ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Π.Μ.Σ. Πληροφοριακά Συστήματα και Υπηρεσίες
Ειδίκευση Μεγάλα Δεδομένα και Αναλυτική

Μελέτη νέων τεχνολογιών για τη διαχείριση μεγάλου όγκου ροών δεδομένων

Καπότης Χρήστος

Επιβλέπων Καθηγητής:

Χαλκίδη Μαρία

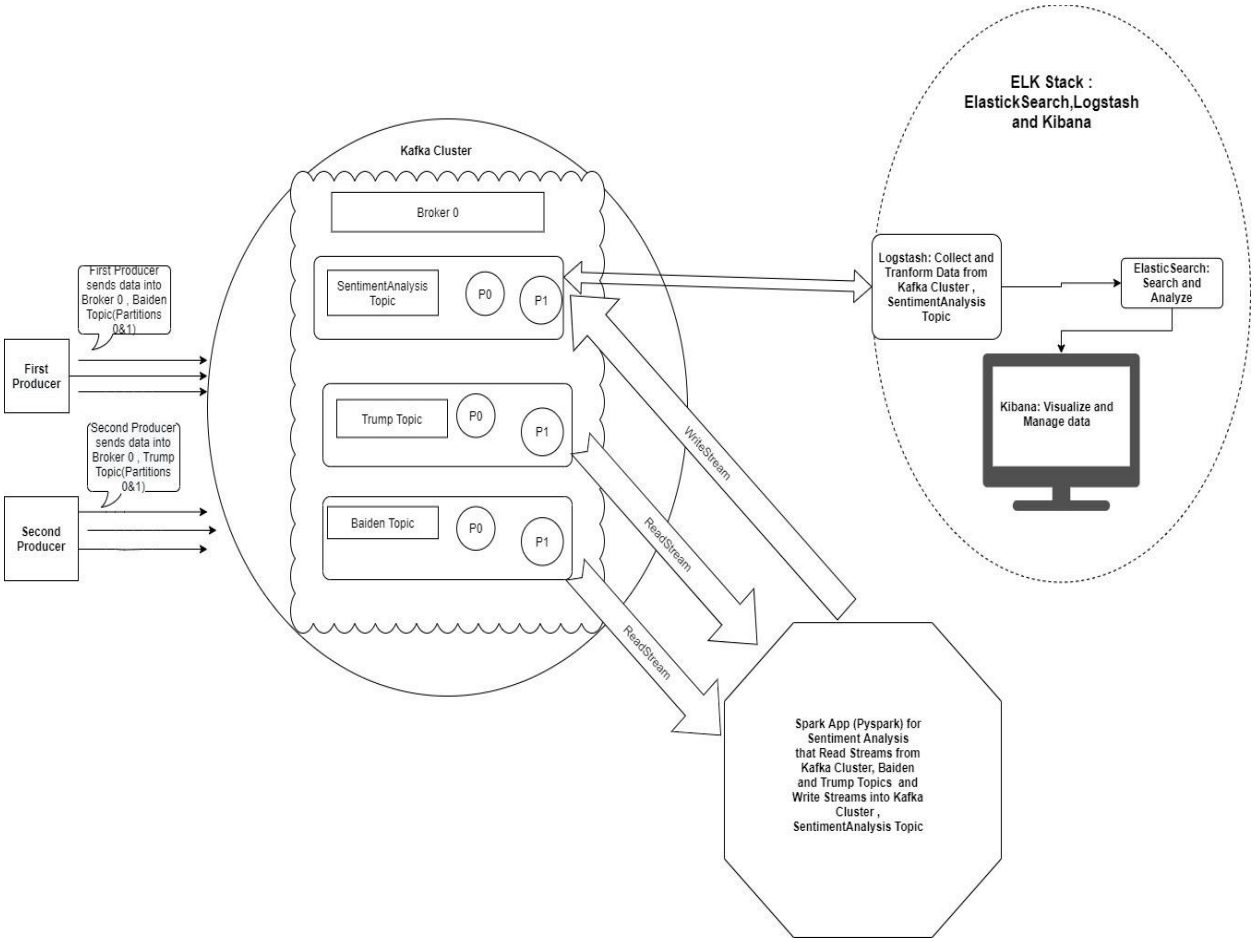
Φεβρουάριος 2021

Ευχαριστίες

Στην οικογένειά μου, σε όσους με στήριξαν να ολοκληρώσω τις σπουδές μου και στη καθηγήτριά μου τη κ. Χαλκίδη.

Περιεχόμενα

Αρχιτεκτονική Συστήματος



Σχήμα 1. Αρχιτεκτονική Συστήματος
και να αναφέρεσαι σε αυτό όπου χρειάζεται μέσα στο κείμενο

Abstract

This paper implements a realtime pipeline, which depicts the momentum of the two candidates for the US presidency, using some of the most popular big data technologies such as Apache Spark, Streaming, Kafka and the ELK Stack (Elasticsearch, Logstash and Kibana).

More specifically, as shown in the picture 1, two python producers have been implemented who generate fake proposals and send them to our Kafka Cluster. Then our Spark infrastructure reads in the form of a stream from the Kafka Cluster the proposals concerning the 2 candidates and implements sentiment analysis to determine if the proposals are positive, negative or neutral. Once the sentiment analysis is implemented and after we have all the data we need and how we need it, Spark writes the results to the Kafka Cluster. Finally, with the help of Logstash we transfer our data from the Kafka Cluster to Elasticsearch with a final destination in Kibana to visualize the results by creating the appropriate diagrams.

Περίληψη

Στην παρούσα εργασία υλοποιείται ένα realtime (πραγματικού χρόνου) pipeline, το οποίο απεικονίζει κάθε στιγμή τη δημοτικότητα των δύο υποψηφίων για την προεδρία της Αμερικής, χρησιμοποιώντας κάποιες από τις πιο δημοφιλή big data τεχνολογίες όπως Apache Spark, Streaming, Kafka και το ELK Stack(Elasticsearch, Logstash και Kibana).

Πιο συγκεκριμένα, όπως φαίνεται και στο Σχήμα 1*, έχουν υλοποιηθεί δύο python producers οι οποίοι γεννούν ψεύτικες προτάσεις και τις στέλνουν στο Kafka Cluster μας. Εν συνεχεία η Spark υποδομή μας διαβάζει σε μορφή stream από το Kafka Cluster τις προτάσεις που αφορούν τους 2 υποψηφίους και υλοποιεί sentiment analysis ώστε να διαπιστωθεί εάν οι προτάσεις είναι θετικές, αρνητικές ή ουδέτερες. Μόλις υλοποιηθεί και το sentiment analysis και αφού έχουμε όλα τα δεδομένα που χρειαζόμαστε και όπως τα χρειαζόμαστε το Spark γράφει τα αποτελέσματα στο Kafka Cluster. Τέλος, με τη βοήθεια του Logstash μεταφέρουμε τα δεδομένα μας από το Kafka Cluster στο Elasticsearch με τελικό προορισμό το Kibana ώστε να οπτικοποιήσουμε τα αποτελεσματα δημιουργώντας τα κατάλληλα διαγράμματα.*

Κεφάλαιο 1: Εισαγωγή στην Ανάλυση Μεγάλων Δεδομένων

Η δημιουργία ενός συστήματος όπως αυτό που υλοποιήθηκε σε αυτή την εργασία, σκοπό έχει να παρουσιάσει στην πράξη και σε πραγματικό πρόβλημα, τις πλέον διαδεδομένες τεχνολογίες Μεγάλων Δεδομένων που χρησιμοποιούνται για Εξαγωγή, Επεξεργασία, Γράψιμο και τέλος Οπτικοποίησης των δεδομένων. Πιο συγκεκριμένα αυτό που κάνει η παρούσα εργασία είναι συλλέγοντας προτάσεις που αφορούν τους 2 υποψηφίους των Αμερικανικών εκλογών, να υπάρχει συνεχής ενημέρωση για την δημοφιλία και των δύο, για κάθε χρονική στιγμή.

Στην συνέχεια, θα περιγραφούν σε βάθος οι δυνατότητες της υλοποίησης μας, ενώ θα γίνει αναφορά στις τεχνολογίες που χρησιμοποιήθηκαν. Για αρχή να πούμε ότι χρησιμοποιήθηκαν για την υλοποίηση η γλώσσα προγραμματισμού Python καθώς επίσης και τεχνολογίες Μεγάλων Δεδομένων όπως το Kafka, το Spark και το ELK Stack. Συνοπτικά, αυτό που συμβίνει είναι ότι με την βοήθεια της Python δημιουργούμε προτάσεις οι οποίες καταλήγουν στο Kafka. Εν συνεχεία, η Spark υποδομή μας ακούει σε μορφή streaming τα δεδομένα από το Kafka και κάνοντας sentiment analysis καταλήγει στο εάν οι προτάσεις είναι θετικές αρνητικές ή ουδέτερες και γράφει τα αποτελέσματα στο Kafka. Αφού έχει ολοκληρωθεί η παραπάνω διαδικασία το Logstash (τεχνολογία του ELK Stack) διαβάζει τα δεδομένα από το Kafka και τα μεταφέρει στο Elasticsearch (τεχνολογία του ELK Stack) για να τα οπτικοποιήσουμε στο τέλος στο Kibana (τεχνολογία του ELK Stack).

Τέλος, ιδιαίτερη σημασία έχει δοθεί στην απόδοση των επιμέρους τεχνολογιών που χρησιμοποιούμε όπως και στην ορθότητα και ασφάλεια των δεδομένων για τα οποία έχουν γίνει μετρήσεις και παρουσιάζονται στη συνέχεια.

1.1 Στόχοι της Εργασίας

Το θέμα της εργασίας, είναι η μελέτη στη συμπεριφορά των τεχνολογιών που χρησιμοποιούνται για τη διαχείρηση μεγάλου όγκου ροών δεδομένων. Οπότε, για να το επιτύχουμε αυτό προχωρήσαμε στην επιμέρους παρατήρηση της κάθε τεχνολογίας στη διαχείρηση διαφορετικών πλήθος δεδομένων.

Πιο συγκεκριμένα, δημιουργούσαμε με διαφορετικές συχνότητες δεδομένα για να δούμε σε πρώτη φάση πως συμπεριφέρετε το Kafka, δηλαδή αν η διαδρομή των δεδομένων προς το Spark είναι ομαλή ή υπαρχουν καθυστερήσεις και ακόμα χειρότερα απώλεια δεδομένων, το οποίο είναι ίσως ο σημαντικότερος στόχος σε τέτοιου είδους εφαρμογές. Στη συνέχεια στόχος της εργασίας ήταν να επιτύχουμε όσο το δυνατόν περισσότερη ισοκατανομή των δεδομένων μας στο Spark ώστε να ετιτύχουμε τη μεγαλύτερη απόδοση και αυτού του συστήματος. Τέλος, στο Elasticearch ο στόχος ήταν να μην υπάρχει καθυστέρη στην ανανέωση των δεδομέων και φυσικά να μην υπάρχουν απώλειες στα δεδομένα μας.

1.2: Μεγάλα Δεδομένα

Στο παρόν κεφάλαιο υπάρχει μια γενικότερη περιγραφή των Μεγάλων Δεδομένων, πως προέκυψαν με τέτοια μεγάλη αύξηση τα τελευταία χρόνια, που χρησιμοποιούνται και σε τι έχουν βοηθήσει. Επίσης αναφερόμαστε και στη χρησιμότητα του Streaming, το οποίο χρησιμοποιείται κατά κόρον στα Μεγάλα δεδομένα.

Η επιστήμη των Big Data ή όπως θα λέγαμε στα ελληνικά των «Μεγάλων Δεδομένων», γνώρισε τη δημοσιότητα, και από το ευρύτερο κοινό, κυρίως τα τελευταία χρόνια, καθώς κυριάρχησε στον παγκόσμιο ιστό, βρίσκοντας μια υψηλή θέση στις προτιμήσεις των εταιρειών και οργανισμών. Ο ορός είναι πλέον τόσο διαδεδομένος, ώστε ακόμη και σε θεωρητικούς κλάδους να γίνονται σχετικές συζητήσεις, αφού δεν είναι λίγοι αυτοί που υπογραμμίζουν ότι πρόκειται για την κατ' εξοχήν μελλοντική επιστήμη[9].

Εικόνα 1. Μεγάλα Δεδομένα[1]

Αρχικά, ως Big Data ορίζεται ο τεράστιος όγκος δεδομένων. Ο όγκος αυτός των δεδομένων είναι τόσο περίπλοκος και αναπτύσσει τόσο μεγάλες ταχύτητες, ώστε είναι αδύνατον να επεξεργαστεί μέσω παραδοσιακών μεθόδων. Τα «Μεγάλα Δεδομένα», λοιπόν, προσδιορίζονται από 6 πολύ βασικά χαρακτηριστικά- στα αγγλικά τα 6 'V's, Volume, Velocity, Variety, Viability, Veracity και Value (εικόνα 2). Το πρώτο είναι ο όγκος δεδομένων, μιλώντας πάντοτε για το μέγεθος κι όχι ως προς το δείγμα, αλλά ως προς το σύνολο, π.χ. των δημοσιεύσεων των χρηστών του facebook. Δεύτερο είναι η ταχύτητα, αυτή με την οποία τα δεδομένα αυτά παράγονται κι επεξεργάζονται στο αυξανόμενων απαιτήσεων περιβάλλον -την real time ανάλυσή τους, με άλλα λόγια. Στη συνέχεια, ακολουθεί η ποικιλία, από ένα γραπτό κείμενο ή μήνυμα έως ένα φωνητικό, μια φωτογραφία, ένα βίντεο, σε ζωντανή μετάδοση ή και μαγνητοσκοπημένο. Η δυνατότητα του κάθε χρήστη ατομικά να παρακολουθήσει σε παγκόσμιο επίπεδο έναν άλλον χρήστη «ζωντανά». Τέταρτον, σημαντική είναι η μεταβλητότητα, η συνεχής αλλαγή τόσο των δεδομένων όσο και του νοήματός τους. Η αποκρυπτογράφηση των συναισθημάτων των χρηστών, των οποίων η διατύπωση γίνεται με διάφορους τρόπους στα μέσα κοινωνικής δικτύωσης αποτελεί τη νέα μεγαλύτερη πρόκληση για τον αυτόματο εντοπισμό. Φυσικά, σημαντικό ρόλο διαδραματίζει και η ακρίβεια. Τα δεδομένα αυτά υπάρχουν με σκοπό την εξαγωγή συμπερασμάτων ακρίβειας για τη συμπεριφορά ενός χρήστη, λ.χ. μια επικείμενη αγορά του. Η ποιότητά τους σαφώς επηρεάζει την ανάλυση, η οποία με τη σειρά της επηρεάζεται από τα ανακριβή ή ακριβή στοιχεία. Τέλος, και σπουδαιότερο όλων, είναι η αξία τους. Τα δεδομένα αυτά δε μπορούν να έχουν αξία, εάν δε μπορούμε να τα «μετατρέψουμε» εμείς οι ίδιοι σε αξία. Εν ολίγοις, η αξία τους δεν αφορά την ποσότητα των δεδομένων αλλά πως αξιοποιείται αυτή η όποια ποσότητα[9].

[1] http://ecmetrics.com/big-data-analytic-tools-market-research/

Για παράδειγμα, με δεδομένα από πολλές πηγές μπορούν να δοθούν απαντήσεις ως προς τη μείωση κόστους για μια εταιρία, η μείωση κόστους παραγωγής, ή γρηγορότερη παραγωγή προϊόντων ή ακόμη και σοφότερη επιλογή προϊόντων.

Εικόνα 2. Βασικά χαρακτηριστικά των Μεγάλων Δεδομένων[2]

Συνοπτικά, τα «Μεγάλα Δεδομένα» μπορούν να αξιοποιηθούν από τις εταιρίες, ώστε να καθορίζουν την πηγή της αποτυχίας, των ανωμαλιών και λοιπών ζητημάτων πριν από τον πραγματικό χρόνο, να υπολογίζουν τους κινδύνους των εγχειρημάτων, να εντοπίζουν τις λανθάνουσες συμπεριφορές πριν αυτές δημιουργήσουν προβλήματα και να βοηθήσουν στην πρόβλεψη των αγοραστικών συνηθειών των πελατών. Για τους υπόλοιπους απλούς καθημερινούς ανθρώπους, τα οφέλη τα οποία ήδη αρκετοί απολαμβάνουν αφορούν στην απλούστευση των δραστηριοτήτων μας, όπως για παράδειγμα ο υπολογισμός των παλμών και των βημάτων μας μέσω.

Μερικά παραδείγματα «Μεγάλων Δεδομένων» είναι το social media listening, δηλαδή η διαδικασία συλλογής πληροφοριών από τους χρήστες ως προς το τι λέγεται για διάφορα προϊόντα, μέσω παρατήρησης της δραστηριότητας των χρηστών. Ένα ακόμη θα ήταν το marketing analysis, οι πληροφορίες που χρησιμοποιούνται για προώθηση νέων προϊόντων, υπηρεσιών και πρωτοβουλιών, φυσικά πιο εμπλουτισμένων. Πολύ σημαντικό παράδειγμα είναι και το customer satisfaction and sentiment analysis, η διαδικασία με την οποία όλες οι συλλεγμένες πληροφορίες από πολλές διαφορετικές πηγές υποδεικνύουν πως νιώθει ο κάθε χρήστης ή πελάτης για μια εταιρία, μια επωνυμία ή ένα προϊόν.

[2] https://www.ijettcs.org/Volume6Issue4/IJETTCS-2017-07-14-17.pdf

Για την αποθήκευση των Big Data, πράγμα το οποίο αποτελεί και μεγάλη πρόκληση, λόγω της απαίτησης κεφάλαιου, καθώς η επένδυση σε έναν μεγάλης έκτασης server, θα μπορούσε να αποβεί επιζήμια για έναν ιδιοκτήτη, χρησιμοποιείται ως επί το πλείστον το cloud. Το cloud προσφέρει, επίσης, δυνατότητα αποθήκευσης των «Μεγάλων Δεδομένων» σε managed services όπως Amazon EMR(formerly Elastic MapReduce), Microsoft Azure και Google Cloud Dataproc.

Σε περιβάλλοντα cloud, τα δεδομένα αυτά υπάρχει η δυνατότητα να αποθηκευτούν σε Hadoop Distributed File System, relational databases, Amazon Simple Storage Space και NoSQL databases.

Εν κατακλείδι, κατανοητό είναι αφ' ενός πως τα «Μεγάλα Δεδομένα» αποτελούν το μέλλον αφού ο όλο και μεγαλύτερος όγκος δεδομένων στο διαδίκτυο λόγω της αύξησης χρηστών καθιστά αναγκαίο να «διαβάζονται» οι συμπεριφορές των, ιδίως από εταιρίες εμπορευμάτων, ώστε το μάρκετινγκ και η διαφήμιση να είναι πιο στοχευμένα και το κέρδος σαφώς μεγαλύτερο. Αφ'ετέρου, η συλλογή και πόσο μάλλον η τεράστια συλλογή ογκωδών δεδομένων από διάφορες εταιρίες εγκυμονεί πάσης φύσεως κινδύνους καθότι ενδυναμώνει τη διαδικτυακή παρακολούθηση των χρηστών, τους μετατρέπει σε καταναλωτικά υποχείρια και προμηνύει ένα μέλλον, στο οποίο όλα θα είναι ελεγχόμενα «άνωθεν» παρά από τα ίδια τα άτομα.

1.2.1 Τεχνικές Ανάλυσης Μεγάλων Δεδομένων

Λόγω της ταχύτατης ανάπτυξης του όγκου των δεδομένων, αυξάνεται και η ανάγκη για χρήση κατάλληλων τεχνικών, εργαλείων και αλγορίθμων που υπάρχουν για τη διαχείρηση και την αξιοποίησή τους. Η συγκεκριμένη ανάγκη έχει οδηγήσει στην ανάπτυξη τεχνολογιών μέσα από τις τεχνολογίες που ήδη υπάρχουν, για συλλογή, επεξεργασία και οπτικοποίηση των Μεγάλων Δεδομένων[9]. Τέτοιες τεχνολογίες είναι:

- Βελτιστοποίηση (Optimization)
- Στατιστική (Statistics)
- Εξόρυξη Δεδομένων (Data Mining)
- Μηχανική Μάθηση (MachineLearning)
- Τεχνικές Οπτικοποίησης (Visualization Approaches)
- Ανάλυση Δικτύου (Network Analysis)
- Σημασιολογική Ανάλυση (Semantic Analysis)

1.2.1.1 Βελτιστοποίηση

Υπάρχει η ανάγκη βελτιστοποίησης των μεγάλων δεδομένων για τη διαχείρισή τους με τρόπο που βελτιώνει την ποιότητά τους, επιταχύνει τη λήψη αποφάσεων, εκμεταλλεύεται τις νέες αναλυτικές δυνατότητες και βελτιστοποιεί τις επιχειρηματικές διαδικασίες μαζί με τη μείωση του συνολικού κόστους που σχετίζεται με μια παραδοσιακή αποθήκη δεδομένων. Για να επιτευχθεί αυτή η βελτιστοποίηση χρειάζεται ένα σύστημα με:

Επεκτασιμότητα

Το σύστημα θα πρέπει να είναι εύκολα επεκτάσιμο και η επέκταση του συστήματος δεν θα πρέπει να επηρεάζει το υπάρχον σύστημα.

Ανοχή σε σφάλματα

Το σύστημα θα πρέπει να είναι ικανό να αντιμετωπίσει καταστάσεις όπως πρόβλημα σε ένα μέρος του συστήματος χωρίς σημαντικά αποτελέσματα.

Κατανομή δεδομένων

Η κατανομή των δεδομένων θα πρέπει να γίνεται με τέτοιο τρόπο ώστε το ίδιο μηχάνημα να επεξεργάζεται τα δεδομένα όπου είναι αποθηκευμένα. Εάν η αποθήκευση και η επεξεργασία δεδομένων συμβαίνουν σε διαφορετικά μηχανήματα, θα χρειαστεί επιπλέον κόστος και χρόνος για τη μετάδοση δεδομένων.

1.2.1.2 Στατιστική

Είναι η επιστήμη που συλλέγει, οργανώνει και ερμηνεύει δεδομένα. Στατιστικές τεχνικές χρησιμοποιούνται συχνά για να κάνουν εκτιμήσεις για το τι σχέσεις ανάμεσα στις μεταβλητές θα μπορούσαν να είχαν συμβεί κατά τύχη και για το τι σχέσεις θα μπορούσαν να είναι αποτέλεσμα από κάποια υποκείμενη αιτιώδη σχέση. Ωστόσο, οι πρότυπες στατιστικές τεχνικές συνήθως δεν είναι κατάλληλες για τη διαχείρηση των Μεγάλων Δεδομένων και έτσι έχουν υλοποιηθεί επεκτάσεις στις ήδη υπάρχουσες τεχνικές και σε καποιες περιπτώσεις έχουν υλοποιηθεί εντελώς καινούργιες.

1.2.1.3 Εξόρυξη Δεδομένων

Γενικά, ο στόχος της εξόρυξης δεδομένων είναι είτε η ταξινόμηση είτε η πρόβλεψη. Στην ταξινόμηση, η ιδέα είναι να ταξινομηθούν τα δεδομένα σε ομάδες. Οπότε περιορίζεται η μελέτη των μεγάλων δεδομένων σε υποσύνολα και γίνεται πιο εύκολη. Τεχνικές εξόρυξης δεδομένων είναι:

Δέντρα ταξινόμησης (Classification trees)

Μια δημοφιλής τεχνική εξόρυξης δεδομένων που χρησιμοποιείται για την ταξινόμηση μιας εξαρτημένης κατηγορικής μεταβλητής με βάση τις μετρήσεις μιας ή περισσότερων μεταβλητών πρόβλεψης. Το αποτέλεσμα είναι ένα δέντρο με κόμβους και συνδέσμους μεταξύ των κόμβων που μπορεί να διαβαστεί για να σχηματίσει κανόνες if-then.

Λογιστική παλινδρόμηση (Logistic regression)

Μια στατιστική τεχνική που είναι μια παραλλαγή της τυπικής παλινδρόμησης αλλά επεκτείνει την έννοια για να ασχοληθεί με την ταξινόμηση. Παράγει έναν τύπο που προβλέπει την πιθανότητα εμφάνισης ως συνάρτηση των ανεξάρτητων μεταβλητών.

Νευρωνικά δίκτυα (Neural networks)

Ένας αλγόριθμος λογισμικού που διαμορφώνεται σύμφωνα με την παράλληλη αρχιτεκτονική των ζωικών εγκεφάλων. Το δίκτυο αποτελείται από κόμβους εισόδου, κρυμμένα επίπεδα και κόμβους εξόδου. Κάθε μονάδα έχει ένα βάρος. Τα δεδομένα δίδονται στον κόμβο εισόδου και από ένα σύστημα δοκιμής και σφάλματος, ο αλγόριθμος προσαρμόζει τα βάρη έως ότου πληροί συγκεκριμένα κριτήρια διακοπής. Μερικοί άνθρωποι το έχουν παρομοιάσει με μια προσέγγιση μαύρου κουτιού.

Τεχνικές ομαδοποίησης όπως K-nearest (Clustering techniques like K-nearest)

Μια τεχνική που προσδιορίζει ομάδες παρόμοιων εγγραφών. Η τεχνική Κ-πλησιέστερου γείτονα υπολογίζει τις αποστάσεις μεταξύ της εγγραφής και των σημείων στα ιστορικά (εκπαιδευτικά) δεδομένα. Στη συνέχεια εκχωρεί αυτήν την εγγραφή στην κατηγορία του πλησιέστερου γείτονα σε ένα σύνολο δεδομένων.

1.2.1.4 Μηχανική Μάθηση

Παρέχει προβλέψεις που θα ήταν αδύνατες για τους ανθρώπινους αναλυτές σε τόσο μεγάλο όγκο δεδομένων.

1.2.1.5 Τεχνικές Οπτικοποίησης

Η οπτικοποίηση στα Μεγάλα Δεδομένα είναι ένα από τα σημαντικότερα συστατικά. Μόλις η ροή των ακατέργαστων δεδομένων αντιπροσωπεύεται με εικόνες, η λήψη αποφάσεων γίνεται πολύ πιο εύκολη. Τα εργαλεία για την οπτικοποίηση των Big Data προκειμένου να ανταποκριθούν θα πρέπει να παρέχουν ένα συγκεκριμένο σύνολο χαρακτηριστικών όπως:

- Δυνατότητα επεξεργασίας πολλαπλών τύπων εισερχόμενων δεδομένων
- Δυνατότητα εφαρμογής διαφόρων φίλτρων για προσαρμογή των αποτελεσμάτων
- Δυνατότητα αλληλεπίδρασης με τα σύνολα δεδομένων κατά την ανάλυση
- Δυνατότητα σύνδεσης με άλλο λογισμικό για λήψη εισερχόμενων δεδομένων ή παροχή εισόδου για αυτά
- Δυνατότητα παροχής επιλογών συνεργασίας για τους χρήστες

1.2.1.6 Ανάλυση Δικτύου

Αποτελεί ένα σύνολο τεχνικών που χρησιμοποιούνται για να περιγράψουν σχέσεις και συνδέσεις μεταξύ των κόμβων ενός δικτύου ή γραφίματος. Η πιο γνωστή τεχνική εδώ είναι η ανάλυσηκοινωνικών δικτύων (Social Network Analysis SNA).

1.2.1.7 Σημασιολογική Ανάλυση

Η σημασιολογική ανάλυση είναι η διαδικασία σύλληψης νοήματος από το κείμενο. Επιτρέπει στους υπολογιστές να κατανοούν και να ερμηνεύουν προτάσεις, παραγράφους ή ολόκληρα έγγραφα, αναλύοντας τη γραμματική τους δομή και προσδιορίζοντας τις σχέσεις μεταξύ μεμονωμένων λέξεων σε ένα συγκεκριμένο πλαίσιο.

Είναι ένα ουσιαστικό δευτερεύον καθήκον της Επεξεργασίας Φυσικής Γλώσσας (NLP) και η κινητήρια δύναμη πίσω από εργαλεία μηχανικής εκμάθησης όπως chatbots, μηχανές αναζήτησης και ανάλυση κειμένου.

Τα εργαλεία σημασιολογικής ανάλυσης μπορούν να βοηθήσουν τις εταιρείες να εξαγάγουν αυτόματα σημαντικές πληροφορίες από μη δομημένα δεδομένα, όπως email, εισιτήρια υποστήριξης και σχόλια πελατών.

Τεχνικές σημασιολογικής ανάλυσης

Ανάλογα με τον τύπο των πληροφοριών που θέλουμε να λάβουμε από τα δεδομένα, μπορούμε να χρησιμοποιήσουμε μία από τις δύο τεχνικές σημασιολογικής ανάλυσης, ένα μοντέλο ταξινόμησης κειμένου (το οποίο εκχωρεί προκαθορισμένες κατηγορίες στο κείμενο) ή ένα εργαλείο εξαγωγής κειμένου (το οποίο βγάζει συγκεκριμένες πληροφορίες από το κείμενο).

Σημασιολογικά μοντέλα ταξινόμησης

Ταξινόμηση θέματος (Topic Classification)

Ταξινόμηση κειμένου σε προκαθορισμένες κατηγορίες με βάση το περιεχόμενό του.

Ανάλυση συναισθημάτων (Sentiment Analysis)

Ανίχνευση θετικών, αρνητικών ή ουδέτερων συναισθημάτων σε ένα κείμενο που υποδηλώνει τον χαρακτήρα.

Ταξινόμηση πρόθεσης (Intent Classification)

Ταξινόμηση κειμένου βάσει του τι θα γίνει στη συνέχεια

1.3 Streaming

Το streaming (ροή συμβάντων) είναι το ψηφιακό ισοδύναμο του κεντρικού νευρικού συστήματος του ανθρώπινου σώματος.

Από τεχνική άποψη, η ροή συμβάντων είναι η πρακτική της καταγραφής δεδομένων σε πραγματικό χρόνο από πηγές συμβάντων όπως βάσεις δεδομένων, αισθητήρες, κινητές συσκευές, υπηρεσίες cloud και εφαρμογές λογισμικού με τη μορφή ροών συμβάντων, αποθήκευση αυτών των ροών συμβάντων για μελλοντική ανάκτηση, επεξεργασία και αντίδραση στις ροές συμβάντων σε πραγματικό χρόνο καθώς και αναδρομικά και δρομολόγηση ροών συμβάντων σε διαφορετικές τεχνολογίες προορισμού. Συνεπώς, η ροή συμβάντων εξασφαλίζει συνεχή ροή και ερμηνεία των δεδομένων έτσι ώστε οι σωστές πληροφορίες να βρίσκονται στο σωστό μέρος, την κατάλληλη στιγμή[9].

1.4 Γιατί χρησιμοποιείται το streaming

Η ροή συμβάντων εφαρμόζεται σε μια μεγάλη ποικιλία περιπτώσεων, όπως:

- Για την επεξεργασία πληρωμών και χρηματοοικονομικών συναλλαγών σε πραγματικό χρόνο, όπως σε χρηματιστήρια, τράπεζες και ασφάλειες.
- Για παρακολούθηση σε αυτοκίνητα, φορτηγά, στόλους και αποστολές σε πραγματικό χρόνο, όπως η εφοδιαστική και η αυτοκινητοβιομηχανία.
- Για συνεχή λήψη και ανάλυση δεδομένων αισθητήρων από συσκευές IoT ή άλλο εξοπλισμό, όπως σε εργοστάσια και αιολικά πάρκα.
- Για συλλογή και αντίδραση άμεση σε αλληλεπιδράσεις και παραγγελίες πελατών, όπως σε καταστήματα λιανικής, στον κλάδο ξενοδοχείων και ταξιδιών, καθώς και σε εφαρμογές για κινητά.
- Παρακολούθηση ασθενών σε νοσοκομειακή περίθαλψη και πρόβλεψη αλλαγών στην κατάσταση ώστε να διασφαλιστεί έγκαιρη θεραπεία σε καταστάσεις έκτακτης ανάγκης
- Για σύνδεση, αποθήκευση και διάθεση δεδομένων που παράγονται από διαφορετικά τμήματα μιας εταιρείας.
- Χρησιμεύσει ως βάση για πλατφόρμες δεδομένων.

Κεφάλαιο 2: Τεχνολογίες Μεγάλων Δεδομένων

Στο παρόν κεφάλαιο περιγράφονται αναλυτικά οι τεχνολογίες που χρησιμοποιούμε για την υλοποίηση της εργασίας μας, ώστε να γίνει σαφές ο λόγος χρησιμοποίησής τους και γιατί οι συγκεκριμένες τεχνολογίες είναι ευρέως διαδεδομένες στον κόσμο των Μεγάλων Δεδομένων. Οι τεχνολογίες που περιγράφονται είναι:

- Το Spark
- Το Kafka
- Το ELK Stack (Elasticsearch, Logstash, Kibana)
- Το Docker

2.1 Apache Spark

Το Apache Spark είναι ένα πλαίσιο επεξεργασίας δεδομένων που μπορεί να εκτελεί γρήγορα εργασίες επεξεργασίας σε πολύ μεγάλα σύνολα δεδομένων και μπορεί επίσης να διανείμει εργασίες επεξεργασίας δεδομένων σε πολλούς υπολογιστές, είτε μόνος του είτε σε συνδυασμό με άλλα κατανεμημένα υπολογιστικά εργαλεία. Αυτές οι δύο ιδιότητες είναι το κλειδί για το κόσμο των μεγάλων δεδομένων και της μηχανικής μάθησης, οι οποίες απαιτούν τη συγκέντρωση τεράστιας υπολογιστικής ισχύος για να προσπελάσουν τα μεγάλα δεδομένων. Το Spark παίρνει επίσης μερικά από τα βάρη προγραμματισμού αυτών των εργασιών από τους ώμους των προγραμματιστών με ένα εύχρηστο API που αφαιρεί μεγάλο μέρος της βαρύτητας του κατανεμημένου υπολογιστή και της μεγάλης επεξεργασίας δεδομένων[12].

Από τις αρχές του στο AMPLab στο U.C. Μπέρκλεϊ το 2009, το Apache Spark έχει γίνει ένα από τα βασικά μεγάλα διανεμημένα πλαίσια επεξεργασίας δεδομένων στον κόσμο. Το Spark μπορεί να αναπτυχθεί με διάφορους τρόπους, παρέχει εγγενείς συνδέσεις για τις γλώσσες προγραμματισμού Java, Scala, Python και R και υποστηρίζει SQL, ροή δεδομένων, μηχανική εκμάθηση και επεξεργασία γραφημάτων. Χρησιμοποιείται από τράπεζες, εταιρείες τηλεπικοινωνιών, εταιρείες παιχνιδιών, κυβερνήσεις και όλους τους μεγάλους τεχνολογικούς οργανισμούς όπως η Apple, το Facebook, η IBM και η Microsoft.

2.1.1 Αρχιτεκτονική Apache Spark

Σε θεμελιώδες επίπεδο, μια εφαρμογή Apache Spark αποτελείται από δύο βασικά στοιχεία: ένα πρόγραμμα οδήγησης, το οποίο μετατρέπει τον κώδικα του χρήστη σε πολλαπλές εργασίες που μπορούν να διανεμηθούν σε κόμβους εργαζομένων (workers) και εκτελεστές (masters), οι οποίοι εκτελούνται σε αυτούς τους κόμβους και εκτελούν τις εργασίες που τους έχουν ανατεθεί.

Το Spark μπορεί να εκτελεστεί σε αυτόνομη λειτουργία που απαιτεί απλώς το πλαίσιο Apache Spark και ένα Java Virtual Machine(JVM) σε κάθε υπολογιστή του συμπλέγματος σας (cluster). Ωστόσο, είναι πιο πιθανό να χρειάζεται ένα πιο ισχυρό σύστημα διαχείρισης πόρων ή συστάδων για να φροντίζεται η κατανομή[12].

2.2.2 Spark Core

Το Spark Core είναι ο βασικός κινητήρας για παράλληλη και κατανεμημένη επεξεργασία δεδομένων μεγάλης κλίμακας.

Είναι υπεύθυνο για:

- διαχείριση μνήμης και αποκατάσταση σφαλμάτων
- προγραμματισμός, διανομή και παρακολούθηση εργασιών σε ένα σύμπλεγμα
- αλληλεπιδρά με συστήματα αποθήκευσης

2.1.3 RDD Ανθεκτικά κατανεμημένα σύνολα δεδομένων

Τα Resilient Distributed Datasets (RDD) είναι μια βασική δομή δεδομένων του Spark. Είναι μια αμετάβλητη κατανεμημένη συλλογή αντικειμένων. Κάθε σύνολο δεδομένων στο RDD χωρίζεται σε λογικά διαμερίσματα, τα οποία μπορούν να υπολογιστούν σε διαφορετικούς κόμβους του συμπλέγματος. Τα RDD μπορούν να περιέχουν οποιονδήποτε τύπο αντικειμένων Python, Java ή Scala, συμπεριλαμβανομένων κατηγοριών που καθορίζονται από το χρήστη. Επισήμως, ένα RDD είναι μια συλλογή εγγραφών μόνο για ανάγνωση, κατατμημένη. Τα RDD μπορούν να δημιουργηθούν μέσω ντετερμινιστικών λειτουργιών είτε σε δεδομένα σταθερής αποθήκευσης είτε σε άλλα RDD. Το RDD είναι μια ανθεκτική σε σφάλματα συλλογή στοιχείων που μπορούν να λειτουργήσουν παράλληλα. Υπάρχουν δύο τρόποι για να δημιουργήσετε RDD - παραλληλίζοντας μια υπάρχουσα συλλογή στο πρόγραμμα οδήγησης, ή παραπέμποντας ένα σύνολο δεδομένων σε ένα εξωτερικό σύστημα αποθήκευσης,

όπως ένα κοινόχρηστο σύστημα αρχείων, HDFS, HBase ή οποιαδήποτε πηγή δεδομένων που προσφέρει μια μορφή εισαγωγής Hadoop. Το Spark χρησιμοποιεί την έννοια του RDD για να επιτύχει ταχύτερες και αποδοτικότερες λειτουργίες MapReduce. Ας συζητήσουμε πρώτα πώς πραγματοποιούνται οι λειτουργίες του MapReduce και γιατί δεν είναι τόσο αποτελεσματικές[11].

Η κοινή χρήση δεδομένων είναι αργή στο MapReduce

Το MapReduce υιοθετείται ευρέως για την επεξεργασία και τη δημιουργία μεγάλων συνόλων δεδομένων με έναν παράλληλο, κατανεμημένο αλγόριθμο σε ένα σύμπλεγμα. .

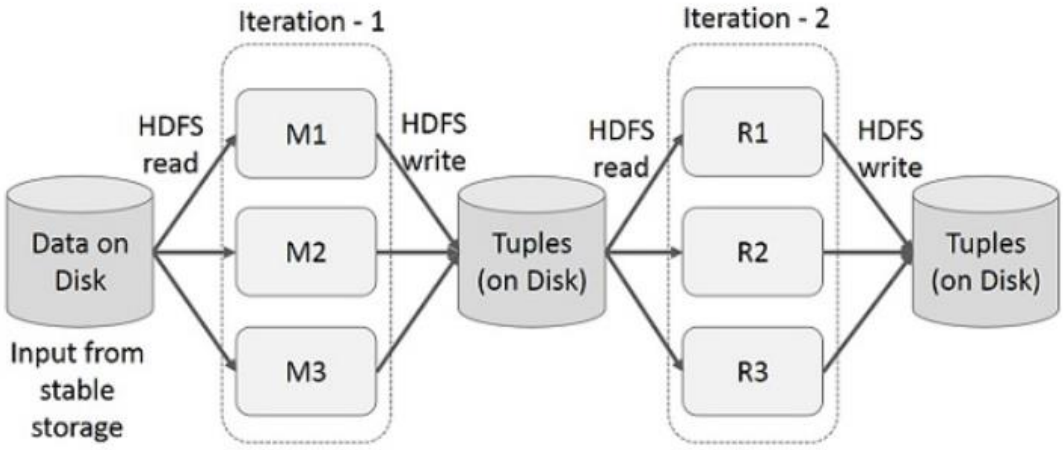
Επιτρέπει στους χρήστες να γράφουν παράλληλους υπολογισμούς, χρησιμοποιώντας ένα σύνολο χειριστών υψηλού επιπέδου, χωρίς να χρειάζεται να ανησυχούν για τη διανομή της εργασίας και την ανοχή σφαλμάτων.

Δυστυχώς, στα περισσότερα τρέχοντα πλαίσια, ο μόνος τρόπος επαναχρησιμοποίησης δεδομένων μεταξύ υπολογισμών (Π.χ. - μεταξύ δύο εργασιών MapReduce) είναι να τα γράψετε σε ένα εξωτερικό σταθερό σύστημα αποθήκευσης (Ex - HDFS). Αν και αυτό το πλαίσιο παρέχει πολλές αφαιρέσεις για την πρόσβαση στους υπολογιστικούς πόρους ενός συμπλέγματος, οι χρήστες θέλουν ακόμα περισσότερα.

Τόσο οι επαναληπτικές όσο και οι διαδραστικές εφαρμογές απαιτούν ταχύτερη κοινή χρήση δεδομένων σε παράλληλες εργασίες. Η κοινή χρήση δεδομένων είναι αργή στο MapReduce λόγω αναπαραγωγής, σειριοποίησης και δίσκου IO. Όσον αφορά το σύστημα αποθήκευσης, οι περισσότερες από τις εφαρμογές Hadoop, ξοδεύουν περισσότερο από το 90% του χρόνου κάνοντας εργασίες ανάγνωσης εγγραφής HDFS.

Επαναληπτικές λειτουργίες στο MapReduce

Επαναχρησιμοποίηση ενδιάμεσων αποτελεσμάτων σε πολλούς υπολογισμούς σε εφαρμογές πολλαπλών σταδίων. Η παρακάτω εικόνα (εικόνα 3) εξηγεί πώς λειτουργεί το τρέχον πλαίσιο, ενώ πραγματοποιεί τις επαναληπτικές λειτουργίες στο MapReduce. Αυτό δημιουργεί σημαντικά έξοδα λόγω αναπαραγωγής δεδομένων, I / O δίσκου και σειριοποίησης, γεγονός που καθιστά το σύστημα αργό.

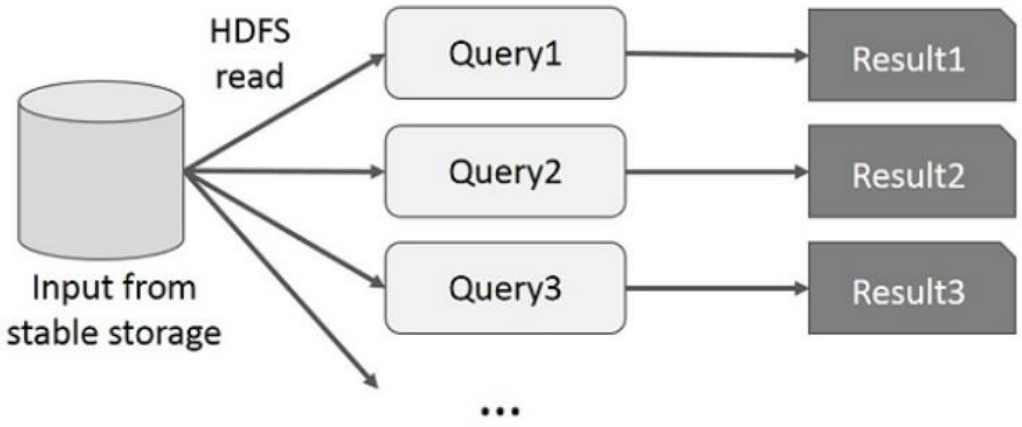


Εικόνα 3. Επαναληπτικές λειτουργίες στο Mapreduce[3]

Διαδραστικές λειτουργίες στο MapReduce

Ο χρήστης εκτελεί ad-hoc ερωτήματα στο ίδιο υποσύνολο δεδομένων. Κάθε ερώτημα θα κάνει το δίσκο I / O στο σταθερό χώρο αποθήκευσης, ο οποίος μπορεί να κυριαρχήσει στο χρόνο εκτέλεσης της εφαρμογής.

Η παρακάτω εικόνα (εικόνα 4) εξηγεί πώς λειτουργεί το τρέχον πλαίσιο, ενώ κάνετε τα διαδραστικά ερωτήματα στο MapReduce.



Εικόνα 4. Διαδραστικές λειτουργίες στο MapReduce[4]

[3] https://www.tutorialspoint.com/apache_spark/apache_spark_rdd.htm
[4] https://www.tutorialspoint.com/apache_spark/apache_spark_rdd.htm

Κοινή χρήση δεδομένων χρησιμοποιώντας Spark RDD

Η κοινή χρήση δεδομένων είναι αργή στο MapReduce λόγω αναπαραγωγής, σειριοποίησης και δίσκου IO. Οι περισσότερες από τις εφαρμογές Hadoop, ξοδεύουν περισσότερο από το 90% του χρόνου κάνοντας λειτουργίες ανάγνωσης εγγραφής HDFS.

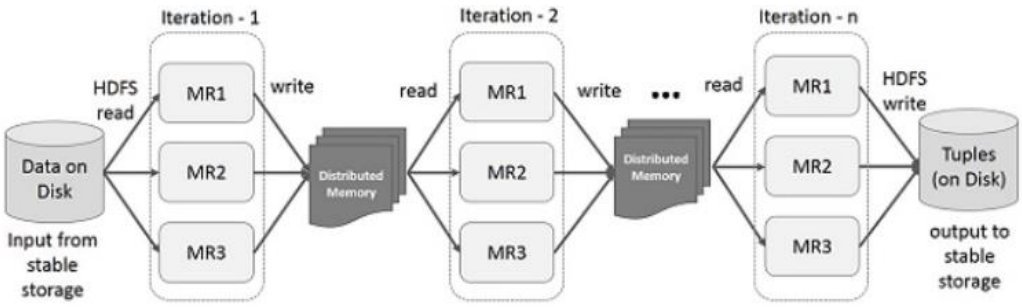
Αναγνωρίζοντας αυτό το πρόβλημα, οι ερευνητές ανέπτυξαν ένα εξειδικευμένο πλαίσιο που ονομάζεται Apache Spark. Η βασική ιδέα του Spark είναι τα ανθεκτικά κατανεμημένα σύνολα δεδομένων (RDD). υποστηρίζει υπολογισμό επεξεργασίας στη μνήμη. Αυτό σημαίνει, αποθηκεύει την κατάσταση της μνήμης ως αντικείμενο στις εργασίες και το αντικείμενο είναι κοινόχρηστο μεταξύ αυτών των εργασιών. Η κοινή χρήση δεδομένων στη μνήμη είναι 10 έως 100 φορές ταχύτερη από το δίκτυο και το Δίσκο.

Πώς επαναλαμβάνονται και διαδραστικές λειτουργίες στο Spark RDD ?

Επαναληπτικές λειτουργίες στο Spark RDD

Η παρακάτω εικόνα (εικόνα 5) δείχνει τις επαναληπτικές λειτουργίες στο Spark RDD. Θα αποθηκεύσει τα ενδιάμεσα αποτελέσματα σε μια κατανεμημένη μνήμη αντί για Σταθερό χώρο αποθήκευσης (Δίσκος) και θα κάνει το σύστημα πιο γρήγορο.

Σημείωση - Εάν η Κατανεμημένη μνήμη (RAM) δεν επαρκεί για την αποθήκευση των ενδιάμεσων αποτελεσμάτων (State of the JOB), τότε θα αποθηκεύσει αυτά τα αποτελέσματα στο δίσκο.

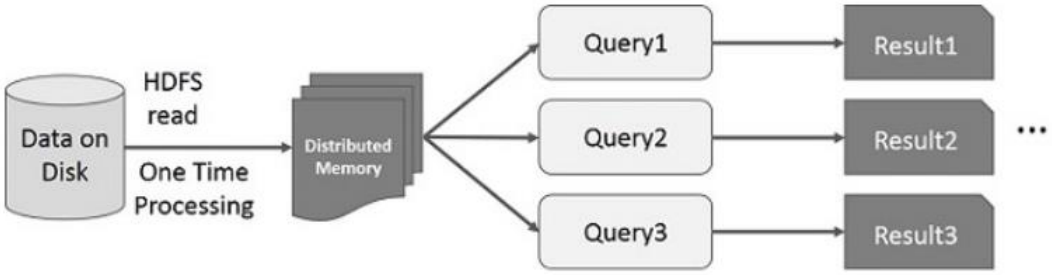


Εικόνα 5. Επαναληπτικές λειτουργίες στο Spark RDD[5]

[5] https://www.tutorialspoint.com/apache_spark/apache_spark_rdd.htm

Διαδραστικές λειτουργίες στο Spark RDD

Η παρακάτω εικόνα (εικόνα 6) δείχνει διαδραστικές λειτουργίες στο Spark RDD. Εάν εκτελούνται επανειλημμένα διαφορετικά ερωτήματα στο ίδιο σύνολο δεδομένων, αυτά τα συγκεκριμένα δεδομένα μπορούν να διατηρηθούν στη μνήμη για καλύτερους χρόνους εκτέλεσης.



Εικόνα 6. Διαδραστικές λειτουργίες στο Spark RDD[6]

Από προεπιλογή, κάθε μετασχηματισμένο RDD μπορεί να υπολογίζεται εκ νέου κάθε φορά που εκτελείτε μια ενέργεια σε αυτό. Ωστόσο, ενδέχεται επίσης να διατηρήσετε ένα RDD στη μνήμη, οπότε το Spark θα διατηρήσει τα στοιχεία γύρω στο σύμπλεγμα για πολύ ταχύτερη πρόσβαση, την επόμενη φορά που θα ζητηθεί. Υπάρχει επίσης υποστήριξη για επίμονα RDDs στο δίσκο ή για αναπαραγωγή σε πολλούς κόμβους.

2.1.4 RDD Lineage

Εισαγωγή στην καταγωγή RDD

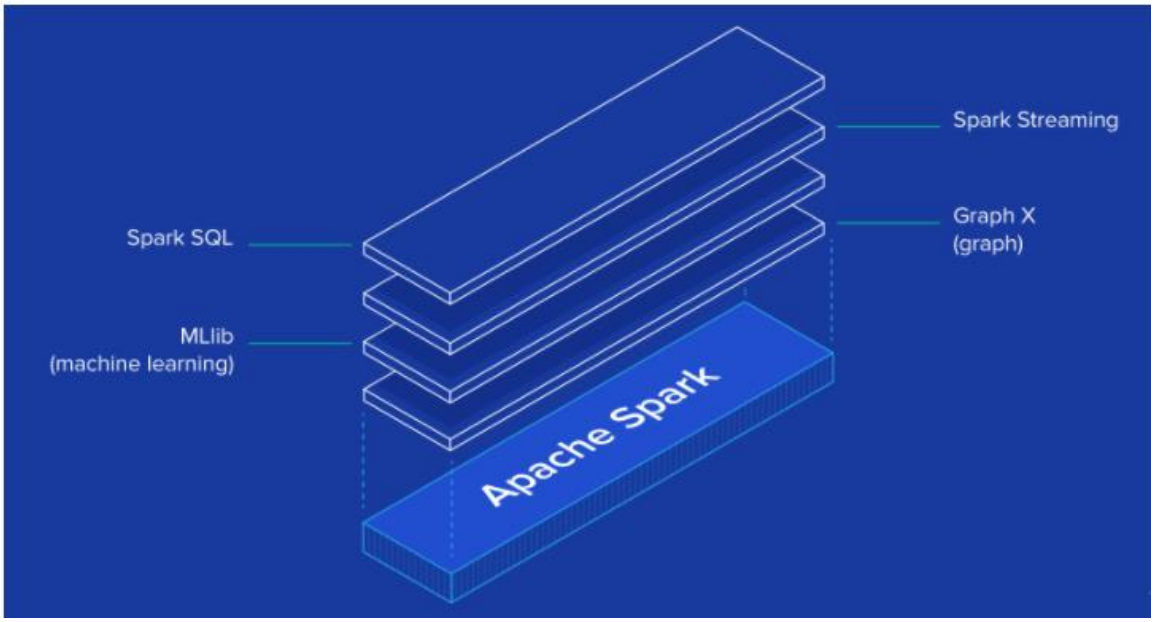
Βασικά, η αξιολόγηση του RDD είναι «τεμπέλης φύσης» (lazy in nature) . Αυτό σημαίνει ότι μια σειρά μετασχηματισμών εκτελούνται σε ένα RDD, οι οποίοι δεν αξιολογούνται καν αμέσως.

Ενώ δημιουργούμε ένα νέο RDD από ένα υπάρχον Spark RDD, αυτό το νέο RDD μεταφέρει ένα δείκτη στο γονικό RDD στο Spark. Αυτό είναι το ίδιο με όλες τις εξαρτήσεις μεταξύ των RDD που είναι συνδεδεμένες σε ένα γράφημα, αντί για τα πραγματικά δεδομένα. Είναι αυτό που ονομάζουμε γράφημα γενεαλογίας.

Η γενεαλογία RDD δεν είναι παρά το γράφημα όλων των γονικών RDDs ενός RDD. Το ονομάζουμε επίσης γράφημα χειριστή RDD ή γράφημα εξάρτησης RDD. Για να είμαστε πολύ συγκεκριμένοι, είναι ένα αποτέλεσμα εφαρμογής μετασχηματισμών στο Spark. Στη συνέχεια, δημιουργεί ένα λογικό σχέδιο εκτέλεσης[11].

[6] https://www.tutorialspoint.com/apache_spark/apache_spark_rdd.htm

Επίσης, το σχέδιο φυσικής εκτέλεσης ή το DAG εκτέλεσης είναι γνωστό ως DAG των σταδίων (DAG of stages).



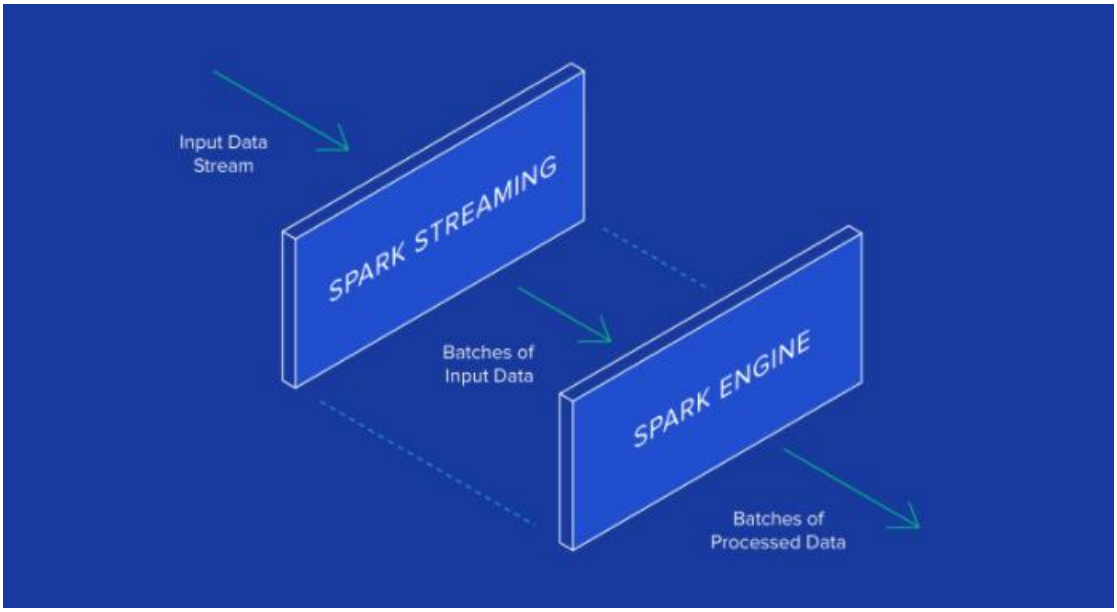
Εικόνα 7. Σχέδιο φυσικής εκτέλεσης[7]

2.1.5 Spark Streaming

Το Spark Streaming υποστηρίζει την επεξεργασία δεδομένων ροής σε πραγματικό χρόνο, όπως αρχεία καταγραφής διακομιστή web παραγωγής (π.χ. Apache Flume και HDFS / S3), κοινωνικά μέσα όπως το Twitter και διάφορες ουρές μηνυμάτων όπως το Kafka. Κάτω από την κουκούλα, το Spark Streaming λαμβάνει τις ροές δεδομένων εισόδου και διαιρεί τα δεδομένα σε παρτίδες[12].

Στη συνέχεια, υποβάλλονται σε επεξεργασία από τον κινητήρα Spark και δημιουργούν τελική ροή αποτελεσμάτων σε παρτίδες, όπως απεικονίζεται παρακάτω (εικόνα 8).

[7] https://www.toptal.com/spark/introduction-to-apache-spark



Εικόνα 8. Spark Streaming[8]

2.1.6 Spark SQl

Το Spark SQL είναι μια μονάδα (module) Spark για επεξεργασία δομημένων δεδομένων. Σε αντίθεση με το βασικό API Spark RDD, οι διεπαφές που παρέχονται από το Spark SQL παρέχουν στο Spark περισσότερες πληροφορίες σχετικά με τη δομή τόσο των δεδομένων όσο και του υπολογισμού που εκτελείται. Εσωτερικά, το Spark SQL χρησιμοποιεί αυτές τις επιπλέον πληροφορίες για να πραγματοποιήσει επιπλέον βελτιστοποιήσεις. Υπάρχουν διάφοροι τρόποι αλληλεπίδρασης με το Spark SQL, συμπεριλαμβανομένου του SQL και του API συνόλου δεδομένων. Κατά τον υπολογισμό ενός αποτελέσματος, χρησιμοποιείται η ίδια μηχανή εκτέλεσης, ανεξάρτητα από το API / γλώσσα που χρησιμοποιείτε για να εκφράσετε τον υπολογισμό. Αυτή η ενοποίηση σημαίνει ότι οι προγραμματιστές μπορούν εύκολα να εναλλάσσονται μεταξύ διαφορετικών API βάσει των οποίων παρέχει τον πιο φυσικό τρόπο για να εκφράσετε μια δεδομένη μετατροπή[12].

Μία χρήση του Spark SQL είναι η εκτέλεση ερωτημάτων SQL. Το Spark SQL μπορεί επίσης να χρησιμοποιηθεί για την ανάγνωση δεδομένων από μια υπάρχουσα εγκατάσταση Hive. χρησιμοποιώντας τη γραμμή εντολών ή μέσω JDBC / ODBC.

[8] https://www.toptal.com/spark/introduction-to-apache-spark

Σύνολα δεδομένων και πλαίσια δεδομένων DATASET & DATAFRAME

Ένα σύνολο δεδομένων είναι μια κατανεμημένη συλλογή δεδομένων. Το σύνολο δεδομένων είναι μια νέα διεπαφή που προστίθεται στο Spark 1.6 που παρέχει τα οφέλη των RDD (ισχυρή πληκτρολόγηση, ικανότητα χρήσης ισχυρών λειτουργιών lambda) με τα πλεονεκτήματα της βελτιστοποιημένης μηχανής εκτέλεσης του Spark SQL Ένα σύνολο δεδομένων μπορεί να κατασκευαστεί από αντικείμενα JVM και στη συνέχεια να χειριστεί χρησιμοποιώντας λειτουργικούς μετασχηματισμούς (map, flatMap, filter κ.λπ.). Το Dataset API είναι διαθέσιμο σε Scala και Java. Η Python δεν έχει την υποστήριξη για το Dataset API. Ωστόσο, λόγω της δυναμικής φύσης της Python, πολλά από τα πλεονεκτήματα του Dataset API είναι ήδη διαθέσιμα (δηλαδή πρόσβαση στο πεδίο μιας σειράς με το όνομα φυσικά row.columnName). Η περίπτωση για το R είναι παρόμοια.

Το DataFrame είναι ένα σύνολο δεδομένων οργανωμένο σε στήλες με όνομα. Είναι εννοιολογικά ισοδύναμο με έναν πίνακα σε μια σχεσιακή βάση δεδομένων ή ένα πλαίσιο δεδομένων στο R / Python, αλλά με πλουσιότερες βελτιστοποιήσεις κάτω από την κουκούλα.

Τα DataFrames μπορούν να κατασκευαστούν από ένα ευρύ φάσμα πηγών όπως: αρχεία δομημένων δεδομένων, πίνακες στην ομάδα, εξωτερικές βάσεις δεδομένων ή υπάρχουσες RDD [1].

Το DataFrame API είναι διαθέσιμο σε Scala, Java, Python και R. Σε Scala και Java, ένα DataFrame αντιπροσωπεύεται από ένα σύνολο δεδομένων σειρών. Στο Scala API, το DataFrame είναι απλώς ένα ψευδώνυμο τύπου του συνόλου δεδομένων [Row]. Ενώ, στο Java API, οι χρήστες πρέπει να χρησιμοποιούν το σύνολο δεδομένων <Row> για να αντιπροσωπεύουν ένα DataFrame.

2.1.7 Spark ML

Το MLlib είναι μια βιβλιοθήκη μηχανικής μάθησης που παρέχει διάφορους αλγόριθμους που έχουν σχεδιαστεί για να κλιμακώνονται σε ένα σύμπλεγμα για ταξινόμηση, παλινδρόμηση, ομαδοποίηση, συνεργατικό φιλτράρισμα και ούτω καθεξής (ανατρέξτε στο άρθρο του Toptal σχετικά με τη μηχανική μάθηση για περισσότερες πληροφορίες σχετικά με αυτό το θέμα) Μερικοί από αυτούς τους αλγόριθμους λειτουργούν επίσης με ροή δεδομένων, όπως η γραμμική παλινδρόμηση χρησιμοποιώντας συνηθισμένα ελάχιστα τετράγωνα ή k-σημαίνει ομαδοποίηση (και περισσότερα στο δρόμο). Το Apache Mahout (μια βιβλιοθήκη μηχανικής μάθησης για το Hadoop) έχει ήδη απομακρυνθεί από το MapReduce και ένωσε τις δυνάμεις του στο Spark MLlib[12].

2.1.8 GraphX

Το GraphX είναι μια βιβλιοθήκη για το χειρισμό γραφημάτων και την εκτέλεση παράλληλων γραφημάτων. Παρέχει ένα ομοιόμορφο εργαλείο για ETL, διερευνητική ανάλυση και επαναληπτικούς υπολογισμούς γραφημάτων. Εκτός από τις ενσωματωμένες λειτουργίες για χειρισμό γραφημάτων, παρέχει μια βιβλιοθήκη κοινών αλγορίθμων γραφημάτων όπως το PageRank.

Συνοψίζοντας, το Spark βοηθά στην απλοποίηση της απαιτητικής και υπολογιστικά εντατικής εργασίας της επεξεργασίας μεγάλων όγκων δεδομένων πραγματικού χρόνου ή αρχειοθετημένων, τόσο δομημένων όσο και μη δομημένων, ενσωματώνοντας απρόσκοπτα σχετικές πολύπλοκες δυνατότητες, όπως μηχανική εκμάθηση και αλγόριθμοι γραφημάτων. Το Spark φέρνει την επεξεργασία Big Data στις μάζες[12].

2.2 Apache Kafka

Το Apache Kafka είναι μια επεκτάσιμη πλατφόρμα ροής συμβάντων.

Το Kafka συνδυάζει τρεις βασικές δυνατότητες, ώστε να μπορούν να εφαρμοστούν οι περιπτώσεις χρήσης σας για ροή συμβάντων από άκρο σε άκρο με μία μόνο δοκιμασμένη λύση:

- Για εγγραφή και ανάγνωση ροών συμβάντων, συμπεριλαμβανομένης της συνεχούς εισαγωγής / εξαγωγής των δεδομένων από άλλα συστήματα.
- Για αποθήκευση ροών δεδομένων με διάρκεια και αξιοπιστία για όσο διάστημα χρειάζεται.
- Για να επεξεργασία ροών δεδομένων που συμβαίνουν ή αναδρομικά.

Και όλη αυτή η λειτουργικότητα παρέχεται με κατανεμημένο, εξαιρετικά επεκτάσιμο, ελαστικό, ανθεκτικό σε σφάλματα και ασφαλή τρόπο. Το Kafka μπορεί να αναπτυχθεί σε μεταλλικό hardware, εικονικές μηχανές και κοντέινερ, καθώς και εντός του χώρου, καθώς και στο cloud [7].

2.2.1 Η λειτουργεία του Kafka

Το Kafka είναι ένα κατανεμημένο σύστημα που αποτελείται από διακομιστές (Servers) και πελάτες (Clients)που επικοινωνούν μέσω ενός πρωτοκόλλου δικτύου υψηλής απόδοσης TCP.

Διακομιστές (Servers): Το Kafka εκτελείται ως σύμπλεγμα ενός ή περισσότερων διακομιστών που μπορούν να εκτείνονται σε πολλά κέντρα δεδομένων ή περιοχές cloud. Μερικοί από αυτούς τους διακομιστές σχηματίζουν το επίπεδο αποθήκευσης, που ονομάζεται μεσίτες (brokers).

Άλλοι διακομιστές εκτελούν το Kafka Connect για συνεχή εισαγωγή και εξαγωγή δεδομένων ως ροές συμβάντων για την ενσωμάτωση του Kafka με τα υπάρχοντα συστήματά, όπως σχεσιακές βάσεις δεδομένων, καθώς και άλλες συστάδες Kafka. Για να επιτρέψει την εφαρμογή κρίσιμων περιπτώσεων χρήσης, ένα σύμπλεγμα Kafka είναι εξαιρετικά επεκτάσιμο και ανεκτικό σε σφάλματα: εάν κάποιος από τους διακομιστές του αποτύχει, οι άλλοι διακομιστές θα αναλάβουν τη δουλειά τους για να εξασφαλίσουν συνεχείς λειτουργίες χωρίς απώλεια δεδομένων [7].

Πελάτες (Clients): Επιτρέπουν τη γραφή κατανεμημένων εφαρμογών και μικροϋπηρεσιών που διαβάζουν, γράφουν και επεξεργάζονται ροές συμβάντων παράλληλα, σε κλίμακα και με ανεκτικό σφάλμα ακόμη και σε περίπτωση προβλημάτων δικτύου ή αστοχιών του μηχανήματος [7]. Το Kafka παρέχεται με μερικούς τέτοιους πελάτες, οι οποίοι αυξάνονται από την κοινότητα του Kafka, οι πελάτες είναι διαθέσιμοι για Java και Scala, συμπεριλαμβανομένης της βιβλιοθήκης Kafka Streams υψηλότερου επιπέδου, για Go, Python, C / C ++ και πολλές άλλες γλώσσες προγραμματισμού καθώς και REST API.

2.2.2 Κύριες έννοιες και ορολογία

Ένα συμβάν καταγράφει το γεγονός ότι "συνέβη κάτι" στον κόσμο ή στην επιχείρηση. Ονομάζεται επίσης εγγραφή ή μήνυμα στην τεκμηρίωση. Το διαβασμα και η γραφή στο Kafka, γίνεται με τη μορφή εκδηλώσεων. Εννοιολογικά, ένα συμβάν έχει κλειδί, τιμή, χρονική σήμανση και προαιρετικές κεφαλίδες μεταδεδομένων. Ακολουθεί ένα παράδειγμα συμβάντος:

Κλειδί εκδήλωσης: "Alice"

Τιμή συμβάντος: "Πραγματοποιήθηκε πληρωμή 200 € στον Bob"

Χρονική σήμανση εκδήλωσης: "25 Ιουνίου 2020 στις 2:06 μ.μ."

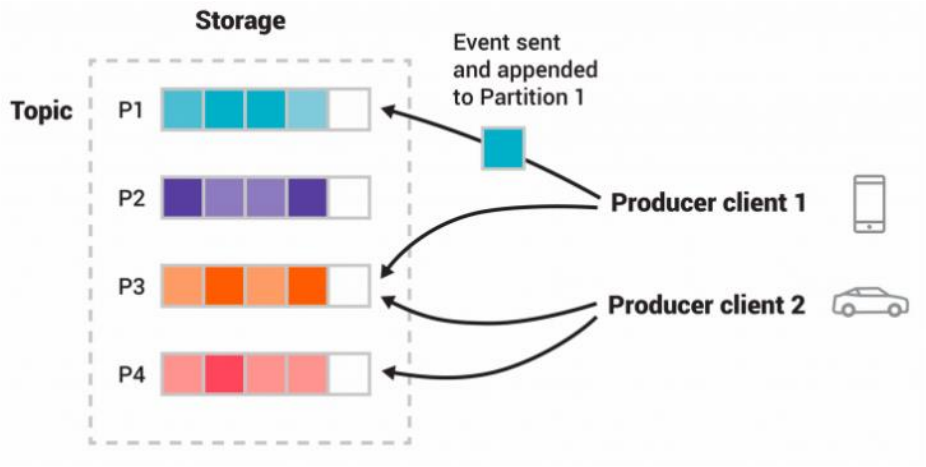
Οι παραγωγοί (Producers) είναι εκείνες οι εφαρμογές που δημοσιεύουν (γράψτε) συμβάντα στο Kafka και οι καταναλωτές (consumers) είναι εκείνοι που εγγράφονται σε (διαβάζουν και επεξεργάζονται) αυτά τα συμβάντα. Στο Kafka, οι παραγωγοί και οι καταναλωτές είναι πλήρως αποσυνδεμένοι και άγνωστοι μεταξύ τους, κάτι που αποτελεί βασικό στοιχείο σχεδιασμού για την επίτευξη της υψηλής κλιμάκωσης για την οποία είναι γνωστό το Kafka. Για παράδειγμα, οι παραγωγοί δεν χρειάζεται ποτέ να περιμένουν τους καταναλωτές. Το Kafka παρέχει διάφορες εγγυήσεις, όπως τη δυνατότητα επεξεργασίας γεγονότων ακριβώς μία φορά [7].

Οι εκδηλώσεις οργανώνονται και αποθηκεύονται διαρκώς σε θέματα (topics). ‘Ενα θέμα είναι παρόμοιο με ένα φάκελο σε ένα σύστημα αρχείων και τα συμβάντα είναι τα αρχεία σε αυτόν το φάκελο. Ένα παράδειγμα ονόματος θέματος θα μπορούσε να είναι "πληρωμές. Ένα θέμα στο Kafka μπορεί να έχει μηδέν, έναν ή πολλούς παραγωγούς που γράφουν συμβάντα σε αυτό, καθώς και μηδενικούς, έναν ή πολλούς καταναλωτές που εγγράφονται σε αυτά τα συμβάντα. Τα συμβάντα σε ένα θέμα μπορούν να διαβαστούν όσο συχνά χρειάζεται.

Σε αντίθεση με τα παραδοσιακά συστήματα ανταλλαγής μηνυμάτων, τα συμβάντα δεν διαγράφονται μετά την κατανάλωση.

Αντ 'αυτού, καθορίζετε για πόσο χρονικό διάστημα το Kafka θα διατηρεί τα συμβάντα σας μέσω μιας ρύθμισης διαμόρφωσης ανά θέμα, μετά την οποία τα παλιά συμβάντα θα απορριφθούν. Η απόδοση του Kafka είναι ουσιαστικά σταθερή σε σχέση με το μέγεθος των δεδομένων, οπότε η αποθήκευση δεδομένων για μεγάλο χρονικό διάστημα είναι απολύτως διαχειρίσιμη.

Τα θέματα είναι κατατμημένα (partitioned) (εικόνα 9), που σημαίνει ότι ένα θέμα κατανέμεται σε διάφορους "κουβάδες" που βρίσκονται σε διαφορετικούς μεσίτες του Kafka. Αυτή η κατανεμημένη τοποθέτηση των δεδομένων είναι πολύ σημαντική για τη δυνατότητα κλιμάκωσης, διότι επιτρέπει στις εφαρμογές διαβάζουν και να γράφουν τα δεδομένα ταυτόχρονα από και προς πολλούς μεσίτες. Όταν ένα νέο συμβάν δημοσιεύεται σε ένα θέμα, προσαρτάται πραγματικά σε ένα από τα διαμερίσματα του θέματος. Τα συμβάντα με το ίδιο κλειδί συμβάντος (π.χ. πελάτης ή αναγνωριστικό οχήματος) γράφονται στο ίδιο διαμέρισμα και το Kafka εγγυάται ότι οποιοσδήποτε καταναλωτής ενός συγκεκριμένου διαμερίσματος θέματος θα διαβάζει πάντα τα συμβάντα αυτού του διαμερίσματος με την ίδια ακριβώς σειρά όπως γράφτηκαν.



Εικόνα 9. Παράδειγμα εισαγωγής δεδομένων στο Kafka[9]

Παράδειγμα εικόνας 9: Αυτό το παράδειγμα θέματος έχει τέσσερα διαμερίσματα (partitions) P1 – P4. Δύο διαφορετικοί παραγωγοί δημοσιεύουν, ανεξάρτητα ο ένας από τον άλλο, νέα συμβάντα στο θέμα. Τα συμβάντα με το ίδιο κλειδί (που υποδηλώνεται με το χρώμα τους στην εικόνα) γράφονται στο ίδιο διαμέρισμα. Σημειώστε ότι και οι δύο παραγωγοί μπορούν να γράψουν στο ίδιο διαμέρισμα εάν είναι απαραίτητο.

[9] https://kafka.apache.org/documentation/#introduction

Για να γίνουν τα δεδομένα ανεκτά σε σφάλματα και εξαιρετικά διαθέσιμα, κάθε θέμα μπορεί να αναπαραχθεί, ακόμη και σε γεωγραφικές περιοχές ή κέντρα δεδομένων, έτσι ώστε να υπάρχουν πάντα πολλοί μεσίτες που έχουν ένα αντίγραφο των δεδομένων σε περίπτωση που τα πράγματα πάνε στραβά. Μια κοινή ρύθμιση παραγωγής είναι ένας συντελεστής αναπαραγωγής (Replication Factor) του 3, δηλαδή, θα υπάρχουν πάντα τρία αντίγραφα των δεδομένων σας. Αυτή η αναπαραγωγή εκτελείται στο επίπεδο των διαμερισμάτων θέματος.

2.2.3 Περιπτώσεις όπου χρησιμοποιείται το Kafka.

Μερικές από τις δημοφιλείς περιπτώσεις χρήσης για το Apache Kafka είναι:

- Μηνύματα

  Το Kafka λειτουργεί καλά ως αντικατάσταση ενός πιο παραδοσιακού μεσίτη μηνυμάτων. Οι μεσίτες μηνυμάτων χρησιμοποιούνται για διάφορους λόγους (για την αποσύνδεση της επεξεργασίας από παραγωγούς δεδομένων, για την αποθήκευση μη επεξεργασμένων μηνυμάτων κ.λ.π. Σε σύγκριση με τα περισσότερα συστήματα ανταλλαγής μηνυμάτων, το Kafka έχει καλύτερη απόδοση, ενσωματωμένο διαχωρισμό, αναπαραγωγή και ανοχή σφαλμάτων που το καθιστούν μια καλή λύση για εφαρμογές επεξεργασίας μηνυμάτων μεγάλης κλίμακας. Οι χρήσεις ανταλλαγής μηνυμάτων είναι συχνά συγκριτικά χαμηλής απόδοσης, αλλά ενδέχεται να απαιτούν χαμηλό λανθάνοντα χρόνο από άκρο σε άκρο και συχνά εξαρτώνται από τις ισχυρές εγγυήσεις αντοχής που παρέχει το Kafka.

- Παρακολούθηση δραστηριότητας ιστότοπου (Website activity tracking)

  Η αρχική περίπτωση χρήσης για το Kafka ήταν να είναι σε θέση να αναδημιουργήσει έναν αγωγό παρακολούθησης δραστηριότητας χρήστη ως ένα σύνολο ροών δημοσίευσης-εγγραφής σε πραγματικό χρόνο. Αυτό σημαίνει ότι η δραστηριότητα του ιστότοπου (προβολές σελίδας, αναζητήσεις ή άλλες ενέργειες που μπορούν να κάνουν οι χρήστες) δημοσιεύεται σε κεντρικά θέματα με ένα θέμα ανά τύπο δραστηριότητας. Αυτές οι ροές είναι διαθέσιμες για συνδρομή για μια σειρά περιπτώσεων χρήσης, όπως επεξεργασία σε πραγματικό χρόνο, παρακολούθηση σε πραγματικό χρόνο και φόρτωση σε συστήματα αποθήκευσης δεδομένων Hadoop/Spark ή εκτός σύνδεσης για επεξεργασία και αναφορά εκτός σύνδεσης.

  Η παρακολούθηση δραστηριότητας είναι συχνά πολύ μεγάλης έντασης, καθώς δημιουργούνται πολλά μηνύματα δραστηριότητας για κάθε προβολή σελίδας χρήστη.

- Μετρήσεις

  Το Kafka χρησιμοποιείται συχνά για επιχειρησιακά δεδομένα παρακολούθησης. Αυτό περιλαμβάνει τη συγκέντρωση στατιστικών στοιχείων από κατανεμημένες εφαρμογές για την παραγωγή κεντρικών ροών επιχειρησιακών δεδομένων.

- Συγκέντρωση καταγραφής

  Συχνά το Kafka χρησιμοποιείται ως υποκατάστατο μιας λύσης συγκέντρωσης αρχείων καταγραφής. Η συνάθροιση αρχείων καταγραφής συλλέγει συνήθως φυσικά αρχεία καταγραφής από διακομιστές και τα τοποθετεί σε κεντρική θέση (ίσως ένας διακομιστής αρχείων ή HDFS) για επεξεργασία. Το Kafka αφαιρεί τις λεπτομέρειες των αρχείων και δίνει μια πιο καθαρή αφαίρεση δεδομένων καταγραφής ή συμβάντων ως ροή μηνυμάτων. Αυτό επιτρέπει την επεξεργασία χαμηλότερου λανθάνοντος χρόνου και ευκολότερη υποστήριξη για πολλές πηγές δεδομένων και κατανεμημένη κατανάλωση δεδομένων. Σε σύγκριση με τα log-centric συστήματα όπως το Scribe ή το Flume, το Kafka προσφέρει εξίσου καλή απόδοση, ισχυρότερες εγγυήσεις αντοχής λόγω αναπαραγωγής (replication) και πολύ χαμηλότερη καθυστέρηση από άκρο σε άκρο.

- Επεξεργασία ροής

  Πολλοί χρήστες επεξεργάζονται δεδομένα σε αγωγούς επεξεργασίας που αποτελούνται από πολλαπλά στάδια, όπου τα ακατέργαστα δεδομένα εισαγωγής καταναλώνονται από θέματα του Kafka και στη συνέχεια συγκεντρώνονται, εμπλουτίζονται ή μετατρέπονται με άλλο τρόπο σε νέα θέματα για περαιτέρω κατανάλωση ή παρακολούθηση. Για παράδειγμα, ένας αγωγός επεξεργασίας για τη σύσταση ειδησεογραφικών άρθρων ενδέχεται να ανιχνεύσει περιεχόμενο άρθρου από ροές RSS και να το δημοσιεύσει σε ένα θέμα "άρθρα". Η περαιτέρω επεξεργασία ενδέχεται να ομαλοποιήσει ή να αντιγράψει αυτό το περιεχόμενο και να δημοσιεύσει το καθαρισμένο περιεχόμενο του άρθρου σε ένα νέο θέμα. ένα τελικό στάδιο επεξεργασίας μπορεί να επιχειρήσει να προτείνει αυτό το περιεχόμενο στους χρήστες. Τέτοιοι αγωγοί επεξεργασίας δημιουργούν γραφήματα ροών δεδομένων σε πραγματικό χρόνο με βάση τα μεμονωμένα θέματα. Ξεκινώντας από 0.10.0.0, μια ελαφριά αλλά ισχυρή βιβλιοθήκη επεξεργασίας ροής που ονομάζεται Kafka Streams είναι διαθέσιμη στο Apache Kafka για να εκτελεί τέτοια επεξεργασία δεδομένων όπως περιγράφεται παραπάνω. Εκτός από τα Kafka Streams, εναλλακτικά εργαλεία επεξεργασίας ροής ανοιχτού κώδικα περιλαμβάνουν τα Apache Storm και Apache Samza.

- Προμήθεια εκδηλώσεων

  Η προμήθεια συμβάντων είναι ένα στυλ σχεδιασμού εφαρμογών όπου καταγράφονται οι αλλαγές κατάστασης ως μια ακολουθία εγγραφών με χρονοδιάταξη. Η υποστήριξη του Kafka για πολύ μεγάλα αποθηκευμένα δεδομένα καταγραφής το καθιστά ένα εξαιρετικό backend για μια εφαρμογή ενσωματωμένη σε αυτό το στυλ.

- Δέσμευση καταγραφής

  Το Kafka μπορεί να χρησιμεύσει ως ένα είδος εξωτερικού αρχείου καταγραφής δεσμεύσεων για ένα κατανεμημένο σύστημα. Το αρχείο καταγραφής βοηθά στην αναπαραγωγή δεδομένων μεταξύ κόμβων και λειτουργεί ως μηχανισμός επανασυγχρονισμού για αποτυχημένους κόμβους για την επαναφορά των δεδομένων τους. Η δυνατότητα συμπίεσης καταγραφής στο Kafka βοηθά στην υποστήριξη αυτής της χρήσης. Σε αυτή τη χρήση το Kafka είναι παρόμοιο με το έργο Apache BookKeeper.

2.3 ELK Stack

Το "ELK" είναι το ακρονύμιο τριών έργων ανοιχτού κώδικα: Elasticsearch, Logstash και Kibana. Το Elasticsearch είναι μια μηχανή αναζήτησης και ανάλυσης. Το Logstash είναι ένας αγωγός επεξεργασίας δεδομένων από διακομιστή που απορροφά δεδομένα από πολλές πηγές ταυτόχρονα, τα μετατρέπει και στη συνέχεια τα στέλνει σε ένα "stash" όπως το Elasticsearch. Το Kibana επιτρέπει στους χρήστες να οπτικοποιούν τα δεδομένα στο Elasticsearch [4].

2.3.1 Elasticsearch

Το Elasticsearch είναι μια μηχανή διανομής και ανάλυσης ανοιχτού κώδικα για όλους τους τύπους δεδομένων, συμπεριλαμβανομένων κειμένων, αριθμητικών, γεωχωρικών, δομημένων και μη δομημένων. Το Elasticsearch βασίζεται στην Apache Lucene και κυκλοφόρησε για πρώτη φορά το 2010 από την Elasticsearch N.V. (τώρα γνωστή ως Elastic). Γνωστό για τα απλά API REST, την κατανεμημένη φύση, την ταχύτητα και την επεκτασιμότητα, το Elasticsearch είναι το κεντρικό στοιχείο του Elastic Stack, ένα σύνολο εργαλείων ανοιχτού κώδικα για απορρόφηση δεδομένων, εμπλουτισμό, αποθήκευση, ανάλυση και οπτικοποίηση [4].

2.3.1.1 Η χρησιμοτητα του Elasticsearch

Η ταχύτητα και η επεκτασιμότητα του Elasticsearch και η ικανότητά του να ευρετηριάζει πολλούς τύπους περιεχομένου σημαίνει ότι μπορεί να χρησιμοποιηθεί για πολλές περιπτώσεις χρήσης όπως [4]:

- Αναζήτηση εφαρμογών (Aplication search)
- Αναζήτηση ιστότοπου (Website search)
- Εταιρική αναζήτηση (Enterprise search)
- Καταγραφή και αναλυτικά στοιχεία καταγραφής (Logging and log analytics)
- Μετρήσεις υποδομής και παρακολούθηση κοντέινερ (Infrastructure metrics and container monitoring)

- Παρακολούθηση απόδοσης εφαρμογής (Application performance monitoring)
- Ανάλυση και οπτικοποίηση γεωχωρικών δεδομένων (Geospatial data analysis and visualization)
- Αναλυτικά στοιχεία ασφαλείας (Security analytics)
- Επιχειρηματική ανάλυση (Business analytics)

Το Elasticsearch είναι γρήγορο επειδή είναι χτισμένο πάνω από το Lucene και υπερέχει στην αναζήτηση πλήρους κειμένου. Το Elasticsearch είναι επίσης μια πλατφόρμα αναζήτησης σχεδόν σε πραγματικό χρόνο, που σημαίνει ότι ο χρόνος καθυστέρησης από τη στιγμή που ένα έγγραφο ευρετηριάζεται μέχρι να γίνει αναζήτηση είναι πολύ σύντομος - συνήθως ένα δευτερόλεπτο. Ως αποτέλεσμα, το Elasticsearch είναι κατάλληλο για περιπτώσεις ευαίσθητες στο χρόνο, όπως αναλυτικά στοιχεία ασφαλείας και παρακολούθηση υποδομής.

Τα έγγραφα που αποθηκεύονται στο Elasticsearch διανέμονται σε διαφορετικά δοχεία (containers) γνωστά ως shards, τα οποία αναπαράγονται για να παρέχουν περιττά αντίγραφα των δεδομένων σε περίπτωση βλάβης υλικού. Η κατανεμημένη φύση του Elasticsearch του επιτρέπει να κλιμακώσει εκατοντάδες (ή και χιλιάδες) διακομιστές και να χειριστεί petabytes δεδομένων.

Το Elasticsearch διαθέτει ένα ευρύ φάσμα χαρακτηριστικών. Εκτός από την ταχύτητα, την επεκτασιμότητα και την ανθεκτικότητα, το Elasticsearch διαθέτει μια σειρά από ισχυρά ενσωματωμένα χαρακτηριστικά που καθιστούν την αποθήκευση και την αναζήτηση δεδομένων ακόμη πιο αποτελεσματική, όπως συλλογές δεδομένων και διαχείριση κύκλου ζωής ευρετηρίου [4].

2.3.1.2 Η λειτουργεία του Elasticsearch

Τα ακατέργαστα δεδομένα ρέουν στην Elasticsearch από μια ποικιλία πηγών, όπως αρχεία καταγραφής, μετρήσεις συστήματος και εφαρμογές ιστού. Η απορρόφηση δεδομένων είναι η διαδικασία με την οποία αυτά τα ακατέργαστα δεδομένα αναλύονται, ομαλοποιούνται και εμπλουτίζονται προτού ευρετηριαστούν στο Elasticsearch. Μόλις ευρετηριαστούν στο Elasticsearch, οι χρήστες μπορούν να εκτελέσουν σύνθετα ερωτήματα έναντι των δεδομένων τους και να χρησιμοποιήσουν συγκεντρώσεις για να ανακτήσουν περίπλοκες περιλήψεις των δεδομένων τους. Από το Kibana, οι χρήστες μπορούν να δημιουργήσουν ισχυρές απεικονίσεις των δεδομένων τους, να μοιραστούν πίνακες ελέγχου και να διαχειριστούν το Elastic Stack [4].

Το ευρετήριο Elasticsearch είναι μια συλλογή εγγράφων που σχετίζονται μεταξύ τους. Το Elasticsearch αποθηκεύει δεδομένα ως έγγραφα JSON. Κάθε έγγραφο συσχετίζει ένα σύνολο κλειδιών (ονόματα πεδίων ή ιδιοτήτων) με τις αντίστοιχες τιμές τους (συμβολοσειρές, αριθμούς, Booleans, ημερομηνίες, πίνακες τιμών, γεωγραφικές τοποθεσίες ή άλλους τύπους δεδομένων).

Το Elasticsearch χρησιμοποιεί μια δομή δεδομένων που ονομάζεται ανεστραμμένο ευρετήριο, το οποίο έχει σχεδιαστεί για να επιτρέπει πολύ γρήγορες αναζητήσεις πλήρους κειμένου.

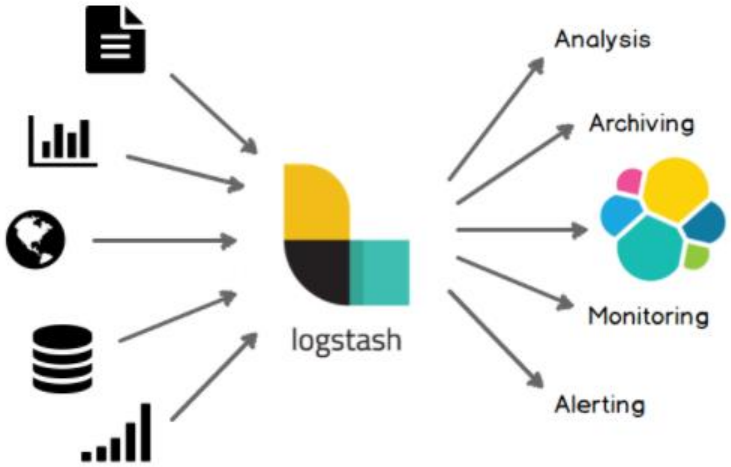
Ένα ανεστραμμένο ευρετήριο παραθέτει κάθε μοναδική λέξη που εμφανίζεται σε οποιοδήποτε έγγραφο και προσδιορίζει όλα τα έγγραφα στα οποία εμφανίζεται κάθε λέξη. Κατά τη διάρκεια της διαδικασίας ευρετηρίασης, το Elasticsearch αποθηκεύει έγγραφα και δημιουργεί ένα ανεστραμμένο ευρετήριο για να κάνει τα δεδομένα εγγράφων αναζήτηση σε σχεδόν πραγματικό χρόνο. Η δημιουργία ευρετηρίου ξεκινά με το API ευρετηρίου, μέσω του οποίου μπορείτε να προσθέσετε ή να ενημερώσετε ένα έγγραφο JSON σε ένα συγκεκριμένο ευρετήριο [4].

2.3.2 Logstash

Το Logstash είναι μια μηχανή συλλογής δεδομένων ανοιχτού κώδικα και προσφέρει real-time pipelines. Το Logstash μπορεί να ενοποιήσει δυναμικά δεδομένα από διαφορετικές πηγές και να ομαλοποιήσει τα δεδομένα σε προορισμούς της επιλογής μας.

Ενώ το Logstash αρχικά ήταν μια καινοτομία στη συλλογή αρχείων καταγραφής, οι δυνατότητές του πλέον επεκτείνονται πολύ πέρα από τη συγκεκριμένη περίπτωση χρήσης.

Το Logstash είναι μία από τις πιο χρήσιμες εφαρμογές του Elasticsearch όπως και για άλλες διότι, είναι ένα οριζόντιο επεκτάσιμο pipeline επεξεργασίας δεδομένων με ισχυρή συνέργια μεταξύ του Elasticsearch και των διαφόρων πηγών άντλησης δεδομένων (εικόνα 10) [4].



Εικόνα 10. Logstash[10]

2.3.3 Kibana

Το Kibana είναι μια εφαρμογή ανοιχτού κώδικα που βρίσκεται στην κορυφή του Elastic Stack, παρέχοντας δυνατότητες αναζήτησης και οπτικοποίησης δεδομένων για δεδομένα που ευρετηριάζονται στο Elasticsearch. Αναπτύχθηκε το 2013 από την κοινότητα Elasticsearch [4].

2.3.3.1 Η χρησιμότητα του Kibana

Η σφιχτή ενσωμάτωση του Kibana με το Elasticsearch και το μεγαλύτερο Elastic Stack το καθιστούν ιδανικό για την υποστήριξη των παρακάτω [4]:

1. Αναζήτηση, προβολή και οπτικοποίηση δεδομένων που ευρετηριάζονται στο Elasticsearch και ανάλυση των δεδομένων μέσω της δημιουργίας ραβδογραμμάτων, γραφημάτων πίτας, πινάκων, ιστογραμμάτων και χαρτών.
2. Μια προβολή πίνακα ελέγχου συνδυάζει αυτά τα οπτικά στοιχεία για να κοινοποιηθεί στη συνέχεια μέσω προγράμματος περιήγησης για να παρέχει αναλυτικές προβολές σε πραγματικό χρόνο σε μεγάλους όγκους δεδομένων για την υποστήριξη περιπτώσεων χρήσης όπως :

[10] https://www.elastic.co/guide/en/logstash/current/introduction.html

- Καταγραφή και αναλυτικά στοιχεία καταγραφής
- Μετρήσεις υποδομής και παρακολούθηση κοντέινερ
- Παρακολούθηση απόδοσης εφαρμογών (APM) Ανάλυση και οπτικοποίηση γεωχωρικών δεδομένων
- Αναλυτικά στοιχεία ασφαλείας
- Επιχειρηματική ανάλυση

3. Παρακολούθηση, διαχείριση και προστασία μιας παρουσίας Elastic Stack μέσω διεπαφής ιστού.
4. Συγκέντρωση της πρόσβασης για ενσωματωμένες λύσεις που αναπτύχθηκαν στο Elastic Stack για εφαρμογές παρατηρησιμότητας, ασφάλειας και εταιρικής αναζήτησης.

2.3.3.2 Η λειτουργεία της αναζήτηση και η οπτικοποίηση δεδομένων στο Kibana

Το Kibana επιτρέπει την οπτική ανάλυση δεδομένων από ένα ευρετήριο Elasticsearch ή πολλαπλούς δείκτες. Οι δείκτες δημιουργούνται όταν το Logstash απορροφά μη δομημένα δεδομένα από αρχεία καταγραφής και άλλες πηγές και τα μετατρέπει σε δομημένη μορφή για λειτουργίες αποθήκευσης και αναζήτησης στο Elasticsearch [4].

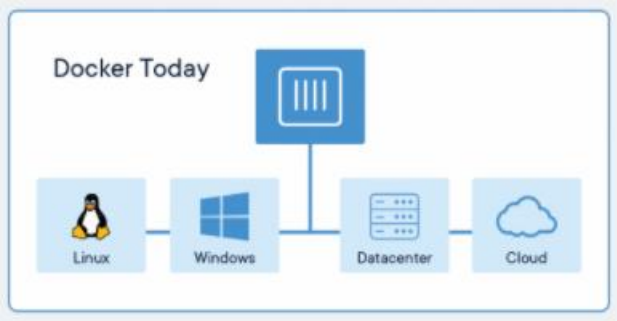
Η διεπαφή του Kibana επιτρέπει στους χρήστες να υποβάλλουν ερωτήματα στα δεδομένα σε δείκτες Elasticsearch και, στη συνέχεια, να απεικονίζουν τα αποτελέσματα μέσω τυπικών επιλογών γραφήματος ή ενσωματωμένων εφαρμογών όπως Lens, Canvas και Maps.

Οι χρήστες μπορούν να επιλέξουν μεταξύ διαφορετικών τύπων γραφημάτων, να αλλάξουν τις συγκεντρώσεις αριθμών και να φιλτράρουν σε συγκεκριμένα τμήματα δεδομένων.

2.4 Η πλατφόρμα Docker

Το Docker παρέχει τη δυνατότητα συσκευασίας και εκτέλεσης μιας εφαρμογής σε ένα απομονωμένο περιβάλλον που ονομάζεται κοντέινερ. Η απομόνωση και η ασφάλεια σάς επιτρέπουν να εκτελείτε πολλά κοντέινερ ταυτόχρονα σε έναν συγκεκριμένο κεντρικό υπολογιστή. Τα κοντέινερ τρέχουν απευθείας μέσα στον πυρήνα του κεντρικού υπολογιστή. Αυτό σημαίνει ότι μπορείτε να εκτελέσετε περισσότερα κοντέινερ σε έναν δεδομένο συνδυασμό υλικού από ό, τι εάν χρησιμοποιούσατε εικονικές μηχανές (virtual machines). Μπορεί να γίνει ακόμη και να εκτελέση κοντέινερ Docker σε κεντρικές συσκευές που είναι πραγματικά εικονικές μηχανές [5]!

Τα κοντέινερ Docker είναι παντού: Linux, Windows, Data center, Cloud, Serverless (εικόνα 11).



Εικόνα 11. Docker και λειτουργικά συστήματα[11]

Η τεχνολογία κοντέινερ Docker κυκλοφόρησε το 2013 ως μηχανή ανοιχτού κώδικα Docker Engine. Αξιοποίησε τις υπάρχουσες υπολογιστικές έννοιες γύρω από κοντέινερ και συγκεκριμένα στον κόσμο του Linux, πρωτόγονες γνωστές ως cgroups και namespaces.

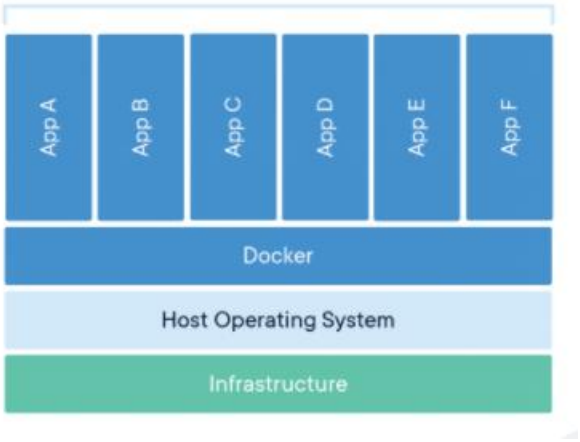
Η τεχνολογία του Docker είναι μοναδική επειδή εστιάζει στις απαιτήσεις των προγραμματιστών και των διαχειριστών συστημάτων για το διαχωρισμό των εξαρτήσεων εφαρμογών από την υποδομή. Η επιτυχία στον κόσμο του Linux οδήγησε σε μια συνεργασία με τη Microsoft που έφερε κοντέινερ Docker και τη λειτουργικότητά του στον Windows Server (μερικές φορές αναφέρεται ως κοντέινερ Windows Docker). Τεχνολογία διαθέσιμη από το Docker και το έργο ανοιχτού κώδικα, το Moby έχει αξιοποιηθεί από όλους τους μεγάλους προμηθευτές κέντρων δεδομένων και παρόχους cloud. Πολλοί από αυτούς τους παρόχους αξιοποιούν το Docker για τις προσφορές IaaS που προέρχονται από κοντέινερ.

Επιπλέον, τα κορυφαία πλαίσια χωρίς διακομιστές ανοιχτού κώδικα χρησιμοποιούν την τεχνολογία κοντέινερ Docker.

[11] https://www.docker.com/resources/what-container

2.4.1 Docker κοντέινερ

Είναι μια τυποποιημένη μονάδα λογισμικού (εικόνα 12) για ανάπτυξη και αποστολή [6].



Εικόνα 12. Το Docker στη πράξη[12]

Το κοντέινερ είναι μια τυπική μονάδα λογισμικού που συσκευάζει κώδικα και όλες τις εξαρτήσεις του, έτσι ώστε η εφαρμογή να εκτελείται γρήγορα και αξιόπιστα από το ένα υπολογιστικό περιβάλλον στο άλλο. Η εικόνα κοντέινερ (image container) Docker είναι ένα ελαφρύ, αυτόνομο, εκτελέσιμο πακέτο λογισμικού που περιλαμβάνει όλα όσα χρειάζονται για την εκτέλεση μιας εφαρμογής: κωδικός, χρόνος εκτέλεσης, εργαλεία συστήματος, βιβλιοθήκες συστήματος και ρυθμίσεις [6].

Οι εικόνες κοντέινερ γίνονται κοντέινερ κατά το χρόνο εκτέλεσης και στην περίπτωση των Docker κοντέινερ - οι εικόνες γίνονται κοντέινερ όταν εκτελούνται στο Docker Engine. Το λογισμικό με κοντέινερ θα λειτουργεί πάντα το ίδιο, ανεξάρτητα από την υποδομή. Τα κοντέινερ απομονώνουν το λογισμικό από το περιβάλλον του και διασφαλίζουν ότι λειτουργεί ομοιόμορφα παρά τις διαφορές, για παράδειγμα, μεταξύ ανάπτυξης και σταδιοποίησης.

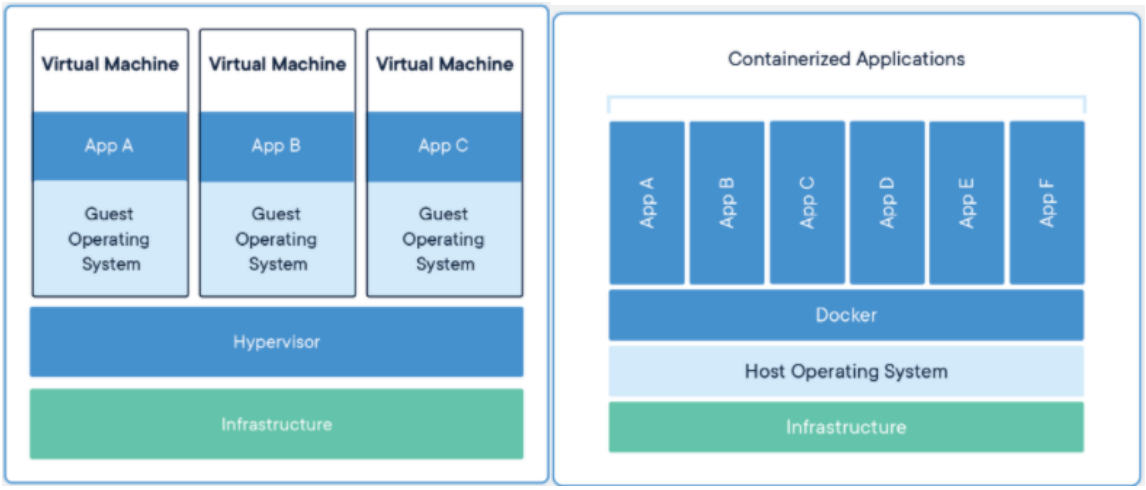
[12] https://www.docker.com/resources/what-container

Docker κοντέινερ που λειτουργούν στο Docker Engine [6]:

- Στάνταρ: μπορούν να είναι φορητά οπουδήποτε
- Ελαφρύ: Τα κοντέινερ μοιράζονται τον πυρήνα του συστήματος OS του μηχανήματος και επομένως δεν απαιτούν λειτουργικό σύστημα ανά εφαρμογή, αυξάνοντας την αποδοτικότητα του διακομιστή και μειώνοντας το κόστος διακομιστή και αδειοδότησης
- Ασφαλής: Οι εφαρμογές είναι ασφαλέστερες σε κοντέινερ και το Docker παρέχει τις ισχυρότερες προεπιλεγμένες δυνατότητες απομόνωσης στον κλάδο.

2.4.2 Σύγκριση κοντέινερ και εικονικών μηχανών (virtual machines)

Τα κοντέινερ και οι εικονικές μηχανές έχουν παρόμοια οφέλη απομόνωσης πόρων και κατανομής, αλλά λειτουργούν διαφορετικά επειδή τα κοντέινερ εικονικοποιούν το λειτουργικό σύστημα αντί του υλικού (εικόνα 13). Τα κοντέινερ είναι πιο φορητά και αποτελεσματικά.



Εικόνα 13. Σύγκριση Docker – Virtul Machines[13]

[13] https://www.docker.com/resources/what-container

Κοντέινερ

Τα κοντέινερ είναι μια αφαίρεση στο επίπεδο της εφαρμογής που συνδυάζει κώδικα και εξαρτήσεις μαζί. Πολλά κοντέινερ μπορούν να τρέχουν στον ίδιο υπολογιστή και να μοιράζονται τον πυρήνα του OS με άλλα κοντέινερ, το καθένα λειτουργεί ως απομονωμένη διαδικασία στο χώρο του χρήστη. Τα κοντέινερ καταλαμβάνουν λιγότερο χώρο από τα VM (οι εικόνες κοντέινερ είναι συνήθως δεκάδες MB σε μέγεθος), μπορούν να χειριστούν περισσότερες εφαρμογές και απαιτούν λιγότερα VM και λειτουργικά συστήματα.

Εικονικές Μηχανές

Οι εικονικές μηχανές (VM) είναι μια αφαίρεση φυσικού υλικού που μετατρέπει έναν διακομιστή σε πολλούς διακομιστές. Το hypervisor επιτρέπει την εκτέλεση πολλαπλών VM σε ένα μόνο μηχάνημα. Κάθε VM περιλαμβάνει ένα πλήρες αντίγραφο ενός λειτουργικού συστήματος, της εφαρμογής, των απαραίτητων δυαδικών αρχείων και βιβλιοθηκών - καταλαμβάνοντας δεκάδες GB. Η εκκίνηση των VM μπορεί επίσης να είναι αργή.

Επίσης μπορεί να γίνει και συνδυασμός των δύο παρέχοντας μεγάλη ευελιξία στην ανάπτυξη και διαχείριση της εφαρμογής.

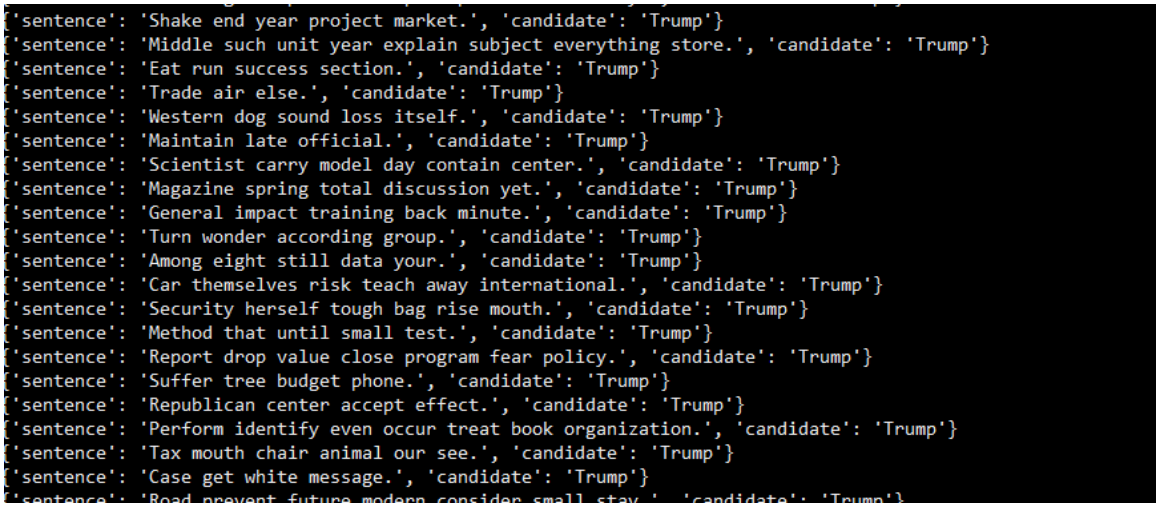
Κεφάλαιο 3: Παρουσίαση συστήματος ανάλυσης συναισθήματος σε πραγματικό χρόνο

Στο παρόν κεφάλαιο γίνεται εκτενής περιγραφή του τρόπου με τον οποίο λειτουργεί η εφαρμογή μας όπως αυτή απεικονίζεται στο Σχήμα 1. Αρχικά εξηγούμε το τρόπο με τον οποίο δημιουργούμε τις ψεύτικες προτάσεις. Στη συνέχεια περιγράφουμε το τρόπο λειτουργίας του Kafka συστήματος μας. Αμέσως μετά εξηγούμε τις ενέργεις που πραγματοποιεί το Spark. Τέλος, αναφερόμαστε στο Elk Stack όπου και καταλήγουν τα δεδομένα μας για να τα οπτικοποιήσυμε και να βγάλουμε τα συμπεράσματά μας.

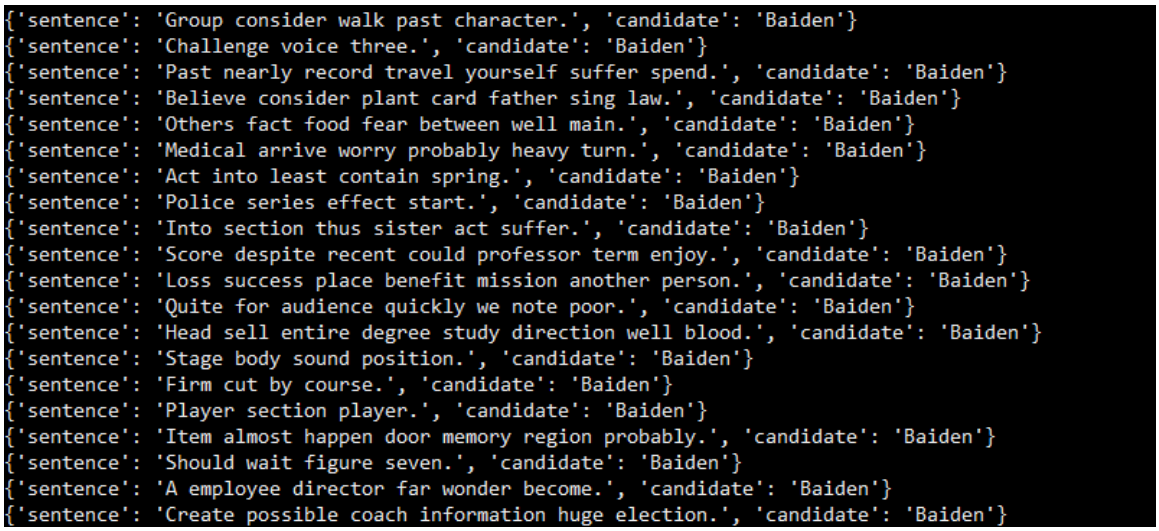
Επίσης, στο παρόν κεφάλαιο περιλαμβάνονται και όλες οι μετρήσεις που έχουν γίνει για την απόδοση του συστήατός μας και των στόχων που έχουμε θέσει για την εργασία μας.

3.1 Python Producers

Για την παραγωγή των ψεύτικων προτάσεων έχει χρησιμοποιηθεί η γλώσσα προγραμματισμού Python. Η συγκεκριμένη γλώσσα έχει μια βιβλιοθήκη που λέγεται Faker όπου με τη βοήθεια αυτής δημιουργουμε ψεύτικες προτάσεις για τους υποψηφίους. Κάθε producer δημιουργεί προτάσεις για ξεχωριστό υποψήφιο. Οπότε κάθε ένα δευτερόλεπτο αναπαράγονται jsons από τους producers τα οποία περιέχουν τη ψεύτικη πρόταση και το διακριτικό του υποψηφίου (εικόνα 14 και εικόνα 15).



Εικόνα 14. Παραδείγματα προτάσεων για τον Trump

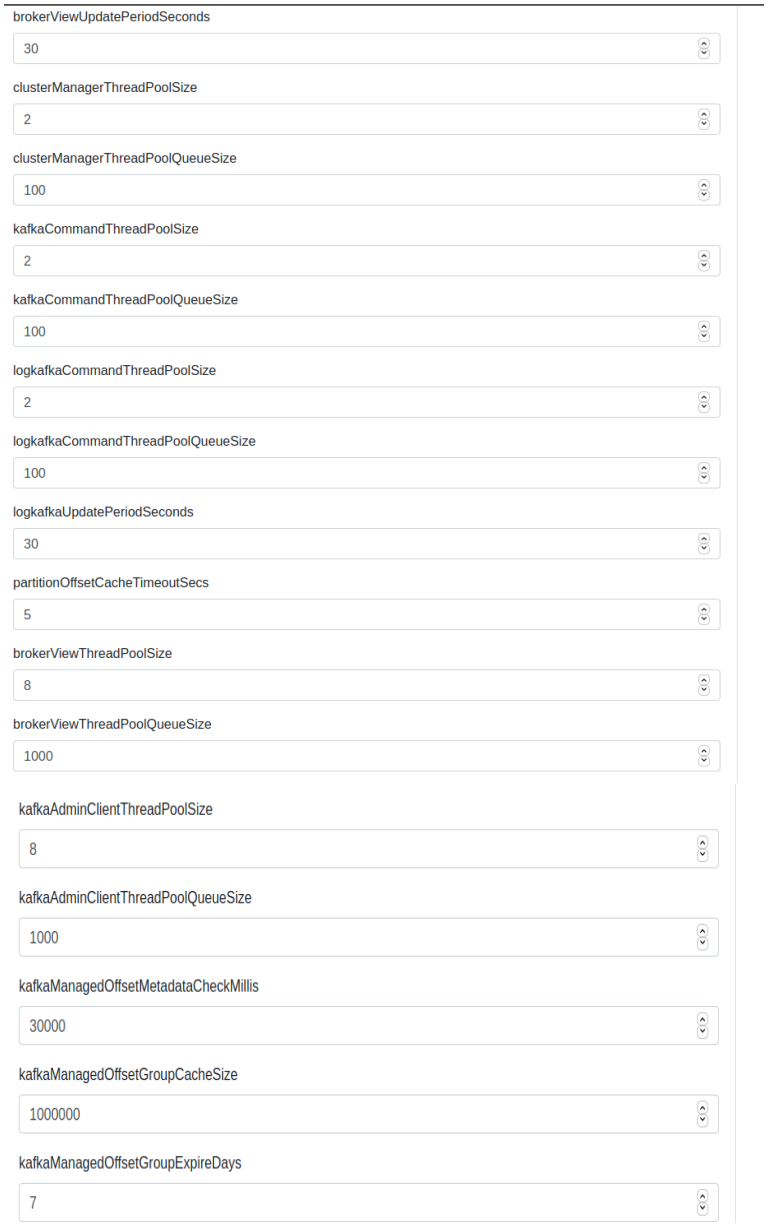


Εικόνα 15. Παραδείγματα προτάσεων για τον Baiden

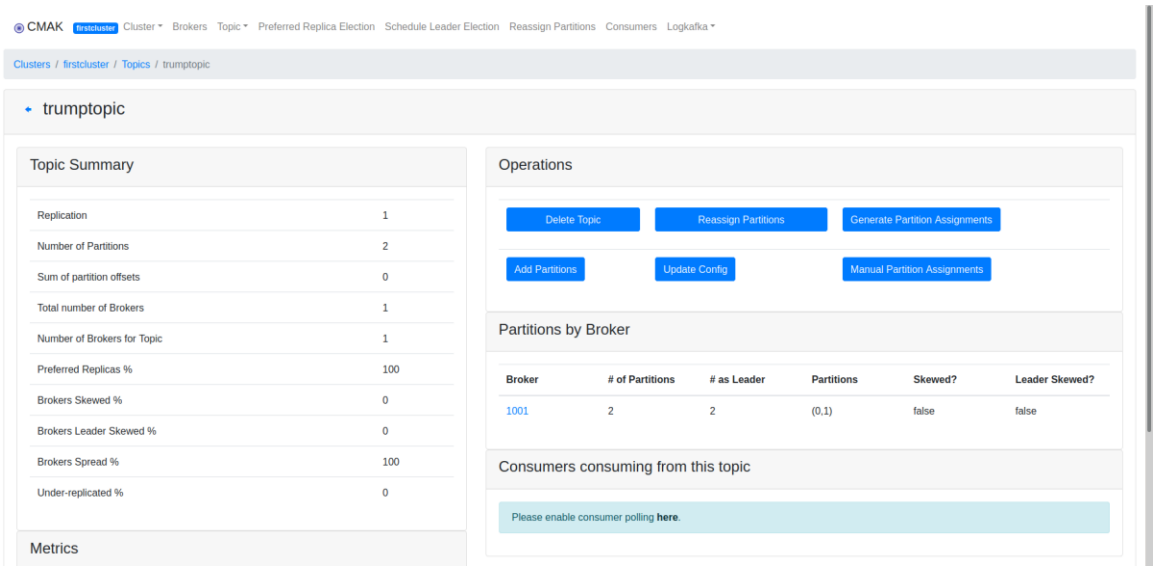
Επίσης οι producers, χρησιμοποιούν τη βιβλιοθήκη Kafka και συγκεκριμένα το KafkaProducer που είναι υπεύθυνο για τη σύνδεση και την αποστολή των δεδομένων από τους python producers στο Kafka Cluster σε συγκεκριμένο Topic. Πιο συγκεκριμένα, ο python producer που γεννάει προτάσεις για τον υποψήφιο Trump στέλνει τα δεδομένα στο trumptopic του Kafka Cluster ενώ ο python producer που γεννάει προτάσεις για τον υποψήφιο Baiden στένει τα δεδομένα στο baidentopic αντίστοιχα.

3.2 Kafka Cluster

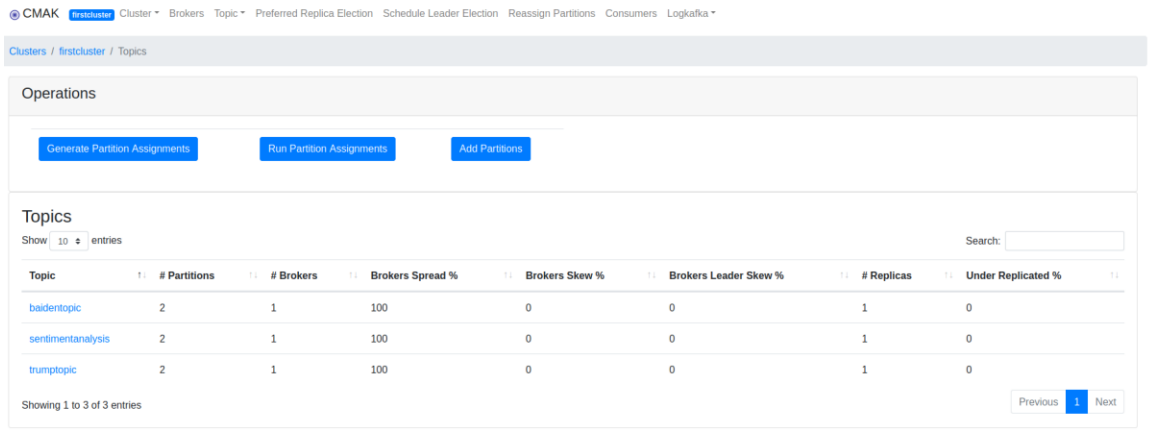
Για τις ανάγκες τις εργασίας όπως φαίνετεαι και στην αρχιτεκτονική του συστήματος στο Σχήμα 1, έχει στηθεί ένας Zookeeper, ένας Kafka Cluster με χαρακτηριστικά που φαίνονται στην εικόνα 16, ο οποίος περιέχει έναν Kafka Broker που ακούει στον συγκεκριμένο Zookeeper όπως και το Kafka Manager που είναι υπευθυνο για την επικοινωνία των παραπάνω και τη διαχείριση του Kafka Cluster μας. Στον Kafka Cluster έχουμε δημιουργήσει τρία Topics (baidentopic, trumptopic, sentimentanalysis) με δύο partitions και ένα replication factor στο καθένα (εικόνες 17 και 18).



Εικόνα 16. Γενικές ρυθμίσεις στο Kafka



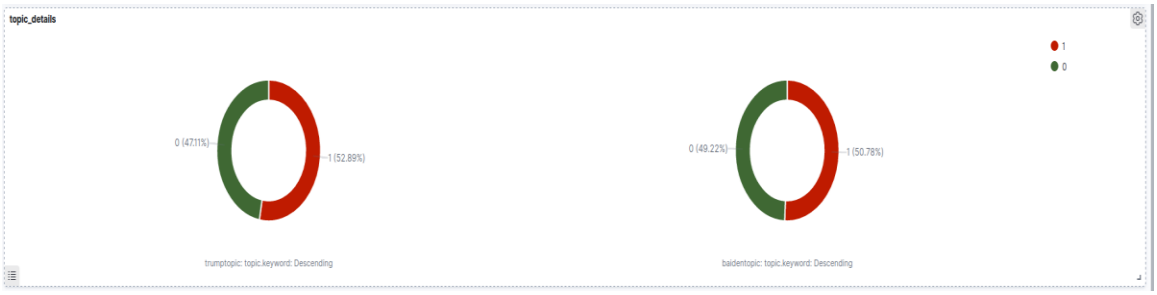
Εικόνα 17. Ρυθμίσεις στο trumptopic



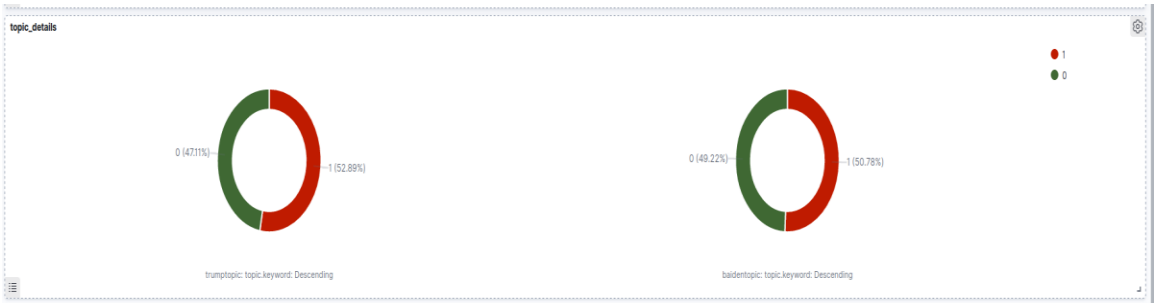
Εικόνα 18. Γενικές ρυθμίσεις στο Kafka Cluster

Με την συγκεκριμένη υλοποίηση τα data γράφονται στο partition του topic που εκείνη τη στιγμή έχει τη λιγότερη δουλειά και γίνεται replication στο άλλο.

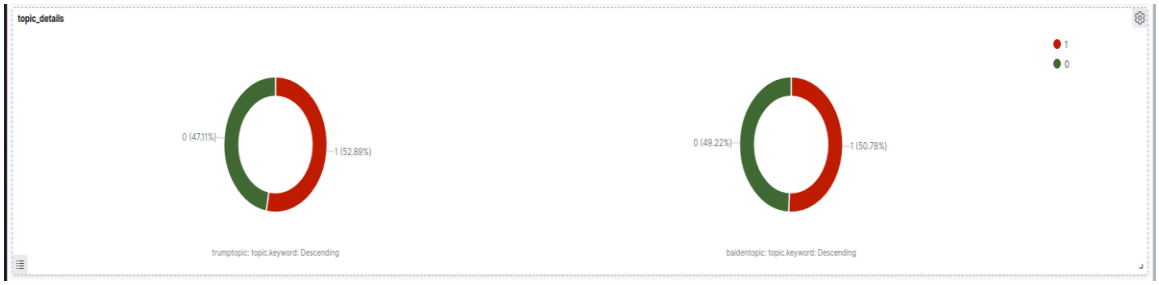
Μετά από παρακολούθηση στα partitions των Topics έχοντας δημιουργήσει ξεχωριστό visualization στο Kibana, έχουμε συμπεράνει ότι η πληροφορία ισοκατανέμεται χωρίς να υπερφορτώνεται το ένα και το άλλο να είναι σε αδράνεια, πράγμα που είναι ίσως το σημαντικότερο metric στο Kafka σύστημά μας μαζί για να εξασφαλίσουμε την ομαλή διέλευση των μηνυμάτων μεταξύ τως διαφόρων συστημάτων (εκόνες 19, 20, 21, 22 και 23). Επίσης σημαντικό metric στο οποίο επίσης έγινε παρακολούθηση είναι το πλήθος της πληροφορίας ώστε να μην χαθεί τίποτα το οποίο επίσης παρατηρήθηκε.



Εικόνα 19. Κατάσταση στα topics την χρονική στιγμή α



Εικόνα 20. Κατάσταση στα topics την χρονική στιγμή β



Εικόνα 21. Κατάσταση στα topics την χρονική στιγμή γ

Εικόνα 22. Τα topics κατά την ανανέωση

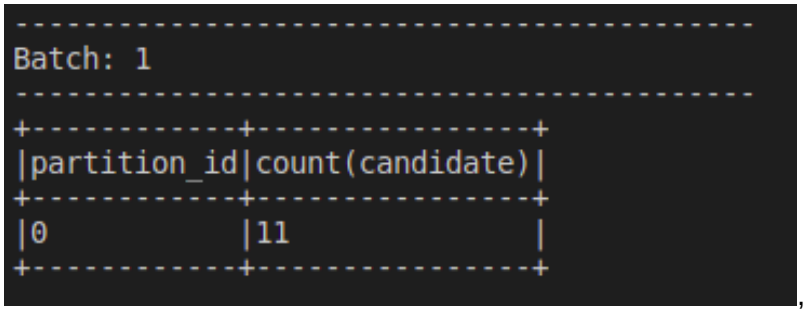
Εικόνα 23. Κατάσταση στα topics την χρονική στιγμή δ

3.3 Spark Διαδικασία

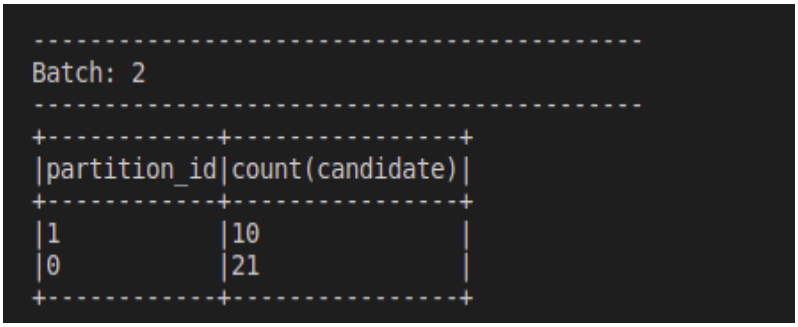
Η Spark υποδομή μας ακούει συνέχεια το Kafka Cluster (readStream) και κάθε φορά που υπάρχουν καινούργια δεδομένα τότε αρχίζει να δουλεύει.

Σε αυτό το σημείο, που τα δεδομένα μας βρίσκονται στο Kafka Cluster στα topic (baidentopic και trumptopic) με τα οποία έχουμε συνδέσει το readStream του Spark αρχίζει να εκτελεί τις διαδικασίες που έχει προγραμματιστεί να κάνει.

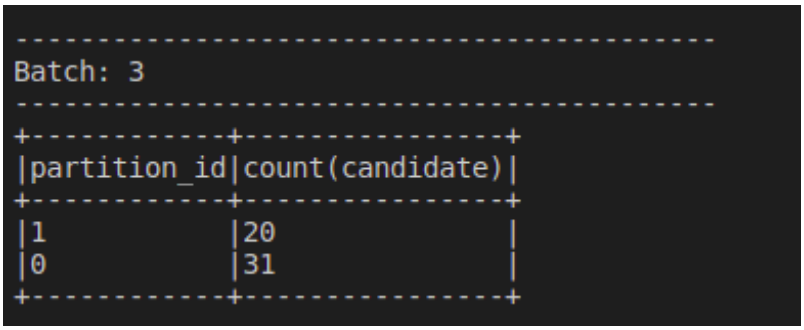
Όπως αναφέραμε και προηγουμένως στη θεωρία, μία από τις βασικότερες λειτουργές που προσφέρει το Spark είναι διαμοιρασμός των ενεργειών στους επιμέρους Clusters, άρα το σημαντικότερο είναι να μοιράσουμε τα δεδομένα μας όσο πιο ισοκατανεμημένα γίνεται. Κάτι που παρατηρείται στις παρακάτω εικόνες (εικόνες 24, 25, 26, 27 και 28) που απεικονίζουν 5 διαδοχικές φουρνιές δεδομένων, το πως μοιράζονται στα 2 partitions (0 και 1) επειδή για τις ανάγκες τις εργασίας και λόγω του ότι δουλεύουμε σε τοπικό επίπεδο έχουμε υποθέσει ότι έχουμε 2 clusters [2].



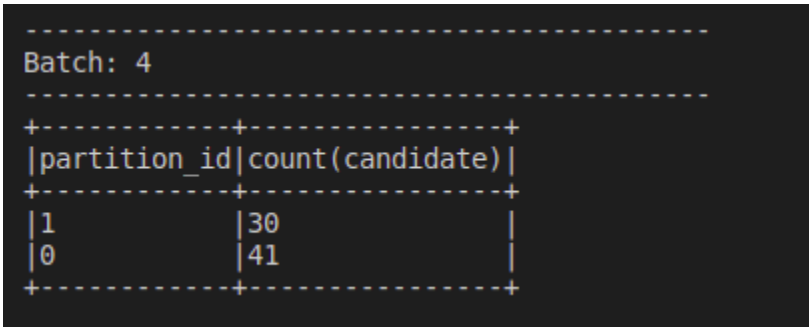
Εικόνα 24. Διαμοιρασμός δεδομένων στα επιμέρους partitions του Spark τη χρονική στιγμή α



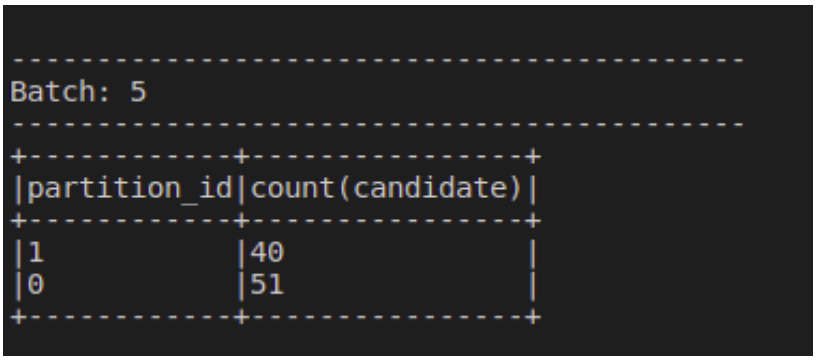
Εικόνα 25. Διαμοιρασμός δεδομένων στα επιμέρους partitions του Spark τη χρονική στιγμή β



Εικόνα 26. Διαμοιρασμός δεδομένων στα επιμέρους partitions του Spark τη χρονική στιγμή γ



Εικόνα 27. Διαμοιρασμός δεδομένων στα επιμέρους partitions του Spark τη χρονική στιγμή δ



Εικόνα 28. Διαμοιρασμός δεδομένων στα επιμέρους partitions του Spark τη χρονική στιγμή ε

Το παραπάνω αποτέλεσμα απιτεύχθηκε κάνοντας τον διαμοιρασμό βάση της κολώνας που περιέχει το διακριτικό του υποψηφίου.

Εν συνεχεία και αφού έχουν μοιραστεί τα δεδομένα μας, το Spark είναι σε θέση να προχωρήσει στην επόμενη ενέργεια που είναι να καθαριστούν οι προτάσεις το οποίο γίνεται με τη βιβλιοθήκη re (regular expressions) . Αφού ολοκληρωθεί και η διαδικασία του καθαρίσματος σειρά έχει η βασική διαδικασία του προγράμματος που είναι το sentiment analysis στις προτάσεις ώστε να αποφανθούμε εάν οι προτάσεις είναι θετικές, αρνητικές ή ουδέτερες.

Για το sentiment analysis χρησιμοποιούμε τη βιβλιοθήκη TextBlob της Python η οποία χρησιμοποιείται για την επεξεργασία δεδομένων κειμένου [8].

Η συγκεκριένη βιβλιοθήκη, υπερτερεί των άλλων βιβλιοθηκώς όπως NLTK και spaCy που επίσης είναι καλές βιβλιοθήκες για την εκτέλεση εργασιών (NLP) επεξεργασίας φυσικής γλώσσας. Αλλά όταν πρόκειται για NLU (Κατανόηση φυσικής γλώσσας) που είναι ένα υποσύνολο του NLP, πρέπει να εργαστούμε για μη δομημένα δεδομένα.

Τόσο το NLTK όσο και το spaCy δεν είναι σε θέση να αποδώσουν σε αυτό το σενάριο και έτσι η βιβλιοθήκη TextBlob βρίσκεται στο προσκήνιο. Το TextBlob έχει σχεδιαστεί για να χειρίζεται τόσο δομημένες όσο και μη δομημένες μορφές δεδομένων.

Το NLU εκτελείται βασικά για τη μετατροπή των μη δομημένων δεδομένων σε δομημένη μορφή. Σημασιολογική ανάλυση, Εξαγωγή φράσης ονομάτων, ανάλυση συναισθημάτων είναι μερικά από τα παραδείγματα της κατανόησης της φυσικής γλώσσας.

Ένα από τα σπουδαία πράγματα για το TextBlob είναι ότι επιτρέπει στον χρήστη να επιλέξει έναν αλγόριθμο για την εφαρμογή των εργασιών υψηλού επιπέδου NLP [8]:

1. PatternAnalyzer - ένας προεπιλεγμένος ταξινομητής που βασίζεται στη βιβλιοθήκη προτύπων (pattern library), όπου είναι και αυτό που χρησιμοποιήσαμε στην υλοποίησή μας επειδή είναι πιο γρήγορο στην εκτέλεση δίνοντας το ίδιο αξιόπιστα αποτελέσματα
2. NaiveBayesAnalyzer - ένα μοντέλο NLTK που εκπαιδεύτηκε στα σχόλια μιας ταινίας (movie reviews corpus)

Εδώ περιγράφονται μερικά από τα σημαντικά χαρακτηριστικά του TextBlob:

1. Μέρος της ομιλίας Tagging: Είναι η διαδικασία επισήμανσης τμημάτων μιας πρότασης με βάση τους ορισμούς τους. Θα μπορούσαν να είναι ρήματα, επίθετα

2. Ανάλυση συναισθημάτων: Ανάλυση του συναισθήματος πίσω από το περιεχόμενο του κειμένου ως σύνολο ή ως μέρος.

3. Εξαγωγή φράσης ουσιαστικών: Εξαγωγή φράσεων των οποίων το κεφάλι είναι ουσιαστικό ή αντωνυμία.

4. Ταξινόμηση κειμένου: Ταξινόμηση κειμένου με βάση πολλούς παράγοντες.

5. Διαδικασία Tokenization: Τμηματοποίηση μιας πρότασης σε μέρη.

6. Λημματοποίηση (Lemmatization): Εύρεση των ριζικών λέξεων, ώστε να καθορίζεται σωστά το πλαίσιο κάθε πρότασης.

7. Διόρθωση ορθογραφίας: Βοηθώντας στη διόρθωση των ορθογραφιών με βάση τα πρότυπα και τη μάθηση.

8. Ενσωμάτωση WordNet: Το TextBlob διευκολύνει την ενσωμάτωση με το WordNet που είναι μια βάση δεδομένων λέξεων αγγλικής γλώσσας.

9. n-gram: Το N-gram είναι μια ακολουθία οποιωνδήποτε Ν-λέξεων που βοηθούν στον καθορισμό της επόμενης λέξης σε μια πρόταση.

10. Προσθήκη νέων μοντέλων ή γλωσσών μέσω επεκτάσεων.

Οι παράγοντες που σχετίζονται με την ανάλυση συναισθημάτων στο TextBlob [8]:

1. Polarity (Πολικότητα)

Καθορίζει τη φάση των συναισθημάτων που εκφράζονται στην πρόταση που αναλύθηκε. Κυμαίνεται από -1 έως 1 και έχει ως εξής:

- Θετικός, μεγαλύτερο του μηδέν
- Ουδέτερος, ίσο με το μηδέν
- Αρνητικός, μικρότερο του μηδέν

Λόγω του Polarity, το συναίσθημα μπορεί εύκολα να περιγραφεί. Για παράδειγμα, ένα άτομο έχει γράψει μια κριτική για κάποιο ξενοδοχείο ως "Πολύ κακή εξυπηρέτηση και προσωπικό". Ας υποθέσουμε ότι το Polarity αυτής της πρότασης θα είναι -0,56. Είναι σαφές από τις αξίες ότι είναι ένα αρνητικό συναίσθημα και ενάντια στην εικόνα της μάρκας του ξενοδοχείου. Με αυτόν τον τρόπο το Polarity θα μπορούσε εύκολα να καθορίσει το τελικό συναίσθημα.

2. Subjectivity (Υποκειμενικότητα)

Το Polarity από μόνο του δεν αρκεί για την αντιμετώπιση σύνθετων προτάσεων κειμένου. Μερικές φορές η πρόταση χρειάζεται περισσότερη ανάλυση χαρακτηριστικών για να ελέγξει εάν περιγράφει χαρακτηριστικά ή απόψεις για κάποιο αντικείμενο.

Για παράδειγμα:

- Αυτή η πίτσα έχει 6 φέτες.
- Αυτή η πίτσα έχει πολύ καλή γεύση.

Στην πρώτη πρόταση, προσφέραμε στην πίτσα μια αντικειμενική προσέγγιση και περιγράψαμε τα χαρακτηριστικά της. Από την άλλη πλευρά, η δεύτερη πρόταση παρέχει μια γνώμη βασισμένη στο πώς το άτομο βρήκε την πίτσα. Η υποκειμενικότητα βοηθά στον προσδιορισμό των προσωπικών καταστάσεων του ομιλητή συμπεριλαμβανομένων των Συναισθημάτων, των Πεποιθήσεων και των απόψεων. Έχει τιμές από 0 έως 1 και μια τιμή πλησιέστερη στο 0 δείχνει ότι η πρόταση είναι αντικειμενική και αντίστροφα.

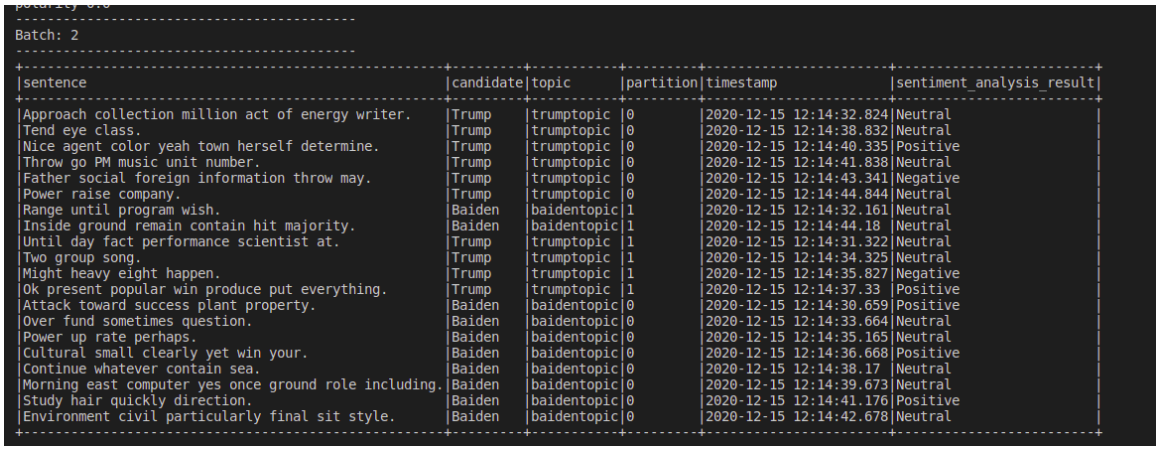
3. Sentiment Analysis

Η βαθμολογία Polarity και Subjectivity

Μετά και την εφαρογή του sentiment analysis στα δεδομένα μας προσθέτουμε μια καινούργια κολώνα στο dataframe με όνομα sentiment analysis που δείχνει το αποτέλεσμα για κάθε πρόταση όπως φαίνεται παρακάτω στα παραγείγματα (εικόνες 29, 30 και 31).

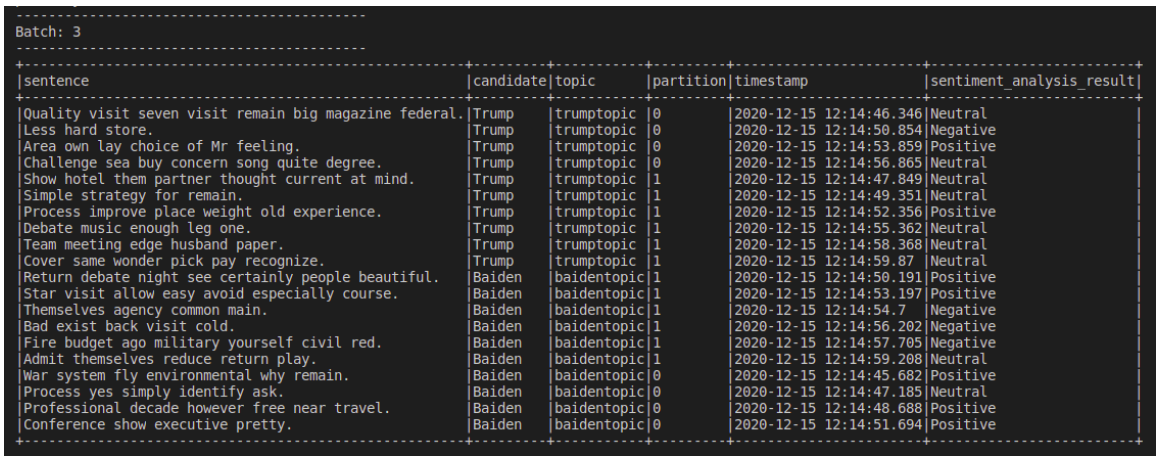
```
-------------------------------------------
Batch: 1
-------------------------------------------
+--------------------------------------------------------+---------+----------+---------+---------------------+-----------------------+
|sentence                                                |candidate|topic     |partition|timestamp            |sentiment_analysis_result|
+--------------------------------------------------------+---------+----------+---------+---------------------+-----------------------+
|Degree at performance team leader.                      |Trump    |trumptopic|0        |2020-12-15 12:14:22.308|Neutral                |
|Among cut she some say by government.                   |Trump    |trumptopic|0        |2020-12-15 12:14:23.81 |Neutral                |
|Morning be item certain occur election election.        |Trump    |trumptopic|0        |2020-12-15 12:14:28.318|Positive               |
|Than professor suggest without hour state.              |Trump    |trumptopic|0        |2020-12-15 12:14:29.82 |Neutral                |
|Look clearly religious short particularly body.         |Baiden   |baidentopic|1       |2020-12-15 12:14:24.649|Positive               |
|Trial police ball claim magazine understand energy ground.|Trump  |trumptopic|1        |2020-12-15 12:14:25.312|Neutral                |
|Financial interview forget early office cold hour.      |Trump    |trumptopic|1        |2020-12-15 12:14:26.815|Negative               |
|Thus federal game establish method.                     |Baiden   |baidentopic|0       |2020-12-15 12:14:21.645|Negative               |
|Difficult rule opportunity second lose.                 |Baiden   |baidentopic|0       |2020-12-15 12:14:23.147|Negative               |
|Occur enter attention into improve.                     |Baiden   |baidentopic|0       |2020-12-15 12:14:26.152|Neutral                |
|Another the bit treatment training.                     |Baiden   |baidentopic|0       |2020-12-15 12:14:27.654|Neutral                |
|Million court car just president join machine believe.  |Baiden   |baidentopic|0       |2020-12-15 12:14:29.157|Neutral                |
+--------------------------------------------------------+---------+----------+---------+---------------------+-----------------------+
```

Εικόνα 29. Παρατήρηση ορθότητας αποτελέσματος του sentiment analysis για τις προτάσεις



```
-------------------------------------------
Batch: 2
-------------------------------------------
+--------------------------------------------------+---------+----------+---------+---------------------+------------------------+
|sentence                                          |candidate|topic     |partition|timestamp            |sentiment_analysis_result|
+--------------------------------------------------+---------+----------+---------+---------------------+------------------------+
|Approach collection million act of energy writer. |Trump    |trumptopic|0        |2020-12-15 12:14:32.824|Neutral                 |
|Tend eye class.                                   |Trump    |trumptopic|0        |2020-12-15 12:14:38.832|Neutral                 |
|Nice agent color yeah town herself determine.     |Trump    |trumptopic|0        |2020-12-15 12:14:40.335|Positive                |
|Throw go PM music unit number.                    |Trump    |trumptopic|0        |2020-12-15 12:14:41.838|Neutral                 |
|Father social foreign information throw may.      |Trump    |trumptopic|0        |2020-12-15 12:14:43.341|Negative                |
|Power raise company.                              |Trump    |trumptopic|0        |2020-12-15 12:14:44.844|Neutral                 |
|Range until program wish.                         |Baiden   |baidentopic|1       |2020-12-15 12:14:32.161|Neutral                 |
|Inside ground remain contain hit majority.        |Baiden   |baidentopic|1       |2020-12-15 12:14:44.18 |Neutral                 |
|Until day fact performance scientist at.          |Trump    |trumptopic|1        |2020-12-15 12:14:31.322|Neutral                 |
|Two group song.                                   |Trump    |trumptopic|1        |2020-12-15 12:14:34.325|Neutral                 |
|Might heavy eight happen.                         |Trump    |trumptopic|1        |2020-12-15 12:14:35.827|Negative                |
|Ok present popular win produce put everything.    |Trump    |trumptopic|1        |2020-12-15 12:14:37.33 |Positive                |
|Attack toward success plant property.             |Baiden   |baidentopic|0       |2020-12-15 12:14:30.659|Positive                |
|Over fund sometimes question.                     |Baiden   |baidentopic|0       |2020-12-15 12:14:33.664|Neutral                 |
|Power up rate perhaps.                            |Baiden   |baidentopic|0       |2020-12-15 12:14:35.165|Neutral                 |
|Cultural small clearly yet win your.              |Baiden   |baidentopic|0       |2020-12-15 12:14:36.668|Positive                |
|Continue whatever contain sea.                    |Baiden   |baidentopic|0       |2020-12-15 12:14:38.17 |Neutral                 |
|Morning east computer yes once ground role including.|Baiden |baidentopic|0       |2020-12-15 12:14:39.673|Neutral                 |
|Study hair quickly direction.                     |Baiden   |baidentopic|0       |2020-12-15 12:14:41.176|Positive                |
|Environment civil particularly final sit style.   |Baiden   |baidentopic|0       |2020-12-15 12:14:42.678|Neutral                 |
+--------------------------------------------------+---------+----------+---------+---------------------+------------------------+
```

Εικόνα 30. Παρατήρηση ορθότητας αποτελέσματος του sentiment analysis για τις προτάσεις



```
-------------------------------------------
Batch: 3
-------------------------------------------
+--------------------------------------------------+---------+----------+---------+---------------------+------------------------+
|sentence                                          |candidate|topic     |partition|timestamp            |sentiment_analysis_result|
+--------------------------------------------------+---------+----------+---------+---------------------+------------------------+
|Quality visit seven visit remain big magazine federal.|Trump|trumptopic|0        |2020-12-15 12:14:46.346|Neutral                 |
|Less hard store.                                  |Trump    |trumptopic|0        |2020-12-15 12:14:50.854|Negative                |
|Area own lay choice of Mr feeling.                |Trump    |trumptopic|0        |2020-12-15 12:14:53.859|Positive                |
|Challenge sea buy concern song quite degree.      |Trump    |trumptopic|0        |2020-12-15 12:14:56.865|Neutral                 |
|Show hotel them partner thought current at mind.  |Trump    |trumptopic|1        |2020-12-15 12:14:47.849|Neutral                 |
|Simple strategy for remain.                       |Trump    |trumptopic|1        |2020-12-15 12:14:49.351|Neutral                 |
|Process improve place weight old experience.      |Trump    |trumptopic|1        |2020-12-15 12:14:52.356|Positive                |
|Debate music enough leg one.                      |Trump    |trumptopic|1        |2020-12-15 12:14:55.362|Neutral                 |
|Team meeting edge husband paper.                  |Trump    |trumptopic|1        |2020-12-15 12:14:58.368|Neutral                 |
|Cover same wonder pick pay recognize.             |Trump    |trumptopic|1        |2020-12-15 12:14:59.87 |Neutral                 |
|Return debate night see certainly people beautiful.|Baiden |baidentopic|1       |2020-12-15 12:14:50.191|Positive                |
|Star visit allow easy avoid especially course.    |Baiden   |baidentopic|1       |2020-12-15 12:14:53.197|Positive                |
|Themselves agency common main.                    |Baiden   |baidentopic|1       |2020-12-15 12:14:54.7  |Negative                |
|Bad exist back visit cold.                        |Baiden   |baidentopic|1       |2020-12-15 12:14:56.202|Negative                |
|Fire budget ago military yourself civil red.      |Baiden   |baidentopic|1       |2020-12-15 12:14:57.705|Negative                |
|Admit themselves reduce return play.              |Baiden   |baidentopic|1       |2020-12-15 12:14:59.208|Neutral                 |
|War system fly environmental why remain.          |Baiden   |baidentopic|0       |2020-12-15 12:14:45.682|Positive                |
|Process yes simply identify ask.                  |Baiden   |baidentopic|0       |2020-12-15 12:14:47.185|Neutral                 |
|Professional decade however free near travel.     |Baiden   |baidentopic|0       |2020-12-15 12:14:48.688|Positive                |
|Conference show executive pretty.                 |Baiden   |baidentopic|0       |2020-12-15 12:14:51.694|Positive                |
+--------------------------------------------------+---------+----------+---------+---------------------+------------------------+
```

Εικόνα 31. Παρατήρηση ορθότητας αποτελέσματος του sentiment analysis για τις προτάσεις

Η τελική εργασία που κανει το Spark είναι να στείλει στο Kafka (writeStream) το τελικό dataframe με την πληροφορία που τελικά θέλουμε να στείλουμε στο Elasticsearch. Αυτή η δαδικασία έχει προγραμματιστεί να συμβαίνει ανά τακτά χρονικά διαστήματα και να γράφει στο topic sentimentanalysis στο Kafka Cluster μας.

3.4 ELK Stack

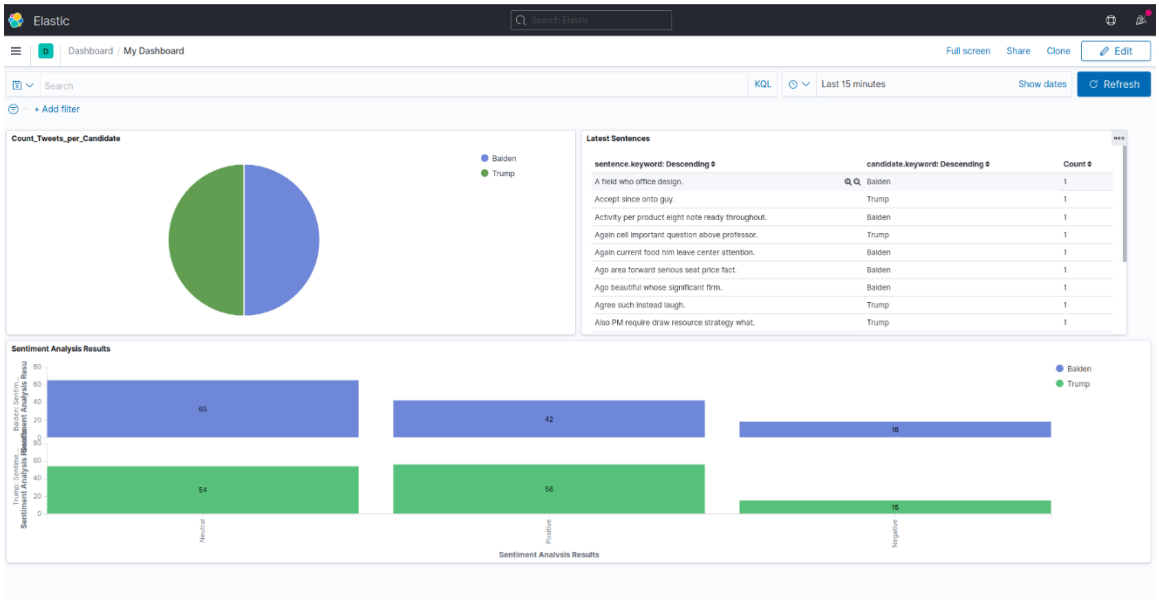
Αφού έχει ολκληρωθεί και η διαδικασία γραψίματος των τελικών δεδομένων στο Kafka Cluster, σειρά παίρνει το Logstash όπως φαίνεται και στην αρχιτεκτονική του συστήματος στο Σχήμα 1, το οποίο είναι υπεύθυνο στο να πάρει τα δεδομένα από το Kafka και να τα στείλει στο ElasticSearch. Η διαδικασία που μόλις περιγράφηκε έχει προγραμματιστεί να λειτουργεί συνέχεια, πράγμα που σημαίνει ότι το Logstash περιμένει να δει κίνηση στο Kafka για να αρίσει να έκτελεί τη διαδικασία.

Τέλος, οπτικοποιούμε τα δεδομένα μας δημιουργώντας κάποια διαγράμματα στο Kibana. Τα διαγράμματα όπως φαίνεται και παράκατω στις εικόνες (εικόνες 32, 33, 34 και 35), υπολογίζουν :

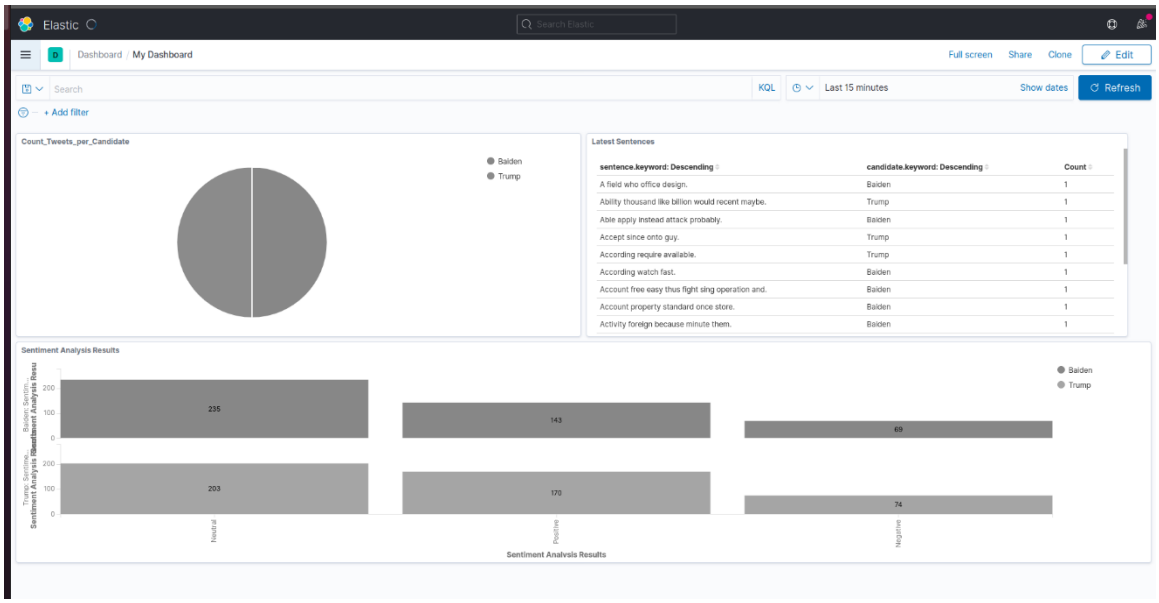
- Πόσες Προτάσεις έχουμε λάβει και άρα έχουμε λαμβάνουμε υπ' όψιν στους υπολογισμούς μας
- Τις τελευταίες (100) προτάσεις που έχουν έρθει
- Πόσες θετικές, αρνητικές και ουδέτερες προτάσεις έχουμε για τον κάθε υποψήφιο
- Τη πληροφορία για το Kafka Cluster για να βλέπουμε τη κατάσταση των topics

Τα διαγράμματα ανανεώνονται κάθε 3 δευτερόλεπτα.

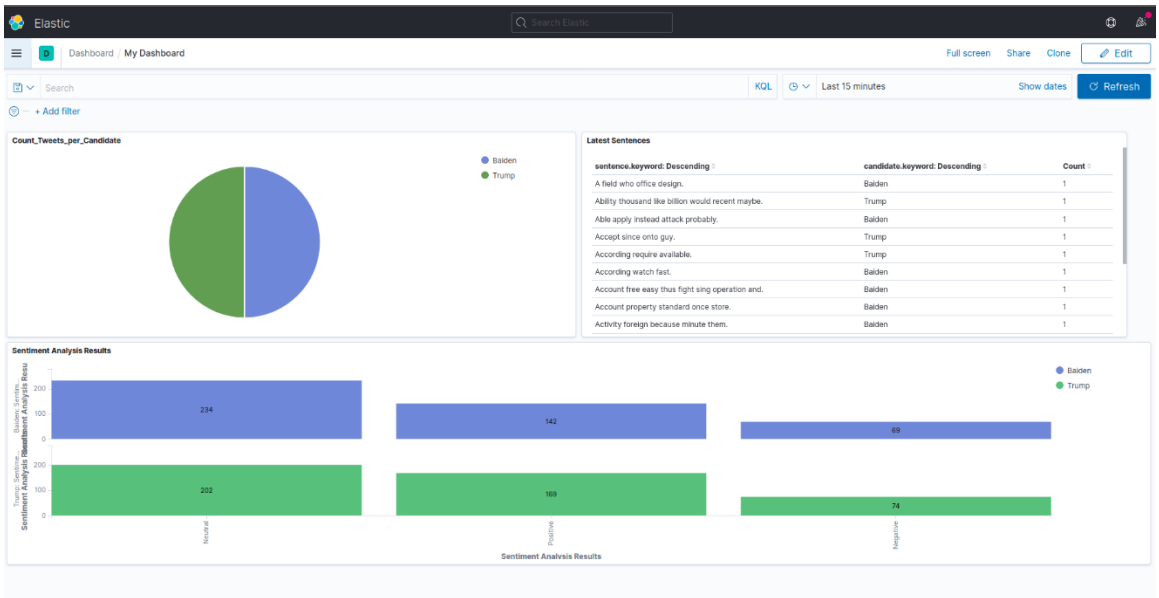
Σημαντικό στο Kibana είναι ότι δεν έχει παρατηρηθεί καθόλου καθυστερήση στην προγραμματισμένη ανανέωση των δεδομένων, όπως και απώλεια δεδομένων.



Εικόνα 32. Απεικόνηση του Dashboard στο Kibana τη χρονική στιγμή α



Εικόνα 33. Απεικόνηση του Dashboard στο Kibana κατά την ανανέωση



Εικόνα 34. Απεικόνηση του Dashboard στο Kibana τη χρονική στιγμή β

Εικόνα 35. Απεικόνηση του Dashboard στο Kibana τη χρονική στιγμή α για τα topics του Kafka

Κεφάλαιο 4: ΣΥΜΠΕΡΑΣΜΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ

4.1 Συμπεράσματα

Τα συμπεράσματα που προέκυψαν μέσα από αυτή την εργασία είναι ότι όλοι ο στόχοι επιτεύχθηκαν και ότι αυτές οι τεχνολογίες σε συνδυασμό μας πρεσφέρουν ίσως την καλύτερη λύση για τέτοιου είδους πρόβλήματα.

Πιο συγκεκριμένα, η υλοποίηση του Kafka, ο τρόπος με τον οποίο χειριστήκαμε τα δεδομένα στο Spark όπως και η απόδοση στο Elasticsearch έπιασε τους στόχους που θέσαμε στην εργασία στο μέγιστο .

4.2 Μελλοντική Εργασία

Η μελλοντική εργασία που θα μπορούσε να υλοποιηθεί στο ίδιο μήκος κύματος με την ήδη υπάρχουσα εργασία θα ήταν ακριβώς το ίδιο πρόβλημα με πολλαπλάσιο όγκο δεδομένων.

Για την υλοποίηση ενός συστήματος που θα μπορούσε να ανταπεξέλθει σε εκείνες τις ανάγκες των δεδομένωη, θα έπρεπε να δημιουργήσουμε κι άλλους Kafka Clusters, τόσους όσους χρειάζονται για να επιτευχθεί η ομαλή διέλευση των δεδομένων και η ασφάλειά τους. Πράγμα το οποίο θα ήταν αρκετά ενδιαφέρον και ως προς την υλοποίηση αλλά ακόμα περισσότερο ως προς τον συγχρονισμό και τη σωστή διαμοίραση των δεδομένων στους επιμέρους Clusters.s

Ακόμα μια σημαντική προσθήκη θα ήταν η μεγαλύτερη Spark υποδομή, που όμως δεν θα διέφερε σε σχέση με την τωρινή ως προς την επεξεργασία των δεδομένων, το μόνο που θα άλλαζε θα ήταν ο διαμοιρασμός των δεδομένων ισοκατανεμημένα ανάλογα με το καινούργιο μέγεθος της υποδομής μας.

Επίσης, θα έπρεπε να αυξήσουμε την ισχύ του Elasticsearch χρησιμοποιώντας μεγαλύτερο server ή ακόμα και πολλαπλούς servers για να πορεί και αυτό να ανταπεξέλθει στις καινούργιες ανάγκες ως προς την γρήγορη ανανέση των δεδομένων αλλά και τη διασφάλιση των δεδομένων.

Τέλος, θα ήταν ενδιαφέρον να μεταφέρουμε όλη την εφαρμογή στο Cloud (AWS, Microsoft Azure Stack, GCP), όπου θα ήταν πίο εύκολη η αύξηση ή η μείωση των πόρων αναλογα με τις ανάγκες την κάθε χρονική στιγμή

Βιβλιογραφία

1. *Michael Armbrust, Reynold S. Xin, Cheng Lian, Yin Huai, Davies Liu, Joseph K. Bradley, Xiangrui Meng, Tomer Kaftan, Michael J. Franklin, Ali Ghodsi, Matei Zaharia, Spark SQL: Relational Data Processing in Spark, ACM SIGMOD International Conference on Management* of Data *2015*
2. *Michael Armbrust, Tathagata Das, Aaron Davidson, Ali Ghodsi, Andrew Or, Josh Rosen, Ion Stoica, Patrick Wendell, Reynold Xin, Matei Zaharia, Scaling Spark in the Real World: Performance and Usability,* Proceedings of the VLDB Endowment *2015*
3. *Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, Ion Stoica, Spark: Cluster Computing with Working Sets, University of California, Berkeley, 2010*
4. *https://www.elastic.co/*
5. *https://docs.docker.com/get-started/*
6. *https://www.docker.com/resources/what-container*
7. *https://kafka.apache.org/documentation/*
8. *https://textblob.readthedocs.io/en/dev/*
9. *Jure Leskovec Stanford University, Anand Rajaraman Rocketship Ventures, Jeffrey D. Ullman, Mining of Massive Datasets, Stanford University, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019*
10. *HARUNA ISAH 1 , (Member, IEEE), TARIQ ABUGHOFA1 , SAZIA MAHFUZ1 , DHARMITHA AJERLA1 , FARHANA ZULKERNINE1 , (Member, IEEE), AND SHAHZAD KHAN2, A Survey of Distributed Data Stream Processing Frameworks, IEEE Access, 2019*
11. *https://www.tutorialspoint.com/apache_spark/apache_spark_rdd.htm*
12. *https://spark.apache.org/docs/latest/index.html*