ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Σύγκριση Streaming Big Data πλατφόρμων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Βραχασωτάκη Μιχαήλ

Επιβλέπων: Νεκτάριος Κοζύρης
Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος 2017

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΗΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Σύγκριση Streaming Big Data πλατφόρμων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Βραχασωτάκη Ν. Μιχαήλ

Επιβλέπων: Νεκτάριος Κοζύρης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 10η Νοεμβρίου 2017.

.....	.....	.....
Νεκτάριος Κοζύρης Καθηγητής Ε.Μ.Π.	Γεώργιος Γκούμας Επίκουρος Καθηγητής Ε.Μ.Π.	Δημήτριος Τσουμάκος Αν. Καθηγητής Ιόνιου Πανεπιστημίου

Αθήνα, Νοέμβριος 2017

.....

Μιχαήλ Ν. Βραχασωτάκης
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών

Περίληψη

Η επέλαση των Big Data και ο ρυθμός που αυτά παράγονται κάθε στιγμή έχει κάνει επιτακτική ανάγκη την χρήση κατανεμημένων streaming engines και πλέον υπάρχουν διάφορα εργαλεία στο ανοιχτό λογισμικό. Δύο γνωστά από αυτά, τα Apache Spark και Apache Flink, θα συγκριθούν για την συγκεκριμένη λειτουργία που προσφέρουν. Η υλοποίηση γίνεται σε ένα μικρό cluster με την βοήθεια έτοιμων benchmarks και μετρικές είναι η μέση καθυστέρηση και το ποσοστό των δεδομένων που προκαλούν την υψηλότερη καθυστέρηση.

Λέξεις-κλειδιά:Κατανεμήμένα συστήματα, Big Data, streaming engines, real-time δεδομένα, Apache Spark, Apache Flink

Abstract

The oncoming assault of Big Data and their creation rate every moment have rendered necessary the utilisation of distributed streaming engines. There exist many open source tools for this purpose and two heavily-utilised, Apache Spark and Apache Flink, will be compared to determine the best one for a particular streaming scenario. A small cluster will be used for this experiment with the help of open benchmarks and the metrics are median latency and the percentage of data that contribute to the highest latency.

Keywords: Distributed systems, Big Data, streaming engines, real-time data, Apache Spark, Apache Flink

Ευχαριστίες

Ευχαριστώ την κυρία Κατερίνα Δόκα για την υπομονή και την καθοδήγησή της, τον κύριο Κοζύρη και τους φίλους και συναδέλφους που συνέβαλαν όλα αυτά τα χρόνια της φοίτησης.

Περιεχόμενα

1 **9**
1.1 Εισαγωγή 9
1.1.1 Εισαγωγικές πληροφορίες 9
1.1.2 Εργαλεία 10
1.1.3 Δοκιμαστικά Προγράμματα 10
1.1.4 Μετρική 10

2 **11**
2.1 Εργαλεία 11
2.1.1 Apache Spark 11
2.1.2 Apache Flink 12

3 **14**
3.1 Πειράματα 14
3.2 Αρχιτεκτονική 14
3.3 Προγράμματα 16
3.4 Μετρήσεις 18
3.5 Διαγράμματα 18

4 **25**
4.1 Συμπεράσματα 25

Κεφάλαιο 1

1.1 Εισαγωγή

1.1.1 Εισαγωγικές πληροφορίες

Οι σημερινές ταχύτητες παραγωγής δεδομένων λόγω των δισεκατομμυρίων ανθρώπων που καθημερινά κατεβάζουν και ανεβάζουν ασύλληπτα μεγέθη, γνωστά ως Big Data, έχουν δημιουργήσει την ανάγκη ύπαρξης εργαλείων για την άμεση και χωρίς λάθη επεξεργασία αυτών και η προγραμματιστική κοινότητα έχει φροντίσει να υπάρχουν εργαλεία ανοιχτού λογισμικού. Με αυτόν τον τρόπο, μπορούν οι διάφοροι οργανισμοί, εταιρίες αλλά και άτομα να δημιουργήσουν τα δικά τους εξατομικευμένα συστήματα με ιδιαίτερη ευκολία και να εξάγουν ενδιαφέροντα συμπεράσματα για τις συμπεριφορές χρηστών, για τάσεις που δημιουργούνται ανά διάφορες χρονικές περιόδους και ως ενδείξεις του τρόπου προσαρμογής των υπηρεσιών τους.

Λόγω αυτού του φαινομένου, την τελευταία δεκαπενταετία μια ιδέα απο το 1970 βελτιώθηκε και χρησιμοποιήθηκε για να βοηθήσει στην εκμετάλλευση των τεράστιων φορτίων με δεδομένα που παράγονται κάθε μέρα λόγω της εξάπλωσης των υπολογιστών. Πρόκειται για τα κατανεμημένα δίκτυα, αποτελούμενα από μηχανήματα με μέτρια χαρακτηριστικά, συνδεδεμένα σε ένα δίκτυο, που επικοινωνόντας μεταξύ τους με διάφορες τεχνικές δίνουν την εντύπωση ενός γρήγορου και αξιόπιστου συστήματος. Οι χρήσεις τους ποικίλλουν και συνήθως θα βρεθούν στο cloud computing, ως παράλληλα συστήματα για απαιτητικούς υπολογισμούς, σαν τεράστιες, γρήγορες και ανθεκτικές στις αποτυχίες απομακρυσμένες βάσεις δεδομένων, ως δικτυακά συστήματα αρχείων και γενικά για διάφορες υπηρεσίες.

Οι Streaming Big Data πλατφόρμες είναι κατεξοχήν κατανεμημένες και χρησιμοποιούνται για real-time processing. Αυτή η τακτική επιτρέπει την επεξεργασία δεδομένων την στιγμή που λαμβάνονται, αντίθετα με το batch processing που είναι το κυρίαρχο πρότυπο τα τελευταία χρόνια. Υπάρχουν διάφορες υλοποιήσεις στο ανοιχτό λογισμικό όπως τοπολογίες, γράφοι και μικρο-εργασίες και πολλές χρησιμοποιούνται ευρέως.

1.1.2 Εργαλεία

Στην παρούσα εργασία θα μελετηθούν δύο αρκετά γνωστά open-source εργαλεία, το καθένα με διαφορετική διάρκεια ζωής. Πρόκειται για το Apache Spark, που έχει εδραιωθεί στην αγορά και θεωρείται αρκετά πετυχημένο και ακολουθεί το Apache Flink αποτελώντας το νεώτερο και πολλά υποσχόμενο μέλος αυτής της ομάδας. Σκοπός είναι η σύγκριση τους στο συγκεκριμένο σενάριο όπου τα δεδομενα παράγονται στιγμιαία. Επιπλέον, χρησιμοποιήθηκαν τα open-source frameworks Apache Zookeeper, Redis και Apache Kafka για τις λειτουργίες συγχρονισμού, αποθήκευσης και μεταφοράς δεδομένων αντίστοιχα και το Openstack για την διαχείριση του συστήματος.

1.1.3 Δοκιμαστικά Προγράμματα

Υπάρχει ανοιχτός και ελεύθερος κώδικας για την χρήση των εργαλείων σε τοπική λειτουργία, δηλαδή ένα μηχάνημα, έχοντας έτοιμες τις εξαρτήσεις για το καθένα και την δημιουργία και αποθήκευση των δεδομένων. Το συγκεκριμένο σύστημα θα προσαρμοστεί για τις ανάγκες της εργασίας ώστε να υποστηρίζει πολλά μηχανήματα στο ίδιο δίκτυο.

1.1.4 Μετρική

Τα αποτελέσματα είναι ο χρόνος για την ολοκλήρωση της επεξεργασίας του τελευταίου από τα δεδομένα που εισήλθε στο σύστημα για το συγκεκριμένο χρονικό διάστημα και ο αριθμός των γεγονότων που εισέρχονται σε κάθε χρονική περίοδο στο σύστημα. Από αυτά μπορούμε να εξάγουμαι κάποια άλλα χρήσιμα χαρακτηριστικά όπως η μέση καθυστέρηση και το ποσοστό των δεδομένων που προκαλούν υψηλή καθυστέρηση, η οποία θα χρησιμοποιηθεί για τα συμπεράσματα μαζί με τον ρυθμό παραγωγής των δεδομένων και τον αριθμό των μηχανημάτων.

Κεφάλαιο 2

2.1 Εργαλεία

Αρχικά, θα παρουσιαστούν τα κοινά στοιχεία για το κάθε εργαλείο και θα ακολουθήσει περιγραφή του κάθε εργαλείου και των χαρακτηριστικών του. Το κάθε εργαλείο παρέχει διάφορα βοηθητικά στοιχεία για τον χρήστη. Αυτά αποτελούνται από μία εσωτερική ιστοσελίδα με διάφορες πληροφορίες για την εκάστοτε δουλειά, τους κόμβους, άμεση πρόσβαση στα αρχεία καταγραφής γεγονότων και λαθών και γενικά χαρακτηριστικά για την συνολική λειτουργία του. Επίσης, υπάρχουν μία ιστοσελίδα με εκτενή κείμενα και παραδείγματα για την λειτουργία του καθενός και μία αρκετά μεγάλη κοινότητα που τα χρησιμοποιεί ώστε να είναι δυνατή η ύπαρξη βοήθειας στα περισσότερα ζητήματα. Όλα υποστηρίζουν σύνδεση με άλλα κατανεμημένα εργαλεία που μπορούν να χρησιμοποιηθούν για κάποιες πιο εξειδικευμένες λειτουργίες όπως αποθήκευση σε βάσεις δεδομένων. Για την χρήση τους σε ένα cluster υποστηρίζουν διάφορα εργαλεία διαχείρισης αλλά και ατομική λειτουργία.

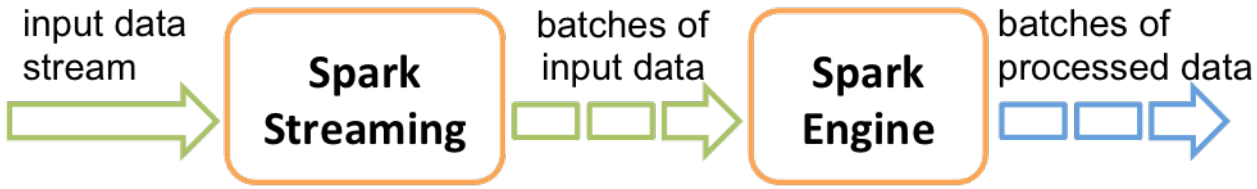
2.1.1 Apache Spark

Κύριο χαρακτηριστικό του είναι τα Resilient Distributed Datasets(RDDs), ένα πολυσύνολο κατανεμημένων δεδομένων που είναι μόνο αναγνώσιμο και ανθεκτικό στα λάθη σημειώνοντας την ακολουθία των πράξεων που το παρήγαγαν και εφαρμόζοντας μία καθυστέρηση στην εφαρμογή κάθε πράξης μέχρις ώτου αυτή χρειάζεται. Παρέχει πολλές γλώσσες για τον χειρισμό του με δικό του κέλυφος για όσες από αυτές είναι γλώσσες σεναρίου και ένα εύκολο σύστημα χρήσης. Τέλος, περιλαμβάνει βιβλιοθήκες για πράξεις σε γραφήματα, μηχανική μάθηση, χρήση ερωτήσεων παρόμοιων με βάσεις δεδομένων και επεξεργασία ζωντανών δεδομένων. Η τελευταία που χρησιμοποιείται σε αυτή την εργασία υλοποιεί την τεχνική του microbatching, δηλαδή της κατάτμησης των δεδομένων σε μικρές ομάδες, με συγκεκριμένο μέγεθος και διάρκεια εργασίας πάνω σε αυτές, την στιγμή που τα λαμβάνει και εκτελεί RDD μετατροπές σε κάθε ομάδα ξεχωριστά γεγονός που εισάγει μία καθυστέρηση στο σύστημα τόση όση διαρκεί η κάθε μικρή δουλειά. Στην συγκεκριμένη βιβλιοθήκη περιέχει πράξεις σε ένα παράθυρο δεδομένων, caching δεδομένων της ροής στην μνήμη ή στον δίσκο και checkpointing που επιτρέπει

την σημασιολογία ακριβώς μιας φοράς για τις πράξεις στα δεδομένα.

Η παρακάτω εικόνα δείχνει την λογική της λειτουργίας της βιβλιοθήκης Spark Streaming. Ουσιαστικά, η ροή των δεδομένων μετατρέπεται από real-time processing σε batch processing με την τεχνική του microbatching.

Figure 2.1: Spark Streaming - microbatching

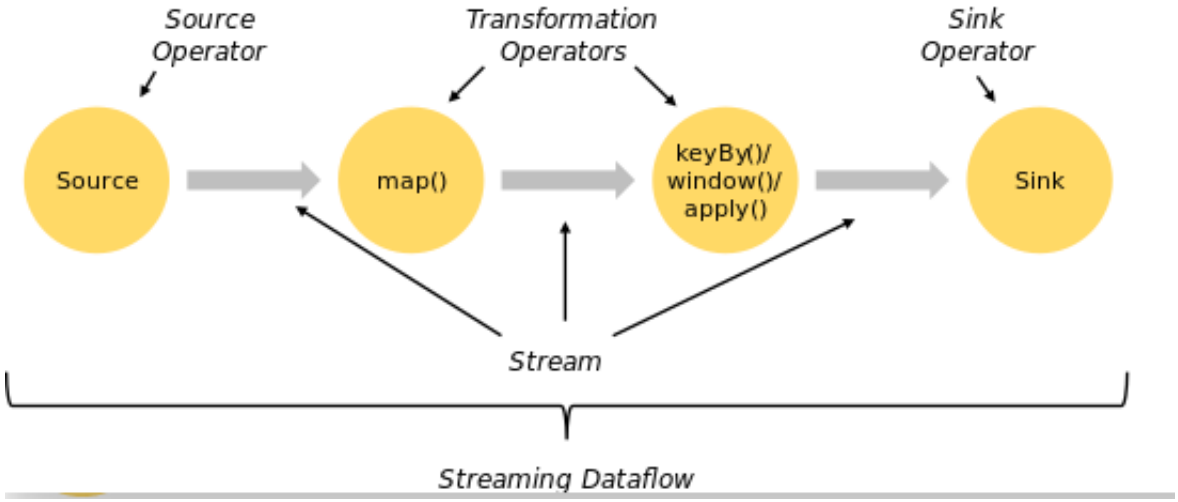


2.1.2 Apache Flink

Διαφέρει λόγω του συστήματος του για dataflow προγράμματα, επεξεργασία γεγονότων σε σχέση με το διάστημα που συμβαίνουν χρησιμοποιώντας παράθυρα χρόνου ή δεδομένων και διαχείριση κατάστασης των συναρτήσεων και τελεστών. Δημιουργεί έναν γράφο για κάθε πρόγραμμα, με κόμβους τις επιμέρους λειτουργίες του, δηλαδή πηγές δεδομένων, υπολογισιμούς στα δεδομένα και καταβόθρες δεδομένων. Είναι ανθεκτικό σε σφάλματα λόγω της χρήσης checkpoints, δηλαδή ασύγχρονα snapshots της κατάστασης της εργασίας και της θέσης στην ροή των δεδομένων ανά τακτά χρονικά διαστήματα αυτόματα. Σε περίπτωση που ο χρήστης θέλει να ανανεώσει το πρόγραμμα και τις ρυθμίσεις του cluster μπορεί να χρησιμοποιήσει savepoints που θέτει αυτός για να συνεχιστεί από εκεί η εκτέλεση. Διαθέτει ακριβώς μία φορά σημασιολογία για τις πράξεις στα δεδομένα και έχει την δυνατότητα για batch εργασίες αν η ροή των δεδομένων που δέχεται έχει όρια όπως το Spark.

Ακολουθεί ένα παράδειγμα της διαχείρισης ενός προγράμματος ως γράφο από το Apache Flink. Τονίζεται η διαφορά με το Apache Spark ως προς την επεξεργασία των δεδομένων σε ροή και όχι σε μικρο-εργασίες.

Figure 2.2: Flink Dataflow



Κεφάλαιο 3

3.1 Πειράματα

Τα πειράματα διαφέρουν στα χαρακτηριστικά του κάθε εργάτη για το κάθε εργαλείο, στον αριθμό των εργατών και στον ρυθμό που στέλνονται τα δεδομένα. Πάντα υπάρχει ο κύριος κόμβος και οι εργάτες είναι τρία, έξι ή εννέα μηχανήματα με δύο ή τέσσερα νήματα εργασίας το καθένα. Η μνήμη που μπορεί να χρησιμοποιεί ο κάθε εργάτης καθορίζεται στο αρχείο ρυθμίσεων του κάθε εργαλείου και είναι 1 GB, καθώς τα προγράμματα δεν αποθηκεύουν μεγάλες ποσότητες δεδομένων στην μνήμη. Η αρχική τιμή δημιουργίας γεγονότων είναι 1000 events/sec και οι υπόλοιπες είναι 5000, 10000, 20000, 50000, 90000, 150000 events/sec. Για το Apache Spark έγιναν μερικά επιπλέον πειράματα με παράμετρο το batchtime, δηλαδή τον χρόνο που διαρκεί κάθε δουλειά πάνω στην κάθε ομάδα δεδομένων, για να διαπιστωθεί η καλύτερη τιμή. Οι τιμές που δοκιμάστηκαν είναι σε milliseconds και βρίσκονται ανάμεσα σε 1000 και 5000.

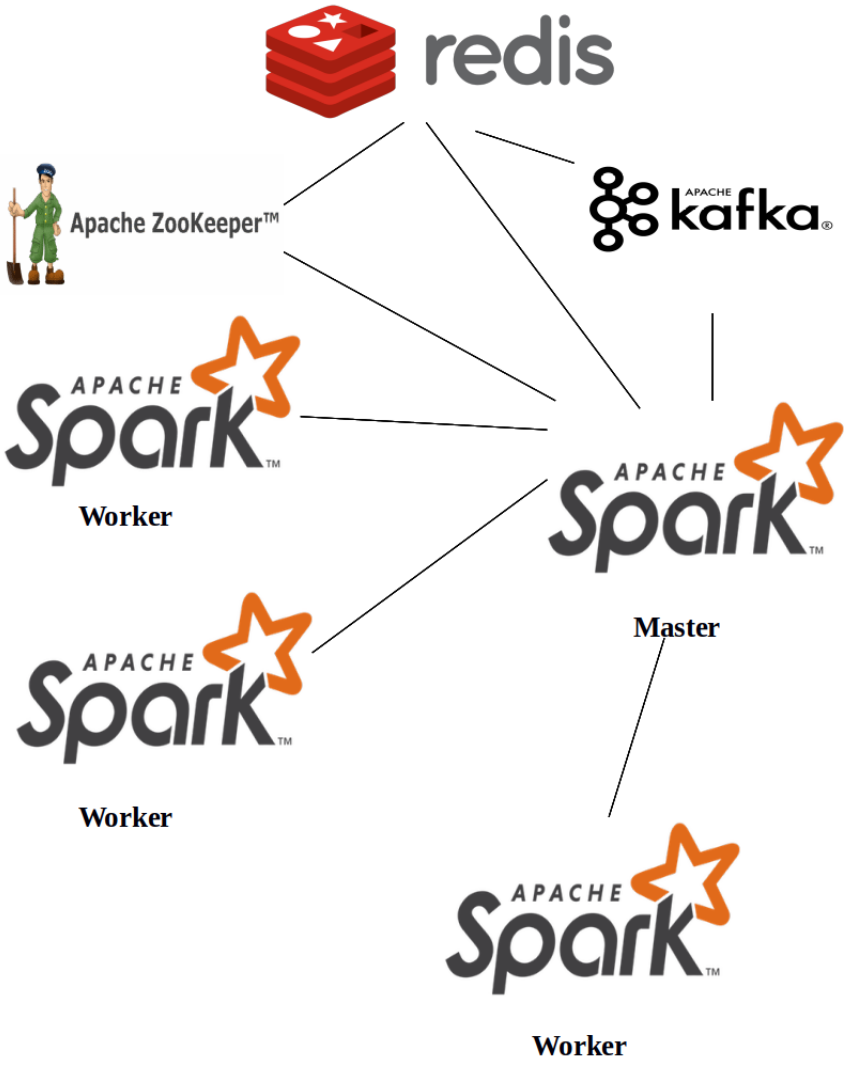
3.2 Αρχιτεκτονική

Η αρχιτεκτονική περιλαμβάνει ένα μηχάνημα που θα είναι ο server για τα Apache Zookeeper, Apache Kafka, Redis και ο κύριος κόμβος για κάθε μηχανή και 3 έως 9 μηχανήματα που θα αποτελούν τους εργάτες. Τα μηχανήματα τρέχουν Ubuntu 16.04 και ανήκουν σε ενα εσωτερικό δίκτυο όπου μόνο το βασικό από αυτά δέχεται περιορισμένη εισερχόμενη κίνηση από το ίντερνετ. Η σύνδεση σε αυτά γίνεται αρχικά με openvpn σε μηχάνημα που ανήκει στο εσωτερικό δίκτυο των μηχανημάτων και μετά με ssh keys στον βασικό server χρησιμοποιώντας τον χρήστη ubuntu που έχει δικαίωμα χρήσης του sudo σε κάθε μηχάνημα. Η ρύθμιση τους γίνεται μέσω του Openstack, του διαδικτυακού εργαλείου για την διαχείριση πολλαπλών clusters που προσφέρει τόσο ένα περιβάλλον για τον χρήστη μέσω του browser όσο και εργαλεία από την γραμμή εντολών. Δύο security groups έχουν δημιουργηθεί στο Openstack, το default και το master, που δίνουν στα εσωτερικά μηχανήματα την δυνατότητα να επικοινωνεί μαζί τους το κύριο μηχάνημα και στο μηχάνημα αυτό την εισερχόμενη κίνηση από το ίντερνετ αντίστοιχα. Το hardware είναι 2 VCPUS ή 4 VCPUS, 4 GB RAM και 40 GB σκληρό δίσκο για τα μηχανήματα-εργάτες,

ενώ το μηχάνημα-εξυπηρετητής διαφέρει μόνο στην μνήμη που χρειάζεται για να τρέξει όλα τα μη streaming εργαλεία και έχει 8 GB RAM. Να σημειώσουμε ότι ο σκληρός δίσκος και η μνήμη δεν διαδραματίζουν ιδιαίτερο ρόλο καθώς τα δεδομένα που αποθηκεύονται είναι πολύ μικρά σε σύγκριση με τον χώρο αυτό και η περιοδική αποθήκευση δεν είναι κύριο κομμάτι των προγραμμάτων.

Ακολουθεί ένα σχεδιάγραμμα της αρχιτεκτονικής του συστήματος.

Figure 3.1: Αρχιτεκτονική



Για να μετατρέψουμε την εκτέλεση από τοπική σε κατανεμημένη πρέπει να αλλάξουμε

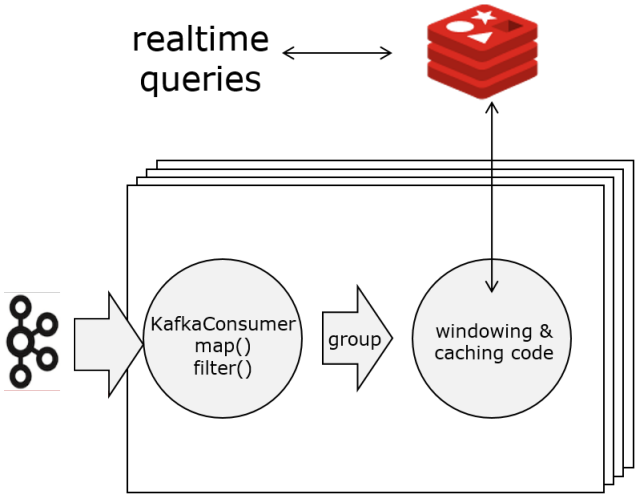
τα αρχεία ρυθμίσεων κάθε εργαλείου για λειτουργία σε cluster και να δηλωθούν ως εργάτες τα μηχανήματα χρησιμοποιώντας τις IPs τους. Επίσης, χρειάζεται η εγκατάσταση της Java 1.8, της Scala 2.10 και το περιβάλλον εκτέλεσης και μεταγλώττισης της γλώσσας Closure, leiningen, στο κεντρικό μηχάνημα για την μεταγλώττιση του κώδικα. Τα υπόλοιπα έχουν εγκαταστημένη μόνο την Java 1.8 για την εκτέλεση του κώδικα. Για να σχηματιστεί το cluster πρέπει κάθε μηχάνημα να έχει τα αρχεία της κάθε μηχανής και το σωστό configuration ανάλογα με το αν είναι ο εξυπηρετητής ή ένας εργάτης. Οι εκδόσεις του κάθε εργαλείου που χρησιμοποιήθηκαν είναι κοντά στις παρούσες που κυκλοφόρουν.

3.3 Προγράμματα

Τα yahoo benchmarks και οι επεκτάσεις τους υλοποιούν ένα κλασσικό σενάριο επεξεργασίας εισερχόμενων διαφημίσεων σε μορφή JSON και διαφέρουν μόνο στον φόρτο των γεγονότων που δημιουργούνται, στον τρόπο που αποθηκεύουν τα αποτελέσματα και πως αυτά ανακτώνται. Ο κώδικας των benchmarks βρίσκεται στο διαδικτυακό τόπο διαχείρισης αποθετήριων του Version Control System εργαλείου Git, Github και περιέχει οδηγίες χρήσης για την εγκατάσταση, την ρύθμιση και την χρήση σε τοπική λειτουργία. Η όλη λειτουργία γίνεται από ενα bash script που ενεργοποιεί τα κατάλληλα προγράμματα για το κάθε σενάριο. Πρώτα, ενεργοποιείται ο server του Apache Zookeeper για να μπορούν να συγχρονιστούν μέσω αυτού τα υπόλοιπα. Έπειτα, έχουμε το Redis όπου περιοδικά αποθηκεύονται τα αποτελέσματα, μετά ο broker(server) του Apache Kafka και αμέσως μετά δημιουργείται το θέμα ad-events για τα μηνύματα που θα σταλούν μέσω Kafka και θα περιέχουν τα δεδομένα. Σε αυτό το σημείο, ανάλογα με το εργαλείο που θέλουμε να δοκιμάσουμε, ενεργοποιούνται με κατάλληλη εντολή ο εξυπηρετητής και οι εργάτες. Το επόμενο βήμα είναι η φόρτωση της δουλειάς που θα εκτελεστεί και το ξεκίνημα της και τέλος ξεκινάει η δημιουργία διαφημίσεων που στέλνονται στο Apache Kafka. Μόλις περάσει ο χρόνος δοκιμής που είναι 4 λεπτά, τα προγράμματα τερματίζονται με την ανάποδη σειρά που αυτά ξεκίνησαν.

Πρώτα παρουσιάζεται η γενική εικόνα του σεναρίου και έπειτα η περιγραφή και παρουσίαση των επιμέρους λειτουργιών του.

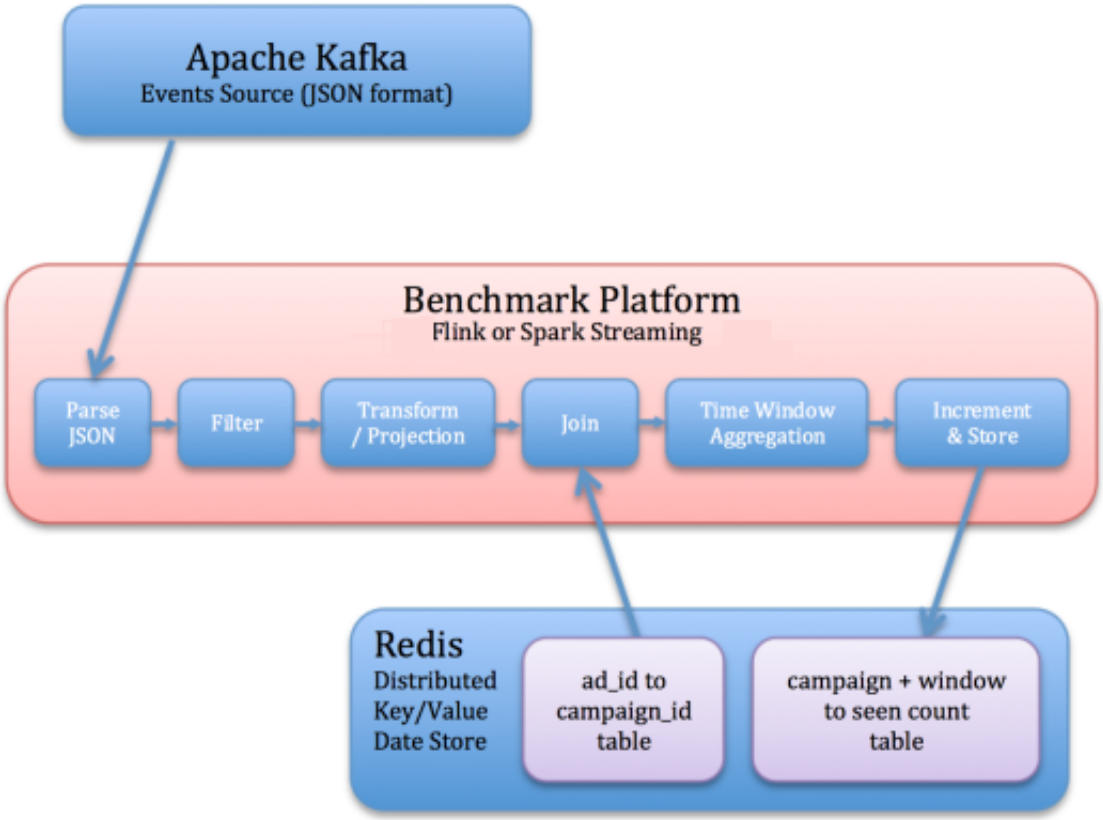
Figure 3.2: Γενική εικόνα σεναρίου



Yahoo! Streaming benchmark job

Το κάθε πρόγραμμα χρησιμοποιώντας το API του εργαλείου στο οποίο απευθύνεται διαβάζει ένα γεγονός από το Apache Kafka, το αποσειριοποιεί από την μορφή συμβολοσειράς JSON και χρησιμοποιεί φίλτρα για να απομακρύνει τα άσχετα γεγονότα. Από αυτά που διατηρεί, εφαρμόζει μία προβολή των σχετικών πεδίων(αναγνωρισιτκό διαφήμισης και τύπος γεγονότος), τα ενώνει με την διαφημιστική καμπάνια στην οποία ανήκει το καθένα και αποθηκεύει αυτή την πληροφορία στο Redis. Τέλος, τα γεγονότα ομαδοποιούνται σε παράθυρα ανά καμπάνια και το άθροισμα τους αποθηκεύεται στο Redis μαζί με μια χρονοσφραγίδα τελευταίας ενημέρωσης του, λαμβάνοντας υπόψην και τυχόν καθυστερήσεις γεγονότων.

Figure 3.3: Επειμέρους λειτουργίες σεναρίου



3.4 Μετρήσεις

Τα αρχεία αποτελεσμάτων πρέπει να αποθηκεύονται μετά από κάθε εκτέλεση, γιατί τα προηγούμενα θα σβηστούν ώστε να γραφτούν τα τρέχοντα. Έπειτα, επεξεργάζονται για να προκύψουν οι μετρικές και συγκρίνονται μεταξύ τους για να προκύψουντα διαγράμματα που βρίσκονται στην επόμενη ενότητα. Οι τιμές της καθυστέρησης είναι σε milliseconds. Για την μέση καθυστέρηση χρησιμοποιήθηκε η συνάρτηση Median στο Excel. Στοιχεία για τις επιμέρους συνθήκες λειτουργίες όπως μνήμη που τελικά χρησιμοποιήθηκε, χρόνος για κάθε πράξη και άλλα μπορούμε να πάρουμε από την εσωτερική ιστοσελίδα του κάθε εργαλείου.

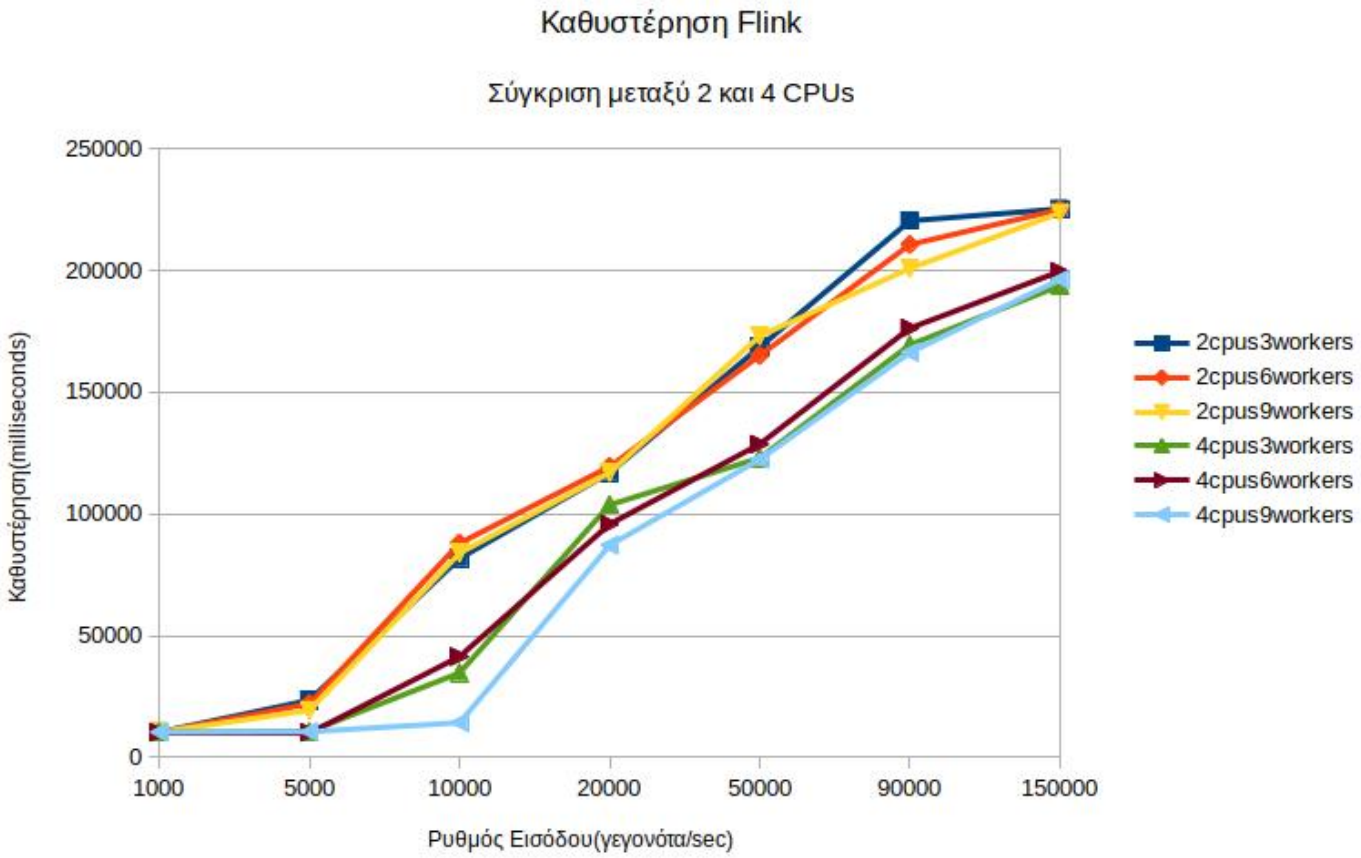
3.5 Διαγράμματα

Οι αξονές των διαγραμμάτων είναι η καθυστέρηση προς τον ρυθμό εισαγωγής δεδομένων στο σύστημα για κάθε εργαλείο. Ο αριθμός των VCPUs και των μηχανημάτων

αναφέρεται στον υπότιτλο του γραφήματος ή στο όνομα της κάθε συνάρτησης. Για την περίπτωση του batchtime στο Spark, οι άξονες είναι καθυστέρηση προς διάρκεια microbatch. Παρακάτω παρατίθονται οι γραφικές με περιγραφή για την καθεμιά.

Αρχικά, θα παρατεθεί η μέση καθυστέρηση και η καθυστέρηση ανά ποσοστό ολοκληρωμένων επεξεργασμένων δεδομένων για κάθε εργαλείο ξεχωριστά για διαφορετικούς αριθμούς μηχανημάτων και πυρήνων και μετά θα γίνει η συνολική σύγκριση.

Figure 3.4: Σύγκριση Καθυστέρησης στο Flink για 2 και 4 VCPUs

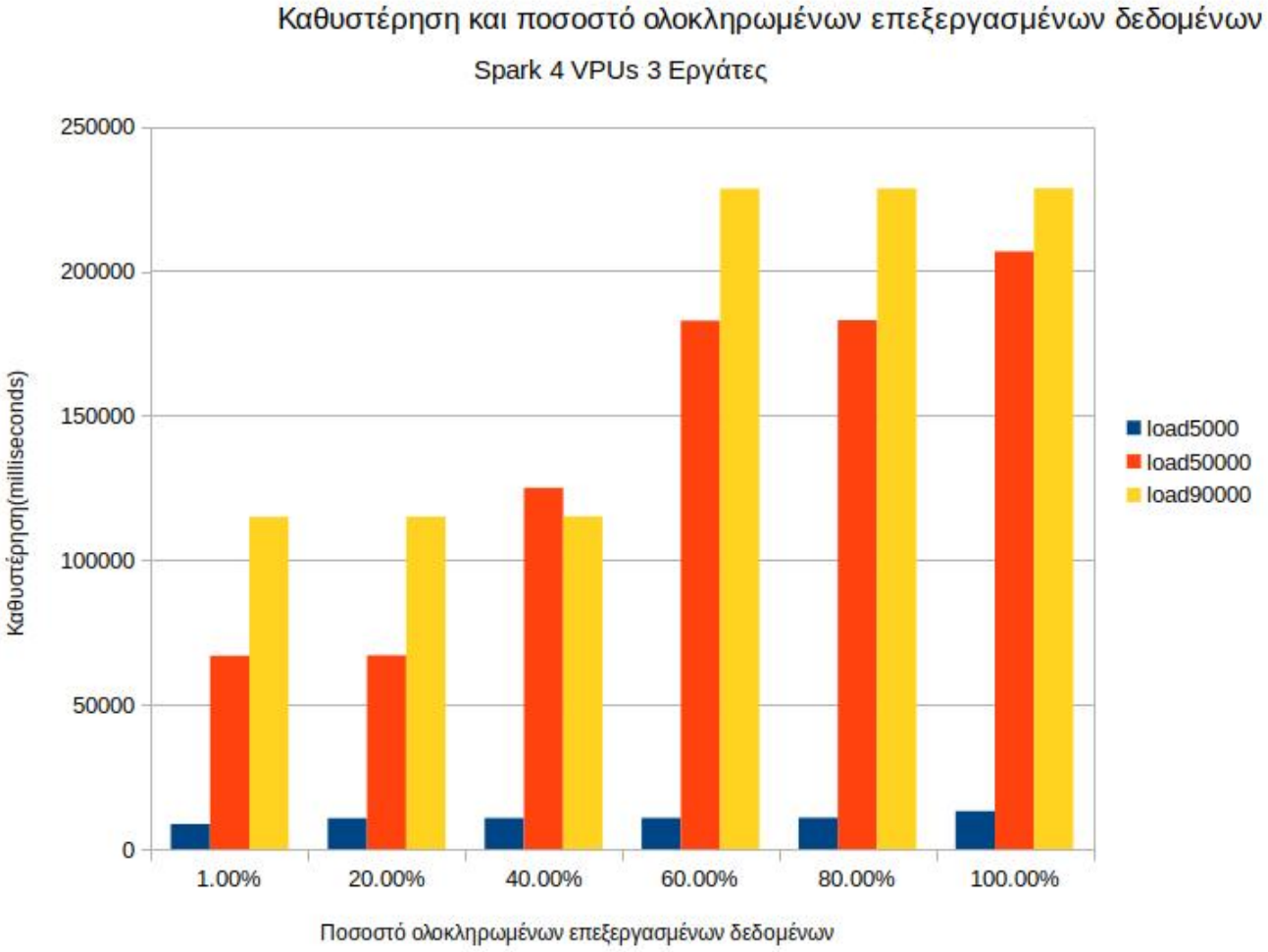


Στο διάγραμμα αυτό απεικονίζεται η σύγκριση της μέσης καθυστέρησης του Apache Flink για μηχανήματα με 2 και 4 VCPUs. Η μπλε γραμμή αντιστοιχεί στο σενάριο τριών εργατών, η κόκκινη στους έξι και η κίτρινη στους εννέα για τους δύο πυρήνες σε κάθε μηχάνημα. Η πράσινη, η μωβ και η γαλάζια στους τρεις, έξι και εννέα εργάτες για τους τέσσερις πυρήνες ανά μηχάνημα. Είναι εμφανής η βελτίωση που υπάρχει με την αύξηση των πυρήνων, ενώ τα περισσότερα μηχανήματα επηρεάζουν ελάχιστα την ταχύτητα του συστήματος έως καθόλου για δύο πυρήνες.

Ακολουθεί το διαγραμμα για την διακύμανση της καθυστέρησης στο Flink κατά την

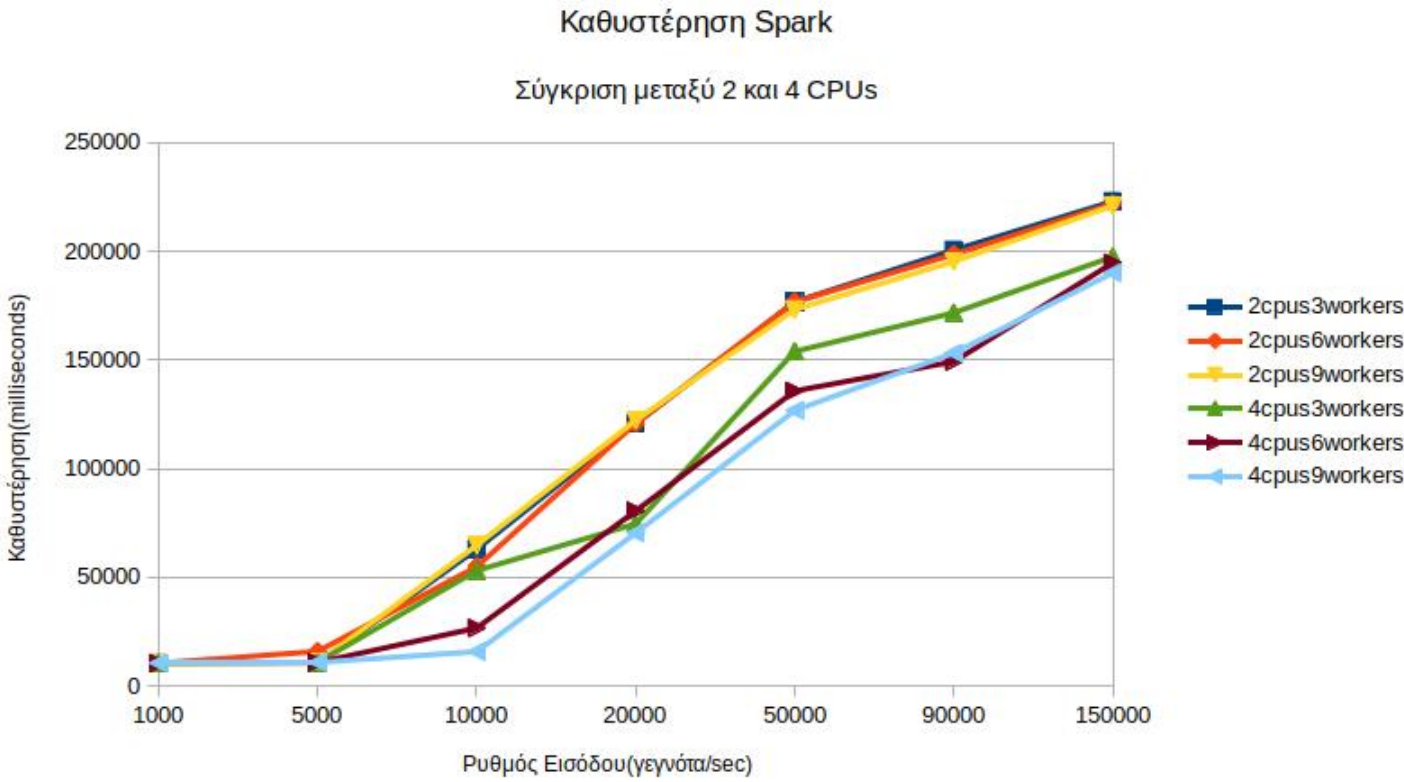
διάρκεια της εκτέλεσης του κάθε σεναρίου, με μεταβλητή τον ρυθμό παραγωγής γεγονότων από την πηγή. Τα αποτελέσματα είναι ανάλογα για διαφορετικά μηχανήματα και πυρήνες οπότε θα παρατεθεί μόνο ένα χαρακτηριστικό παράδειγμα για κάθε πλατφόρμα.

Figure 3.5: Ποσοστό Καθυστέρησης στο Flink για 4 VCPUs



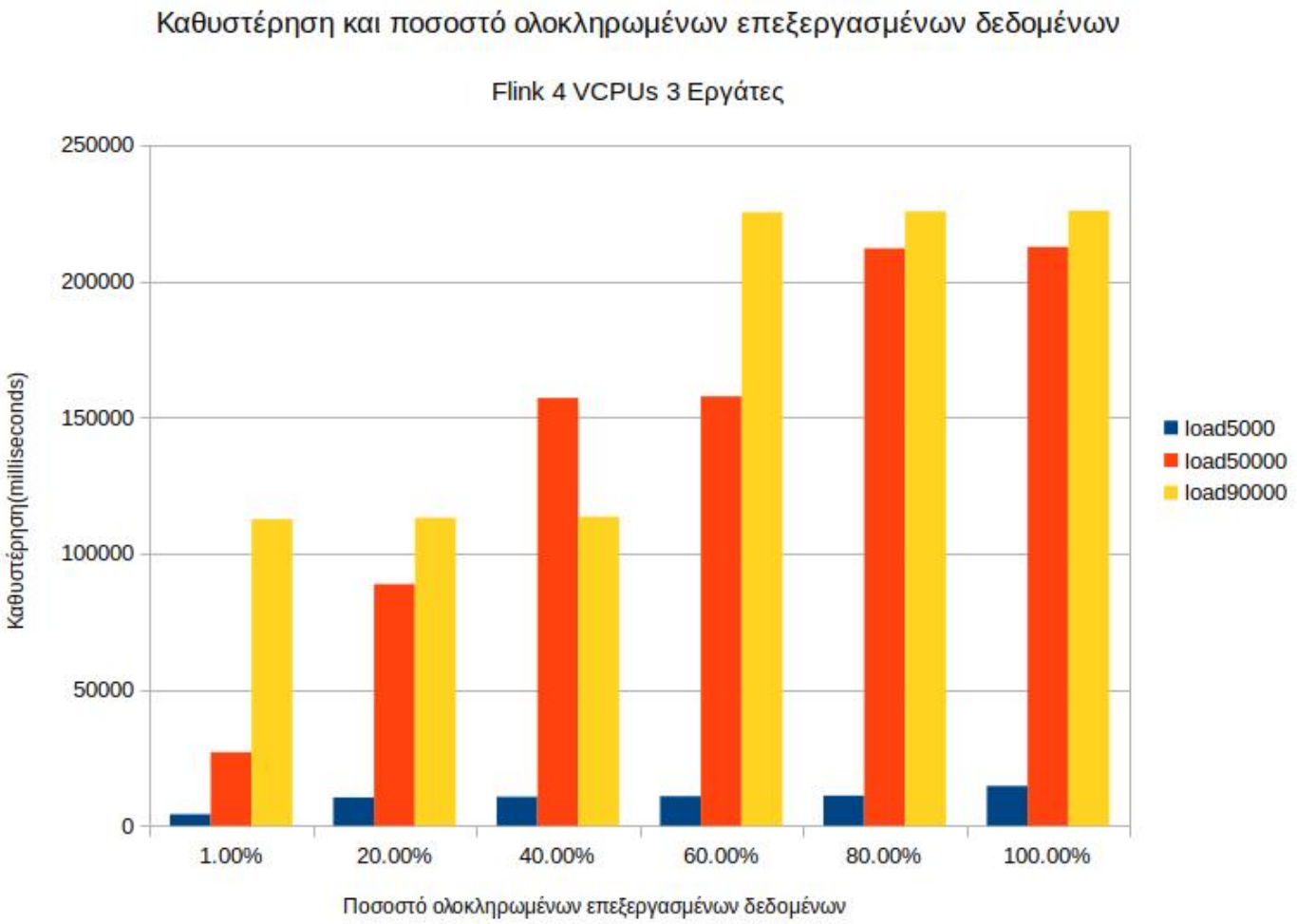
Στο διάγραμμα αυτό απεικονίζεται η σύγκριση των τιμών καθυστέρησης του Apache Flink για τρία μηχανήματα με 4 VCPUs. Η μπλε γραμμή αντιστοιχεί στο σενάριο ρυθμού εισόδου στο σύστημα 5000 γεγονότων/sec, η κόκκινη στα 50000 γεγονότα/sec και η κίτρινη στα 90000 γεγονότα/sec. Η διακύμανση των τιμών είναι ελάχιστη για την μικρότερη τιμή εισόδου και μέγιστη για την μεσαία τιμή, η οποία εμφανίζει και τις περισσότερες διαφορετικές τιμές ενδιάμεσα. Η μέγιστη τιμή εισόδου παρουσιάζει μεγάλη διακύμανση, αλλά όχι ενδιάμεσες τιμές.

Figure 3.6: Σύγκριση Καθυστέρησης στο Spark για 2 και 4 VCPUs



Στο διάγραμμα αυτό απεικονίζεται η σύγκριση της μέσης καθυστέρησης του Apache Spark για μηχανήματα με 2 και 4 VCPUs. Η μπλε γραμμή αντιστοιχεί στο σενάριο τριών εργατών, η κόκκινη στους έξι και η κίτρινη στους εννέα για τους δύο πυρήνες σε κάθε μηχάνημα. Η πράσινη, η μωβ και η γαλάζια στους τρεις, έξι και εννέα εργάτες για τους τέσσερις πυρήνες ανά μηχάνημα. Στους μικρούς ρυθμούς φαίνεται μια ταύτιση όλων των τιμών, γεγονός που οφείλεται στο ότι κάθε microbatch έχει συγκεκριμένη διάρκεια. Συνεπώς, εάν έχουν γίνει οι υπολογισμοί στα δεδομένα ενός microbatch, δεν μπορεί ο ίδιος πυρήνας να αρχίσει την επεξεργασία άλλου microbtach πριν περάσει χρόνος ίσος με το batchtime. Οι υπόλοιπες παρατηρήσεις είναι όμοιες με το Flink.

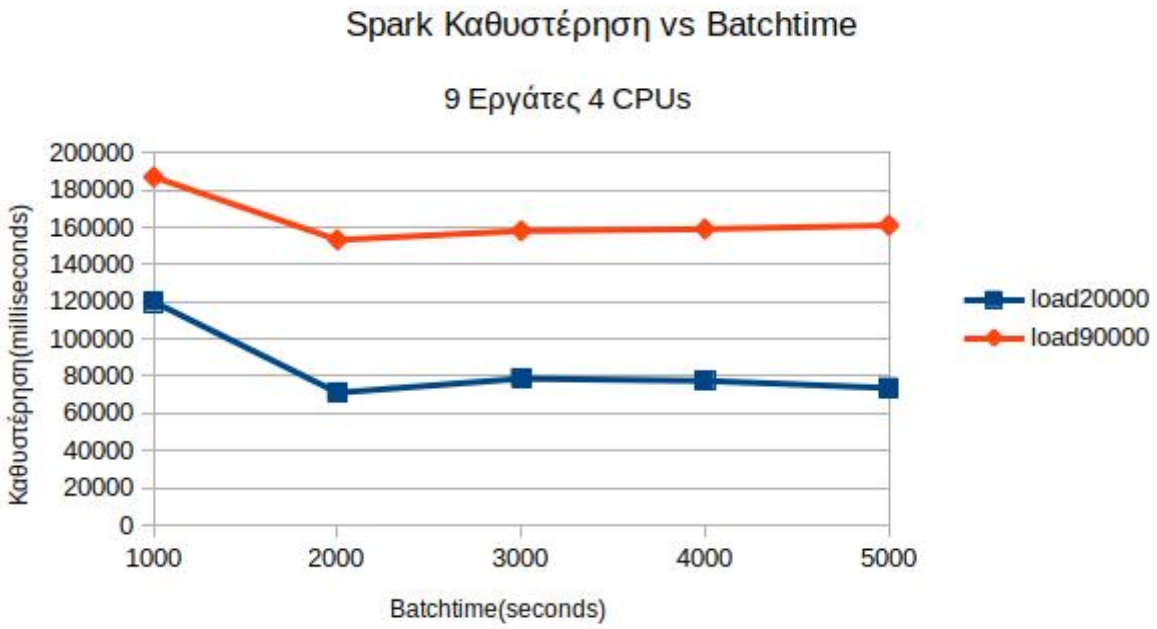
Figure 3.7: Ποσοστό Καθυστέρησης στο Spark για 4 VCPUs



Στο διάγραμμα αυτό απεικονίζεται η σύγκριση των τιμών καθυστέρησης του Apache Spark για τρία μηχανήματα με 4 VCPUs. Η μπλε γραμμή αντιστοιχεί στο σενάριο ρυθμού εισόδου στο σύστημα 5000 γεγονότων/sec, η κόκκινη στα 50000 γεγονότα/sec και η κίτρινη στα 90000 γεγονότα/sec. Η διακύμανση έχει όμοια συμπεριφορά με το Apache Flink.

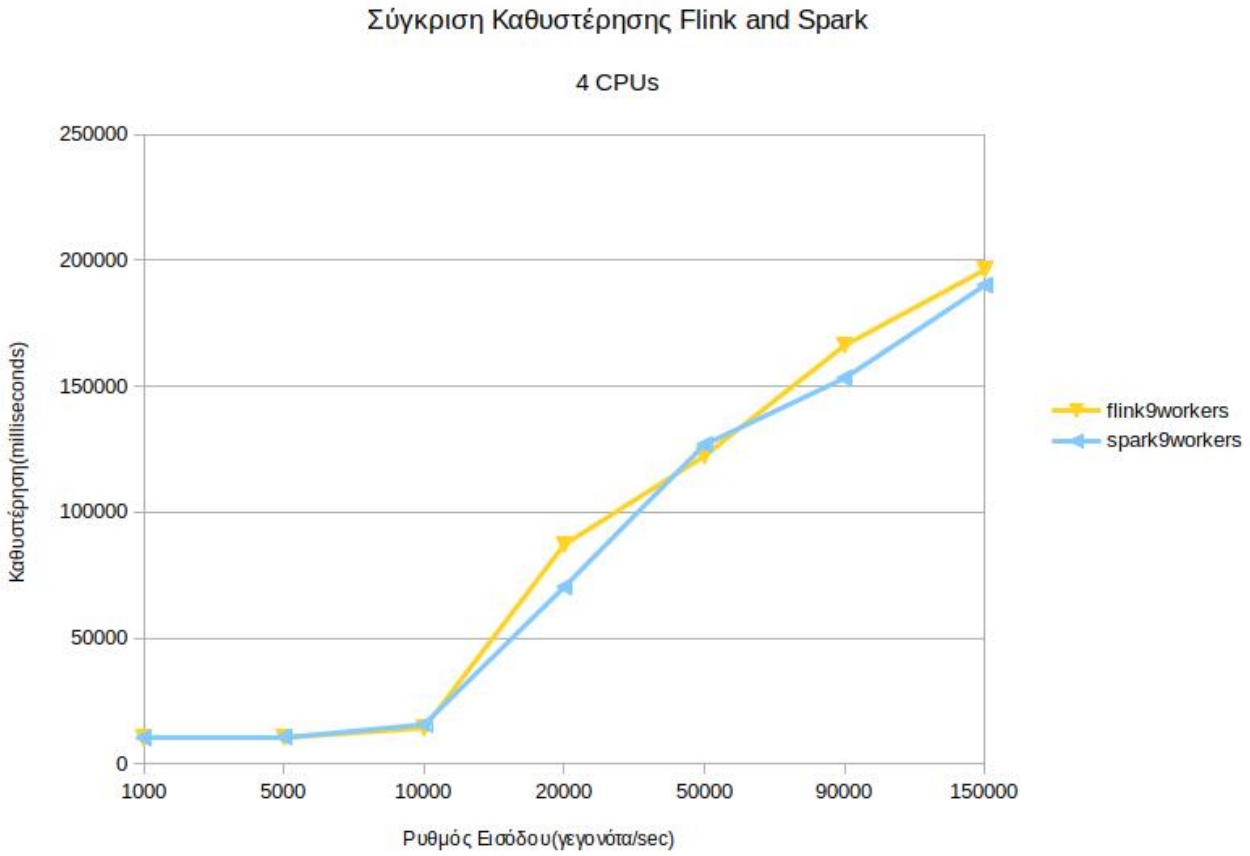
Για το batchtime θα παρατεθούν στοιχεία μόνο για ένα σενάριο καθώς παρατηρείται η ίδια συμπεριφορά σε όλα.

Figure 3.8: Καθυστέρηση προς Batchtime στο Spark για 4 VCPUs



Στο διάγραμμα αυτό απεικονίζεται η σύγκριση της μέσης καθυστέρησης και του batchtime του Apache Spark για εννέα μηχανήματα με 4 VCPUs. Η μπλε γραμμή αντιστοιχεί στο σενάριο όπου η πηγή δημιουργεί δεδομένα με ρυθμό 20000/sec και η κόκκινη σε ρυθμό 90000/sec. Και στις δύο περιπτώσεις παρατηρείται μεγαλύτερη καθυστέρηση όταν τα micro-batches έχουν διάρκεια ένα δευτερόλεπτο, ενώ στις υπόλοιπες τιμές η διαφορά τιμών είναι σχεδόν αδιάφορη.

Figure 3.9: Συνολική σύγκριση καθυστέρησης για 4 VPUs



Στο διάγραμμα αυτό απεικονίζεται η σύγκριση της μέσης καθυστέρησης των Apache Flink και Spark για μηχανήματα με 4 VCPUs και 9 εργάτες. Επιλέχθηκαν αυτές οι παράμετροι, καθώς είναι η περίπτωση που είναι πιο εμφανής η διαφορά των δύο εργαλείων. Στα υπόλοιπα σενάρια, η συμπεριφορά των εργαλείων είναι ίδια με μικρότερη διαφορά τιμών έως ελάχιστη. Οι διαφορές στις τιμές μεταξύ των εργαλείων είναι πιο μεγάλες σε σχέση με την περίπτωση των λιγότερων πυρήνων, όχι σε τεράστιο βαθμό, αλλά αρκετά ώστε να δείχνουν καλύτερο το Spark.

Κεφάλαιο 4

4.1 Συμπεράσματα

Το πρώτο συμπέρασμα που παρατηρείται είναι η μικρή και σχεδόν κοινή απόδοση των δύο μηχανών όταν οι πυρήνες σε κάθε μηχάνημα είναι δύο ανεξαρτήτως των αριθμών των εργατών. Αυτό το γεγονός μας οδηγεί να υποθέσουμε ότι η ορθή και πλήρης λειτουργία του κάθε εργαλείου απαιτεί ένα συγκεκριμένο αριθμό CPUs ως ελάχιστο ανάλογα το σενάριο για να αναλαμβάνουν την αποστολή και λήψη των δεδομένων και τον συγχρονισμό με τον κύριο κόμβο.

Οι 2 VCPUs είναι αρκετές μόνο για τις μικρές τιμές ρυθμού εισαγωγής δεδομένων, εφόσον μόνο σε αυτά τα σενάρια δεν παρατηρείται διαφορά μετά την αύξηση των πυρήνων. Ωστόσο, τέτοια σενάρια συναντιούνται σήμερα μόνο σε μικρού μεγέθους συστήματα, όπου οι χρήστες ή οι πελάτες είναι μικροί σε αριθμό. Η ουσιαστική συμβολή εργαλείων όπως αυτά που εξετάσαμε είναι στους τεράστιους ρυθμούς κοντά στην μέγιστη τιμή των πειραμάτων μας και οι οποίοι αυξάνονται πολύ γρήγορα.

Στην συνέχεια, συγκρίνοντας τις επιδόσεις της κάθε μίας μηχανής για 2 και 4 πυρήνες, μπορούμε να δούμε μια ξεκάθαρη βελτίωση. Αυτό συμβαίνει ειδικά για τις μεσαίες και μεγάλες τιμές του ρυθμού εισαγωγής των δεδομένων στο σύστημα, στις οποίες το ποσοστό της διαφοράς είναι μεγαλύτερο και η χρήση περισσότερων εργατών δίνει ένα μικρό επιπλέον προβάδισμα.

Από το ποσοστό της καθυστέρησης ανάλογα με τον ρυθμό εισαγωγής των δεδομένων στο σύστημα ότι πάνω από το μισά δεδομένα προκαλούν την μεγάλη καθυστέρηση στο κάθε εργαλείο. Αυτό οδηγεί στο συμπέρασμα ότι το σύστημα πρέπει να γίνει πιο δυνατό για να φτάσει το ποσοστό αυτό σε μικρότερες τιμές. Με βάση τα άλλα αποτελέσματα, κύρια αλλαγή πρέπει να είναι η αύξηση των πυρήνων σε κάθε μηχάνημα και έπειτα η αύξηση των μηχανημάτων-εργατών.

Συνολικά, το εργαλείο που έδειξε ότι αντιμετωπίζει καλύτερα την κίνηση είναι το Apache Spark, ωστόσο η διαφορά δεν είναι μεγάλη. Υπερτερεί στο σενάριο λόγω της κατανομής των μικρο-εργασιών που δημιουργεί μόλις δεχθεί την εισερχόμενη ροή των δεδομένων.

Το batchtime είναι μία σημαντική παράμετρος του Spark και πρέπει να ορίζεται στην βέλτιστη τιμή για να λειτουργεί το σύστημα στο έπακρο των δυνατοτήτων του. Στο

παράδειγμά αυτή της εργασίας παρατηρείται ότι η τιμή των 1000 milliseconds προκαλεί σημαντική καθυστέρηση στο σύστημα και συνεπώς είναι αποτρεπτική. Οι υπόλοιπες διατηρούν την καθυστέρηση στην ίδια τάξη μεγέθους με την βέλτιστη τιμή να εμφανίζεται για 2000 milliseconds.

Αυτή είναι μία αρχική σύγκριση των δύο εργαλείων και δικαιώνει την επικράτηση του Spark στην κοινότητα των προγραμματιστών. Υπάρχουν διάφορα ακόμα σενάρια για μια πιο ολοκληρωμένη σύγκριση σε απλές και πολύπλοκες λειτουργίες των μηχανών αυτών.

Bibliography

[1] Apache Spark *Apache Spark Documentation* https://spark.apache.org/docs/1.5.2 2015.

[2] Apache Flink *Apache Flink Documentation* https://ci.apache.org/projects/flink/flink-docs-release-1.0 2016.

[3] Jamie Grier *Extending the Yahoo! Streaming Benchmark* https://data-artisans.com/extending-the-yahoo-streaming-benchmark 2016.

[4] Revans *Benchmarking Streaming Computation Engines at Yahoo!* https://yahooeng.tumblr.com/post/135321837876/benchmarking-streaming-computation-engines-at 2015.