί ένα χαρακτηριστικό ζεύγος τιμών key/value και το σχετικό περιεχόμενό του από μηδέν ή περισσότερα bytes.
FlowFile Επεξεργαστής	Black Box	Η εργασία εκτελείται στην πραγματικότητα από τους επεξεργαστές. Σε EIP όρους, ένας επεξεργαστής κάνει κάποιους συνδυασμούς δρομολόγησης, μετασχηματισμού ή διαμεσολάβησης δεδομένων μεταξύ των συστημάτων. Οι επεξεργαστές έχουν πρόσβαση στα χαρακτηριστικά και τα περιεχόμενα των FlowFiles και μπορούν να λειτουργήσουν σε μηδέν ή περισσότερα FlowFiles σε μία μονάδα εργασίας, εκτελώντας είτε το έργο, είτε επαναφορά.
Σύνδεση (Connection)	Bounded Buffer	Οι συνδέσεις παρέχουν την επικοινωνία μεταξύ των επεξεργαστών. Ενεργούν ως ουρές και επιτρέπουν σε διάφορες διαδικασίες να αλληλεπιδρούν με διαφορετικούς ρυθμούς. Μπορεί να δοθεί προτεραιότητα στις ουρές δυναμικά και άνω φράγματα για το φορτίο.
Ελεγκτής Ροής (Flow Controller)	Χρονοπρογραμματιστής (Scheduler)	Ο Ελεγκτής Ροής διατηρεί τη γνώση του πώς οι διαδικασίες συνδέονται πραγματικά και διαχειρίζεται τα νήματα και τις κατανομές που χρησιμοποιούν αυτές. Ενεργεί ως μεσίτης, διευκολύνοντας την ανταλλαγή των FlowFiles μεταξύ των επεξεργαστών.
Ομάδα Επεξεργασίας (Process Group)	Υποδίκτυο (Subnet)	Μία Ομάδα Επεξεργασίας είναι ένα συγκεκριμένο σύνολο διαδικασιών και των συνδέσεων τους, η οποία μπορεί να λαμβάνει δεδομένα μέσω θυρών εισόδου και να στέλνει δεδομένα μέσω θυρών εξόδου. Με τον τρόπο αυτό, οι Ομάδες Επεξεργασίας επιτρέπουν τη δημιουργία εντελώς νέων συστατικών απλά με τη σύνθεση άλλων συστατικών.

Πίνακας 1 : Διαφορές NiFi – FBP

Αυτό το σχεδιαστικό μοντέλο, επίσης παρόμοιο με το [SEDA](#), παρέχει πολλές ευεργετικές συνέπειες που επιτρέπουν στο NiFi να είναι μια πολύ αποτελεσματική πλατφόρμα για την οικοδόμηση ισχυρών και επεκτάσιμων ροών δεδομένων. Μερικά από αυτά τα οφέλη είναι:

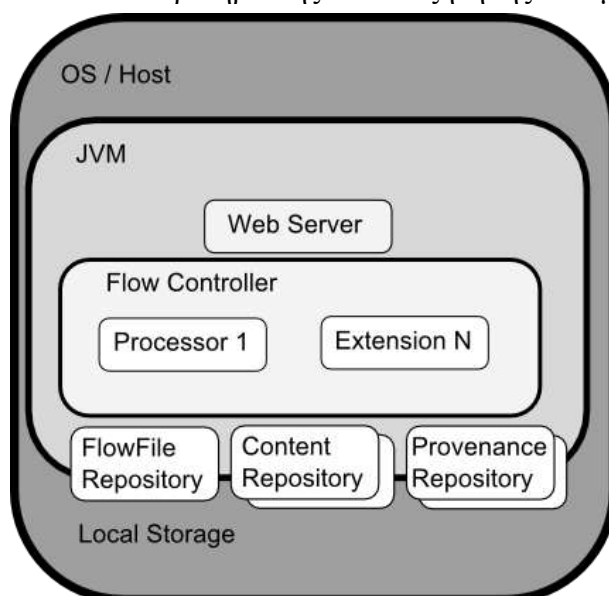
- βοηθάει στην εικονική δημιουργία και τη διαχείριση κατευθυνόμενων γράφων επεξεργαστών
- Είναι εγγενώς ασύγχρονη, πράγμα που επιτρέπει την πολύ υψηλή απόδοση και το φυσικό buffering ακόμη και αν ο βαθμός επεξεργασίας και ροής δεν είναι σταθερός
- Παρέχει ένα ιδιαίτερα σύμφωνο μοντέλο χωρίς να χρειάζεται ο προγραμματιστής να ανησυχεί για τα τυπικά εμπόδια συγχρονισμού
- Προωθεί την ανάπτυξη συνεκτικών και χαλαρά συνδεδεμένων συστατικών τα οποία μπορούν να επαναχρησιμοποιηθούν σε άλλα πλαίσια και προωθεί ελέγξιμες μονάδες
- Οι περιορισμένους συνδέσεις των πόρων κάνουν κρίσιμες λειτουργίες, όπως back-pressure και απελευθερώνουν την πίεση πολύ φυσικά και διαισθητικά
- Ο χειρισμός λαθών γίνεται πολύ φυσικά
- Τα σημεία στα οποία τα δεδομένα εισέρχονται και εξέρχονται από το σύστημα, καθώς επίσης και το πώς ρέουν διαμέσου του συστήματος είναι αρκετά κατανοητά και παρακολουθούνται εύκολα.

3.2.3 Αρχιτεκτονική

Το NiFi εκτελείται μέσα σε ένα JVM (Java Virtual Machine), το οποίο φιλοξενείται σε ένα host λειτουργικό σύστημα. Τα κύρια στοιχεία του NiFi που λειτουργούν στο JVM είναι τα ακόλουθα:

- **Web Server** : Σκοπός του web server είναι να φιλοξενήσει τις εντολές HTTP-based και τον έλεγχο API του NiFi.
- **Flow Controller (Ελεγκτής Ροής)** : Ο Flow Controller είναι ο εγκέφαλος της λειτουργίας. Παρέχει τα νήματα για να τρέχουν οι επεκτάσεις και διαχειρίζεται το χρονοδιάγραμμά τους για το πότε θα λάβουν πόρους προς εκτέλεση.

- **Extensions (Επεκτάσεις) :** Υπάρχουν διάφοροι τύποι επεκτάσεων για το NiFi, οι οποίες λειτουργούν/εκτελούνται εντός του JVM.
- **FlowFile Repository (Αποθετήριο) :** Το FlowFile Repository είναι το μέρος όπου το NiFi παρακολουθεί την κατάσταση των FlowFiles που είναι μία δεδομένη στιγμή ενεργά στη ροή. Η υλοποίησή του είναι συνδεόμενη (pluggable) και η προκαθορισμένη προσέγγιση είναι ένα επίμονο Write-Ahead αρχείο καταγραφής (Log), που βρίσκεται σε ένα συγκεκριμένο κομμάτι δίσκου.
- **Content Repository (Αποθετήριο Περιεχομένου) :** Το Content Repository είναι το μέρος όπου αποθηκεύονται τα πραγματικά bytes του περιεχομένου ενός FlowFile. Η υλοποίησή του είναι επίσης συνδεόμενη και η προκαθορισμένη προσέγγιση είναι ένας αρκετά απλός μηχανισμός, ο οποίος αποθηκεύει block δεδομένων στο σύστημα αρχείων. Μπορούν να καθοριστούν περισσότερες από μία τοποθεσίες αποθήκευσης στο σύστημα αρχείων, έτσι ώστε να δεσμευτούν διαφορετικές φυσικές κατατιμήσεις προκειμένου να μειωθεί η διαμάχη για κάθε ενιαίο τμήμα.
- **Provenance Repository (Αποθετήριο Προέλευσης) :** Το Provenance Repository είναι το μέρος όπου αποθηκεύονται όλα τα δεδομένα προέλευσης συμβάντων. Η κατασκευή του repository είναι και αυτή συνδεόμενη, με την προεπιλεγμένη υλοποίηση να είναι η χρησιμοποίηση ενός ή περισσότερων φυσικών τμημάτων δίσκου. Μέσα σε κάθε τμήμα υπάρχει δυνατότητα ευρετηρίασης και αναζήτησης δεδομένων συμβάντων.

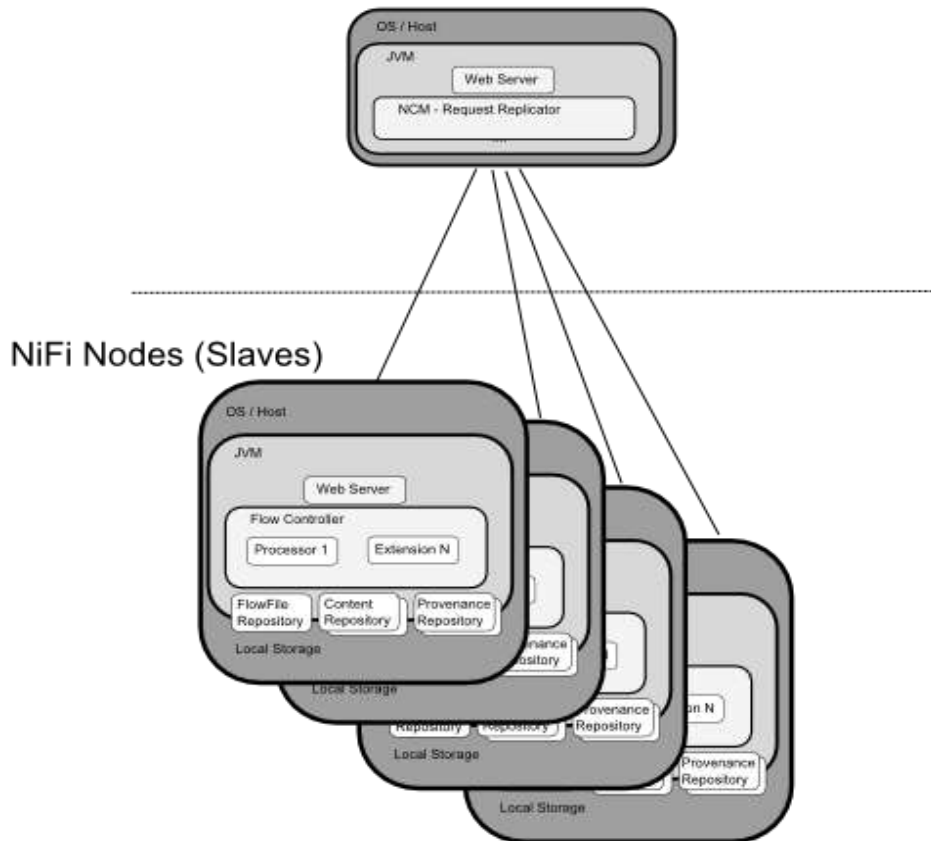


Εικόνα 8 :

Αρχιτεκτονική Κόμβου NiFi

Το NiFi μπορεί επίσης να λειτουργεί και σε σύμπλεγμα υπολογιστών, όπως φαίνεται στην Εικόνα 9.

NiFi Cluster Manager (Master)



Εικόνα 9 : Αρχιτεκτονική Σύμπλεγματος NiFi

Ένα σύμπλεγμα NiFi αποτελείται από έναν ή περισσότερους NiFi κόμβους (Nodes), οι οποίοι ελέγχονται από έναν ενιαίο NiFi Cluster Manager (NCM). Ο σχεδιασμός του συμπλέγματος ακολουθεί την αρχιτεκτονική master/slave, όπου ο NCM έχει το ρόλο του master και οι κόμβοι είναι οι σκλάβοι. Ο NCM έχει σχεδιαστεί για να παρακολουθεί ποιοι κόμβοι είναι στο σύμπλεγμα και την κατάστασή τους, καθώς και για να αναπαράγει αιτήματα τροποποίησης ή παρατήρησης της ροής. Ουσιαστικά, λοιπόν, το NCM κρατά σταθερή την κατάσταση του συμπλέγματος. Ενώ το μοντέλο αρχιτεκτονικής είναι master/slave, εάν ο master πεθάνει οι κόμβοι έχουν όλοι την εντολή να εξακολουθήσουν να λειτουργούν, για να εξασφαλιστεί ότι η ροή δεδομένων παραμένει ενεργή. Η απουσία του NCM σημαίνει απλώς ότι δεν μπορούν να ενταχθούν νέοι κόμβοι στο σύμπλεγμα και ότι δεν μπορούν να συμβούν αλλαγές στη ροή του συμπλέγματος μέχρι να αποκατασταθεί ο NCM.

3.2.4 Προσδοκώμενη επίδοση και χαρακτηριστικά του NiFi

Το NiFi έχει σχεδιαστεί για να αξιοποιήσει πλήρως τις δυνατότητες του υποκείμενου host συστήματος στο οποίο λειτουργεί. Αυτή η μεγιστοποίηση των πόρων είναι ιδιαίτερα ισχυρή σε σχέση με τη CPU και το δίσκο.

- **Είσοδος/Εξοδος (IO) :** Συνήθως η απόδοση ή η καθυστέρηση διαφέρει σε μεγάλο βαθμό ανάλογα με το πώς έχει ρυθμιστεί το σύστημα. Δεδομένου ότι υπάρχουν συνδεδεμένες προσεγγίσεις για τα περισσότερα από τα μεγάλα NiFi υποσυστήματα η απόδοση εξαρτάται από την υλοποίηση. Οι προεπιλεγμένες ρυθμίσεις υλοποίησης είναι όλες επίμονες, με εγγυημένη διανομή και μάλιστα χρησιμοποιώντας τον τοπικό δίσκο, επιτυγχάνοντας υψηλή απόδοση για το NiFi. Αυτό συμβαίνει καθώς αναμένεται γραμμική (linear) ανάπτυξη για κάθε φυσικό τμήμα και προστίθεται το Content Repository στο NiFi, επιφέροντας συμφόρηση σε κάποιο σημείο στο FlowFile Repository και στο Provenance Repository. Παρέχεται ένα template αξιολόγησης/τεστ απόδοσης στην κατασκευή, το οποίο επιτρέπει στους χρήστες να δοκιμάσουν εύκολα το σύστημά τους και να προσδιορίσουν που υπάρχει συμφόρηση και σε ποιο σημείο θα μπορούσε να δημιουργηθεί πρόβλημα.
- **CPU :** Ο Ελεγκτής Ροής λειτουργεί ως η συσκευή υπαγόρευσης όταν δοθεί σε ένα συγκεκριμένο επεξεργαστή ένα νήμα προς εκτέλεση. Οι επεξεργαστές θα πρέπει να σχεδιασμένοι να επιστρέφουν το νήμα μόλις τελειώσουν την εκτέλεση της εργασίας τους. Στον Ελεγκτή Ροής δίνεται μία τιμή διαμόρφωσης η οποία δείχνει πόσα νήματα θα πρέπει να υπάρχουν στις διάφορες περιοχές νημάτων που διατηρεί. Ο ιδανικός αριθμός των νημάτων προς χρησιμοποίηση εξαρτάται από τους πόρους του συστήματος υποδοχής (host), δηλαδή τον αριθμό των πυρήνων, εάν το σύστημα αυτό λειτουργεί και άλλες υπηρεσίες, καθώς και τη φύση της επεξεργασίας στη ροή. Για τυπικές IO βαριές ροές όμως, ορίζονται πολλές δεκάδες νημάτων να είναι διαθέσιμα.
- **RAM :** Το NiFi λειτουργεί εντός του JVM και γι' αυτό γενικά περιορίζεται στο χώρο μνήμης που παρέχεται από το JVM. Η συλλογή σκουπιδιών του JVM είναι ένας πολύ σημαντικός παράγοντας τόσο για τον περιορισμό του συνολικού πρακτικού μέγεθος του σωρού, όσο και για το πόσο καλά θα τρέχει η εφαρμογή με την πάροδο του χρόνου.

3.2.5 Πλεονεκτήματα από τη χρήση του NiFi

- **Εγγυημένη διανομή**

Βασική φιλοσοφία του NiFi είναι ότι ακόμη και σε πολύ υψηλή επεκτασιμότητα, η διανομή πρέπει να είναι εγγυημένη. Αυτό επιτυγχάνεται μέσω της αποτελεσματικής χρήσης ενός κατασκευαστικού σκοπού, επίμονου write-ahead αρχείου καταγραφής και content repository, τα οποία έχουν σχεδιαστεί με τέτοιο τρόπο ώστε να επιτρέπουν πολύ υψηλά ποσοστά συναλλαγής, αποτελεσματικό φορτίο διασποράς, copy-on-write και να συμβαδίζουν με τις αντοχές των παραδοσιακών αναγνώσεων/εγγραφών δίσκου.

- **Buffering Δεδομένων / Back Pressure και Pressure Release**

Το NiFi υποστηρίζει buffering όλων των δεδομένων ουράς, καθώς και την ικανότητα να παρέχει back pressure όσο αυτές οι ουρές φτάνουν τα καθορισμένα όρια ή να «γερνάει» τα δεδομένα όσο φθάνουν σε μια ορισμένη ηλικία (η αξία τους έχει χαθεί).

- **Ουρές με προτεραιότητα**

Το NiFi επιτρέπει την επιβολή ενός ή περισσοτέρων σχημάτων προτεραιότητας για το πώς τα δεδομένα ανακτώνται από μία ουρά. Η προεπιλεγμένη επιλογή είναι πρώτα τα παλαιότερα δεδομένα, αλλά υπάρχουν φορές που θα πρέπει πρώτα να τραβηχτούν τα πιο πρόσφατα δεδομένα, τα μεγαλύτερα ή με βάση κάποια άλλη προτεραιότητα.

- **Flow QoS (αφάνεια κατά απόδοσης, ανοχή στην απώλεια κ.λ.π.)**

Υπάρχουν σημεία σε μία ροή δεδομένων όπου τα δεδομένα είναι απολύτως κρίσιμης σημασίας και η απώλεια τους μη ανεκτή. Υπάρχουν επίσης φορές που θα πρέπει τα δεδομένα να υποβάλλονται σε επεξεργασία και να διανεμηθούν μέσα σε δευτερόλεπτα. Το NiFi επιτρέπει τη λεπτομερή, ειδική διαμόρφωση της ροής για τέτοιες περιπτώσεις.

- **Προέλευση των δεδομένων**

Το NiFi αυτόματα καταγράφει, κατατάσσει σε πίνακες και καθιστά διαθέσιμα στοιχεία προέλευσης όσο τα δεδομένα ρέουν μέσα στο σύστημα. Η πληροφορία αυτή καθίσταται εξαιρετικά σημαντική για την ενίσχυση της συμμόρφωσης, την αντιμετώπιση προβλημάτων, τη βελτιστοποίηση και άλλα σενάρια.

- **Ανάκτηση/Εγγραφή κυλιόμενου buffer ιστορικού**

Το Content Repository του NiFi έχει σχεδιαστεί για να λειτουργεί ως κυλιόμενο buffer ιστορικού. Τα δεδομένα αφαιρούνται μόνο καθώς γερνούν στο content repository ή εάν απαιτείται χώρος. Αυτό σε συνδυασμό με την ικανότητα για δεδομένα προέλευσης δημιουργούν μία εξαιρετικά χρήσιμη βάση για να επιτραπούν click-to-content, λήψη του περιεχομένου και επανάληψη, όλα σε ένα συγκεκριμένο σημείο του κύκλου ζωής ενός αντικειμένου.

- **Εντολές και Έλεγχος Απεικόνισης**

Οι ροές δεδομένων μπορούν να γίνουν αρκετά πολύπλοκες, γι' αυτό το να μπορούν να απεικονιστούν και να εκφραστούν οπτικά μπορεί να συμβάλει σημαντικά στη μείωση αυτής της πολυπλοκότητας και στον εντοπισμό περιοχών που πρέπει να απλουστευθούν. Το NiFi όχι μόνο επιτρέπει την οπτική καθιέρωση των ροών δεδομένων, αλλά το κάνει και σε πραγματικό χρόνο. Αν γίνει κάποια αλλαγή στη ροή δεδομένων, αυτή η αλλαγή παίρνει αμέσως δράση. Οι αλλαγές είναι λεπτομερείς και απομονωμένες από τα συστατικά που επηρεάζονται. Δεν χρειάζεται να σταματήσει μία ολόκληρη ροή ή ένα σύνολο ροών μόνο και μόνο για να πραγματοποιηθεί κάποια συγκεκριμένη τροποποίηση.

- **Διαγράμματα ροής (Flow Templates)**

Οι ροές δεδομένων τείνουν να είναι εξαιρετικά προσανατολισμένες σε μοτίβα και ενώ υπάρχουν συχνά πολλοί διαφορετικοί τρόποι να λυθεί ένα πρόβλημα, βοηθά σε μεγάλο βαθμό να μοιραστούν αυτές οι βέλτιστες πρακτικές. Τα διαγράμματα επιτρέπουν στους ειδικούς επί του θέματος να κατασκευάσουν και να δημοσιεύσουν τα σχέδια ροής τους και σε άλλους να επωφεληθούν και να συμβάλουν σε αυτά.

- **Ασφάλεια**

- **Από σύστημα σε σύστημα**

Μία ροή δεδομένων είναι χρήσιμη εφόσον είναι ασφαλής. Το NiFi σε κάθε σημείο μίας ροής δεδομένων προσφέρει ασφαλή ανταλλαγή μέσω της χρήσης πρωτοκόλλων με κρυπτογράφηση, όπως αμφίδρομη SSL. Επιπλέον, επιτρέπει στη ροή να κρυπτογραφεί και να αποκρυπτογραφεί το περιεχόμενο και να χρησιμοποιεί κοινόχρηστα κλειδιά ή άλλους μηχανισμούς είτε στην πλευρά του αποστολέα, είτε στην πλευρά του παραλήπτη.

- **Από χρήστη σε σύστημα**

Το NiFi επιτρέπει αμφίδρομη ταυτοποίηση SSL και παρέχει συνδεδεμένη εξουσιοδότηση, έτσι ώστε να μπορεί να ελέγξει σωστά την πρόσβαση ενός χρήστη και σε συγκεκριμένα επίπεδα (read-only, dataflow manager, admin). Αν ένας χρήστης εισάγει ευαίσθητα στοιχεία στη ροή, όπως ένας κωδικός πρόσβασης, αυτά κρυπτογραφούνται αμέσως στην πλευρά του server και δεν εκτίθενται ποτέ ξανά στην πλευρά του πελάτη, ακόμη και σε κρυπτογραφημένη μορφή.

- **Σχεδιασμένο για Επέκταση**

Το NiFi είναι σχεδιασμένο για να επεκτείνεται, όπως το ίδιο είναι και μια πλατφόρμα πάνω στην οποία οι διεργασίες ροής δεδομένων μπορούν να εκτελούνται και να αλληλεπιδρούν με ένα προβλέψιμο και επαναλαμβανόμενο τρόπο.

- **Σημεία επέκτασης**

Επεξεργαστές, Ελεγκτής Υπηρεσιών, Εργασίες Αναφοράς, Prioritizers, Διεπαφές Χρηστών Πελατών

- **Classloader Isolation**

Σε οποιοδήποτε component-based σύστημα μπορούν πολύ γρήγορα να προκύψουν προβλήματα εξαρτήσεων. Το NiFi το αντιμετωπίζει αυτό παρέχοντας ένα τροποποιημένο μοντέλο class loader, το οποίο εξασφαλίζει ότι κάθε δέσμη επέκτασης εκτίθεται σε ένα πολύ περιορισμένο σύνολο εξαρτήσεων. Με αυτό το τρόπο, οι επεκτάσεις μπορούν να κατασκευαστούν χωρίς να χρειάζεται να λαμβάνεται σοβαρά υπόψη αν θα μπορούσαν να έρθουν σε σύγκρουση με κάποια άλλη επέκταση. Η ιδέα αυτών των δεσμών επέκτασης ονομάζεται *NiFi Archives*.

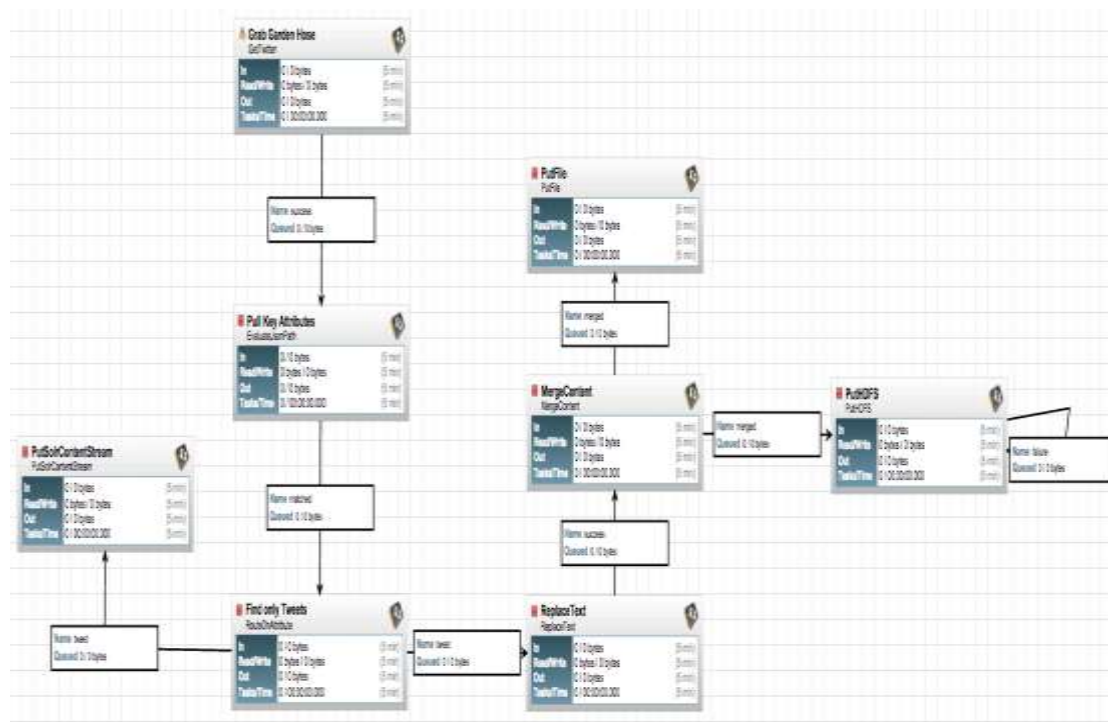
- **Ομαδοποίηση (scale-out)**

Το NiFi είναι σχεδιασμένο επεκτείνεται μέσα από την ομαδοποίηση πολλών κόμβων μαζί, όπως περιγράφηκε παραπάνω. Εάν τροφοδοτείται ένας μόνο κόμβος, ο οποίος έχει ρυθμιστεί για να χειρίζεται εκατοντάδες MBs, τότε ένα μικρό σύμπλεγμα θα μπορούσε να ρυθμιστεί ώστε να χειρίζεται GBs. Αυτό επιφέρει ενδιαφέρουσες προκλήσεις ως προς την εξισορρόπηση φορτίου (load balancing) και αποτυχίας (fail-over) μεταξύ του NiFi και των συστημάτων από τα οποία εξάγει τα δεδομένα. Η χρήση πρωτοκόλλων βασισμένων σε ασύγχρονες ουρές, όπως υπηρεσίες μηνυμάτων, Kafka κλπ, μπορεί να βοηθήσει. Η χρήση της λειτουργίας *site-to-site* του NiFi είναι επίσης

πολύ αποτελεσματική, δεδομένου ότι είναι ένα πρωτόκολλο που επιτρέπει στο NiFi και σε έναν πελάτη (μπορεί να είναι ένα άλλο σύμπλεγμα NiFi) να συνομιλούν, να μοιράζονται πληροφορίες σχετικά με το φορτίο και να ανταλλάζουν δεδομένων μέσω συγκεκριμένων εξουσιοδοτημένων θυρών.

3.2.6 Περιβάλλον Χρήστη

Το NiFi παρέχει επίσης ένα γραφικό περιβάλλον χρήστη (Web-based User Interface) για την απρόσκοπτη εμπειρία μεταξύ του σχεδιασμού, του ελέγχου, της ανατροφοδότησης και την παρακολούθηση των τρεχόντων τοπολογιών, όπως φαίνεται στην Εικόνα 10.



Εικόνα 10 : Γραφικό Περιβάλλον Χρήστη NiFi

Κεφάλαιο 4 : Hadoop

4.1 Γενικές πληροφορίες

Κατά τη διάρκεια των τελευταίων ετών, η βιομηχανία των Big Data έχει γνωρίσει τέτοια τεράστια άνοδο που οι περισσότερες από τις παγκόσμιες εταιρίες επενδύουν όλο και περισσότερο στην έρευνα και την προσέγγισή τους. Με απλά λόγια, ο όρος Big Data περιλαμβάνει τη διαχείριση πολύ μεγάλων όγκων δεδομένων, την επεξεργασία τους με έναν



*Εικόνα 11 :
Big Data και Hadoop*

χρονικά αποτελεσματικό τρόπο και την αποθήκευση τους σε μαζική κλίμακα. Ωστόσο, το αληθινό κίνητρο πίσω από τις μεγάλες επενδύσεις των εταιριών στο συγκεκριμένο τομέα δεν είναι μόνο η συλλογή των δεδομένων αλλά η ανάλυση και η αποσαφήνισή τους. Για το λόγο αυτό αναπτύσσονται διάφορες τεχνολογίες που μας απασχολούν όλο και περισσότερο το τελευταίο καιρό, όπως το Hadoop και το Spark, γνωστό και ως ξάδερφο του Hadoop, με προοπτική να το επισκιάσει στο εγγύς μέλλον.

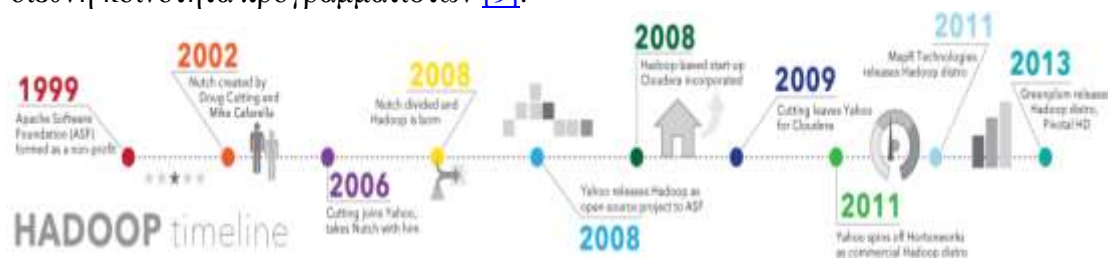
Το Hadoop γνώρισε τεράστια επιτυχία τα τελευταία δύο χρόνια και πλέον είναι αδιαμφισβήτητο το πιο καυτό θέμα των ημερών μας. Τα προηγούμενα χρόνια όλοι μιλούσαν γι' αυτό, ενώ τώρα όλοι καταπιάνονται με αυτό. Μερικοί βρίσκονται στη φάση της εγκατάστασης και της ανάπτυξης και άλλοι στα πρώτα στάδια υλοποίησης και προσομοίωσης. Και φυσικά άλλοι το χρησιμοποιούν ήδη για την αντιμετώπιση πραγματικών και δυσεπίλυτων, μέχρι προηγουμένως, επιχειρηματικών προβλημάτων.

Σήμερα, χρησιμοποιείται ευρέως στις περισσότερες βιομηχανίες, όπως στην οικονομία, στα μέσα ενημέρωσης και ψυχαγωγίας, από κυβερνήσεις, στην υγειονομική περίθαλψη, στις υπηρεσίες πληροφοριών, στο λιανικό εμπόριο και σε άλλες βιομηχανίες με μεγάλες απαιτήσεις δεδομένων. Είναι σχεδιασμένο για την επεξεργασία πολύ μεγάλου όγκου δεδομένων από terabytes σε petabytes και ακόμη μεγαλύτερων.

Το Hadoop είναι η κινητήρια δύναμη πίσω από την ανάπτυξη της βιομηχανίας των Big Data και χωρίς αμφιβολία είναι το “biggest thing” αυτή τη στιγμή αλλάζοντας ριζικά τον τρόπο με τον οποίο οι επιχειρήσεις αποθηκεύουν, επεξεργάζονται και αναλύουν δεδομένα [8].

4.2 Ιστορική αναδρομή

Το 2002 οι Doug Cutting και Mike Cafarella έφτιαξαν μία μηχανή αναζήτησης ανοιχτού κώδικα, το Nutch. Η ιδέα εμπνεύστηκε από την ίδια ιδέα που δημιούργησε τη μηχανή αναζήτησης Google και βασιζόταν στην αποθήκευση και επεξεργασία δεδομένων με ένα καταναμημένο και αυτοματοποιημένο τρόπο έτσι ώστε σχετικά αποτελέσματα αναζήτησης στον ιστό να επιστρέφονται γρηγορότερα. Το 2006 ο Cutting εντάσσεται στη Yahoo, παίρνοντας μαζί του το έργο Nutch και οδηγώντας στην απόσπαση του τμήματος καταναμημένου υπολογισμού και επεξεργασίας με την ονομασία Hadoop το 2008. Η ονομασία προήλθε από ένα παιχνίδι-ελέφαντα του γιου του Cutting. Ουσιαστικά, το Hadoop ήταν η απόρροια της προσπάθειας από το Yahoo να αντιμετωπίσει το πρόβλημα των υπέρογκων συνόλων δεδομένων, σπάζοντάς τα σε μικρότερα κομμάτια τα οποία θα μπορούσαν να υφίστανται παράλληλη επεξεργασία. Σήμερα, το πλαίσιο και το οικοσύστημα των τεχνολογιών του Hadoop διαχειρίζονται και συντηρούνται από το μη κερδοσκοπικό Apache Software Foundation (ASF) και μία διεθνή κοινότητα προγραμματιστών [9].



Εικόνα 12 : Ιστορική Αναδρομή

4.3 Ορισμός

Το Hadoop είναι ένα προγραμματιστικό πλαίσιο ανοιχτού κώδικα (open-source software framework), γραμμένο σε Java, για την καταναμημένη αποθήκευση

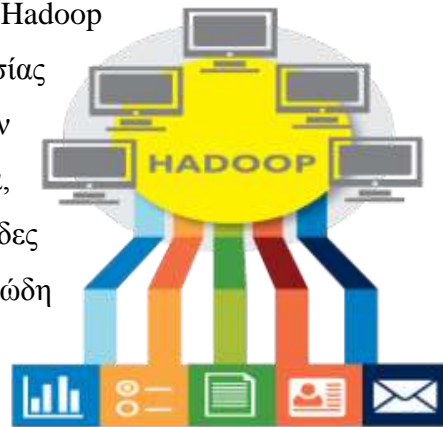


Εικόνα 13 : Hadoop Logo

και επεξεργασία πολύ μεγάλων συνόλων, δομημένων και αδόμητων, δεδομένων σε συστάδες υπολογιστών (clusters) με τη χρήση απλών μοντέλων προγραμματισμού. Παρέχει επεκτάσιμη (scalable) και αξιόπιστη μαζική αποθήκευση για κάθε είδος δεδομένων, τεράστια επεξεργαστική ισχύ και την ικανότητα να χειριστεί εικονικά απεριόριστες συντρέχουσες εργασίες [10].

4.4 Σχεδιασμός

Σε αντίθεση με τα παραδοσιακά συστήματα, το Hadoop επιτρέπει σε πολλαπλές αναλυτικές μονάδες εργασίας (multiple types of analytic workloads) να έχουν πρόσβαση στα ίδια, δομημένα και αδόμητα, δεδομένα, την ίδια χρονική στιγμή. Όλες οι μονάδες του Hadoop έχουν σχεδιαστεί με τη θεμελιώδη παραδοχή ότι οι αποτυχίες υλικού (hardware failures) είναι συχνές και θα πρέπει αυτόματα να αντιμετωπίζονται από το πλαίσιο λογισμικού (framework). Έτσι, αντί να παρέχεται υψηλή διαθεσιμότητα από το υλικό, η βιβλιοθήκη από μόνη της είναι σχεδιασμένη να εντοπίζει και να χειρίζεται τυχόν αποτυχίες σε επίπεδο εφαρμογής, προσφέροντας έτσι υψηλής διαθεσιμότητας υπηρεσίες πάνω από ένα σύμπλεγμα υπολογιστών (cluster), καθένας από τους οποίους είναι επιρρεπής σε αποτυχίες [11].



Εικόνα 14 : Αλληλεπίδραση του Hadoop

4.5 Λειτουργία

Το Hadoop είναι σχεδιασμένο να λειτουργεί ταυτόχρονα σε ένα μεγάλο αριθμό κόμβων, που δε μοιράζονται μνήμη ή δίσκους.

Έτσι, για την αποθήκευση των δεδομένων, τα αρχεία χωρίζονται σε μεγάλα τεμάχια (blocks) και διανέμονται σε πολλαπλούς κόμβους ενός συμπλέγματος υπολογιστών, με το Hadoop να παρακολουθεί που τοποθετούνται τα δεδομένα. Αν ένας κόμβος βγει εκτός σύνδεσης ή υποστεί βλάβη τα δεδομένα που βρίσκονται αποθηκευμένα σε αυτόν μπορούν να επανακτηθούν από κάποιο άλλο γνωστό, καλό αντίγραφο, αποφεύγοντας, έτσι, πιθανά σφάλματα υλικού [12].

Επιπλέον, για την επεξεργασία των δεδομένων, το Hadoop μεταφέρει στους κόμβους JAR αρχεία (Java Archive files), με βάση τα δεδομένα που χρήζουν επεξεργασίας, έτσι ώστε να τα επεξεργάζονται παράλληλα. Αυτή η προσέγγιση εκμεταλλεύεται την τοπικότητα των δεδομένων (data locality) – οι κόμβοι χειρίζονται μόνο τα δεδομένα στα οποία έχουν πρόσβαση – για να επιτρέψει στο σύνολο των

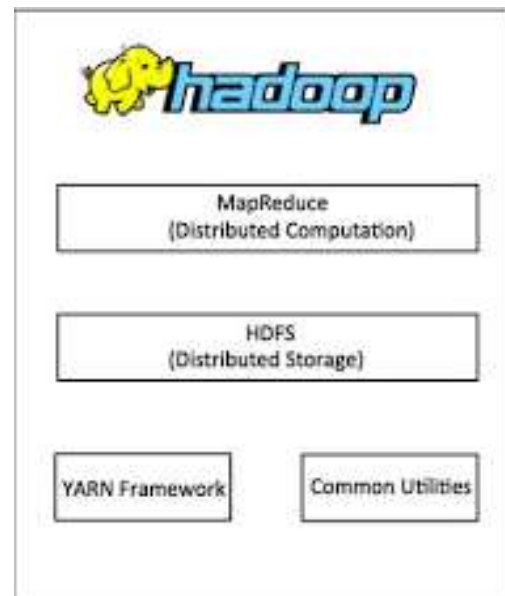
δεδομένων να υποστεί πιο γρήγορη και αποτελεσματική επεξεργασία απ' ότι θα μπορούσε σε μία πιο συμβατική αρχιτεκτονική βασιζόμενη σε ένα παράλληλο σύστημα αρχείων (Parallel File System, PFS), όπου υπολογισμοί και δεδομένα διανέμονται μέσω υψηλής ταχύτητας δικτύωσης (high-speed networking).

Αρχιτεκτονικά, ο λόγος που το Hadoop μπορεί να ασχοληθεί με πολλά δεδομένα ταυτόχρονα είναι επειδή τα διαμοιράζει στους διάφορους κόμβους. Και ο λόγος που είναι σε θέση να ζητάει πολύπλοκες υπολογιστικές ερωτήσεις είναι γιατί έχει όλους αυτούς τους επεξεργαστές των κόμβων, να εργάζονται παράλληλα προσδεδεμένοι μεταξύ τους.

4.6 Βασικό πλαίσιο

Ο πυρήνας του Apache Hadoop αποτελείται από τις εξής τέσσερις τεχνολογίες :

- *Hadoop Common* : Αποτελείται από Java βιβλιοθήκες και βοηθητικά προγράμματα που απαιτούνται για τη λειτουργία των υπολοίπων τμημάτων του Hadoop.
- *Hadoop Distributed File System (HDFS)* : Ένα κατακευματισμένο σύστημα αρχείων σε Java που παρέχει κλιμακώσιμη (scalable) και αξιόπιστη αποθήκευση δεδομένων και υψηλής απόδοσης πρόσβαση σε δεδομένα εφαρμογής.
- *Hadoop YARN (Yet Another Resource Negotiator)* : Μία πλατφόρμα για τη διαχείριση των πόρων του συμπλέγματος υπολογιστών και το χρονοπρογραμματισμό των εργασιών του χρήστη.
- *Hadoop MapReduce* : Ένα προγραμματιστικό μοντέλο, βασισμένο στο YARN, για την παράλληλη επεξεργασία μεγάλων συνόλων δεδομένων.



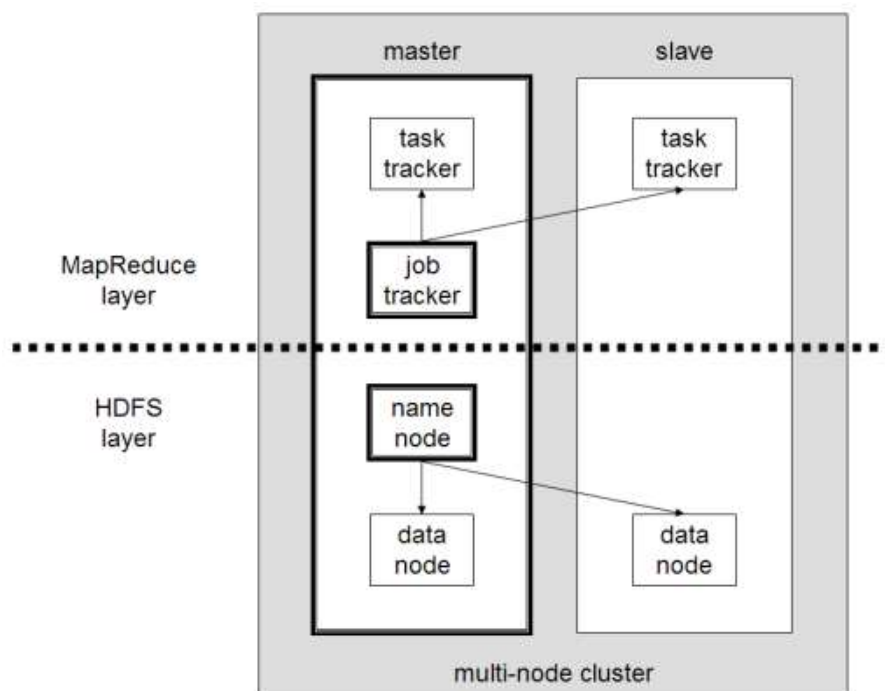
Εικόνα 15 : Βασικό Πλαίσιο του Hadoop

4.7 Αρχιτεκτονική

Το βασικό πλαίσιο του Hadoop, όπως αναφέρθηκε προηγουμένως, περιλαμβάνει το Hadoop Common πακέτο, το οποίο παρέχει το σύστημα αρχείων και το στρώμα αφαίρεσης του λειτουργικού συστήματος (OS abstraction layer). Σε αυτό περιέχονται οι απαραίτητες API διεπαφές (ρουτίνες, πρωτόκολλα και εργαλεία) για την ανάπτυξη λογισμικού και εφαρμογών, και τα απαιτούμενα JAR αρχεία και scripts για την εκκίνηση του Hadoop.

Το Hadoop περιλαμβάνει, επίσης, μία MapReduce μηχανή (είτε MapReduce/MR1, είτε YARN/MR2) και ένα σύστημα αρχείων σε επίπεδο χρήστη που διαχειρίζεται τους πόρους αποθήκευσης σε όλο το σύμπλεγμα. Για φορητότητα ανάμεσα σε πλατφόρμες – Linux, FreeBSD, Mac OS/X, Solaris και Windows – και ευκολία εγκατάστασης, τα δύο αυτά βασικά συστατικά είναι γραμμένα σε Java και απαιτούν μόνο χαμηλού κόστους υλικό (commodity hardware) [13].

Το Hadoop χρησιμοποιεί master/slave αρχιτεκτονική τόσο για κατανεμημένη αποθήκευση, όσο και για κατανεμημένο υπολογισμό. Ένα μικρό σύμπλεγμα Hadoop περιλαμβάνει ένα κύριο κόμβο (master) και πολλαπλούς κόμβους-εργάτες (slaves). Ο κύριος κόμβος αποτελείται από JobTracker, TaskTracker, NameNode και DataNode. Ένας κόμβος-εργάτης δρα τόσο ως DataNode όσο και ως TaskTracker (Εικόνα 16).



Εικόνα 16 : Αρχιτεκτονική Hadoop

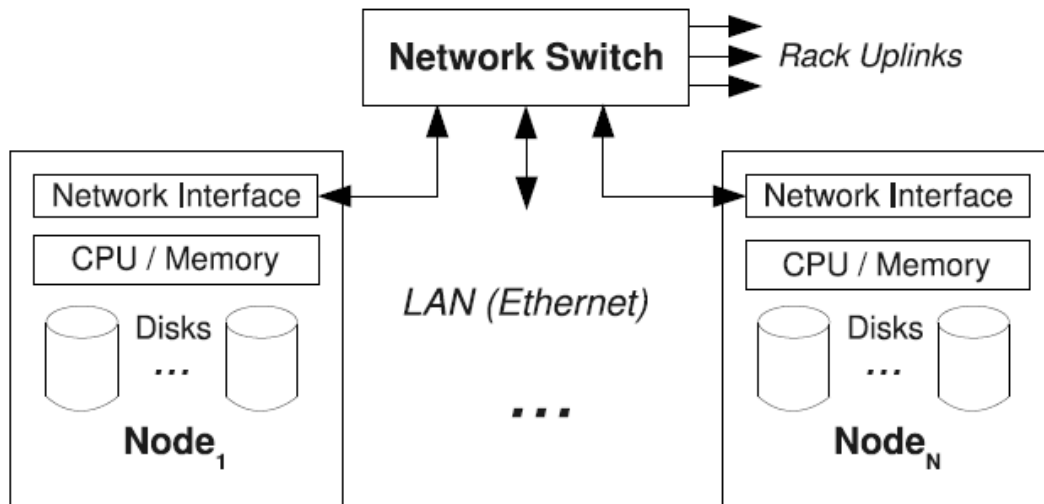
4.7.1 Hadoop Cluster

Λόγω της εξαιρετικά παραλληλοποιήσιμης φύσης του υπολογιστικού μοντέλου MapReduce, τα μεγάλα συμπλέγματα υπολογιστών γίνονται ευκολότερα εκμεταλλεύσιμα για την επίτευξη υψηλότερης απόδοσης στην εφαρμογή. Ως εκ τούτου, σε ένα Data-intensive Computing (DC) σύμπλεγμα, η δυναμική κάθε κόμβου μεμονωμένα είναι λιγότερο σημαντική από τη συνολική ικανότητά τους για κλιμάκωση του αριθμού των κόμβων στο σύμπλεγμα. Με βάση αυτή τη φιλοσοφία, τα DC συστήματα, που είναι σχεδιασμένα να τρέχουν υπολογιστικά πλαίσια όπως το Hadoop, έχουν κατασκευαστεί με τις ακόλουθες τεχνολογίες χαμηλού κόστους: επεξεργαστές x86, δίκτυα Ethernet και Serial ATA (SATA) σκληρούς δίσκους. Οποιαδήποτε άλλη επιλογή τεχνολογίας αυξάνει το ανά κόμβο κόστος του συμπλέγματος και τελικά περιορίζει τον αριθμό των κόμβων που μπορούν να αγοραστούν και να χρησιμοποιηθούν, οικονομικά.

Σε ένα τέτοιο σύμπλεγμα, η συνολική απόδοση του συστήματος καθορίζεται από το εύρος ζώνης αποθήκευσης. Κάθε Map ή Reduce εργασία πρέπει να έχει επαρκές διαθέσιμο εύρος ζώνης αποθήκευσης για να ολοκληρώσει αποτελεσματικά το έργο της σε ένα δεδομένο κόμβο υπολογισμού. Η απαίτηση αυτή έχει οδηγήσει σε μια αρχιτεκτονική συμπλέγματος κατά την οποία οι τοπικοί δίσκοι στους κόμβους υπολογισμού χρησιμοποιούνται ως μέρος ενός κατανεμημένου συστήματος αρχείων για την αποθήκευση των blocks. Για να εκμεταλλευτεί αποτελεσματικά το εύρος ζώνης των τοπικών δίσκων, το Hadoop επιχειρεί να αναθέσει τις εργασίες στους κόμβους εκείνους όπου είναι αποθηκευμένα τα δεδομένα εισόδου της εκάστοτε εργασίας.

Ένα παράδειγμα αρχιτεκτονικής ενός συμπλέγματος Hadoop σε ένα ενιαίο ράφι (rack) φαίνεται στην Εικόνα 17, όπου κάθε κόμβος υπολογισμού είναι εξοπλισμένος με έναν ή περισσότερους δίσκους και οι κόμβοι διασυνδέονται σε ένα δίκτυο Ethernet. Ένα ράφι είναι πιθανό να είναι διασυνδεδεμένο με έναν ενιαίο, υψηλής απόδοσης διακόπτη Ethernet, που προσφέρει πλήρες εύρος ζώνης μεταξύ όλων των κόμβων εντός του ραφιού. Κάθε ράφι στη συνέχεια συνδέεται με τα υπόλοιπα ράφια μέσω μιας ιεραρχίας διακοπών Ethernet, με πολύ μικρότερο, όμως, εύρος ζώνης μεταξύ των ραφιών, λόγω του κόστους και των περιορισμών καλωδίωσης. Ενώ, κάθε κόμβος μπορεί να επικοινωνεί με οποιονδήποτε άλλο κόμβο (στο ίδιο ή σε διαφορετικό ράφι),

υπάρχει πολύ μεγαλύτερο διαθέσιμο εύρος ζώνης μέσα σε ένα ράφι από ό,τι ανάμεσα στα ράφια [14].



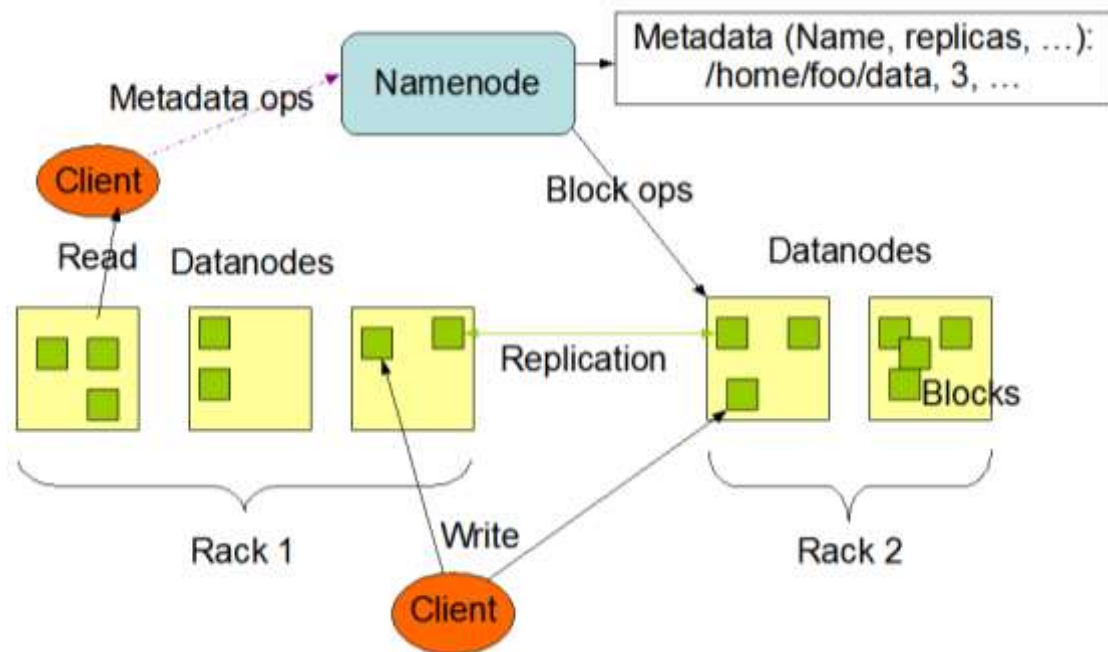
Εικόνα 17 : Αρχιτεκτονική Συμπλέγματος Hadoop

Το Hadoop απαιτεί Java Runtime Environment (JRE) 1.6 ή κάποια νεότερη έκδοση. Τα τυποποιημένα σενάρια (scripts) εκκίνησης και τερματισμού απαιτούν μία SSH σύνδεση (Secure Shell) μεταξύ των κόμβων του συμπλέγματος.

4.7.2 HDFS

Το Hadoop Distributed System File είναι ένα κλιμακώσιμο και αξιόπιστο καταναμημένο σύστημα αποθήκευσης που συναθροίζει την μνήμη κάθε κόμβου ενός συμπλέγματος υπολογιστών σε ένα ενιαίο σύστημα αρχείων. Το HDFS αποθηκεύει μεμονωμένα αρχεία σε μεγάλα τεμάχια, πετυχαίνοντας έτσι την αποτελεσματική αποθήκευση πολύ μεγάλων και/ή πολυάριθμων αρχείων σε πολλές μηχανές και την παράλληλη πρόσβαση σε μεμονωμένα κομμάτια δεδομένων, χωρίς να χρειάζεται να αναγνωστεί ολόκληρο το αρχείο στη μνήμη ενός και μόνο υπολογιστή. Η αξιοπιστία του επιτυγχάνεται με την αντιγραφή των δεδομένων σε πολλούς και διαφορετικούς κόμβους, με κάθε τεμάχιο δεδομένων να αποθηκεύεται σε τουλάχιστον τρεις. Εάν ένας κόμβος αποτύχει για κάποιο λόγο, τα δεδομένα παραμένουν διαθέσιμα και ένα επιπλέον αντίγραφο των τεμαχίων που κατέχει μπορεί να δημιουργηθεί σε άλλο κόμβο για την προστασία απέναντι σε μελλοντικές αποτυχίες [15].

Το HDFS ακολουθεί την αρχιτεκτονική Master/Slave. Αποτελείται από ένα NameNode (master) και πολλούς DataNodes (slaves). Ο χώρος αποθήκευσης μπορεί να βρίσκεται σε ετερογενή λειτουργικά συστήματα και είναι ενιαίος για όλο το σύμπλεγμα υπολογιστών. Οι πελάτες (clients) μπορούν να βρουν τις τοποθεσίες των τεμαχίων και να προσπελούν τα δεδομένα απευθείας.



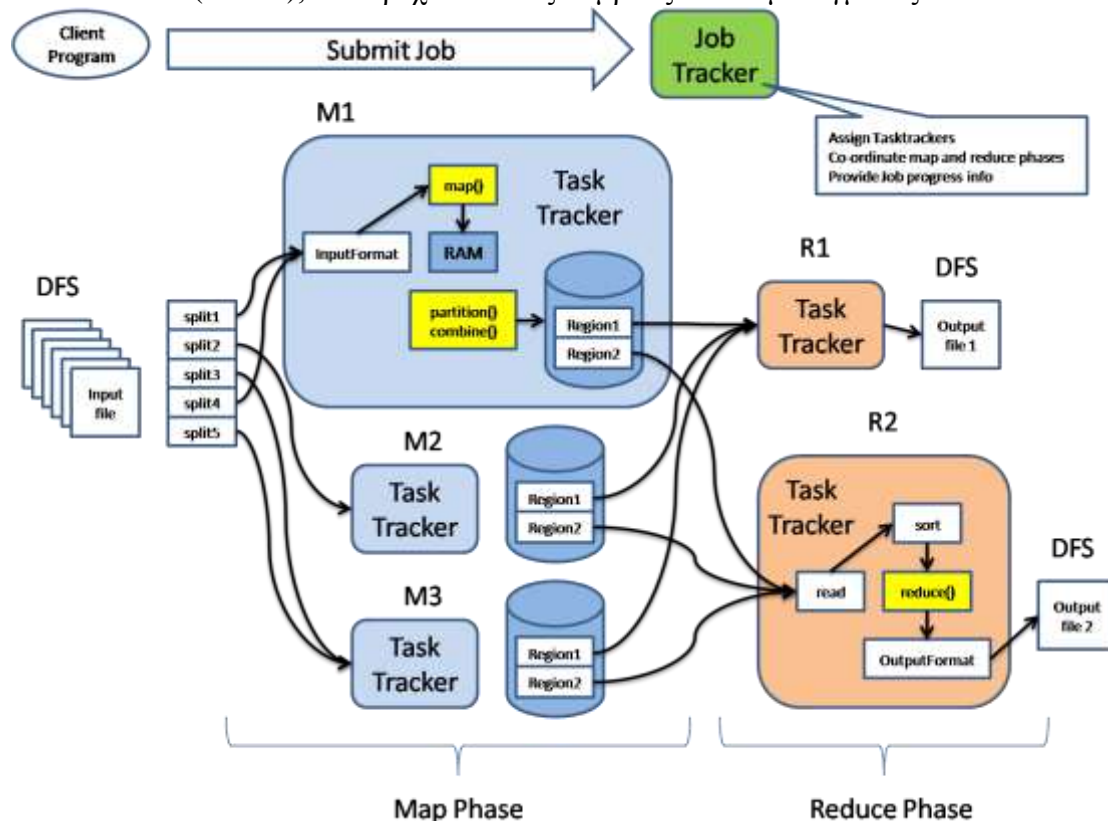
Εικόνα 18 : Αρχιτεκτονική HDFS

Συνολικά, το HDFS εξασφαλίζει διαθεσιμότητα (availability) και υψηλής απόδοσης πρόσβαση (high throughput access) στα δεδομένα. Παρέχει γνωστοποίηση τοποθεσίας (location awareness): το όνομα του ραφιού (του διακόπτη δικτύου ακριβέστερα) όπου βρίσκεται κάθε κόμβος, για τον αποτελεσματικό προγραμματισμό των εργασιών. Φροντίζει για τα αντίγραφα (replication) και την τοπικότητα των δεδομένων (data locality), ενώ παρέχει ανοχή σε σφάλματα (fault tolerance) και διαθέτει την ικανότητα να εξυπηρετεί ένα μεγάλο σύνολο από πελάτες. Χρησιμοποιείται σαν είσοδος δεδομένων και έξοδος αποτελεσμάτων από το MapReduce.

Η αρχιτεκτονική του HDFS περιγράφεται αναλυτικά στο [Υπόμνημα Α](#)

4.7.3 MapReduce

Πάνω από το σύστημα αρχείων έρχεται η MapReduce μηχανή, η οποία είναι υπεύθυνη για τη διαχείριση των εργασιών. Όπως φαίνεται στην Εικόνα 19, το σχεδιαστικό μοντέλο γύρω από την υλοποίηση του MapReduce ακολουθεί την αρχιτεκτονική Master/Slave, όπου οι εφαρμογές πελάτη αναθέτουν τις MapReduce εργασίες τους στο JobTracker (Master) και αυτός με τη σειρά του τις μοιράζει στους TaskTrackers (Slaves), που τρέχουν στους κόμβους του συμπλέγματος.



Εικόνα 19 : Αρχιτεκτονική MapReduce

Το Hadoop βελτιστοποιεί αυτόματα την εκτέλεση των MapReduce προγραμμάτων, καθώς σε ένα σύστημα αρχείων με επίγνωση θέσης (location-aware filesystem), όπως το HDFS, ο JobTracker γνωρίζει ποιοι κόμβοι περιέχουν τα δεδομένα, καθώς και ποιοι από αυτούς βρίσκονται σε κοντινή απόσταση. Έτσι, προωθεί την εργασία στους διαθέσιμους TaskTrackers προσπαθώντας να την κρατήσει όσο πιο κοντά γίνεται στα δεδομένα. Εάν ο κόμβος όπου βρίσκονται τα δεδομένα είναι ήδη κατειλημμένος, δίνεται προτεραιότητα στους κοντινότερους κόμβους. Εάν ένας TaskTracker αποτύχει ή εκπνεύσει, η εργασία επαναπρογραμματίζεται. Ο TaskTracker σε κάθε κόμβο δημιουργεί από ένα ξεχωριστό Java Virtual Machine (JVM) για να εκτελέσει την εργασία, με σκοπό να αποφευχθεί η αποτυχία του ίδιου του TaskTracker εάν η

τρέχουσα εργασία οδηγεί σε σφάλμα το JVM. Κάθε TaskTracker επικοινωνεί περιοδικά με το JobTracker μέσω ενός heartbeat μηνύματος για να αναφέρει την πρόοδο διεκπεραίωσης της εργασίας και να ζητήσει πρόσθετες εργασίες όταν βρίσκεται σε περίοδο αδράνειας.

Το MapReduce είναι η καρδιά του Hadoop. Είναι το μοντέλο προγραμματισμού που επιτρέπει στο Hadoop τη μαζική επέκταση σε εκατοντάδες ή χιλιάδες servers σε ένα σύμπλεγμα υπολογιστών και δημιουργήθηκε με σκοπό την εύκολη ανάπτυξη εφαρμογών που επεξεργάζονται γρήγορα, παράλληλα και καταναμημένα τεράστιες ποσότητες δεδομένων, με έναν αξιόπιστο και ανεκτικό σε βλάβες τρόπο. Το MapReduce σπάει μεγάλα προβλήματα επεξεργασίας δεδομένων σε πολλαπλά στάδια, δηλαδή σε μια σειρά από Map και Reduce βήματα, όπου το καθένα έχει ζευγάρια key/value (κλειδιού/τιμής) για είσοδο και έξοδο και μπορεί να εκτελεστεί την ίδια στιγμή (παράλληλα) σε πολλούς υπολογιστές.

Στο MapReduce μοντέλο, ο υπολογισμός χωρίζεται στα ακόλουθα τμήματα [\[16\]](#):

- Input Reader

Ο αναγνώστης εισόδου χωρίζει την είσοδο σε κατάλληλου μεγέθους «splits» (κομμάτια συνήθως 64-128 MB) και το μοντέλο εκχωρεί ένα split σε κάθε συνάρτηση Map. Ο αναγνώστης εισόδου διαβάζει δεδομένα από κάποιο σταθερό μέσο αποθήκευσης (όπως ένα καταναμημένο σύστημα αρχείων) και δημιουργεί ζεύγη key/value. Για παράδειγμα, θα διαβάσει ένα φάκελο γεμάτο με αρχεία κειμένου και θα επιστρέψει κάθε πρόταση ως ένα ξεχωριστό αρχείο.

- Map Function

Η συνάρτηση Map λαμβάνει μια σειρά από ζεύγη key/value, επεξεργάζεται το καθένα από αυτά και παράγει μηδενικά ή περισσότερα ζεύγη εξόδου key/value. Οι τύποι εισόδου και εξόδου της Map μπορεί να είναι (και συχνά είναι) διαφορετικοί μεταξύ τους.

Εάν η εφαρμογή κάνει καταμέτρηση λέξεων, η συνάρτηση Map θα σπάσει την πρόταση σε λέξεις και θα εξάγει ένα ζεύγος key/value για κάθε λέξη. Κάθε ζεύγος εξόδου θα περιέχει ως key τη λέξη και ως value τον αριθμό των εμφανίσεων αυτής της λέξης στην πρόταση.

- Partition Function

Κάθε έξοδος της συνάρτησης Map κατανέμεται σε ένα συγκεκριμένο reducer (κόμβος όπου θα εκτελεστεί η συνάρτηση Reduce) από τη συνάρτηση Partition της εφαρμογής, για λόγους θρυμματοποίησης. Η συνάρτηση Partition δέχεται το κλειδί και τον αριθμό των reducers και επιστρέφει το δείκτη (index) του επιθυμητού reducer.

Μεταξύ του Map και του Reduce βήματος, τα δεδομένα ανακατεύονται (shuffle) – ανταλλάσσονται μεταξύ των κόμβων και παράλληλα ταξινομούνται – για να μετακινηθούν από το Map κόμβο, όπου παράχθηκαν, στον κόμβο Reduce.

- Comparison Function

Η είσοδος για κάθε Reduce τραβιέται από το μηχάνημα όπου έτρεξε η Map και ταξινομείται με τη χρήση της συνάρτησης Comparison της εφαρμογής.

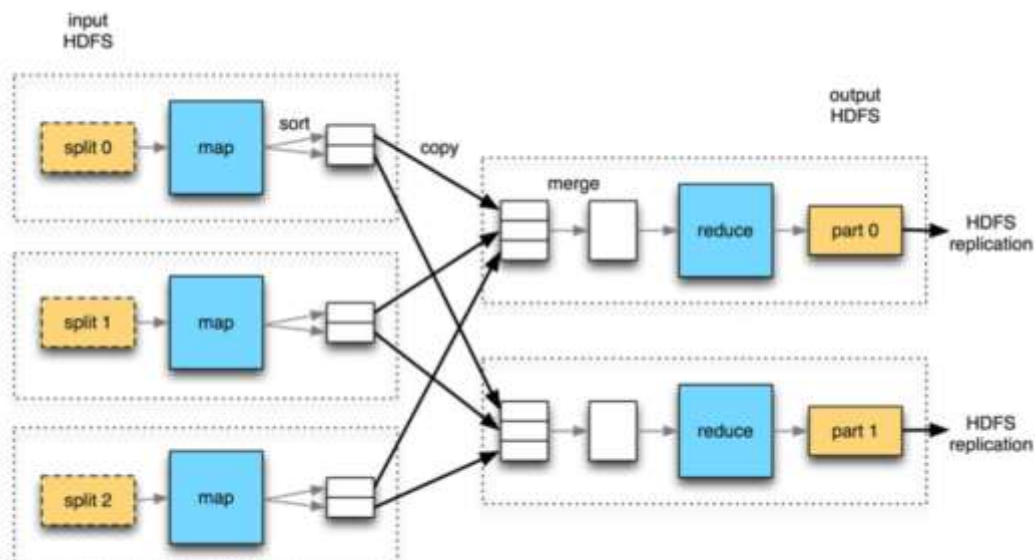
- Reduce Function

Το μοντέλο καλεί με ταξινομημένη σειρά τη συνάρτηση Reduce της εφαρμογής, μία φορά για κάθε μοναδικό key. Η Reduce μπορεί να επανακληθεί αναδρομικά μέσα από τα values που σχετίζονται με αυτό το key και παράγει μηδέν ή περισσότερες εξόδους.

Στο προηγούμενο παράδειγμα, η συνάρτηση Reduce παίρνει τα values εισόδου, τα συνοψίζει και δημιουργεί μια ενιαία έξοδο της λέξης και του τελικού αθροίσματος.

- Output Writer

Ο συγγραφέας εξόδου εγγράφει την έξοδο από τη Reduce στο HDFS



Εικόνα 20 : Τμήματα MapReduce

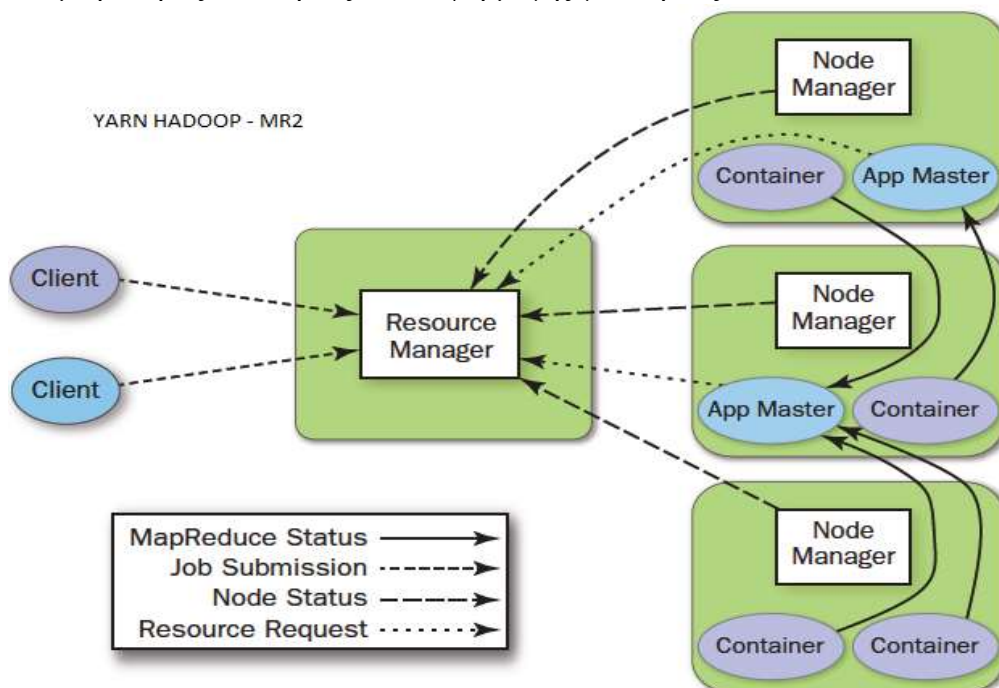
4.7.4 YARN

Στη 2^η γενιά του Hadoop, το MapReduce έχει υποστεί μια πλήρη αναμόρφωση και πλέον διαμορφώνεται σε αυτό που αναφέρεται ως MapReduce 2.0 (MRv2) ή YARN.

Η βασική ιδέα πίσω από το YARN είναι να μοιραστούν οι δύο κύριες λειτουργίες του JobTracker, διαχείριση πόρων και χρονοπρογραμματισμός/εποπτεία, σε ξεχωριστές οντότητες, χρησιμοποιώντας έναν ενιαίο ResourceManager και έναν ApplicationMaster ανά εφαρμογή [17].

Ο ResourceManager και ο NodeManager διαμορφώνουν το νέο σύστημα για τη διαχείριση των εφαρμογών με ένα κατανεμημένο τρόπο. Ο ResourceManager είναι η απόλυτη αρχή που διαιτητεύει του πόρους μεταξύ όλων των εφαρμογών του συστήματος. Ο ApplicationMaster είναι μία ειδική οντότητα λογισμικού που διαπραγματεύεται πόρους από το ResourceManager και συνεργάζεται με τους NodeManagers για την εκτέλεση και την παρακολούθηση των συστατικών των εργασιών.

Ο ResourceManager έχει ένα Scheduler, ο οποίος είναι υπεύθυνος για την κατανομή των πόρων στις διάφορες εφαρμογές που τρέχουν στο σύμπλεγμα, σύμφωνα με περιορισμούς όπως χωρητικότητα ουράς και όρια χρηστών. Ο χρονοπρογραμματισμός γίνεται με βάση τις απαιτήσεις κάθε εφαρμογής για πόρους.



Εικόνα 21 : Αρχιτεκτονική YARN

Κάθε ApplicationMaster έχει την ευθύνη για τη διαπραγμάτευση κατάλληλων δοχείων (Container) πόρων από το Scheduler, παρακολουθώντας την κατάστασή τους και την πρόοδό τους.

Ο NodeManager είναι ο ανά-μηχανή σκλάβος, ο οποίος είναι υπεύθυνος για την δρομολόγηση των Containers των εφαρμογών, παρακολουθώντας τη χρήση των πόρων τους (CPU, μνήμη, δίσκο, δίκτυο) και αναφέροντάς την στο ResourceManager.

Το YARN ενισχύει ένα σύμπλεγμα υπολογισμού με τους ακόλουθους τρόπους:

- Πολυμετοχικότητα (Multi-tenancy)

Επιτρέπει σε πολλαπλές μηχανές πρόσβασης (είτε open-source ή ιδιόκτητες) να χρησιμοποιούν το Hadoop ως το κοινό πρότυπο για batch, διαδραστικές και πραγματικού χρόνου μηχανές που μπορούν να έχουν πρόσβαση ταυτόχρονα στο ίδιο σύνολο δεδομένων.

- Χρησιμοποίηση του συμπλέγματος (Cluster Utilization)

Η δυναμική κατανομή των πόρων του συμπλέγματος βελτιώνει τη χρησιμοποίηση έναντι των περισσότερο στατικών MapReduce κανόνων που χρησιμοποιούνται σε παλαιότερες εκδόσεις του Hadoop.

- Επεκτασιμότητα (Scalability)

Η επεξεργαστική ισχύς των κέντρων δεδομένων συνεχίζει να επεκτείνεται με ταχείς ρυθμούς. Ο ResourceManager του YARN επικεντρώνεται αποκλειστικά στον προγραμματισμό και κρατά ρυθμό όσο το σύμπλεγμα επεκτείνεται σε χιλιάδες κόμβους που διαχειρίζονται petabytes δεδομένων.

- Συμβατότητα (Compatibility)

Το YARN διατηρεί την API συμβατότητα με την προηγούμενη έκδοση του Hadoop. Αυτό σημαίνει ότι υφιστάμενες MapReduce εφαρμογές μπορούν να τρέξουν το YARN χωρίς καμία αναστάτωση στις διεργασίες που ήδη λειτουργούν.

Το YARN είναι η προϋπόθεση για το επιχειρηματικό Hadoop και παρέχει διαχείριση πόρων και μια κεντρική πλατφόρμα όπου προσφέρονται συνεπείς εργασίες, ασφάλεια και εργαλεία διαχείρισης δεδομένων σε συμπλέγματα Hadoop. Επεκτείνει επίσης τη δύναμη του Hadoop σε παλιές και νέες τεχνολογίες που βρίσκονται μέσα στο κέντρο δεδομένων, ώστε να μπορούν να επωφεληθούν από πλευράς κόστους, γραμμικής κλίμακας αποθήκευσης και επεξεργασίας. Είναι το μέλλον του Hadoop.

4.8 Το Οικοσύστημα

Ο όρος Hadoop δεν αναφέρεται μόνο στο βασικό πλαίσιο που περιγράφηκε παραπάνω, αλλά και στο οικοσύστημα ή συλλογή επιπρόσθετων πακέτων λογισμικού που μπορούν να εγκατασταθούν παράλληλα με το Hadoop ή στην κορυφή του. Τα εργαλεία αυτά μπορούν να ταξινομηθούν σε Hadoop Data Services (Υπηρεσίες Δεδομένων) και σε Hadoop Operational Services (Λειτουργικές Υπηρεσίες) με βάση την λειτουργικότητα που προσφέρουν [18].

Hadoop Data Services είναι τα εργαλεία εκείνα που επιτρέπουν στους χρήστες να χειρίζονται και να επεξεργάζονται δεδομένα ευκολότερα. Μερικά από αυτά είναι:

- Apache Hive

Το Hive είναι υποδομή αποθήκης δεδομένων, χτισμένη στην κορυφή του Hadoop για την παροχή σύνοψης δεδομένων, ad-hoc query και ανάλυσης μεγάλων συνόλων δεδομένων. Παρέχει ένα μηχανισμό για μετάδοση δομής πάνω στα δεδομένα και για την αναζήτηση αυτών χρησιμοποιώντας μία γλώσσα τύπου SQL που ονομάζεται HiveQL (HQL). Το Hive διευκολύνει την επικοινωνία μεταξύ του Hadoop και διαφόρων εργαλείων απεικόνισης και business intelligence (BI).

- Apache Pig

Το Pig επιτρέπει τη συγγραφή σύνθετων MapReduce μεταμορφώσεων, χρησιμοποιώντας μια απλή scripting γλώσσα που ονομάζεται Pig Latin. Η Pig Latin καθορίζει ένα σύνολο μεταμορφώσεων σε ένα σύνολο δεδομένων, όπως άθροισμα, join και ταξινόμηση. Το Pig μεταφράζει την Pig Latin σε Hadoop MapReduce έτσι ώστε να μπορεί να εκτελεστεί μέσα στο HDFS.

- Apache HCatalog

Το HCatalog είναι ένα στρώμα διαχείρισης πινάκων και αποθηκών για το Hadoop, που επιτρέπει σε χρήστες με διαφορετικά εργαλεία επεξεργασίας δεδομένων – συμπεριλαμβανομένων των Hive, Pig και MapReduce – να πραγματοποιούν εγγραφές και αναγνώσεις δεδομένων ευκολότερα. Οι αφαιρέσεις πινάκων του HCatalog παρουσιάζουν στους χρήστες μια σχεσιακή όψη των δεδομένων στο HDFS και διασφαλίζουν ότι οι χρήστες δεν χρειάζεται να ανησυχούν για το που ή σε ποια μορφή αποθηκεύονται τα δεδομένα τους.

- Apache HBase

Το HBase είναι μία μη σχεσιακή βάση δεδομένων που τρέχει στην κορυφή του HDFS. Παρέχει ανεκτική σε σφάλματα αποθήκευση και γρήγορη πρόσβαση σε μεγάλες ποσότητες αραιών δεδομένων. Προσθέτει, επίσης, στο Hadoop συναλλακτικές δυνατότητες, επιτρέποντας στους χρήστες να διεξάγουν ενημερώσεις, εισαγωγές και διαγραφές.

- Apache Sqoop

Το Sqoop είναι ένα εργαλείο σχεδιασμένο για την αποτελεσματική μεταφορά αδόμητων δεδομένων μεταξύ Hadoop και δομημένων αποθηκών, όπως σχεσιακές βάσεις δεδομένων. Εισάγει δεδομένα από εξωτερικές πηγές, είτε απευθείας στο HDFS είτε σε συστήματα όπως το Hive και το HBase. Το Sqoop μπορεί επίσης να τραβήξει δεδομένα από το Hadoop και να τα εξάγει σε εξωτερικά μέσα αποθήκευσης, όπως σχεσιακές βάσεις δεδομένων και αποθήκες δεδομένων.

- Apache Flume

Το Flume είναι μια υπηρεσία για την αποτελεσματική συλλογή, συνάθροιση και μετακίνηση μεγάλων ποσοτήτων δεδομένων καταγραφής (log data). Επιτρέπει σε δεδομένα καταγραφής από πολλές διαφορετικές πηγές, όπως web servers, να αποθηκεύονται εύκολα σε ένα κεντρικό HDFS αποθετήριο.

- Apache Solr

Το Solr είναι ένα επεκτάσιμο εργαλείο αναζήτησης που προσφέρει ευρετηρίαση, αξιοπιστία, κεντρική διαμόρφωση, failover και ανάκτηση δεδομένων αποθηκευμένων στο HDFS. Το Solr ενεργοποιεί τις δυνατότητες αναζήτησης και πλοήγησης στις περισσότερες από τις μεγαλύτερες ιστοσελίδες του διαδικτύου, επιτρέποντας ισχυρή αναζήτηση πλήρους κειμένου και σχεδόν πραγματικού χρόνου ευρετηρίαση

- Apache Zeppelin

Το Zeppelin είναι ένα open-source, web-based εργαλείο που επιτρέπει analytics και απεικόνιση μεγάλων συνόλων δεδομένων σε κατανοητά, διαδραστικά και συνεργατικά συστήματα, όπως το Hadoop. Παρέχει απορρόφηση, εξερεύνηση, οπτικοποίηση και κοινή χρήση των δεδομένων.

Hadoop Operational Services είναι τα εργαλεία εκείνα τα οποία έχουν προκύψει για να βοηθήσουν στις λειτουργίες και τη διαχείριση ενός συμπλέγματος Hadoop, όπως:

- Apache Ambari

Το Ambari παρέχει ένα διαισθητικό σύνολο εργαλείων για την παρακολούθηση, τη διαχείριση και τον αποτελεσματικό εφοδιασμό ενός συμπλέγματος Hadoop. Απλοποιεί τη λειτουργία και κρύβει την πολυπλοκότητα του Hadoop, κάνοντάς το να δουλεύει ως μια ενιαία, συνεκτική πλατφόρμα δεδομένων.

- Apache Oozie

Το Oozie είναι μία web εφαρμογή σε Java, που χρησιμοποιείται για το χρονοπρογραμματισμό των εργασιών του Hadoop. Συνδυάζει πολλαπλές εργασίες διαδοχικά σε μία λογική μονάδα εργασίας και υποστηρίζει Hadoop εργασίες για το MapReduce, Pig, Hive και Sqoop.

- Apache ZooKeeper

Το ZooKeeper παρέχει λειτουργικές υπηρεσίες για ένα σύμπλεγμα Hadoop, συμπεριλαμβανομένης μίας κατανεμημένης υπηρεσίας διαμόρφωσης, μίας υπηρεσίας συγχρονισμού και ενός μητρώου ονομάτων. Αυτές οι υπηρεσίες επιτρέπουν στις κατανεμημένες διαδικασίες του Hadoop να συντονιστούν πιο εύκολα μεταξύ τους.



Εικόνα 22 : Οικοσύστημα Hadoop

Κεφάλαιο 5 : Εξόρυξη Δεδομένων (Data Mining)

5.1 Εισαγωγή

Εξόρυξη δεδομένων (Data Mining) ή ανακάλυψη γνώσης από βάσεις δεδομένων είναι η εξεύρεση – ενδιαφέρουσας, αυτονόητης, μη προφανούς και πιθανόν χρήσιμης – πληροφορίας ή προτύπων από μεγάλες βάσεις δεδομένων με χρήση αλγορίθμων ομαδοποίησης ή κατηγοριοποίησης και των αρχών της στατιστικής, της τεχνητής νοημοσύνης, της μηχανικής μάθησης και των συστημάτων βάσεων δεδομένων. Στόχος της εξόρυξης δεδομένων είναι η πληροφορία που θα εξαχθεί και τα πρότυπα που θα προκύψουν να έχουν δομή κατανοητή προς τον άνθρωπο έτσι ώστε να τον βοηθήσουν να πάρει τις κατάλληλες αποφάσεις.

5.2 Ορισμός

Ο όρος εξόρυξη δεδομένων είναι μία έννοια που συνήθως παραπέμπει σε κάθε είδος φόρμας με μεγάλη ποσότητα δεδομένων ή επεξεργασία δεδομένων (συλλογή, εξαγωγή δεδομένων, warehouse, ανάλυση δεδομένων και στατιστικής) αλλά επίσης γενικεύεται σε κάθε είδος συστήματος υποστήριξης αποφάσεων συμπεριλαμβανομένου της τεχνητής νοημοσύνης, της εκμάθησης μηχανής και της επιχειρηματικής ευφυΐας. Στην ορθή χρήση του όρου η λέξη κλειδί είναι η ανακάλυψη, που ορίζεται ως η ανίχνευση κάτι καινούριου.

5.3 Στόχος

Ο πραγματικός στόχος της εξόρυξης δεδομένων είναι η αυτόματη ή ημιαυτόματη ανάλυση μεγάλων ποσοτήτων δεδομένα για την εξαγωγή κάποιου ενδιαφέροντος προτύπου που ήταν άγνωστο μέχρι εκείνη τη στιγμή, όπως ομάδες από εγγραφές δεδομένων (συσταδοποίηση), ασυνήθιστες εγγραφές (*anomaly detection*) και

εξαρτήσεις (κανόνες συσχετίσεων). Αυτό συνήθως συμπεριλαμβάνει τη χρήση βάσης δεδομένων όπως χωρικά ευρετήρια. Αυτά τα πρότυπα ύστερα μπορούν να θεωρηθούν ως μία περιγραφή των δεδομένων εισαγωγής και να χρησιμοποιηθούν για περαιτέρω ανάλυση ή για παράδειγμα στην εκμάθηση μηχανής και στην προγνωστική ανάλυση. Για παράδειγμα, η εξόρυξη δεδομένων θα μπορούσε να προσδιορίσει πολλαπλά σύνολα στα δεδομένα, τα οποία μπορούν να χρησιμοποιηθούν μετά για να εξασφαλίσουν περισσότερο ακριβή αποτελέσματα από ένα σύστημα υποστήριξης αποφάσεων. Παρότι η συλλογή δεδομένων και η προετοιμασία δεδομένων, αλλά και η ερμηνεία των αποτελεσμάτων και εκθέσεων δεν αποτελούν μέρος της εξόρυξης δεδομένων, παρ' όλα αυτά ανήκουν στην ανακάλυψη γνώσης από βάσεις δεδομένων σαν κάποια επιπρόσθετα βήματα.

Άλλοι σχετικοί όροι της εξόρυξης δεδομένων είναι οι *data dredging*, *data fishing* και *data snooping*, που αναφέρονται στην χρήση μεθόδων της εξόρυξης δεδομένων για να πάρουν δείγματα από μεγαλύτερη συλλογή δεδομένων που είναι (ή μπορεί να είναι) πολύ μικρά για αξιόπιστα στατιστικά συμπεράσματα που έγιναν σχετικά με τη εγκυρότητα των προτύπων που ανακαλύφθηκαν. Αυτές οι μέθοδοι, επίσης, μπορούν να χρησιμοποιηθούν για την δημιουργία νέων υποθέσεων προς εξέταση έναντι μεγαλύτερων συλλογών δεδομένων.

5.4 Ιστορική Αναδρομή

Η χειροκίνητη εξαγωγή προτύπων από δεδομένα συμβαίνει εδώ και αιώνες. Οι πρώτες μέθοδοι για τον προσδιορισμό προτύπων ήταν αυτές της θεωρίας Bayes και της ανάλυσης της παλινδρόμησης. Ο πολλαπλασιασμός, η ευρεία διαθεσιμότητα και η εξέλιξη της τεχνολογίας των υπολογιστών έχουν αυξήσει τον όγκο των συγκεντρωμένων δεδομένων και την ζήτηση για αποδοτικούς και αποτελεσματικούς χειρισμούς. Καθώς, οι συλλογές δεδομένων αυξήθηκαν τόσο σε όγκο όσο και σε πολυπλοκότητα, η χειρωνακτική ανάλυση των δεδομένων έχει αντικατασταθεί από την αυτόματη επεξεργασία δεδομένων. Σε αυτό συνέβαλαν άλλες ανακαλύψεις της επιστήμης των υπολογιστών, όπως τα νευρωνικά δίκτυα, η συσταδοποίηση, οι γενετικοί αλγόριθμοι, τα δέντρα απόφασης και η μηχανή υποστήριξης διανυσμάτων. Η εξόρυξη δεδομένων είναι η διαδικασία εφαρμογής αυτών των μεθόδων στα δεδομένα

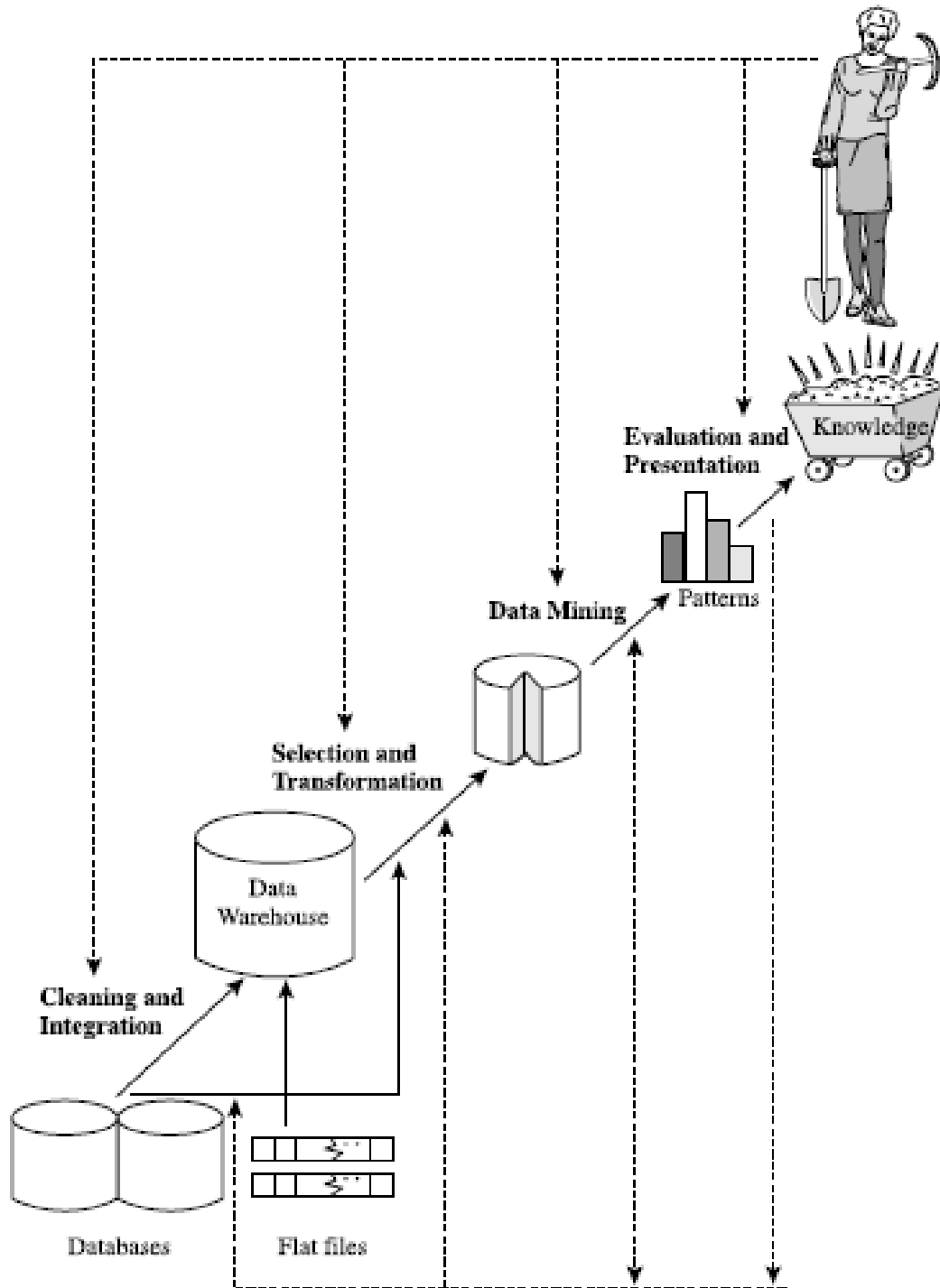
με σκοπό την αποκάλυψη άγνωστων προτύπων σε μεγάλα σύνολα δεδομένων. Αυτό γεφυρώνει το χάσμα της εφαρμοσμένης στατιστικής και της τεχνητής νοημοσύνης (τα οποία συνήθως παρέχουν το μαθηματικό υπόβαθρο) με την διαχείριση βάσεων δεδομένων, κάνοντας χρήση του τρόπου με τον οποίο αποθηκεύονται και κατατάσσονται στη βάση δεδομένων για να εκτελέσουν την θεωρία και τους διαθέσιμους αλγορίθμους περισσότερο αποτελεσματικά, επιτρέποντας σε τέτοιες μεθόδους να εφαρμόζονται σε μεγάλα σύνολα δεδομένων.

5.5 Ανακάλυψη γνώσης

Συχνά, η εξόρυξη δεδομένων αντιμετωπίζεται ως συνώνυμο με έναν άλλο ευρέως χρησιμοποιούμενο όρο, ανακάλυψη γνώσης από δεδομένα (Knowledge Discovery from Data, KDD) [19]. Εναλλακτικά, άλλοι βλέπουν την εξόρυξη δεδομένων ως απλά ένα ουσιαστικό βήμα στη διαδικασία της ανακάλυψης γνώσης. Η ανακάλυψη γνώσης ως μία διαδικασία απεικονίζεται στην Εικόνα 23 και αποτελείται από την επαναληπτική ακολουθία των παρακάτω βημάτων:

- *Καθαρισμός Δεδομένων*, για την απομάκρυνση του θορύβου, εσφάλμενων δεδομένων και περιττών πληροφοριών
- *Ενσωμάτωση Δεδομένων*, όπου μπορούν να συνδυαστούν πολλαπλές πηγές δεδομένων
- *Συλλογή Δεδομένων*, όπου τα δεδομένα που σχετίζονται με την διαδικασία της ανάλυσης ανακτώνται από τη βάση δεδομένων
- *Μετασχηματισμός Δεδομένων*, όπου τα δεδομένα μετασχηματίζονται ή ενοποιούνται σε μορφές κατάλληλες για εξόρυξη, εκτελώντας για παράδειγμα λειτουργίες σύνοψης ή συνάθροισης
- *Εξόρυξη Δεδομένων*, μια ουσιαστική διαδικασία, όπου ευφυείς μέθοδοι εφαρμόζονται προκειμένου να εξαχθούν μοτίβα δεδομένων
- *Αξιολόγηση Μεθόδου*, για τον εντοπισμό των πραγματικά ενδιαφέρον μεθόδων που αναπαριστούν τη γνώση

- Παρουσίαση της γνώσης, όπου οι τεχνικές απεικόνισης και αναπαράστασης της γνώσης χρησιμοποιούνται για να παρουσιάσουν την εξορυσσόμενη γνώση στο χρήστη



Εικόνα 23 : Εξόρυξη Δεδομένων, ως μέρος της διαδικασίας της Ανακάλυψης Γνώσης

Τα βήματα 1 έως 4 είναι διαφορετικές μορφές προεπεξεργασίας δεδομένων, κατά τα οποία τα δεδομένα προετοιμάζονται για εξόρυξη. Το βήμα εξόρυξη δεδομένων μπορεί να αλληλεπιδράσει με το χρήστη ή μία βάση γνώσεων. Τα ενδιαφέροντα μοτίβα παρουσιάζονται στο χρήστη και μπορούν να αποθηκευτούν ως νέα γνώση στη βάση γνώσεων. Η εξόρυξη δεδομένων είναι μόνο ένα βήμα στην συνολική διαδικασία, ωστόσο πολύ σημαντικό, καθώς αποκαλύπτει κρυφά πρότυπα για αξιολόγηση.

5.5.1 Προ-επεξεργασία

Πριν την εφαρμογή των αλγορίθμων εξόρυξης δεδομένων, το ερευνώμενο σύνολο δεδομένων πρέπει να συναρμολογηθεί. Καθώς η εξόρυξη δεδομένων μπορεί να αποκαλύψει μόνο τα πρότυπα που πράγματι εμφανίζονται στα δεδομένα, το σύνολο δεδομένων που ερευνάται πρέπει να είναι αρκετά μεγάλο για να περιέχει αυτά τα πρότυπα, παραμένοντας ωστόσο αρκετά συνοπτικό ώστε να εξορυχθεί σε ένα αποδεκτό χρονικό διάστημα. Η προεπεξεργασία είναι απαραίτητη για την ανάλυση πολυπαραγοντικών συνόλων δεδομένων πριν την εξόρυξη δεδομένων.

Έτσι το ερευνώμενο σύνολο καθαρίζεται. Το καθάρισμα δεδομένων διαγράφει τις παρατηρήσεις που περιέχουν θόρυβο και αυτές με ελλιπή ή ελλείποντα δεδομένα.

5.5.2 Τεχνικές

Η εξόρυξη δεδομένων περιλαμβάνει κάποιες από τις ακόλουθες τάξεις διαδικασιών:

- Ανίχνευση ανωμαλιών (Anomaly detection) : Ο προσδιορισμός ασυνήθιστων εγγραφών δεδομένων, που μπορεί να παρουσιάζουν κάποιο ενδιαφέρον ή λάθη στα δεδομένα που απαιτούν περαιτέρω έρευνα.
- Κανόνες συσχέτισης (Μοντέλο αλληλεξάρτησης) : Αναζητήσεις για σχέσεις μεταξύ των μεταβλητών. Για παράδειγμα, μία εταιρεία μπορεί να συλλέξει δεδομένα που αφορούν τις αγοραστικές συνήθειες των καταναλωτών της και χρησιμοποιώντας κανόνες συσχέτισης, μπορεί να υπολογίσει ποια προϊόντα αγοράζονται συνήθως μαζί και να χρησιμοποιήσει αυτή την πληροφορία για αγοραστικούς σκοπούς.
- Συσταδοποίηση (Clustering) : Η διαδικασία ανακάλυψης ομάδων και δομών στα δεδομένα που είναι "παρόμοια" κατά κάποιο τρόπο, χωρίς να χρησιμοποιούνται γνωστές δομές στα δεδομένα.

- Κατηγοριοποίηση (Classification) : Η διαδικασία γενίκευσης γνωστών δομών για την εφαρμογή τους πάνω σε νέα δεδομένα.
- Παλινδρόμηση (στατιστική) : Εύρεση συναρτήσεων που μοντελοποιούν τα δεδομένα με το λιγότερο λάθος.
- Περιληπτική : Μία πιο συμπαγής αναπαράσταση του συνόλου δεδομένων, συμπεριλαμβανομένης της απεικόνισης και τη δημιουργία αναφορών.

5.5.3 Επικύρωση Αποτελέσματος

Το τελικό βήμα της ανακάλυψης γνώσης από δεδομένα είναι η επικύρωση των προτύπων που εξήχθησαν από τους αλγορίθμους της εξόρυξης δεδομένων που απευθύνονται σε ευρύτερο σύνολο δεδομένων. Δεν είναι απαραίτητα όλα τα πρότυπα, που βρέθηκαν, έγκυρα. Είναι συνηθισμένο για τους αλγορίθμους της εξόρυξης δεδομένων να βρίσκουν πρότυπα κατά τη διαδικασία εκπαίδευσης, τα οποία δεν υπάρχουν στο γενικό σύνολο δεδομένων. Αυτό καλείται υπερφόρτωση (overfitting). Για να ξεπεραστεί αυτό, στην εκτίμηση χρησιμοποιείται ένα δοκιμαστικό σύνολο δεδομένων στο οποίο δεν έχουν εφαρμοστεί οι αλγόριθμοι της εξόρυξης δεδομένων. Τα πρότυπα, που έχουν προκύψει, εφαρμόζονται σε αυτό το δοκιμαστικό σύνολο και το προκύπτον αποτέλεσμα συγκρίνεται με το επιθυμητό. Για παράδειγμα, ένας αλγόριθμος εξόρυξης δεδομένων που ξεχωρίζει τα “ανεπιθύμητα” μηνύματα από τα “επιθυμητά” θα εφαρμοζόταν κατά την εκπαίδευση σε ένα δείγμα ηλεκτρονικών μηνυμάτων. Στη συνέχεια, τα εξαχθέντα πρότυπα θα εφαρμόζονταν στο δοκιμαστικό σύνολο μηνυμάτων στο οποίο δεν είχαν εφαρμοστεί πριν. Η ακρίβεια αυτών των προτύπων μπορεί τώρα να μετρηθεί από τα πόσα μηνύματα έχουν καταταχθεί-ταξινομηθεί σωστά. Διάφορες στατιστικές μέθοδοι μπορούν να χρησιμοποιηθούν για την αξιολόγηση του αλγορίθμου, όπως το ROC curves.

Αν τα πρότυπα δεν ανταποκρίνονται στα επιθυμητά κριτήρια, τότε είναι απαραίτητο να επανεκτιμηθεί και να διαφοροποιηθούν τα βήματα της προεπεξεργασίας και της εξόρυξης δεδομένων. Στην αντίθετη περίπτωση, το τελικό στάδιο είναι να ερμηνευτούν τα εξαχθέντα πρότυπα και να μετατραπούν σε γνώση.

5.6 Εφαρμογές

5.6.1 Ιατρική

Τα τελευταία χρόνια, η εξόρυξη δεδομένων χρησιμοποιείται ευρέως στους τομείς της ιατρικής, όπως η βιοϊατρική, το DNA, η γενετική και η φαρμακευτική. Στον τομέα της γενετικής, ο σκοπός είναι να κατανοήσουμε την χαρτογράφηση της σχέσης μεταξύ της μεταβολής των ακολουθιών του ανθρώπινου DNA και την προδιάθεση στην αρρώστια. Η εξόρυξη δεδομένων είναι ένα σημαντικό εργαλείο που μπορεί να βοηθήσει στην βελτίωση της διάγνωσης, της πρόληψης και της θεραπείας των ασθενειών.

- Εξαιτίας της αύξησης των βιοϊατρικών ερευνών, μία μεγάλη κλίμακα γονιδιακών προτύπων και λειτουργιών πρέπει να εξετασθεί. Τα εργαλεία της εξόρυξης δεδομένων μπορούν να βοηθήσουν σε μεγάλο βαθμό στη μελέτη της σύστασης του DNA και να βρεθούν ποικίλα πρότυπα και λειτουργίες αυτού.
- Ένας από τους κύριους στόχους που σχετίζεται με την ανάλυση δεδομένων του DNA είναι η σύγκριση ποικίλων ακολουθιών και η αναζήτηση ομοιοτήτων μεταξύ των δεδομένων του DNA. Αυτή η ανάλυση βοηθάει στο να βρεθούν ομοιότητες και διαφορές στις γενετικές ακολουθίες.
- Στην βιοϊατρική, ερευνάται αν οι περισσότερες ασθένειες προκαλούνται από ένα συνδυασμό των γονιδίων. Η μέθοδος της συσχέτισης χρησιμοποιείται για να καθορίσει την συνύπαρξη ομάδων των γονιδίων και να εξεταστεί η αλληλεπίδραση και η σχέση μεταξύ των γονιδίων.
- Τα εργαλεία της οπτικοποίησης παίζουν επίσης ένα σημαντικό ρόλο στην εξόρυξη δεδομένων στην βιοϊατρική. Τα εργαλεία αυτά μπορούν να παρουσιάσουν πολύπλοκες δομές γονιδίων σε γράφους, δένδρα και αλυσίδες. Η οπτική παρουσίαση βοηθάει στην καλύτερη κατανόηση αυτών των δομών για ανακάλυψη γνώσης και εξερεύνηση των δεδομένων.
- Υπάρχουν διάφοροι συνδυασμοί γονιδίων που συμβάλλουν στις ασθένειες, αλλά αυτά τα γονίδια ενεργοποιούνται σε διαφορετικά επίπεδα. Η ανάλυση μονοπατιού (path analysis) χρησιμοποιείται για να συνδέει διαφορετικά γονίδια με διαφορετικά στάδια κατά την εξέλιξη της ασθένειας.

5.6.2 Οικονομία

Άλλος τομέας που εφαρμόζεται η εξόρυξη δεδομένων είναι η οικονομία. Τα οικονομικά δεδομένα κυρίως συλλέγονται από τράπεζες και από άλλους οικονομικούς οργανισμούς. Τα δεδομένα αυτά συνήθως είναι αξιόπιστα, ολοκληρωμένα, έχουν υψηλή ποιότητα και απαιτούν συστηματική μέθοδο για την ανάλυση τους. Η συνεισφορά της εξόρυξης δεδομένων στην επιστήμη της οικονομίας συναντάται στην συλλογή και κατανόηση των δεδομένων, στην βελτίωση των δεδομένων (data refinement), στην δημιουργία και εκτίμηση ενός μοντέλου και στην ανάπτυξη αυτού. Η σωστή ανάλυση των οικονομικών δεδομένων διευκολύνει τη λήψη καλύτερων αποφάσεων, σύμφωνα με την ανάλυση της αγοράς. Τα εργαλεία και οι τεχνικές της εξόρυξης δεδομένων βοηθούν στην ανάλυση των οικονομικών δεδομένων με τους παρακάτω τρόπους :

- Τα δεδομένα που συλλέγονται από διάφορα οικονομικά ιδρυτήματα, όπως οι τράπεζες, συγκεντρώνονται αρχικά στην αποθήκη δεδομένων (data warehouse) και αναλύονται χρησιμοποιώντας τεχνικές της πολυδιάστατης ανάλυσης δεδομένων.
- Μέθοδοι της εξόρυξης δεδομένων, όπως η επιλογή χαρακτηριστικών (feature selection), βοηθούν στην ταυτοποίηση ποικίλων χαρακτηριστικών. Οι τεχνικές της συσταδοποίησης και της ταξινόμησης βοηθούν τα οικονομικά ιδρυτήματα να ομαδοποιούν διάφορους πελάτες που έχουν κοινά χαρακτηριστικά. Η αποτελεσματική συσταδοποίηση και οι μέθοδοι φιλτραρίσματος βοηθούν τις τράπεζες να ταυτοποιούν μία ομάδα πελατών, να συσχετίζουν ένα νέο πελάτη με την παρούσα ομάδα και να παρέχουν στους πελάτες μίας ομάδας κοινά οφέλη.
- Τα εργαλεία της εξόρυξης δεδομένων βοηθούν τα οικονομικά ιδρυτήματα να αναγνωρίζουν τις απάτες από παραποιημένα δεδομένα, από τις διάφορες βάσεις δεδομένων και από το ιστορικό συναλλαγών που έγιναν από τους πελάτες. Οι τεχνικές οπτικοποίησης βοηθούν στην παρουσίαση δεδομένων με διαφορετικές μορφές, όπως γράφοι, που βασίζονται σε συγκεκριμένα γνωρίσματα.

5.6.3 Τηλεπικοινωνία

Η τηλεπικοινωνιακή βιομηχανία αναπτύσσεται πολύ γρήγορα όπως και η τεχνολογία. Εξαιτίας των εξελίξεων στις τηλεπικοινωνιακές τεχνολογίες οι τεχνικές της εξόρυξης δεδομένων ενσωματώνονται σε αυτές τις τεχνολογίες για να παράγουν αποδοτικά αποτελέσματα. Η εξόρυξη δεδομένων βοηθάει στην διάκριση τηλεπικοινωνιακών προτύπων, στην καταπολέμηση παράνομων δραστηριοτήτων και επίσης στην καλύτερη χρήση των πόρων και στη βελτίωση της ποιότητας των υπηρεσιών. Η εξόρυξη δεδομένων βελτιώνει τις τηλεπικοινωνιακές υπηρεσίες με τους εξής τρόπους:

- Τα τηλεπικοινωνιακά δεδομένα που συλλέγονται, περιλαμβάνουν τον τύπο κλήσης, την τοποθεσία του καλούντος και του κληθέντος, τον χρόνο κλήσης, την διάρκεια κλήσης κλπ. Η πολυδιάστατη ανάλυση βοηθά στον προσδιορισμό και στη σύγκριση του φορτιού του συστήματος, της κίνησης δεδομένων, του κέρδους κλπ.. Η ανάλυση μπορεί να δείξει διαγράμματα και γράφους των πόρων του συστήματος κάνοντας χρήση των εργαλείων απεικόνισης της εξόρυξης δεδομένων. Τέτοια εργαλεία, όπως η συσχετισμένη, η οπτικοποίηση και η συσταδοποίηση παρέχουν χρήσιμες υπηρεσίες στην ανάλυση των δεδομένων τηλεπικοινωνίας.
- Το κύριο πρόβλημα που παρουσιάστηκε στη βιομηχανία τηλεπικοινωνιών είναι οι παράνομες δραστηριότητες. Αυτές οι δραστηριότητες μπορεί να έχουν να κάνουν με σκόπιμες κλήσεις κατά την ώρα αιχμής, περιοδικές κλήσεις κ.α., με αποτέλεσμα να επιδρούν αρνητικά στην επίδοση του δικτύου. Μέθοδοι, όπως η συσταδοποίηση και η ανάλυση ακραίων τιμών, συνεισφέρουν στην ανίχνευση παράνομων προτύπων, βελτιώνοντας την αποτελεσματικότητα των υπηρεσιών τηλεπικοινωνίας.
- Εκμεταλλευόμενοι τα εργαλεία της εξόρυξης δεδομένων είναι δυνατή η δημιουργία προφίλ των πελατών και ο εντοπισμός βλαβών στο δίκτυο.
- Η ανάλυση συσχετιζόμενων και ακολουθιακών προτύπων ενθαρρύνει την προώθηση νέων και ποικίλων υπηρεσιών τηλεπικοινωνίας.

5.7 Σημασία

Οι εκτεταμένες αλλαγές στην υιοθέτηση και χρησιμοποίηση των νέων τεχνολογιών στις μεγάλες αλλά και στις μικρές επιχειρήσεις, έχει ως αποτέλεσμα την συγκέντρωση μεγάλου αριθμού δεδομένων από τις οικονομικές συναλλαγές. Είναι ευθύνη του αναλυτή να αναλύσει αυτές τις συναλλαγές και να εντοπίσει τις απάτες και τα λάθη μέσα σε αυτές. Λόγω των αλλαγών των τάσεων μέσα στις επιχειρήσεις, είναι δύσκολο να επεξεργαστούν και να αναλυθούν τα δεδομένα με παλαιές μεθόδους. Οι περιορισμοί που εμφανίζουν αυτές οι μέθοδοι έχουν οδηγήσει στην υιοθέτηση των εργαλείων της εξόρυξης για καλύτερα και περισσότερο αξιόπιστα αποτελέσματα.

Η εξόρυξη δεδομένων έχει προσελκύσει μεγάλη προσοχή στη βιομηχανία πληροφοριών και στο σύνολο της κοινωνίας τα τελευταία χρόνια, λόγω της ευρείας διαθεσιμότητας τεράστιων ποσοτήτων δεδομένων και της επικείμενης ανάγκης για μετατροπή αυτών των δεδομένων σε χρήσιμες πληροφορίες και γνώσεις. Οι εξαχθείσες πληροφορίες και γνώσεις μπορούν να χρησιμοποιηθούν για εφαρμογές που κυμαίνονται από την ανάλυση της αγοράς, την ανίχνευση απάτης και τη διατήρηση των πελατών, μέχρι τον έλεγχο της παραγωγής και την εξερεύνηση της επιστήμης.

Κεφάλαιο 6 : Υλοποίηση Συστήματος Ανάλυσης Δεδομένων

6.1 Εισαγωγή – Παρουσίαση του συστήματος

Σκοπός της παρούσας διπλωματικής είναι η εξαγωγή πραγματικού χρόνου δεδομένων από το twitter – tweets που αναφέρονται στη λέξη “refugees” και είναι γραμμένα στην αγγλική γλώσσα – μέσω του Apache NiFi και η διοχέτευσή τους στον αποθηκευτικό χώρο του Hadoop, το HDFS. Με την βοήθεια της Hive και ενός αλγορίθμου κατηγοριοποίησης χτίστηκε ένα μοντέλο δεδομένων (μέσω πινάκων), το οποίο παρέχει πληροφορία ανά χώρα για το αν οι χρήστες εμφανίζουν θετικά, αρνητικά ή ουδέτερα συναισθήματα όσον αφορά το μεταναστευτικό. Τα αποτελέσματα του μοντέλου αναπαραστάθηκαν σε χάρτη (google maps), με το συναίσθημα που χαρακτηρίζει συνολικά κάθε χώρα να εμφανίζεται με διαφορετικό χρώμα. Η γενική αναπαράσταση του συστήματος φαίνεται παρακάτω (Εικόνα 24).

Στην παρούσα διπλωματική εργασία γίνεται χρήση του Hortonworks Sandbox. Το Hortonworks Sandbox είναι ένα προσωπικό, φορητό περιβάλλον Hortonworks Data Platform (HDP) για το Apache Hadoop, το οποίο περιλαμβάνει δεκάδες εφαρμογές αλληλεπίδρασης με το Hadoop. Συγκεκριμένα, στο σύμπλεγμα του συστήματος ο κόμβος που δημιουργήθηκε αποτελείται από ένα εικονικό μηχάνημα με τα παρακάτω χαρακτηριστικά :

Processors	4
RAM	8 GB
Hard Disk	48 GB
CPU speed	2600 MHz

Πίνακας 2 : Χαρακτηριστικά Κόμβου Συστήματος

Παράλληλα με το sandbox, έγινε εγκατάσταση του Hortonworks Data Flow (HDF), το οποίο χρησιμοποιεί την τεχνολογία Apache NiFi για να διοχετεύει πραγματικού χρόνου δεδομένα στο HDFS.

6.2 Αρχιτεκτονική

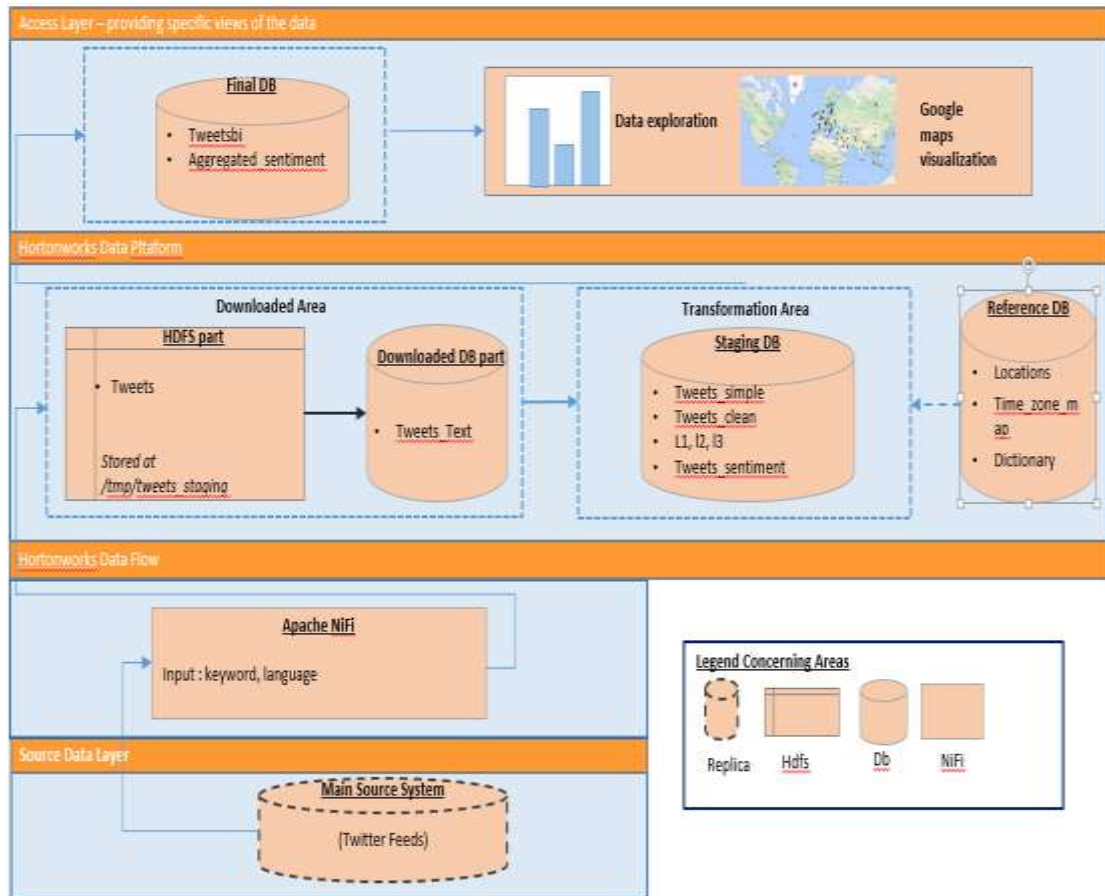
6.2.1 Γενική αρχιτεκτονική του συστήματος

Το σύστημα δέχεται ένα keyword (λέξη κλειδί) ως είσοδο (input) και μετά από κάποιες διαδικασίες για την επεξεργασία των δεδομένων παρουσιάζει κάποια γραφήματα ως έξοδο (output).



Εικόνα 24 : Γραφική Παρουσίαση του Συστήματος Εξόρυξης Δεδομένων

6.2.2 Αρχιτεκτονική φυσικού επιπέδου



Εικόνα 25 : Αρχιτεκτονική Συστήματος Εξόρυξης Δεδομένων

Στην παραπάνω εικόνα απεικονίζονται όλα τα βήματα της διαδικασίας που ακολουθείται για την εξαγωγή γνώσης, από το φιλτράρισμα της πληροφορίας μέχρι το τελικό αποτέλεσμα που είναι η αναπαράσταση των δεδομένων σε χάρτη.

- Στο κατώτερο επίπεδο βρίσκεται η κύρια πηγή πληροφορίας, η οποία είναι το Twitter (Twitter Feeds).
- Στο αμέσως επόμενο επίπεδο βρίσκεται το Apache NiFi, το οποίο είναι υπεύθυνο για τη δρομολόγηση των ροών δεδομένων από το Twitter στον αποθηκευτικό χώρο του Hadoop, όπου και θα γίνει η ανάλυση των raw data (αδόμητων δεδομένων).
- Στο τρίτο επίπεδο, τα δεδομένα (tweets) μεταφέρονται από τη θέση : «/tmp/tweets_staging» όπου βρίσκονται αποθηκευμένα, στον πίνακα Tweets_Text, ο οποίος “ζει” στην downloaded βάση δεδομένων. Με την

βοήθεια των πινάκων *Time_zone_map* και *Dictionary*, οι οποίοι είναι αποθηκευμένοι στην *reference* βάση, και έναν αλγόριθμο κατηγοριοποίησης (*classification*) τα αρχικά δεδομένα μετασχηματίζονται κατάλληλα σύμφωνα με το μοντέλο που δομείται. Οι ενδιαμέσοι πίνακες που απαρτίζουν το μοντέλο ανάλυσης βρίσκονται στην *Staging* βάση και είναι οι εξής : *Tweets_simple*, *Tweets_clean*, *L1*, *L2*, *L3* και *Tweets_sentiment*.

- Στο τελευταίο επίπεδο βρίσκονται οι δύο τελικοί πίνακες, *Tweetsbi* και *aggregated_sentiment*, οι οποίοι περιλαμβάνουν την πληροφορία για το συναίσθημα και την τοποθεσία. Οι πίνακες αυτοί χρησιμοποιήθηκαν για την παρουσίαση της ανάλυσης σε διαγράμματα και απεικόνισής της σε χάρτη.

6.2.3 Δομές Δεδομένων

Όλες οι παραπάνω βάσεις που απεικονίζονται στο σχήμα δημιουργήθηκαν μέσω της γλώσσας *Hive*, προκειμένου να είναι αναγνωρίσιμες από το *Hadoop*. Οι βάσεις του συστήματος, που σχεδιάστηκε, είναι οι εξής :

➤ *Downloaded Database*, η οποία περιλαμβάνει τον πίνακα :

- *Tweets_text*, με στήλες :
 - *Tweet_id*, ένας μοναδικός αριθμός που αποδίδεται σε κάθε tweet
 - *Created_unixtime*, η ώρα δημιουργίας του tweet σε μορφή που καταλαβαίνει το *unix* σύστημα
 - *Created_time*, η ημερομηνία και η ώρα δημιουργίας του tweet σε μορφή που αντιλαμβάνεται ο άνθρωπος
 - *Language*, η γλώσσα στην οποία γράφτηκε το tweet
 - *Displayname*, το όνομα χρήστη του *Twitter* που δημοσίευσε το tweet
 - *Time_zone*, η ζώνη ώρας στην οποία ανήκει η θέση από την οποία δημοσιεύθηκε το tweet (διότι καταγράφονται tweets από διάφορα σημεία του κόσμου)
 - *Message*, το περιεχόμενο (*text*) του tweet
 - *Fulltext*, περιλαμβάνει όλες τις παραπάνω πληροφορίες σε αδόμητη μορφή

- Staging Database, η οποία περιλαμβάνει 3 πίνακες και 3 views :
- Tweets_simple, με στήλες :
 - *Tweet_id*, ο μοναδικός αριθμός που αποδίδεται σε κάθε tweet
 - *Timestamp*, η ημερομηνία δημιουργίας του tweet σε μορφή: yyyy MMM dd hh:mm:ss
 - *Message*, το περιεχόμενο του tweet
 - *Time_zone*, η ζώνη ώρας στην οποία ανήκει το tweet
 - Tweets_clean, με στήλες :
 - *Tweet_id*, ο μοναδικός αριθμός που αποδίδεται σε κάθε tweet
 - *Timestamp*, η ημερομηνία δημιουργίας του tweet σε μορφή: yyyy MMM dd hh:mm:ss
 - *Message*, το περιεχόμενο του tweet
 - *Country*, η χώρα από την οποία δημοσιεύθηκε το tweet
 - L1, (view) με στήλες :
 - *Tweet_id*, ο μοναδικός αριθμός που αποδίδεται σε κάθε tweet
 - *Words*, όλες οι λέξεις ενός Message χωρισμένες με κόμμα
 - L2, (view) με στήλες :
 - *Tweet_id*, ο μοναδικός αριθμός που αποδίδεται σε κάθε tweet
 - *Word*, κάθε λέξη ενός Message, ανά γραμμή
 - L3, (view) με στήλες :
 - *Tweet_id*, ο μοναδικός αριθμός που αποδίδεται σε κάθε tweet
 - *Word*, κάθε λέξη ενός Message, ανά γραμμή
 - *Polarity*, το συναίσθημα (πολικότητα) κάθε λέξης εκφρασμένο αριθμητικά, (1 = θετικό, 0 = ουδέτερο, -1 = αρνητικό)
 - Tweets_sentiment, με στήλες :
 - *Tweet_id*, ο μοναδικός αριθμός που αποδίδεται σε κάθε tweet
 - *Sentiment*, το συνολικό συναίσθημα που χαρακτηρίζει ολόκληρο το tweet

- Reference Database, η οποία περιλαμβάνει τους παρακάτω πίνακες :
 - Locations, με στήλες :
 - *Country*, περιλαμβάνει όλες τις χώρες
 - *Latitude* το γεωγραφικό πλάτος κάθε χώρας
 - *Longitude* το γεωγραφικό μήκος κάθε χώρας
 - Time_zone_map, με στήλες :
 - *Time_zone*, η ζώνη ώρας κάθε χώρας
 - *Country*, η αντίστοιχη χώρα
 - Dictionary, με στήλες :
 - *Word*, μία λίστα από λέξεις
 - *Polarity*, πολικότητα κάθε λέξης

- Final Database, η οποία περιλαμβάνει τους τελικούς πίνακες :
 - *Tweetsbi*, με στήλες :
 - *Tweet_id*, ο μοναδικός αριθμός που αποδίδεται σε κάθε tweet
 - *Timestamp*, η ημερομηνία δημιουργίας του tweet σε μορφή: yyyy MMM dd hh:mm:ss
 - *Message*, το περιεχόμενο του tweet
 - *Country*, η χώρα από την οποία δημοσιεύθηκε το tweet
 - *Sentiment*, το συνολικό συναίσθημα που χαρακτηρίζει ολόκληρο το tweet
 - *Aggregated_sentiment*, με στήλες :
 - *Country*, η χώρα από την οποία δημοσιεύθηκε το tweet
 - *Latitude*, το γεωγραφικό πλάτος της χώρας
 - *Longitude*, το γεωγραφικό μήκος της χώρας
 - *Sentiment*, το συνολικό συναίσθημα που χαρακτηρίζει ολόκληρη τη χώρα

6.3 Περιγραφή δομικών συστατικών

Αρχικά, για να είναι εφικτή η εξαγωγή πραγματικού χρόνου δεδομένων από το Twitter, έγινε χρήση του API του. Το API, όπως αναφέρθηκε στην [παράγραφο 2.4](#), είναι η διεπαφή των προγραμματιστικών διαδικασιών, η οποία παρέχει ένα λειτουργικό σύστημα, βιβλιοθήκη ή εφαρμογή προκειμένου να επιτρέπει να γίνονται προς αυτά αιτήσεις από άλλα προγράμματα ή και ανταλλαγή δεδομένων. Για το λόγο αυτό δημιουργήθηκε ένα twitter application, όπου παρήχθησαν κάποια διαπιστευτήρια (credentials), δηλαδή Consumer Key, Consumer Secret, Access Token και Access Token Secret, όπως φαίνεται στις παρακάτω εικόνες.

Application Details

Name *

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 30 characters max.

Description *

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website *

Your application's publicly accessible home page, where users can go to download, meet use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens.
(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL

Where should we return after successfully authenticating? [OAuth 1.0a](#) applications should explicitly specify their `oauth_callback` URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

Developer Rules of the Road

Last Update: September 16, 2014.

Twitter maintains an open platform that supports the millions of people around the world who are sharing and discovering what's happening now. We want to empower our ecosystem partners to build valuable businesses around the information flowing through Twitter. At the same time, we aim to strike a balance between encouraging interesting development and protecting both Twitter's and users' rights.

So, we've come up with a set of Developer Rules of the Road ("Rules") that describes the policies and philosophy around what type of innovation is permitted with the content and information shared on Twitter.

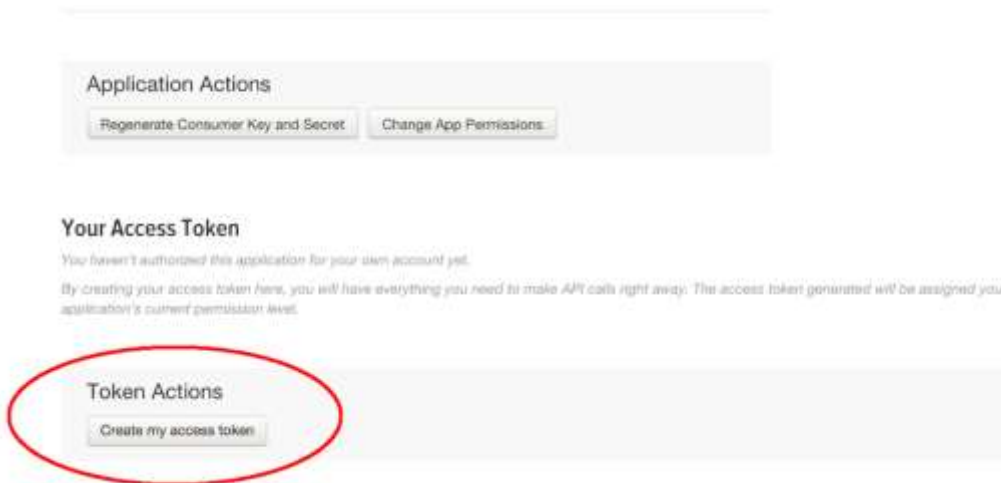
The Rules will evolve along with our ecosystem as developers continue to innovate and find new, creative ways to use the Twitter API, so please check back periodically to see the current version. Don't do anything prohibited by the Rules and talk to us if you think we should make a change or give you an exception.

If your application will eventually need more than 1 million user tokens, or you expect your [embedded Tweets](#) and [embedded timelines](#) to exceed 10 million daily impressions, you will need to talk to us directly about your access to the Twitter API as you may be subject to additional terms. Furthermore, applications that attempt to replicate Twitter's core user experience (as described in Section 1.5 below) will need our permission to have more than 100,000 user tokens and are subject to additional terms.

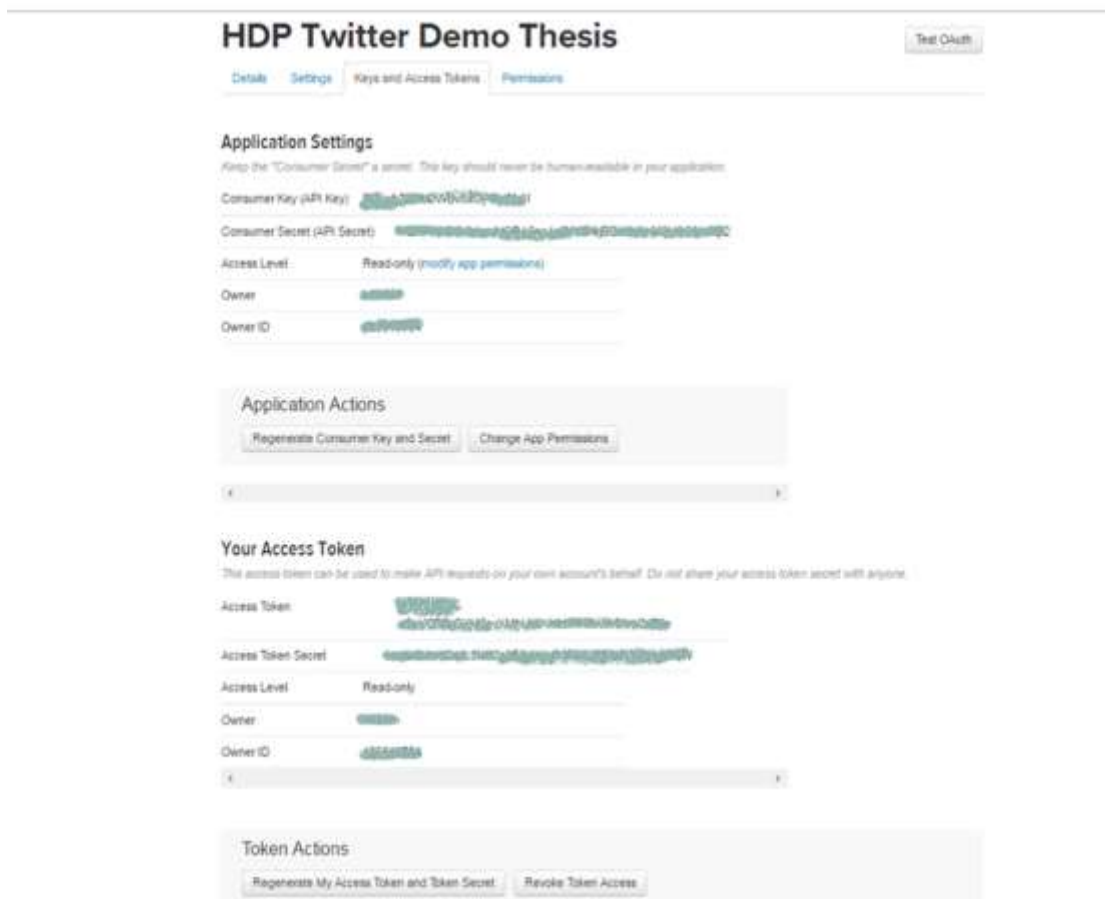
Yes, I agree

Create your Twitter application

Εικόνα 26 : Δημιουργία Twitter Application



Εικόνα 28 : Δημιουργία Access Token

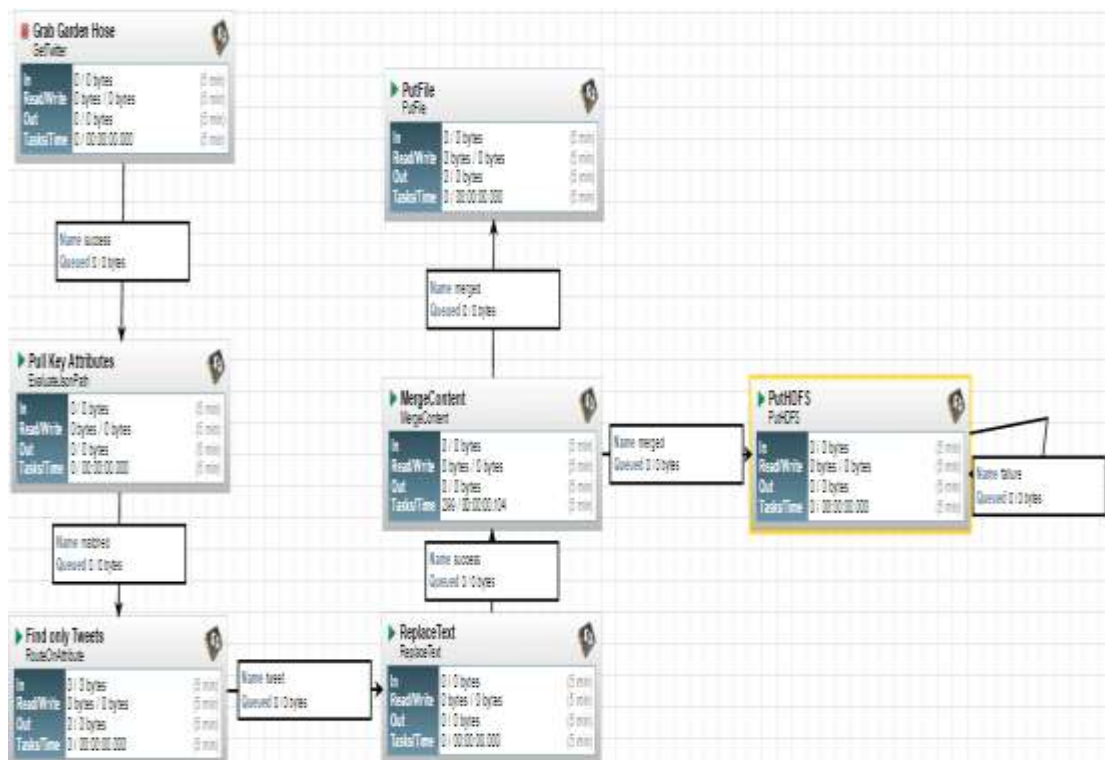


Εικόνα 27 : Twitter Application

6.3.1 Εξαγωγή Δεδομένων (Data Extraction)

Αρχικά, έγινε εγκατάσταση του NiFi, στο οποίο δημιουργήθηκε μία ροή δεδομένων από επεξεργαστές, όπως αναφέρθηκε στην [παράγραφο 3.2.6](#), ώστε να πραγματοποιηθεί η εξαγωγή των δεδομένων. Το NiFi δέχεται σαν είσοδο τα tweets από το Twitter και τα διοχετεύει στο HDFS. Για την υλοποίηση της παρούσας διπλωματικής συγκεντρώθηκαν περίπου 5.000 tweets, τα οποία περιλαμβάνουν στο περιεχόμενό τους το hashtag “#refugees” και είναι γραμμένα στην αγγλική γλώσσα, με σκοπό την ανάλυση των συναισθημάτων (sentiment analysis) των χρηστών όσον αφορά το προσφυγικό θέμα.

Στο παρακάτω διάγραμμα (Εικόνα 29) απεικονίζεται το γραφικό περιβάλλον του Apache NiFi το οποίο αποτελείται από 7 επεξεργαστές (processors), οι οποίοι χρησιμοποιήθηκαν για τη διοχέτευση των δεδομένων από το Twitter στο HDFS. Σε κάθε επεξεργαστή επιλέχθηκαν συγκεκριμένα πεδία ώστε να καθοριστεί η χρήση και η λειτουργία τους, όπως θα φανεί παρακάτω.



Εικόνα 29 : Γραφικό Περιβάλλον NiFi

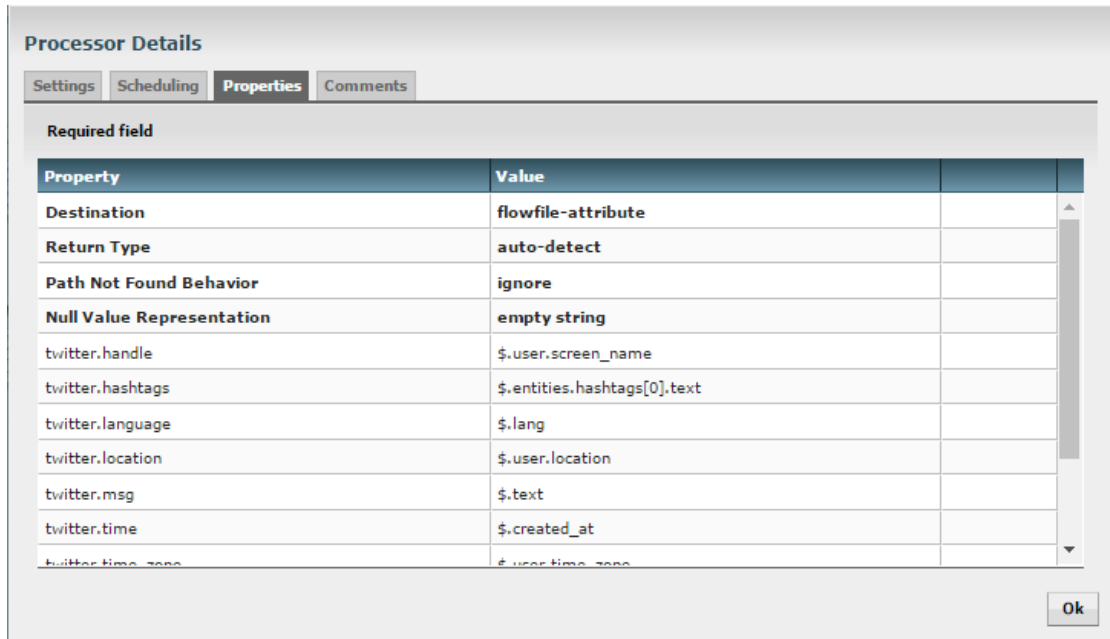
Ο πρώτος επεξεργαστής ονομάζεται GetTwitter και ελέγχει τις αλλαγές κατάστασης των λογαριασμών από το Streaming API του Twitter. Στο πρώτο του πεδίο επιλέχθηκε η λειτουργία «Filter Endpoint», ώστε να μπορούν να φιλτραριστούν tweets με συγκεκριμένα χαρακτηριστικά. Στα επόμενα τέσσερα πεδία εισήχθησαν τα διαπιστευτήρια που δόθηκαν από το Twitter Application, ενώ στα αμέσως δύο επόμενα πεδία επιλέχθηκε η αγγλική γλώσσα και η λέξη κλειδί «refugees» ώστε να εισαχθούν στην ροή τα κατάλληλα tweets.

The screenshot shows the 'Configure Processor' dialog box with the 'Properties' tab selected. It contains a table with the following data:

Property	Value
Twitter Endpoint	Filter Endpoint
Consumer Key	R2xwb7ECbDWBv8aOy4KJ3IVj1
Consumer Secret	Sensitive value set
Access Token	405816754-sSeU0PFyGXM0evXMhUVIXHbb7rkebu...
Access Token Secret	Sensitive value set
Languages	en
Terms to Filter On	refugees
IDs to Follow	No value set
Locations to Filter On	No value set

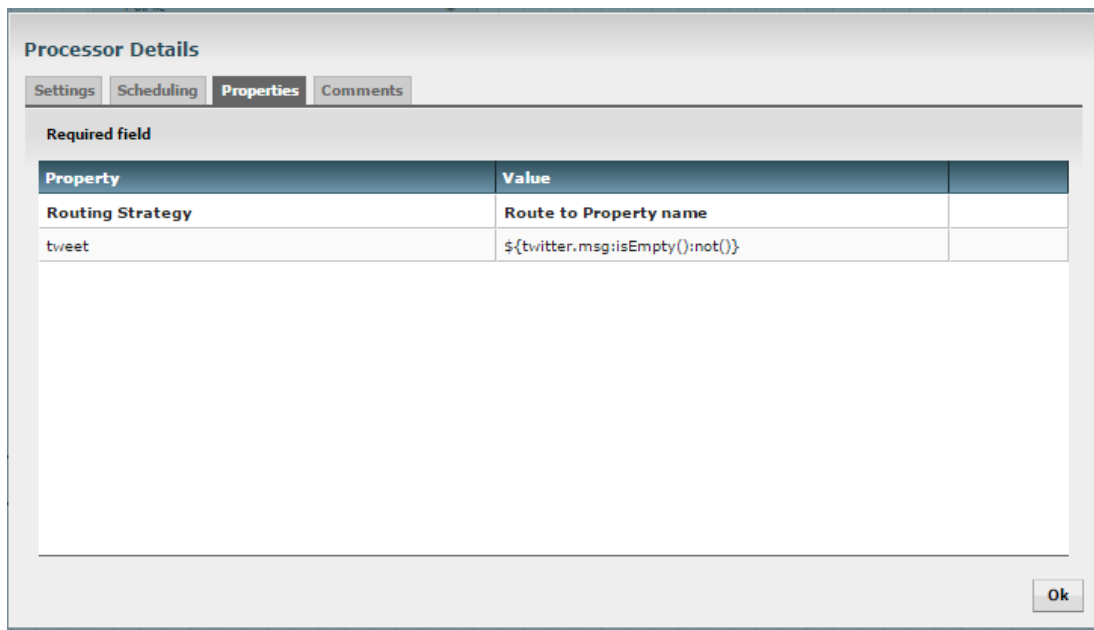
Εικόνα 30 : Επεξεργαστής GetTwitter

Ο δεύτερος επεξεργαστής ονομάζεται EvaluatesJsonPath και αξιολογεί μία ή περισσότερες JsonPath εκφράσεις κατά το περιεχόμενο ενός FlowFile. Τα αποτελέσματα αυτών των εκφράσεων ανατίθενται σε FlowFiles Attributes ή εγγράφονται εσωτερικά στο ίδιο το FlowFile, ανάλογα με την παραμετροποίηση του επεξεργαστή. Στο πρώτο πεδίο επιλέχθηκε τα αποτελέσματα της JsonPath έκφρασης να εγγράφονται σε ένα FlowFile Attribute. Στο δεύτερο, ο τύπος που επιστρέφεται να εντοπίζει αυτόματα αν είναι FlowFile Attribute και να επιστέφει scalar ή FlowFile content σε μορφή Json. Το τρίτο πεδίο αγνοεί τα attributes που λείπουν, ενώ το τέταρτο πεδίο αναπαριστά τις Null τιμές με ένα κενό αλφαριθμητικό. Τα επόμενα πεδία είναι τα Attributes που δημιουργήθηκαν.



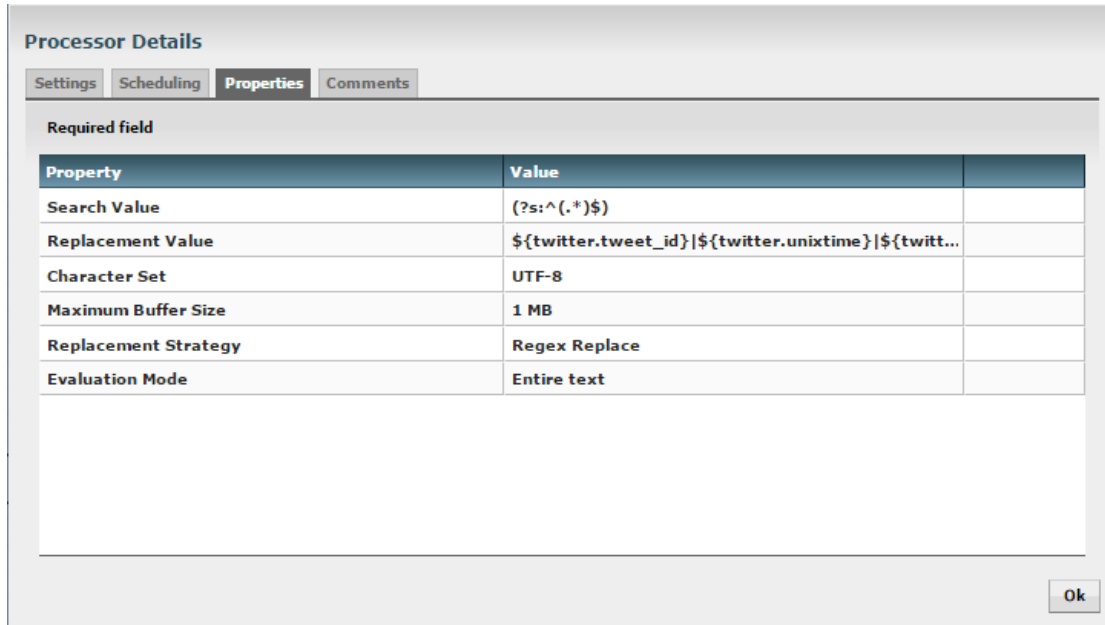
Εικόνα 32 : Επεξεργαστής EvaluateJsonPath

Ο τρίτος επεξεργαστής ονομάζεται RouteOnAttribute και δρομολογεί τα FlowFiles στον επόμενο κόμβο, εάν το Attribute tweet δεν είναι κενό.



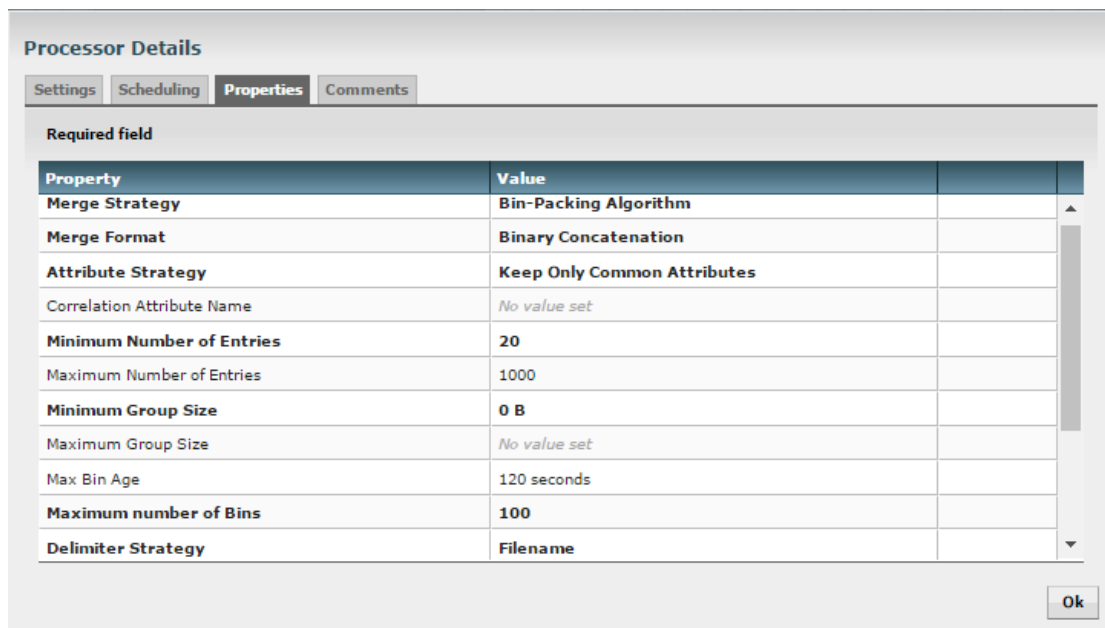
Εικόνα 31 : Επεξεργαστής RouteOnAttribute

Ο τέταρτος επεξεργαστής ονομάζεται ReplaceText και ανανεώνει το περιεχόμενο ενός FlowFile αξιολογώντας το περιεχόμενο μιας Κανονικής Έκφρασης, αντικαθιστώντας το κομμάτι του περιεχομένου που αντιστοιχεί στην τιμή που έχει αποδοθεί στο αντίστοιχο πεδίο.



Εικόνα 33 : Επεξεργαστής ReplaceText

Ο πέμπτος επεξεργαστής ονομάζεται MergeContent και συγχωνεύει μια ομάδα από FlowFiles σε ένα ενιαίο FlowFile σύμφωνα με μία στρατηγική που ορίστηκε και



Εικόνα 34 : Επεξεργαστής MergeContent

φαίνεται παρακάτω. Ο επεξεργαστής αυτός θα πρέπει να δέχεται είσοδο μόνο από μια σύνδεση εναλλακτικά δεν θα μπορεί να δημιουργήσει το ενιαίο αρχείο.

Ο έκτος επεξεργαστής ονομάζεται PutFile και αποθηκεύει τα περιεχόμενα ενός FlowFile στο τοπικό σύστημα αρχείων.

Processor Details

Settings | Scheduling | **Properties** | Comments

Required field

Property	Value
Directory	/tmp/tweets
Conflict Resolution Strategy	fail
Create Missing Directories	true
Maximum File Count	No value set
Last Modified Time	No value set
Permissions	No value set
Owner	No value set
Group	No value set

Ok

Εικόνα 35 : Επεξεργαστής PutFile

Ο έβδομος επεξεργαστής ονομάζεται PutHDFS και αποθηκεύει τα περιεχόμενα ενός FlowFile στο σύστημα αρχείων του Hadoop.

Processor Details

Settings | Scheduling | **Properties** | Comments

Required field

Property	Value
Hadoop Configuration Resources	/etc/hadoop/conf/core-site.xml
Kerberos Principal	No value set
Kerberos Keytab	No value set
Kerberos Relogin Period	4 hours
Directory	/tmp/tweets_staging
Conflict Resolution Strategy	replace
Block Size	No value set
IO Buffer Size	No value set
Replication	1
Permissions umask	No value set
Remote Owner	No value set

Ok

Εικόνα 36 : Επεξεργαστής PutHDFS

6.3.2 Καθαρισμός Δεδομένων (Data Cleansing)

Τα δεδομένα που αποθηκεύτηκαν μέσω του NiFi σε μορφή αρχείου στο φάκελο tweets_staging του HDFS, φορτώθηκαν εν συνεχεία στον πίνακα tweets_text (Εικόνα ?), όπου ξεκίνησε η διαδικασία του καθαρισμού των δεδομένων, μέσω hive εντολών.

Ο καθαρισμός των δεδομένων περιλαμβάνει τρεις αλλαγές στα αρχικά δεδομένα:

- Απαλοιφή περιττής πληροφορίας
- Μορφοποίηση του χρόνου δημιουργίας του tweet
- Δημιουργία πληροφορίας για τη χώρα προέλευσης του tweet

Ως περιττή πληροφορία θεωρήθηκε το όνομα χρήστη, η γλώσσα δημιουργίας του tweet (διότι τα δεδομένα φιλτράρονταν έτσι ώστε να έρχονται μόνο στα αγγλικά) και το fulltext.

Η μορφοποίηση του χρόνου δημιουργίας του tweet πραγματοποιήθηκε με τη βοήθεια του view tweets_simple (του πίνακα tweets_text) χρησιμοποιώντας έτοιμες συναρτήσεις της hive για την τροποποίηση της κολώνας created_time σε timestamp, η οποία απεικονίζει την ώρα δημιουργίας του tweet σε μορφή : χρόνος-μήνας-ημέρα, ώρα-λεπτά-δευτερόλεπτα (YY-MM-DD hh-mm-ss).

Η πληροφορία για τη χώρα προέλευσης του tweet δημιουργήθηκε με τη βοήθεια του reference πίνακα time_zone_map, ο οποίος αντιστοιχίζει το εκάστοτε time zone σε μία χώρα (country). Το αποτέλεσμα της ένωσης του πίνακα time_zone_map με το view tweets_simple εισήχθη, είναι το επόμενο view tweets_clean.

Παρακάτω εμφανίζεται ο προγραμματιστικός κώδικας που χρησιμοποιήθηκε για την υλοποίηση των παραπάνω εντολών.

- Δημιουργία πίνακα για τη μορφοποίηση των tweets :

```
CREATE EXTERNAL TABLE IF NOT EXISTS tweets_text (  
    tweet_id bigint,  
    created_unixtime bigint,  
    created_time string,  
    lang string,  
    displayname string,  
    time_zone string,
```



```
        msg string,  
        fulltext string)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY "|"   
LOCATION "/tmp/tweets_staging";
```

- Δημιουργία πίνακα για την αντιστοίχιση του time zone σε χώρα :

```
CREATE EXTERNAL TABLE if not exists time_zone_map (  
    time_zone string,  
    country string )  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY '\t'  
STORED AS TEXTFILE  
LOCATION '/tmp/data/tables/time_zone_map';
```

- Δημιουργία δύο views πινάκων από tweets, η οποία θα απλοποιήσει τις στήλες των δεδομένων :

```
CREATE VIEW IF NOT EXISTS tweets_simple AS  
SELECT  
    tweet_id,  
    cast ( from_unixtime( unix_timestamp(concat(  
        '2015 ', substring(created_time,5,15)), 'yyyy  
        MMM dd hh:mm:ss')) as timestamp) ts,  
    msg,  
    time_zone  
FROM tweets_text;
```

```
CREATE VIEW IF NOT EXISTS tweets_clean AS  
SELECT  
    t.tweet_id,  
    t.ts,  
    t.msg,  
    m.country  
FROM tweets_simple t  
LEFT OUTER JOIN time_zone_map m  
ON t.time_zone = m.time_zone;
```

6.3.3 Επεξεργασία Δεδομένων (Data Processing)

Η επεξεργασία των δεδομένων πραγματοποιήθηκε με σκοπό τον υπολογισμό των συναισθημάτων των χρηστών, ακολουθώντας τον παρακάτω αλγόριθμο κατηγοριοποίησης (classification) :

- Ανάλυση των tweets σε λέξεις
- Απόδοση συναισθήματος σε κάθε λέξη
- Απόδοση συνολικού συναισθήματος σε κάθε tweet

Με τη βοήθεια έτοιμων συναρτήσεων της Hive, έγινε επεξεργασία του μηνύματος του εκάστοτε tweet, σπάζοντάς το σε λέξεις και κρατώντας παράλληλα για κάθε μία από αυτές το tweet_id που της αντιστοιχεί. Δημιουργήθηκε, λοιπόν, αρχικά το view L1 και στη συνέχεια το view L2, το οποίο περιέχει όλες τις λέξεις που εμφανίστηκαν στα tweets που εξήχθησαν, διατηρώντας πάντα το tweet_id κάθε μίας.

Η απόδοση συναισθήματος σε κάθε λέξη πραγματοποιήθηκε με τη βοήθεια ενός reference πίνακα, ονόματι dictionary. Ο πίνακας αυτός περιέχει ένα μεγάλο αριθμό λέξεων αντιστοιχισμένων με μία πολικότητα σε αριθμητική μορφή ανάλογα με το συναίσθημα που αυτές εκφράζουν, χαρακτηρίζοντάς τις ως θετικές, αρνητικές ή ουδέτερες. Έτσι, για κάθε λέξη του view L2 για την οποία υπήρχε ταύτιση με μία λέξη του πίνακα dictionary αποτιμήθηκε ένα συναίσθημα – με τιμές -1 για αρνητικό, 0 για ουδέτερο και 1 για θετικό – το οποίο καταχωρήθηκε στο view L3 διατηρώντας εκ νέου το tweet_id κάθε λέξης.

Τέλος, συναθροίζοντας την πολικότητα – συναίσθημα των λέξεων που είχαν κοινό tweet_id, αποδόθηκε ένα συνολικό συναίσθημα σε κάθε tweet, χαρακτηρίζοντας το ως θετικό αν το άθροισμα των επιμέρους συναισθημάτων είναι μεγαλύτερο του μηδενός, αρνητικό αν είναι μικρότερο και ουδέτερο αν το άθροισμα είναι ίσο με το μηδέν. Το αποτέλεσμα αποθηκεύτηκε στον πίνακα tweets_sentiment (Εικόνα ?).

Παρακάτω εμφανίζεται ο προγραμματιστικός κώδικας που χρησιμοποιήθηκε για την υλοποίηση των παραπάνω εντολών.

- Υπολογισμός sentiment :

```
CREATE VIEW IF NOT EXISTS l1 AS
SELECT
    tweet_id,
    words
FROM tweets_text lateral view explode(sentences(
lower(msg))) dummy as words;
```

```
CREATE VIEW IF NOT EXISTS l2 AS
SELECT
    tweet_id,
    word
FROM l1 lateral view explode( words ) dummy as word;
```

```
CREATE EXTERNAL TABLE if not exists dictionary (
    word string,
    polarity string )
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE
LOCATION '/tmp/data/tables/dictionary';
```

```
CREATE VIEW IF NOT EXISTS l3 AS
SELECT
    tweet_id,
    l2.word,
    case d.polarity
    when 'negative' then -1
    when 'positive' then 1
    else 0 end as polarity
FROM l2 left outer join dictionary d on l2.word =
d.word;
```

- Απόδοση πολικότητας – συναισθήματος συνολικά σε ένα tweet :

```
CREATE TABLE IF NOT EXISTS tweets_sentiment
STORED AS ORC AS
SELECT
    tweet_id,
    case
        when sum( polarity ) > 0 then 'positive'
        when sum( polarity ) < 0 then 'negative'
        else 'neutral' end as sentiment
FROM 13 group by tweet_id;
```

6.3.4 Ανάλυση Δεδομένων (Data Analysis)

Εφόσον το συναίσθημα ανά tweet υπολογίστηκε, για να γίνει εφικτή η υλοποίηση των αναλύσεων, ενοποιήθηκαν οι πίνακες tweets_clean και tweets_sentiment με βάση το κοινό tweet_id. Παράλληλα, για τη διευκόλυνση των αναλύσεων, οι πολικότητες αντικαταστάθηκαν από θετικούς αριθμούς, έτσι οι θετικές συνολικές πολικότητες αποτιμήθηκαν ως 3, οι ουδέτερες ως 2 και οι αρνητικές πολικότητες ως 1.

- Αντικατάσταση πολικότητας – συναισθήματος :

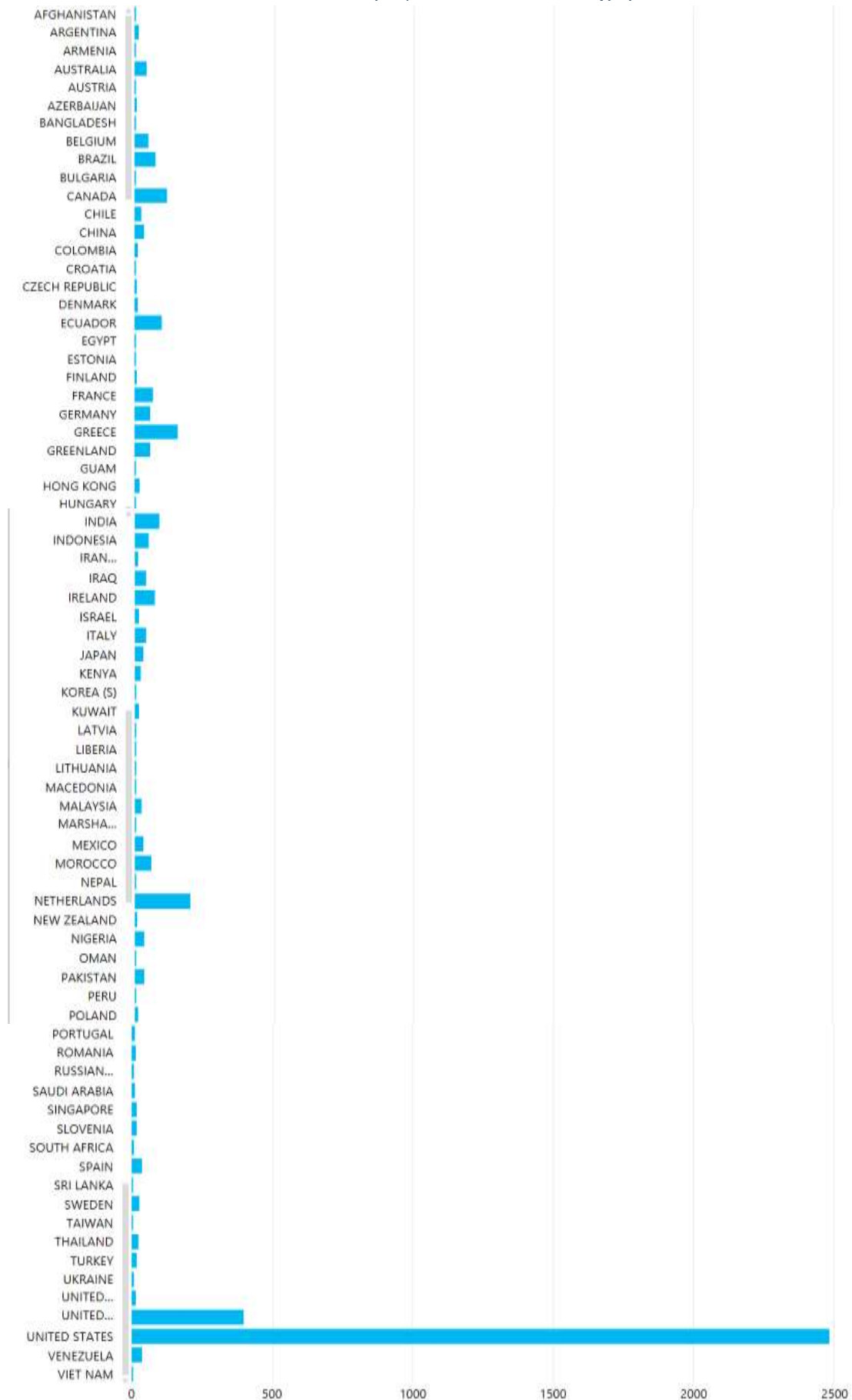
```
CREATE TABLE IF NOT EXISTS tweetsbi
STORED AS ORC
AS SELECT
    t.*,
    case s.sentiment
        when 'positive' then 3
        when 'neutral' then 2
        when 'negative' then 1
    end as sentiment
FROM tweets_clean t
LEFT OUTER JOIN tweets_sentiment s
on t.tweet_id = s.tweet_id;
```

Παρακάτω παρουσιάζονται κάποια διαγράμματα με τα περιεχόμενα του ανωτέρω πίνακα :



Εικόνα 37 : Απεικόνιση Ανάλυσης Συναισθήματος σε Χάρτη

Εικόνα 38 : Αριθμός των tweets ανά χώρα



6.3.5 Απεικόνιση δεδομένων (Data Visualization)

Για την υλοποίηση της απεικόνισης των δεδομένων δημιουργήθηκε ο πίνακας `aggregated_sentiment`, ο οποίος περιλαμβάνει για κάθε χώρα το μέγιστο συναίσθημα. Τα δεδομένα αυτού το πίνακα αποθηκεύτηκαν σε ένα `csv` αρχείο. Η απεικόνιση έγινε με την βοήθεια κώδικα (`html`, `css`, `javascript`), ο οποίος αφού αρχικά διαβάσει το αρχείο, στη συνέχεια το απεικονίζει, μέσω του `api` του `google maps`, σε χάρτη, χρησιμοποιώντας διαφορετικό χρώμα για κάθε πολικότητα – συναίσθημα.

Παρακάτω παρουσιάζονται ο προγραμματιστικός κώδικας που χρησιμοποιήθηκε για την παραπάνω υλοποίηση καθώς και η απεικόνιση των αποτελεσμάτων.

```
<!DOCTYPE html>
<html>
<head>
<link          rel="stylesheet"          type="text/css"
href="mystyle.css">
<script              type="text/javascript"
src="https://maps.googleapis.com/maps/api/js?v=3.exp">
</script>
<script              type="text/javascript"
src="http://alysql.org/console/alasql.min.js"></script
>

<style>
html, body, #map-canvas {
    height: 100%;
    width: 100%;
    margin: 0px;
    padding: 0px
}

</style>
</head>
<body>

<div id="map-canvas"></div>
</body>
</html>
<script type="text/javascript">

    function initialize() {
```

```

var cityCircle;

    var contentString = '<div id="content">'+
    '<div id="siteNotice">'+
    '</div>'+
    '<h1                                id="firstHeading"
class="firstHeading">Global Sentiment Analysis for
Refugees</h1>'+
    '<div id="bodyContent">'+
    '<p>On the map , there are three types of
circles ' +
    '<b>green</b> for positive '+
    '<b>blue</b> for neutreal '+
    '<b>red</b> for negative</p>'+
    '</div>'+
    '</div>';

var infowindow = new google.maps.InfoWindow({
    content: contentString
});

    alasql('SELECT * FROM
CSV("https://rawgit.com/Billntua/sentiment/master/aver
age-latitude-longitude-
countries%20(1).csv",{headers:true}) WHERE Latitude
BETWEEN -100 AND 100', [], function(country){
    var mapOptions = { zoom : 3, center : new
google.maps.LatLng(40, 0),
    mapTypeId : google.maps.MapTypeId.ROADMAP
    };
    var map = new
google.maps.Map(document.getElementById('map-canvas'),
mapOptions);
    for (i = 0; i < (country.length); i++) {
        var opts = {
            strokeColor : '#000000',
            fillColor
            :
["red","blue","green"][(country[i].Sentiment-1)],
            fillOpacity : 0.5,
            map : map,

```



```
                center                :                new  
google.maps.LatLng(country[i].Latitude, country[i].Long  
itude),  
                radius : 100000  
                };  
var                circle1                =                new  
google.maps.Circle(opts);  
                circle1.addListener('click',  
function() {  
                infowindow.open(map, circle1);  
                });  
                }  
                });  
                }  
//                google.maps.event.addDomListener(window,  
'load', initialize);  
initialize();  
</script>
```



Εικόνα 39 : Sentiment Analysis,

Απεικόνιση αποτελεσμάτων Εξόρυξης Δεδομένων στο Google Maps

Κεφάλαιο 7 : Επίλογος

7.1 Σύνοψη – Συμπεράσματα

Στην παρούσα διπλωματική εργασία έγινε προσπάθεια για τη σχεδίαση και την υλοποίηση ενός συστήματος, που σκοπό έχει την αποδοτική εξόρυξη γνώσης μέσα από μεγάλο όγκο δεδομένα. Τα δεδομένα, που χρησιμοποιήθηκαν, εξήχθησαν από τον ιστότοπο κοινωνικής δικτύωσης Twitter. Στόχος της εργασίας ήταν η ανάλυση αυτής της βασικής πληροφορίας σύμφωνα με έναν αλγόριθμο κατηγοριοποίησης και η εξαγωγή γνώσης μέσα από αυτήν. Η διαφορά με προηγούμενες διπλωματικές εργασίες είναι ότι στην παρούσα κατασκευάστηκε ένα ενιαίο σύστημα πραγματικού χρόνου το οποίο επιτελεί συγκεντρωτικά όλες τις επιμέρους λειτουργίες της εξόρυξης δεδομένων ως μία ενιαία διαδικασία και όχι ως μεμονωμένες λειτουργίες αυτής ή χρησιμοποιώντας τις κλασσικές τεχνικές Batch Processing. Έτσι, με πυρήνα την πλατφόρμα Hortonworks Data Platform, σχεδιάστηκε ένα τέτοιου είδους σύστημα, για το οποίο ήταν απαραίτητη η κατασκευή ενός μόλις κόμβου, καθώς ο όγκος των δεδομένων δεν ήταν τόσο μεγάλος ώστε να δικαιολογεί την κατασκευή περισσότερων κόμβων στο υποσύστημα του Hadoop. Λόγο αυτού, δεν έγινε εκμεταλλεύσιμο και δεν παρατηρήθηκε το προτέρημα του Hadoop έναντι άλλων συστημάτων, κατά το οποίο τα δεδομένα αντιγράφονται σε τρεις τουλάχιστον κόμβους, προσφέροντας ασφαλή και αξιόπιστη αποθήκευση.

Στο παραπάνω σύστημα εφαρμόστηκαν λειτουργίες όπως η εξαγωγή, ο καθαρισμός, η επεξεργασία, η τροποποίηση, η ανάλυση και η απεικόνιση των δεδομένων, προκειμένου να φανεί ότι πολλές διαφορετικές λειτουργίες μπορούν να εκφραστούν ως επιμέρους λειτουργίες μίας συνολικής διαδικασίας, όπως η εξόρυξη δεδομένων. Τέλος, με χρήση ενός αλγορίθμου κατηγοριοποίησης (classification) παρουσιάστηκε μία στατιστική ανάλυση των συναισθημάτων που μπορεί να εμπεριέχονται σε μεγάλα σύνολα δεδομένων, κατηγοριοποιώντας το συναίσθημα σε τρεις ευρύτερες ομάδες.

Συγκεκριμένα, η υλοποίηση του εν λόγω συστήματος παρέχει στο χρήστη τη δυνατότητα να εισάγει σε αυτό μία λέξη κλειδί και να πάρει τα αποτελέσματα μίας ανάλυσης των συναισθημάτων των χρηστών ενός κοινωνικού δικτύου, όπως το

Twitter, βασισμένα σε αυτή τη λέξη κλειδί, χωρίς να χρειάζεται να επιτελέσει μία μία ξεχωριστά τις επιμέρους λειτουργίες. Χωρίς να χρειάζεται, δηλαδή, να εξάγει ο ίδιος τα δεδομένα, να τα επεξεργαστεί, να τα αναλύσει ή να τα απεικονίσει. Τα αποτελέσματα σε αυτό το επίπεδο κρίνονται ικανοποιητικά.

Όσον αφορά το κομμάτι της ανάλυσης των δεδομένων, φαίνεται να υπάρχει ένα μικρό στατιστικό σφάλμα το οποίο έγκειται στην συγγραφή του εκάστοτε μηνύματος και στη σημασία που αποδίδεται στις λέξεις από το συγγραφέα. Πολλοί χρήστες τείνουν να γράφουν ειρωνικά ή αστεία μηνύματα, μηνύματα με διαφορετική σημασία ή μηνύματα τα οποία περιέχουν σχόλια για κάτι που σχετίζεται με κάποιο θέμα και όχι για το θέμα αυτό καθ' αυτό. Τέτοιου είδους μηνύματα δεν λήφθηκαν υπόψη ως διαφορετικά μηνύματα στην παρούσα διπλωματική εργασία και ενδέχεται να έχουν επηρεάσει τα αποτελέσματα σε σημαντικό βαθμό.

7.2 Μελλοντικές Εργασίες

Το συγκεκριμένο πεδίο έρευνας κρίνεται εξαιρετικά ενδιαφέρον και θα ήταν ιδιαίτερα χρήσιμο για έναν νέο μηχανικό να μπορέσει να συνδυάσει γνώσεις δικτύων υπολογιστών, πληροφορικής, αρχιτεκτονικής και λειτουργίας υπολογιστικών συστημάτων, οι οποίες αποτελούν μέρος των ακαδημαϊκών σπουδών του, για να φέρει εις πέρας μία τέτοιου είδους έρευνα. Οι επεκτάσεις αυτής της έρευνας μπορεί να είναι πραγματικά πολλές ανάλογα το προσωπικό ενδιαφέρον και τις επιδιώξεις του εκάστοτε ερευνητή. Θα μπορούσε για παράδειγμα να γίνει καταγραφή δεδομένων για ένα μεγάλο και ευρύ χρονικό διάστημα, ώστε η ανάλυση να είναι πιο αποδοτική και να ελαχιστοποιούνται τα στατιστικά σφάλματα. Εναλλακτικά, θα μπορούσε να υλοποιηθεί ένας αντίστοιχος real-time machine learning αλγόριθμος, ο οποίος θα τρέχει σε ένα τέτοιο σύνολο δεδομένων με απώτερο σκοπό την εκτίμηση – πρόβλεψη των ενδιαφερόντων, των κινήσεων ή των ενεργειών μίας ομάδας χρηστών ή την κατηγοριοποίηση των χρηστών με βάση τις κινήσεις τους σε έναν ιστότοπο κοινωνικής δικτύωσης, όπως το Twitter.

Υπόμνημα : HDFS

1. Εισαγωγή

Το Hadoop Distributed File System (HDFS) είναι ένα καταναμημένο σύστημα αρχείων σχεδιασμένο να λειτουργεί σε μεσαίας ισχύος υλικό (commodity hardware). Έχει πολλές ομοιότητες με τα υπάρχοντα καταναμημένα συστήματα αρχείων, ωστόσο υπάρχουν και σημαντικές διαφορές. Το HDFS είναι εξαιρετικά ανεκτικό σε σφάλματα και έχει σχεδιαστεί για να αναπτύσσεται σε υλικό χαμηλού κόστους. Παρέχει υψηλής απόδοσης πρόσβαση στα δεδομένα της εφαρμογής και είναι κατάλληλο για εφαρμογές που έχουν μεγάλα σύνολα δεδομένων. Το HDFS είναι πιο ελαστικό σε μερικές POSIX απαιτήσεις για να επιτρέψει πρόσβαση ροής (streaming access) στο σύστημα αρχείων δεδομένων. Το HDFS χτίστηκε αρχικά ως υποδομή για τη μηχανή αναζήτησης ιστού Apache Nutch. Το HDFS αποτελεί πλέον έργο του Apache Hadoop [\[20\]](#).

2. Παραδοχές και Στόχοι

2.1 Αποτυχίες Υλικού (Hardware Failure)

Οι αποτυχίες υλικού είναι ο κανόνας και όχι η εξαίρεση. Ένα παράδειγμα HDFS μπορεί να αποτελείται από εκατοντάδες ή χιλιάδες servers, με κάθε έναν από αυτούς να αποτελεί αποθηκευτικό χώρο για τα δεδομένα του συστήματος αρχείων. Το γεγονός ότι υπάρχει ένας τεράστιος αριθμός εξαρτημάτων και ότι κάθε εξάρτημα έχει μία σημαντική πιθανότητα αποτυχίας σημαίνει ότι κάποιο από αυτά θα είναι πάντα μη λειτουργικό. Ως εκ τούτου, η ανίχνευση των βλαβών και η γρήγορη, αυτόματη ανάκτηση δεδομένων είναι ένας βασικός αρχιτεκτονικός στόχος του HDFS.

2.2 Πρόσβαση Ροής Δεδομένων (Streaming Data Access)

Οι εφαρμογές που τρέχουν στο HDFS χρειάζονται συνεχή πρόσβαση στα σύνολα δεδομένων τους. Δεν είναι εφαρμογές γενικής χρήσης που συνήθως τρέχουν σε γενικού σκοπού συστήματα αρχείων. Το HDFS έχει σχεδιαστεί περισσότερο για batch επεξεργασία και όχι για διαδραστική χρήση. Η έμφαση δίνεται στην υψηλή απόδοση και όχι στη χαμηλή καθυστέρηση κατά την πρόσβαση των δεδομένων. Το POSIX επιβάλλει πολλές σκληρές απαιτήσεις που δεν χρειάζονται για τις εφαρμογές που

στοχεύουν στο HDFS, έτσι η POSIX σημασιολογία σε μερικές βασικές περιοχές έχει θυσιάσει για την αύξηση του ρυθμού διεκπεραίωσης δεδομένων.

2.3 Μεγάλα Σύνολα Δεδομένων (Large Data Sets)

Οι εφαρμογές που τρέχουν στο HDFS έχουν μεγάλα σύνολα δεδομένων. Ένα τυπικό αρχείο στο HDFS έχει μέγεθος από gigabytes σε terabytes. Έτσι, το HDFS είναι συντονισμένο για να υποστηρίζει μεγάλα αρχεία. Θα πρέπει να παρέχει υψηλό συνολικό εύρος ζώνης δεδομένων και κλιμάκωση εκατοντάδων κόμβων σε ένα κοινό σύμπλεγμα. Θα πρέπει να υποστηρίζει δεκάδες εκατομμύρια αρχεία.

2.4 Απλό Μοντέλο Συνοχής (Simple Coherency Model)

Οι εφαρμογές του HDFS χρειάζονται ένα write-once-read-many μοντέλο πρόσβασης αρχείων. Ένα αρχείο που δημιουργήθηκε, γράφτηκε και έκλεισε δεν χρειάζεται να αλλάξει. Η παραδοχή αυτή απλοποιεί τα προβλήματα συνοχής των δεδομένων και επιτρέπει υψηλής απόδοσης πρόσβαση στα δεδομένα (high throughput data access). Μία MapReduce εφαρμογή ή μία web εφαρμογή ανίχνευσης ταιριάζει απόλυτα με αυτό το μοντέλο. Υπάρχει ένα σχέδιο για να υποστηριχθούν προσαρτημένες-εγγραφές (appending-writes) σε αρχεία στο μέλλον.

2.5 Μετακίνηση υπολογισμών αντί μετακίνησης δεδομένων

Ένας υπολογισμός που έχει ζητηθεί από μία εφαρμογή είναι πολύ πιο αποτελεσματικός εάν εκτελείται κοντά στα δεδομένα πάνω στα οποία λειτουργεί, ειδικά όταν το μέγεθος του συνόλου δεδομένων είναι τεράστιο. Όμως, συχνά είναι καλύτερο να μεταναστεύσει ο υπολογισμός κοντά στο σημείο όπου βρίσκονται τα δεδομένα παρά να μετακινηθούν τα δεδομένα στο σημείο όπου εκτελείται η εφαρμογή. Το HDFS παρέχει διεπαφές για τις εφαρμογές με σκοπό την ικανοποίηση αυτής της παραδοχής, ελαχιστοποιώντας έτσι τη συμφόρηση του δικτύου και αυξάνοντας τη συνολική απόδοση του συστήματος.

2.6 Φορητότητα

Το HDFS έχει σχεδιαστεί ώστε να είναι εύκολα φορητό από μία πλατφόρμα σε άλλη. Αυτό διευκολύνει την ευρεία υιοθέτησή του ως πλατφόρμα επιλογής για ένα μεγάλο σύνολο εφαρμογών.

3. NameNode και DataNodes

Το HDFS έχει μία master/slave αρχιτεκτονική. Ένα HDFS σύμπλεγμα αποτελείται από έναν κοινό NameNode, ο οποίος είναι ένας master server που διαχειρίζεται το namespace του συστήματος αρχείων και ρυθμίζει την πρόσβαση των πελατών στα αρχεία. Επιπλέον, υπάρχουν αρκετοί DataNodes, συνήθως ένας ανά κόμβο του συμπλέγματος, οι οποίοι έχουν την αποθηκευτική διαχείριση των κόμβων στους οποίους τρέχουν. Το HDFS εκθέτει ένα namespace του συστήματος αρχείων και επιτρέπει στα δεδομένα των χρηστών να αποθηκεύονται σε αρχεία. Εσωτερικά, ένα αρχείο διαχωρίζεται σε ένα ή περισσότερα blocks, τα οποία και αποθηκεύονται σε ένα σύνολο από DataNodes. Ο NameNode εκτελεί namespace λειτουργίες στο σύστημα αρχείων, όπως το άνοιγμα, το κλείσιμο και τη μετονομασία αρχείων και φακέλων. Επίσης, καθορίζει τη χαρτογράφηση των blocks στους DataNodes. Οι DataNodes είναι υπεύθυνοι για την εξυπηρέτηση αιτημάτων ανάγνωσης και εγγραφής των πελατών του συστήματος αρχείων. Εκτελούν επίσης τη δημιουργία, διαγραφή και αντιγραφή blocks, κατόπιν εντολής από το NameNode. Ο NameNode και οι DataNodes είναι κομμάτια του λογισμικού που έχει σχεδιαστεί για να λειτουργεί με μηχανές μεσαίας ισχύος. Οι μηχανές αυτές τρέχουν συνήθως ένα λειτουργικό σύστημα GNU/Linux. Το HDFS είναι χτισμένο σε Java και έτσι κάθε μηχανή που υποστηρίζει Java μπορεί να τρέχει το NameNode ή DataNode λογισμικό. Η χρήση της, εξαιρετικά φορητής, γλώσσας Java σημαίνει ότι το HDFS μπορεί να αναπτυχθεί σε ένα ευρύ φάσμα μηχανών. Ένα τυπικό HDFS σύστημα έχει ένα μηχάνημα ειδικής χρήσης (dedicated machine), που τρέχει μόνο το NameNode λογισμικό, και όλα τα υπόλοιπα μηχανήματα του συμπλέγματος εκτελούν μία υπόδειξη του DataNode λογισμικού. Η αρχιτεκτονική δεν αποκλείει την εκτέλεση πολλών DataNodes στο ίδιο μηχάνημα, αν και κάτι τέτοιο συμβαίνει σπάνια σε μια πραγματική εφαρμογή.

Η ύπαρξη ενός μόνο κοινού NameNode σε ένα σύμπλεγμα απλοποιεί σε μεγάλο βαθμό την αρχιτεκτονική του συστήματος. Ο NameNode είναι ο διαιτητής και το αποθετήριο όλων των μεταδεδομένων (metadata) του HDFS. Το σύστημα έχει σχεδιαστεί με τέτοιο τρόπο ώστε τα δεδομένα χρήστη να μη ρέουν ποτέ μέσω του NameNode.

4. Ονομασία και οργάνωση αρχείων (File System Namespace)

Το HDFS υποστηρίζει μια παραδοσιακή ιεραρχική οργάνωση αρχείων. Ένας χρήστης ή μια εφαρμογή μπορεί να δημιουργήσει φακέλους και να αποθηκεύσει αρχεία στο εσωτερικό αυτών των φακέλων. Η ιεραρχία του namespace είναι παρόμοια με τα περισσότερα από τα υπάρχοντα συστήματα αρχείων. Μπορούν να δημιουργηθούν, να αφαιρεθούν και να μετονομαστούν αρχεία ή να μετακινηθεί ένα αρχείο από ένα φάκελο σε έναν άλλο. Οποιαδήποτε αλλαγή στο namespace ή τις ιδιότητές του καταγράφεται από το NameNode. Το HDFS δεν υποστηρίζει προς το παρόν ποσοτώσεις χρήστη (user quotas), hard links ή soft links (symlinks), αν και η αρχιτεκτονική του δεν αποκλείει την εφαρμογή αυτών των χαρακτηριστικών.

5. Αντιγραφή Δεδομένων (Data Replication)

Το HDFS, όπως αναφέρθηκε προηγουμένως, έχει σχεδιαστεί για την αξιόπιστη αποθήκευση πολύ μεγάλων αρχείων σε μηχανήματα ενός μεγάλου συμπλέγματος. Αποθηκεύει κάθε αρχείο ως μια ακολουθία από blocks, τα οποία (εκτός του τελευταίου) έχουν ίδιο μέγεθος, με σκοπό την ανοχή σε σφάλματα. Το μέγεθος του block και ο παράγοντας αντιγραφής είναι παραμετροποιήσιμα ανά αρχείο. Μια εφαρμογή μπορεί να καθορίσει, μέσω του παράγοντα αντιγραφής (replication factor), τον αριθμό αντιγράφων ενός αρχείου που θα πρέπει να διατηρηθούν από το HDFS. Ο παράγοντας αντιγραφής καθορίζεται κατά τη δημιουργία του αρχείου και μπορεί να αλλάξει αργότερα. Τα αρχεία στο HDFS είναι write-once και έχουν αυστηρά ένα συγγραφέα ανά πάσα στιγμή.

Ο NameNode παίρνει όλες τις αποφάσεις σχετικά με την αντιγραφή των blocks. Λαμβάνει περιοδικά ένα Heartbeat και ένα Blockreport από κάθε έναν από τους DataNodes του συμπλέγματος. Η λήψη του Heartbeat υποδηλώνει ότι ο DataNode λειτουργεί κανονικά. Το Blockreport περιέχει μια λίστα με όλα τα blocks σε ένα DataNode.

5.1 Τοποθέτηση Αντιγράφου (Replica Placement)

Η τοποθέτηση των αντιγράφων είναι κρίσιμη για την αξιοπιστία και την απόδοση του HDFS και η βελτιστοποίησή της διακρίνει το HDFS από τα περισσότερα

κατανεμημένα συστήματα αρχείων. Σκοπός της πολιτικής τοποθέτησης αντιγράφων με επίγνωση του ραφιού (rack-aware replica placement policy) είναι να βελτιωθεί η αξιοπιστία των δεδομένων, η διαθεσιμότητα και η χρησιμοποίηση του εύρους ζώνης του δικτύου. Η τρέχουσα εκτέλεση για την εν λόγω πολιτική είναι μια πρώτη προσπάθεια προς αυτή την κατεύθυνση. Οι βραχυπρόθεσμοι στόχοι της είναι η επικύρωσή της σε συστήματα παραγωγής, η εξαγωγή συμπερασμάτων για τη συμπεριφορά της και η οικοδόμηση θεμελίων για δοκιμή και έρευνα πιο εξελιγμένων πολιτικών.

Μεγάλα HDFS συστήματα τρέχουν σε ένα σύμπλεγμα υπολογιστών που συνήθως εκτείνεται σε πολλά ράφια (racks). Η επικοινωνία μεταξύ δύο κόμβων που ανήκουν σε διαφορετικά ράφια πρέπει να περάσει μέσα από διακόπτες (switches). Στις περισσότερες περιπτώσεις, το εύρος ζώνης δικτύου μεταξύ κόμβων στο ίδιο ράφι είναι μεγαλύτερο από αυτό μεταξύ κόμβων διαφορετικών ραφιών.

Ο NameNode προσδιορίζει την ταυτότητα του ραφιού (rack-ID) όπου ανήκει κάθε DataNode μέσω της διαδικασίας που περιγράφεται από το [Hadoop Rack Awareness](#). Μια απλή, αλλά όχι βέλτιστη, πολιτική είναι να τοποθετηθούν αντίγραφα σε ξεχωριστά ράφια. Αυτό αποτρέπει την απώλεια δεδομένων όταν ένα ολόκληρο ράφι αποτυγχάνει, επιτρέπει τη χρήση του εύρους ζώνης από πολλαπλά ράφια κατά την ανάγνωση των δεδομένων και κατανέμει ομοιόμορφα τα αντίγραφα στο σύμπλεγμα, κάτι που καθιστά εύκολο να εξισορροπηθεί το φορτίο σε πιθανή αποτυχία. Ωστόσο, η πολιτική αυτή αυξάνει το κόστος των εγγραφών, διότι μία εγγραφή πρέπει να μεταφέρει blocks σε πολλαπλά ράφια.

Κατά τη συνήθη περίπτωση, όπου ο παράγοντας αντιγραφής είναι ίσος με τρία, η πολιτική τοποθέτησης του HDFS είναι να εναποθέτει ένα αντίγραφο σε έναν κόμβο από το τοπικό ράφι, ένα άλλο σε έναν κόμβο ενός διαφορετικού (απομακρυσμένου) ραφιού και το τελευταίο σε έναν διαφορετικό κόμβο του ίδιου απομακρυσμένου ραφιού. Αυτή η πολιτική μοιράζει την ενδο-ραφική κυκλοφορία εγγραφής, η οποία γενικά βελτιώνει την απόδοση εγγραφής. Η πιθανότητα αποτυχίας ενός ραφιού είναι πολύ μικρότερη από εκείνη ενός κόμβου, έτσι η πολιτική αυτή δεν επηρεάζει την αξιοπιστία των δεδομένων και τις εγγυήσεις διαθεσιμότητας. Ωστόσο, μειώνει το συνολικό εύρος ζώνης του δικτύου που χρησιμοποιείται κατά την ανάγνωση των δεδομένων, αφού το κάθε block τοποθετείται σε δύο μόνο διαφορετικά ράφια αντί για

τρία, έστω και αν με αυτή την πολιτική τα αντίγραφα ενός αρχείου δεν κατανέμονται ομοιόμορφα (τα δύο τρίτα των αντιγράφων είναι σε ένα ράφι). Συνολικά, η τρέχουσα, προεπιλεγμένη πολιτική τοποθέτησης αντιγράφων βελτιώνει την απόδοση εγγραφής χωρίς να θυσιάζεται η αξιοπιστία των δεδομένων ή η απόδοση ανάγνωσης.

5.2 Επιλογή Αντιγράφου (Replica Selection)

Για να ελαχιστοποιηθεί η συνολική κατανάλωση εύρους ζώνης και ο χρόνος απόκρισης κατά την ανάγνωση, το HDFS προσπαθεί να ικανοποιήσει ένα αίτημα ανάγνωσης αρχικά από ένα αντίγραφο που είναι πιο κοντά στον αναγνώστη. Αν υπάρχει ένα αντίγραφο στο ίδιο ράφι με τον κόμβο αναγνώστη, τότε προτιμάται αυτό για να ικανοποιήσει το αίτημα ανάγνωσης. Αν το HDFS σύμπλεγμα εκτείνεται σε πολλαπλά κέντρα δεδομένων τότε ένα αντίγραφο που είναι εγκατεστημένο στο τοπικό κέντρο δεδομένων προτιμάται από οποιαδήποτε άλλο απομακρυσμένο αντίγραφο.

5.3 Ασφαλής Λειτουργία (Safemode)

Κατά την εκκίνηση, ο NameNode εισέρχεται σε μια ιδιαίτερη κατάσταση, που ονομάζεται Safemode. Σε αυτή την κατάσταση δεν είναι δυνατή η αντιγραφή blocks δεδομένων, είναι όμως η λήψη μηνυμάτων Heartbeat και Blockreport από τους DataNodes. Κάθε block έχει ένα καθορισμένο ελάχιστο αριθμό αντιγράφων και θεωρείται ότι έχει αντιγραφεί με ασφάλεια όταν τα αντίγραφα αυτά επιβεβαιώσουν τη δημιουργία τους στο NameNode. Μόλις ένα ρυθμιζόμενο ποσοστό από ασφαλή αντίγραφα των blocks δεδομένων επιβεβαιώσουν στο NameNode (συν 30 δευτερόλεπτα επιπλέον), ο NameNode εξέρχεται από την κατάσταση Safemode. Στη συνέχεια, ενημερώνει στον κατάλογο τα blocks δεδομένων (αν υπάρχουν) που εξακολουθούν να έχουν λιγότερα, από τον προκαθορισμένο αριθμό, αντίγραφα. Ο NameNode, τέλος, αντιγράφει αυτά τα blocks σε άλλους DataNodes.

6. Η ανθεκτικότητα του συστήματος αρχείων μεταδεδομένων

Στο HDFS το namespace αποθηκεύεται από το NameNode. Ο NameNode χρησιμοποιεί ένα αρχείο καταγραφής συναλλαγών, που ονομάζεται EditLog, για να καταγράφει συνεχώς κάθε αλλαγή που υφίσταται στο σύστημα αρχείων μεταδεδομένων. Για παράδειγμα, η δημιουργία ενός νέου αρχείου στο HDFS ή η

αλλαγή του παράγοντα αντιγραφής ενός αρχείου οδηγεί το NameNode να εισάγει μία νέα εγγραφή στο EditLog. Ο NameNode χρησιμοποιεί ένα αρχείο από το λειτουργικό σύστημα του τοπικού συστήματος αρχείων για να αποθηκεύσει το EditLog. Το συνολικό namespace, συμπεριλαμβανομένων της χαρτογράφησης των blocks σε αρχεία και των ιδιοτήτων του συστήματος αρχείων, αποθηκεύεται σε ένα αρχείο που ονομάζεται FsImage. Το FsImage αποθηκεύεται επίσης ως αρχείο στο τοπικό σύστημα αρχείων του NameNode.

Ο NameNode κρατά μια εικόνα του συνολικού namespace και του αρχείου Blockmap στη μνήμη. Αυτό το βασικό στοιχείο μεταδεδομένων έχει σχεδιαστεί να είναι συμπαγής, έτσι ώστε ένας NameNode με 4 GB μνήμης RAM να είναι αρκετός για να υποστηρίξει έναν τεράστιο αριθμό αρχείων και φακέλων. Όταν ο NameNode εκκινεί, διαβάζει το FsImage και το EditLog από το δίσκο, εφαρμόζει όλες τις συναλλαγές από το EditLog ως την εκπροσώπηση στη μνήμη του FsImage και αποθηκεύει αυτή τη νέα έκδοση σε ένα νέο FsImage στο δίσκο. Στη συνέχεια, μπορεί να περικόψει το παλιό EditLog αφού οι συναλλαγές του έχουν εφαρμοστεί στο υπάρχον FsImage. Αυτή η διαδικασία ονομάζεται σημείο ελέγχου (check point) και εμφανίζεται μόνο όταν ο NameNode εκκινήσει.

Ο DataNode αποθηκεύει κάθε block δεδομένων σε διαφορετικό αρχείο στο δικό του τοπικό σύστημα αρχείων και δεν έχει καμία γνώση σχετικά με τα αρχεία του HDFS. Χρησιμοποιεί μια ευρετική (heuristic) για να καθορίσει το βέλτιστο αριθμό αρχείων ανά φάκελο και δημιουργεί κατάλληλα υποφακέλους, καθώς το τοπικό σύστημα αρχείων μπορεί να μην είναι ικανό να υποστηρίξει αποτελεσματικά έναν τεράστιο αριθμό αρχείων σε έναν ενιαίο φάκελο. Όταν ένας DataNode ξεκινά, σαρώνει το τοπικό σύστημα αρχείων, παράγει μια λίστα με όλα τα blocks δεδομένων του HDFS που αντιστοιχούν σε κάθε ένα από αυτά τα τοπικά αρχεία και στέλνει την αναφορά αυτή στο NameNode, η οποία αποτελεί το Blockreport.

7. Πρωτόκολλα Επικοινωνίας

Όλα τα πρωτόκολλα επικοινωνίας του HDFS εντάσσονται στο στρώμα TCP/IP. Ο πελάτης εγκαθιδρύει μία σύνδεση, μέσω μίας ρυθμιζόμενης θύρας TCP, με το NameNode και επικοινωνεί το ClientProtocol μαζί του. Οι DataNodes συνομιλούν με το NameNode με το δικό τους πρωτόκολλο. Μία κλήση απομακρυσμένης διαδικασίας

(Remote Procedure Call - RPC) περικλύει τόσο το πρωτόκολλο πελάτη όσο και αυτό του DataNode. Από σχεδιασμού, ο NameNode ποτέ δεν αρχίζει μία RPC, αντ' αυτού ανταποκρίνεται μόνο σε RPC αιτήματα που εκδίδονται από DataNodes ή πελάτες.

8. Ευρωστία

Πρωταρχικός στόχος του HDFS είναι η αξιόπιστη αποθήκευση δεδομένων ακόμη και υπό την παρουσία αποτυχιών, με τις συνηθέστερες να είναι οι αποτυχίες του NameNode και του DataNode και τα χωρίσματα δικτύου (network partitions).

8.1 Αποτυχία Δίσκου, Heartbeats και Επαναντιγραφή

Κάθε DataNode στέλνει περιοδικά ένα Heartbeat μήνυμα στο NameNode. Ένα χωρίσμα δικτύου μπορεί να οδηγήσει ένα υποσύνολο από DataNodes να χάσει τη σύνδεση του με το NameNode. Ο NameNode αντιλαμβάνεται αυτή την κατάσταση με την απουσία ενός Heartbeat μηνύματος από κάποιον DataNode, τον οποίο θεωρεί πλέον ως κόμβο εκτός λειτουργίας και δεν διαβιβάζει οποιαδήποτε νέα αιτήματα E/E (I/O) προς αυτόν. Όλα τα δεδομένα, που ήταν καταχωρημένα σε έναν νεκρό DataNode, δεν είναι πλέον διαθέσιμα από το HDFS. Η αποτυχία ενός DataNode μπορεί να ρίξει τον παράγοντα αντιγραφής μερικών blocks κάτω από την καθορισμένη τιμή του. Ο NameNode παρακολουθεί συνεχώς τα blocks που πρέπει να αναπαραχθούν και ξεκινά την αντιγραφή όποτε χρειάζεται. Η ανάγκη για εκ νέου αντιγραφή μπορεί να προκύψει εξαιτίας πολλών λόγων, όπως ένας DataNode να μην είναι διαθέσιμος, ένα αντίγραφο να έχει καταστραφεί, ένας σκληρός δίσκος σε έναν DataNode να αποτύχει ή ο παράγοντας αντιγραφής ενός αρχείου να αυξηθεί.

8.2 Επαναπροσαρμογή Συμπλέγματος (Cluster Rebalancing)

Η αρχιτεκτονική του HDFS είναι συμβατή με τα συστήματα αναπροσαρμογής των δεδομένων. Ένα σύστημα μπορεί αυτόματα να μετακινεί τα δεδομένα από τον ένα DataNode στον άλλο, αν ο ελεύθερος χώρος σε ένα DataNode είναι μικρότερος από ένα ορισμένο κατώφλι. Ακόμη, σε περίπτωση αιφνίδιας υψηλής ζήτησης για ένα συγκεκριμένο αρχείο, ένα σύστημα μπορεί δυναμικά να δημιουργήσει πρόσθετα αντίγραφα και να αναπροσαρμόσει άλλα δεδομένα στο σύμπλεγμα.

8.3 Ακεραιότητα Δεδομένων (Data Integrity)

Είναι πιθανό ένα block δεδομένων προερχόμενο από ένα DataNode να φτάσει κατεστραμμένο, είτε λόγω σφαλμάτων σε μια συσκευή αποθήκευσης, είτε λόγω βλαβών στο δίκτυο ή προβληματικού λογισμικού. Το λογισμικό πελάτη πραγματοποιεί άθροισμα ελέγχου (checksum) στο περιεχόμενο των HDFS αρχείων. Όταν ένας πελάτης δημιουργεί ένα HDFS αρχείο, υπολογίζει ένα άθροισμα ελέγχου για το κάθε block του αρχείου και αποθηκεύει αυτά τα αθροίσματα σε ένα ξεχωριστό κρυφό αρχείο στο ίδιο HDFS namespace. Όταν ένας πελάτης ανακτά το περιεχόμενο ενός αρχείου από κάποιο DataNode, επαληθεύει ότι τα δεδομένα που έλαβε ταιριάζουν με το αποθηκευμένο άθροισμα ελέγχου στο σχετικό αρχείο. Αν όχι, ο πελάτης μπορεί να επιλέξει να ανακτήσει αυτό το block από έναν άλλο DataNode, που έχει το ίδιο αντίγραφο.

8.4 Αποτυχία Δίσκου Μεταδεδομένων (Metadata Disk Failure)

Το FsImage και το EditLog είναι βασικές δομές δεδομένων του HDFS και ένα σφάλμα σε αυτά μπορεί να κάνει μη λειτουργικό το HDFS. Για το λόγο αυτό, ο NameNode μπορεί να ρυθμιστεί ώστε να υποστηρίζει τη διατήρηση πολλαπλών αντιγράφων του FsImage και του EditLog. Οποιαδήποτε ενημέρωση για το FsImage ή το EditLog οδηγεί στην συγχρονισμένη ενημέρωση όλων των αντιγράφων τους. Αυτή η συγχρονισμένη ενημέρωση μπορεί να υποβαθμίσει το ποσοστό των συναλλαγών namespace ανά δευτερόλεπτο που μπορεί να υποστηρίξει ο NameNode. Ωστόσο, είναι αποδεκτή διότι ακόμη και αν οι HDFS εφαρμογές είναι από φύση τους εφαρμογές έντασης δεδομένων (data intensive), δεν είναι εφαρμογές έντασης μεταδεδομένων (metadata intensive). Όταν ένας NameNode επανεκκινεί, επιλέγει για χρήση το τελευταίο συνεπές αρχείο FsImage και EditLog. Το μηχάνημα του NameNode είναι ζωτικής σημασίας για ένα σύμπλεγμα HDFS, καθώς αν αποτύχει απαιτείται χειροκίνητη παρέμβαση. Προς το παρόν, αυτόματη επανεκκίνηση και εφεδρεία (failover) του λογισμικού του NameNode σε ένα άλλο μηχάνημα δεν υποστηρίζεται.

8.5 Στιγμιότυπα (Snapshots)

Τα στιγμιότυπα υποστηρίζουν την αποθήκευση ενός αντιγράφου δεδομένων σε μια συγκεκριμένη χρονική στιγμή. Μία χρήση της δυνατότητας στιγμιότυπου μπορεί να

είναι να επανέλθει από ένα κατεστραμμένο περιστατικό HDFS σε ένα προηγουμένως γνωστό καλό χρονικό σημείο.

9. Οργάνωση Δεδομένων

9.1 Τεμάχια Δεδομένων (Data Blocks)

Οι εφαρμογές που είναι συμβατές με το HDFS είναι εκείνες που ασχολούνται με μεγάλα σύνολα δεδομένων. Αυτές οι εφαρμογές εγγράφουν τα δεδομένα τους μόνο μία φορά, αλλά τα διαβάζουν μία ή περισσότερες φορές (write-once-read-many) και απαιτούν αυτές οι αναγνώσεις να ικανοποιούνται από θέμα ταχυτήτων ροής. Ένα τυπικό μέγεθος block που χρησιμοποιείται από το HDFS έχει χωρητικότητα 64 MB. Έτσι, ένα HDFS αρχείο τεμαχίζεται σε κομμάτια των 64 MB και, ει δυνατόν, κάθε κομμάτι διανέμεται σε διαφορετικό DataNode.

9.2 Staging

Ένα αίτημα για δημιουργία αρχείου δεν φθάνει αμέσως στο NameNode. Στην πραγματικότητα, ο HDFS πελάτης αρχικά αποθηκεύει προσωρινά τα δεδομένα του αρχείου σε ένα τοπικό αρχείο και οι εγγραφές της εφαρμογής ανακατευθύνονται σε αυτό το προσωρινό αρχείο. Όταν το αρχείο συσσωρεύσει δεδομένα μεγέθους πάνω από ένα HDFS block, ο πελάτης ειδοποιεί το NameNode. Αυτός εισάγει το όνομα του αρχείου μέσα στην ιεραρχία του συστήματος αρχείων και διαθέτει ένα block δεδομένων για αυτό. Έπειτα, ανταποκρίνεται στο αίτημα του πελάτη με την ταυτότητα του DataNode και το block δεδομένων προορισμού και τότε ο πελάτης δρομολογεί το block δεδομένων από το τοπικό προσωρινό αρχείο στον καθορισμένο DataNode. Όταν ένα αρχείο είναι κλειστό, τα υπόλοιπα δεδομένα στο προσωρινό τοπικό αρχείο, που δεν έχουν δρομολογηθεί ακόμη, μεταφέρονται και αυτά στο DataNode. Ο πελάτης λέει τότε στο NameNode ότι το αρχείο είναι κλειστό και ο NameNode εκτελεί τη λειτουργία δημιουργίας αρχείων σαν μια επίμονη αποθήκευση (persistent store). Εάν ο NameNode πεθάνει πριν κλείσει το αρχείο, αυτό χάνεται.

Η παραπάνω προσέγγιση έχει υιοθετηθεί μετά από προσεκτική εξέταση των εφαρμογών που τρέχουν στο HDFS, οι οποίες χρειάζονται συνεχείς εγγραφές σε αρχεία. Εάν ένας πελάτης γράφει σε ένα απομακρυσμένο αρχείο απευθείας, χωρίς

κανένα buffering από πλευράς πελάτη (client side buffering), η ταχύτητα του δικτύου και η συμφόρηση σε αυτό επιδρά σημαντικά στην απόδοση. Προγενέστερα καταναμημένα συστήματα αρχείων, όπως το AFS, έχουν χρησιμοποιήσει προσωρινή αποθήκευση από πλευράς πελάτη (client side caching) για τη βελτίωση των επιδόσεων.

9.3 Διασωλήνωση Αντιγραφής (Replication Pipelining)

Όταν ένας πελάτης γράφει δεδομένα σε ένα HDFS αρχείο, τα δεδομένα γράφονται πρώτα σε ένα τοπικό αρχείο, όπως εξηγήθηκε στην προηγούμενη ενότητα. Όταν το τοπικό αρχείο συγκεντρώσει ένα πλήρες block δεδομένων χρήστη, ο πελάτης ανακτά από το NameNode μία λίστα με τους DataNodes που δηλώνονται από τον παράγοντα αντιγραφής (συνήθως τρεις), οι οποίοι θα φιλοξενήσουν από ένα αντίγραφο αυτού του block. Ο πελάτης στη συνέχεια στέλνει το block δεδομένων στον πρώτο DataNode. Αυτός αφού αρχίσει να λαμβάνει τα δεδομένα σε μικρά τμήματα (4 KB), γράφει κάθε τμήμα στο τοπικό χώρο αποθήκευσής του και το μεταφέρει προς το δεύτερο DataNode στη λίστα, ο οποίος με τη σειρά του εκτελεί τα ίδια βήματα και στέλνει τα δεδομένα, τμήμα-τμήμα, στον επόμενο κ.ο.κ.. Τέλος, ο τελευταίος DataNode γράφει απλώς τα δεδομένα στο τοπικό αποθηκευτικό του χώρο. Έτσι, ένας DataNode μπορεί να λαμβάνει δεδομένα από τον προηγούμενο μέσω ενός σωλήνα (pipeline) και ταυτόχρονα να διαβιβάζει δεδομένα στον επόμενο μέσω άλλου σωλήνα. Έτσι, τα δεδομένα διασωληνώνονται από τον ένα DataNode στον επόμενο.

10. Προσβασιμότητα

Το HDFS μπορεί να προσπελαστεί από εφαρμογές με πολλούς διαφορετικούς τρόπους. Εγγενώς, το HDFS παρέχει ένα Java API στις εφαρμογές, ωστόσο υπάρχει και μία τέτοια διεπαφή περικλυόμενη σε C γλώσσα. Επιπλέον, ένα HTTP πρόγραμμα περιήγησης μπορεί επίσης να χρησιμοποιηθεί για την πλοήγηση στα αρχεία του HDFS. Ακόμη, γίνονται εργασίες για πρόσβαση στο HDFS μέσω του πρωτοκόλλου WebDAV.

Βιβλιογραφία

- [1] Data, data everywhere, (2010), <http://www.economist.com/node/15557443>
- [2] Hive: a warehousing solution over a map-reduce framework, Thusoo, A., Sarma, J. S., Jain, N., Shao, Z., Chakka, P., Anthony, S., ... & Murthy, R, (2009), Proceedings of the VLDB Endowment, 2(2), 1626-1629
- [3] Mining the Social Web, 2nd Ed., Matthew A. Russell, (2014), ISBN: 978-1-449-36761-9
- [4] Mining Data From Twitter, Abhishanga Upadhyay, Luis Mao & Malavika Goda Krishna, (2014)
- [5] Twitter Documentation, <https://dev.twitter.com/overview/documentation>
- [6] OAuth 2 in Action, Justin Richer & Antonio Sanso, (2015), Manning Publications
- [7] Apache NiFi Overview, Apache NiFi Team, (2014), <https://nifi.apache.org/docs/nifi-docs/html/overview.html>
- [8] What is Apache Hadoop?, (2012), <https://www.oreilly.com/ideas/what-is-apache-hadoop>
- [9] Hadoop, http://www.sas.com/en_us/insights/big-data/hadoop.html
- [10] Welcome to Apache™ Hadoop®, (2014), <https://hadoop.apache.org/>
- [11] Hadoop: What it is, how it works, and what it can do, (2011), <https://www.oreilly.com/ideas/what-is-hadoop>
- [12] Apache Hadoop, <http://hortonworks.com/hadoop/>
- [13] Apache Hadoop, (2011), https://en.wikipedia.org/wiki/Apache_Hadoop
- [14] Apache Hadoop Basics, Hortonworks Inc., (2013)
- [15] Hadoop: The Definitive Guide, Tom White, (2015)

- [16] MapReduce, <https://en.wikipedia.org/wiki/MapReduce>
- [17] Apache Hadoop NextGen MapReduce (YARN), (2015),
<https://hadoop.apache.org/docs/r2.7.1/hadoop-yarn/hadoop-yarn-site/YARN.html>
- [18] Hadoop in Action, Chuck Lam, (2011), ISBN: 9781935182191
- [19] Data Mining: Concepts and Techniques, 2nd Ed., Jiawei Han & Micheline Kamber, (2006), ISBN 13: 978-1-55860-901-3
- [20] HDFS Architecture Guide, Dhruba Borthakur (2008)

