



Χαροκόπειο Πανεπιστήμιο
Τμήμα Πληροφορικής και Τηλεματικής

Πτυχιακή Εργασία

Χαρτογράφηση καμένων εκτάσεων με χρήση map-reduce

Καμμάς Εμμανουήλ

ΑΜ. 20712

Επιβλέπων: **Μιχαήλ Δημήτριος**

Μέλη της Εξεταστικής Επιτροπής

Νικολαΐδου Μάρα, Καθηγήτρια

Βαρλάμης Ηρακλής, Λέκτορας

ΑΘΗΝΑ

ΣΕΠΤΕΜΒΡΙΟΣ 2014

Πρόλογος

Η εκπόνηση της παρούσας πτυχιακής εργασίας πραγματοποιήθηκε στο τμήμα Πληροφορικής και Τηλεματικής του Χαροκόπειου Πανεπιστημίου από τον Ιούνιο 2013 έως τον Σεπτέμβριο 2014.

Επιθυμώ να εκφράσω τις ευχαριστίες μου σε όλους εκείνους που συνέβαλλαν άμεσα ή έμμεσα στην ολοκλήρωση της Πτυχιακής μου εργασίας και κατά συνέπεια των προπτυχιακών μου σπουδών.

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέπον Καθηγητή της παρούσας πτυχιακής, κ Δημήτριο Μιχαήλ για το ενδιαφέρον και την βοήθεια που μου πρόσφερε καθ' όλη την διάρκεια της πτυχιακής εργασίας.

Εν συνεχεία, θα ήθελα να εκφράσω τις ευχαριστίες μου στα υπόλοιπα μέλη της τριμελούς εξεταστικής επιτροπής: τον κ Ηρακλή Βαρλάμη για τις χρήσιμες γνώσεις που μου μετέδωσε στα μαθήματα του καθ' όλη τη διάρκεια της φοίτησης μου και τη κα Μάρα Νικολαΐδου για το υψηλό επίπεδο γνώσεων που παρείχε στα μαθήματα της κατά τη φοίτηση μου, τη ψυχολογική στήριξη, και τη βοήθεια της για την ολοκλήρωση της παρούσας πτυχιακής.

Επιπλέον θα ήθελα να ευχαριστήσω τους φοιτητές Αλεξανδράκη Δημήτρη και Ταφραλή Ζαχαρία για τη βοήθεια και στήριξη που παρείχαν ώστε να ολοκληρωθεί επιτυχώς και εγκαίρως η εν λόγω εργασία.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου για την αγάπη, τη κατανόηση και τη στήριξη που μου επέδειξαν χωρίς την οποία δεν θα ήταν δυνατή η ολοκλήρωση της παρούσας πτυχιακής.

Περιεχόμενα

Περίληψη	7
Abstract	8
Κεφάλαιο 1ο – Εισαγωγή	9
1.1 Αντικείμενο και σκοπός πτυχιακής εργασίας	9
Κεφάλαιο 2ο – Υπόβαθρο	11
2.1 Τηλεπισκόπηση	11
2.1.1 Ορισμός	11
2.1.2 Αισθητήρες ακτινοβολίας και ψηφιακές εικόνες	12
2.1.3 Τηλεπισκόπηση και πληροφορική	15
2.1.4 Βιβλιοθήκη GDAL	15
2.1.5 Εφαρμογές της τηλεπισκόπησης	16
2.1.6 Χαρτογράφηση καμένων περιοχών	16
2.2 Κατανεμημένα Συστήματα	18
2.2.1 Ορισμός	18
2.2.2 Χαρακτηριστικά Κατανεμημένων Συστημάτων	18
2.2.3 Αρχιτεκτονική Επικοινωνίας	19
2.3 Προγραμματιστικό μοντέλο MapReduce	20
2.3.1 Ορισμός	20
2.3.2 Προγραμματισμός με MapReduce	21
2.4 Apache Hadoop	22
2.4.1 Τι είναι το Hadoop	22
2.4.2 Περισσότερα για το Hadoop	22
2.4.3 Πλεονεκτήματα του Hadoop	23
2.4.4 Αρχιτεκτονική του Hadoop	23
2.4.5 Κατανεμημένο σύστημα αρχείων του Hadoop (HDFS)	24
Κεφάλαιο 3ο – Αλγόριθμος χαρτογράφησης καμένων εκτάσεων (BSM_NOA)	27

3.1 Περιγραφή	27
3.2 Διαδικασία επεξεργασίας εικόνας.....	27
3.2.1 Προεπεξεργασία εικόνων	27
3.2.2 Στάδιο βασικής επεξεργασίας.....	28
3.2.3 Ποιοτικός έλεγχος και ερμηνεία εικόνων.....	30
Κεφάλαιο 4ο – Υλοποίηση	32
4.1 Εισαγωγή.....	32
4.2 Είσοδος και μετατροπή δεδομένων.....	32
4.2.1 Είσοδος δεδομένων.....	32
4.2.2 Μετατροπή εικόνας σε μορφή κειμένου.....	34
4.3 Φάση ταξινόμησης	34
4.3.1 Mapper.....	34
4.3.2 Reducer.....	35
4.4 Φάση διάμεσου φίλτρου.....	35
4.4.1 Median filter	35
4.4.2 Περιγραφή και λειτουργικότητα MapReducer	36
4.5 Φάση ομαδοποίησης και διαγραφής	37
4.5.1 Αλγόριθμος σύνδεσης και επισήμανσης στοιχείων.....	37
4.5.2 Λειτουργικότητα MapReducer	39
4.6 Μετατροπή αρχείων κειμένου σε εικόνα	40
Κεφάλαιο 5ο – Λειτουργία και αποτελέσματα	41
5.1 Εισαγωγή.....	41
5.2 Δοκιμή λειτουργίας προγράμματος.....	41
5.3 HDINSIGHT και υπηρεσίες cloud.....	46
5.3.1 Εισαγωγή στο HDInsight	46
5.3.2 Εκτέλεση στο HDInsight	47
5.4 Αναφορά χρόνων εκτέλεσης	49

5.5 Συμπεράσματα	51
5.6 Μελλοντικές κατευθύνσεις	52
Συντομογραφίες	53
Βιβλιογραφία	54
Εικόνες	57

Περίληψη

Η παρατήρηση της Γης από δορυφόρους γίνεται εδώ και πενήντα χρόνια, γεγονός που έχει ως αποτέλεσμα την συσσώρευση τεράστιου όγκου πληροφορίας ψηφιακών δεδομένων. Ο όγκος αυτός των δεδομένων καθιστά την διαδικασία αποθήκευση τους και επεξεργασία τους δύσκολη και χρονοβόρα. Για τον λόγο αυτό ο ιδανικός τρόπος αποθήκευσης και επεξεργασίας τους είναι σε κατανεμημένα συστήματα, με την χρήση παράλληλου και κατανεμημένου προγραμματισμού.

Στην παρούσα πτυχιακή εργασία, υλοποιήθηκε μια εφαρμογή επεξεργασίας δορυφορικών εικόνων για την χαρτογράφηση καμένων περιοχών, χρησιμοποιώντας το προγραμματιστικό μοντέλο MapReduce του συστήματος Hadoop framework, το οποίο είναι σχεδιασμένο για την επεξεργασία μεγάλου όγκου δεδομένων σε κατανεμημένο περιβάλλον και είναι ανοικτού κώδικα.

Η υλοποίηση της εφαρμογής βασίστηκε στον αλγόριθμο χαρτογράφησης καμένων περιοχών που έχει αναπτύξει το Εθνικό Αστεροσκοπείο Αθηνών. Η εφαρμογή χωρίζεται σε πέντε φάσεις επεξεργασίας: μετατροπή δορυφορικών εικόνων σε μορφή κατάλληλη για επεξεργασία από τους mappers, ταξινόμηση των καμένων pixel της εικόνας, εφαρμογή διάμεσου φίλτρου 3x3, απομάκρυνση θορύβου με την βοήθεια του αλγορίθμου “connect components”, μετατροπή αποτελεσμάτων σε εικόνα. Τα τρία ενδιάμεσα στάδια είναι υλοποιημένα σε Hadoop, τα υπόλοιπα σε Java.

Η εφαρμογή έτρεξε στο σύγχρονο περιβάλλον cloud της Microsoft, και τα παραγόμενα αποτελέσματα ήταν ικανοποιητικά, καθώς επεξεργάστηκε παράλληλα οχτώ δορυφορικές εικόνες και στο τέλος εξήγαγε οχτώ εικόνες που έχουν χαρτογραφήσει τις καμένες εκτάσεις.

Abstract

Earth satellite observation is known for more than fifty years, which has resulted the accumulation of huge amount of digital data. This huge amount of data makes the process of storing and processing difficult and time consuming. For this reason, the ideal way to store and process the data is in distributed systems, using parallel and distributed programming.

In this thesis, is implemented an application that processes satellite images for burn scar mapping, using the programming model MapReduce and the Hadoop framework, which is designed to process large amounts of data in a distributed environment and it is open source.

The implementation of the application based on the Burn Scar Mapping algorithm, developed at the National Observatory of Athens. The application is divided into five processing stages: converting satellite images in a format suitable for processing by the mappers, classification of the “burned” pixels of the image, application of a 3x3 median filter, noise reduction using "connect components" algorithm, generation of the final output image. The three intermediate stages are implemented on the Hadoop, the other two in Java.

The application ran into the modern cloud environment of Microsoft, and the produced results were satisfactory, cause of parallel processing of eight satellite images and exporting eight mapped images.

Κεφάλαιο 1ο – Εισαγωγή

1.1 Αντικείμενο και σκοπός πτυχιακής εργασίας

Το σύνολο των καμένων δασικών και αγροτικών εκτάσεων στην Ελλάδα από το 1983 έως και το 2008 ανέρχεται σε 13.613.121 στρέμματα, αριθμός που αναλογεί σε 1,2 στρέμματα ανά κάτοικο, σύμφωνα με έρευνα του Ινστιτούτου Μεσογειακών - Δασικών Οικοσυστημάτων και Τεχνολογίας Δασικών Προϊόντων του ΕΘΙΑΓΕ και του WWF Ελλάς. Κατά συνέπεια, η μελέτη των πυρκαγιών είναι απαραίτητη για να αναπτυχθεί καλύτερη πρόβλεψη αλλά και αντιμετώπιση αυτού του φαινομένου. Γνωρίζοντας για παράδειγμα τις περιοχές που έχουν υψηλή επικινδυνότητα εκδήλωσης πυρκαγιάς, μπορεί να συμβάλει στην καλύτερη οργάνωση, με περισσότερες περιπολίες, αντιπυρικές λωρίδες, δασικούς δρόμους, δασικούς σταθμούς κτλ [1].

Η επιστήμη που μελετάει τα φαινόμενα στην επιφάνεια της γης λέγεται τηλεπισκόπηση (remote sensing) και αναλύεται στο υποκεφάλαιο 2.1. Η τηλεπισκόπηση βοηθάει για την μελέτη των πυρκαγιών, και η μέθοδος που χρησιμοποιείται για την ανάλυση των δορυφορικών εικόνων για την χαρτογράφηση των καμένων εκτάσεων λέγεται Burn Scar Mapping (BSM) που υλοποιήθηκε από το Αστεροσκοπείο Αθηνών, και αναλύεται εκτενέστερα στο κεφάλαιο 3.

Το αντικείμενο της πτυχιακής είναι η επεξεργασία δορυφορικών εικόνων για την χαρτογράφηση των καμένων περιοχών με την χρήση MapReduce σε Java. Για την ανάλυση των δορυφορικών εικόνων σε περιβάλλον προγραμματισμού Java χρειάζεται η Geospatial Data Abstraction Library (GDAL). Η βιβλιοθήκη της GDAL είναι ένα σημαντικό εργαλείο καθώς μετατρέπει πολλούς τύπους δορυφορικών εικόνων σε μορφές δεδομένων που βοηθούν το προγραμματιστή να διαχειριστεί με μεγαλύτερη ευκολία τα δεδομένα. Περισσότερη ανάλυση για τα εργαλεία που προσφέρει η GDAL αναφέρονται στην ενότητα 2.1.4. Η αποθήκευση όλων αυτών των δορυφορικών εικόνων και η επεξεργασία των πληροφοριών που περιέχουν χρειάζονται μεγάλο αποθηκευτικό χώρο και μεγάλη επεξεργαστική ισχύ. Γι αυτό το λόγο προτιμήθηκε ένα μοντέλο παράλληλου προγραμματισμού, το Hadoop framework. Το Hadoop, είναι μία υλοποίηση του προγραμματιστικού μοντέλου Map-Reduce, η οποία περιγράφεται στο υποκεφάλαιο 2.3, και είναι σχεδιασμένο για την

επεξεργασία μεγάλης κλίμακας δεδομένων σε κατανεμημένο περιβάλλον υπολογιστών. Με το Hadoop framework θα επιτευχθεί η ανάλυση όλων των δεδομένων που χρειάζονται για να καταγραφτούν οι καμένες εκτάσεις σε πολύ μικρό χρονικό διάστημα. Στην υλοποίηση του προγράμματος, που διατυπώνονται στο κεφάλαιο 4, αναλύονται υποπρογράμματα που αναπτύχθηκαν σύμφωνα με τα βήματα του αλγόριθμου BSM. Για να διευκολυνθεί η είσοδος δεδομένων στο hadoop αλλά και για να παραχθούν τα αποτελέσματα σε μορφή εικόνων χρησιμοποιήθηκε η βιβλιοθήκη GDAL. Ουσιαστικά, η GDAL, συνέβαλε στις μετατροπές των εικόνων σε πίνακες και αντίστροφα, κάνοντας έτσι πιο εύκολη τη διαδικασία επεξεργασίας των δεδομένων.

Έπειτα στο κεφάλαιο 5 παρουσιάζονται τα αποτελέσματα του προγράμματος, οι εικόνες που προκύπτουν και ο χρόνος εκτέλεσης του προγράμματος με διαφορετικούς παραμέτρους και σε διαφορετικά μηχανήματα. Τέλος στο κεφάλαιο 6 αναγράφονται συμπεράσματα και προτάσεις για επέκταση του προγράμματος.

Για την έκβαση αυτής της πτυχιακής, ήταν χρήσιμες οι πληροφορίες που αντλήθηκαν από τη πτυχιακή των Μ. Γαζάκη και Χ. Λόντος, με τίτλο “Επεξεργασία Δορυφορικών Εικόνων για την Χαρτογράφηση Καμένων Περιοχών με χρήση MapReduce” που έγινε στο Εθνικό και Καποδιστριακό Πανεπιστήμιο [2]. Η εν λόγω πτυχιακή που έχει πολλά κοινά στοιχεία με αυτή καθώς γίνεται μελέτη στο ίδιο αντικείμενο βοήθησε αρκετά στον σχεδιασμό, την υλοποίηση και την εύρεση βιβλιογραφίας. Η διαφορά των δύο πτυχιακών είναι κυρίως στη γλώσσα υλοποίησης, καθώς η παρούσα πτυχιακή υλοποιήθηκε σε Java, με σκοπό να λαμβάνει ως είσοδο πολλές εικόνες και να εξάγει τα αποτελέσματα για όλες τις εικόνες ταυτόχρονα. Αντίθετα η άλλη πτυχιακή είναι υλοποιημένη σε PYTHON και λαμβάνει ως είσοδο μία εικόνα τη φορά, εξάγοντας αντίστοιχα αποτελέσματα μόνο για την εικόνα αυτή.

Κεφάλαιο 2ο – Υπόβαθρο

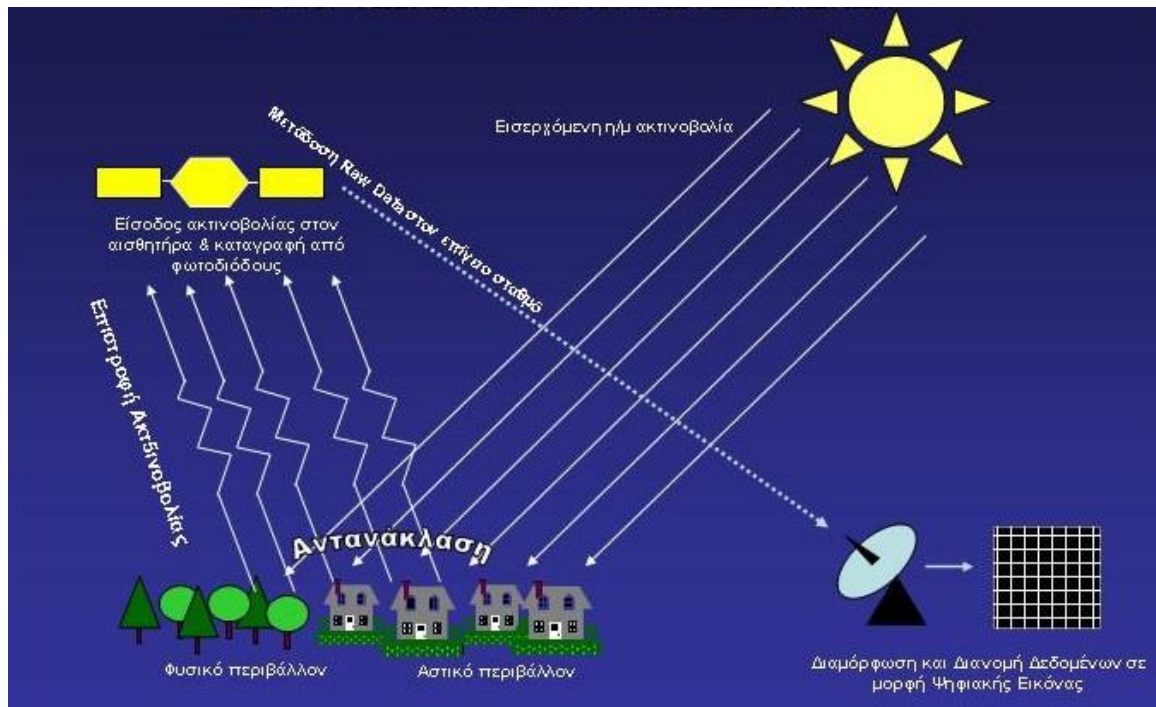
2.1 Τηλεπισκόπηση

2.1.1 Ορισμός

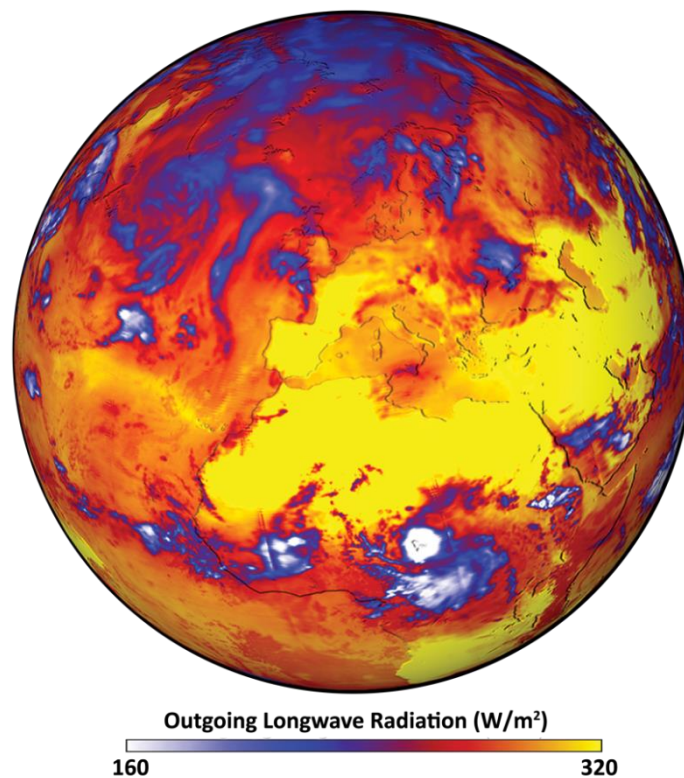
Η επιστήμη της τηλεπισκόπησης, γνωστή διεθνώς ως remote sensing, ασχολείται με την παρατήρηση φαινομένων και χαρακτηριστικών της γήινης επιφάνειας από απόσταση, βάση της αλληλεπίδρασης των υλικών που βρίσκονται επάνω σε αυτή με την ηλεκτρομαγνητική ακτινοβολία. Η παρατήρηση της επιφάνειας της Γης είναι δυνατή με τη χρήση ψηφιακών σαρωτών που ανιχνεύουν την ανάκλαση της ηλεκτρομαγνητικής ακτινοβολίας της γήινης επιφάνειας και την αποδίδουν ως ψηφιακή εικόνα. Κάθε αντικείμενο - επιφάνεια - υλικό που βρίσκεται επάνω στη Γη, έχει ένα μοναδικό τρόπο να αντανακλά την ηλεκτρομαγνητική ακτινοβολία σε διαφορετικά μήκη κύματος, και αυτό ονομάζεται φασματική υπογραφή του αντικειμένου. Για παράδειγμα η χλωροφύλλη, που βρίσκεται στα πράσινα μέρη των φυτών, έχει την ιδιότητα να ανακλά σε μεγάλο βαθμό την ηλεκτρομαγνητική ακτινοβολία στο πράσινο τμήμα του ορατού ηλεκτρομαγνητικού φάσματος και να την απορροφά στο μπλε και κόκκινο τμήμα. Η φασματική αυτή συμπεριφορά έχει ως αποτέλεσμα να αντιλαμβανόμαστε το πράσινο χρώμα των ζωντανών φυτών. Κατά παρόμοιο τρόπο όλα τα υλικά μπορούν να μελετηθούν, να εντοπισθούν και να απεικονισθούν χρησιμοποιώντας την αντανακλαστική τους συμπεριφορά. Εάν χρησιμοποιείται το ορατό τμήμα της ηλεκτρομαγνητικής ακτινοβολίας για την αναπαράσταση, τότε έχουμε μια πραγματική έγχρωμη εικόνα, ισοδύναμη με αυτές που καταγράφουν οι ψηφιακές φωτογραφικές μηχανές. Κατά συνέπεια όλα τα αντικείμενα στην επιφάνεια της γης μπορούν να ανιχνευτούν και να καταγραφτούν ανάλογα με την φασματική τους υπογραφή [3][4].

2.1.2 Αισθητήρες ακτινοβολίας και ψηφιακές εικόνες

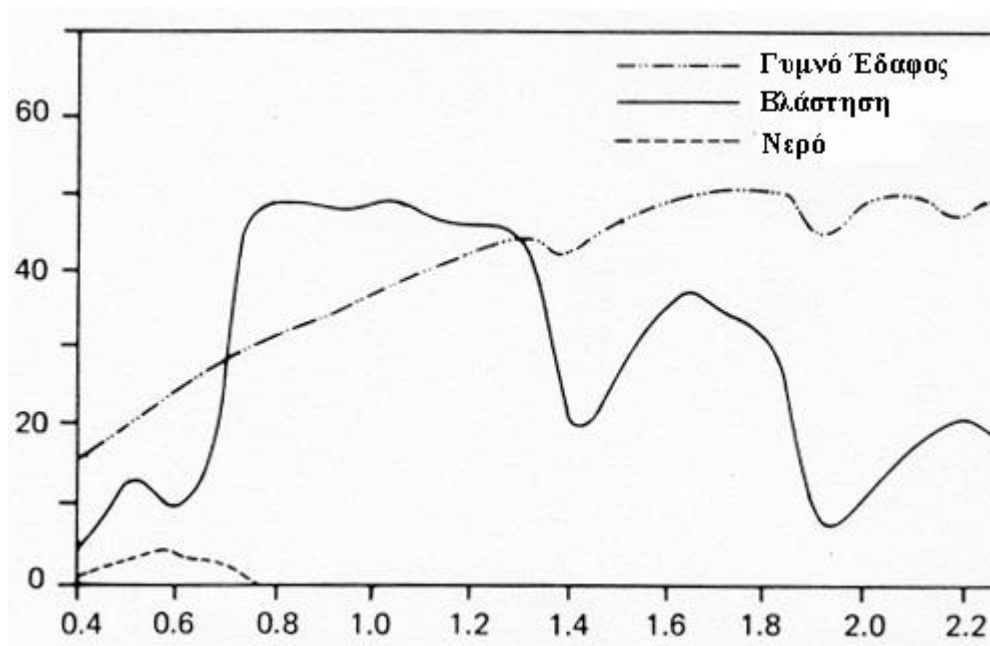
Οι σαρωτές ή αλλιώς αισθητήρες που καταγράφουν τις ακτινοβολίες είναι εγκατεστημένοι σε τεχνητούς δορυφόρους που βρίσκονται σε τροχιά γύρω από τη Γη ή βρίσκονται σε αερομεταφερόμενα μέσα (αεροσκάφη, ελικόπτερα) και μετρούν το ποσοστό της ηλεκτρομαγνητικής ακτινοβολίας που αντανακλάται από τα διάφορα υλικά. Για να κατανοηθεί καλύτερα πώς λειτουργούν οι απομακρυσμένοι αισθητήρες, μπορεί να φανταστεί κάποιος μια πηγή ηλεκτρομαγνητικής ακτινοβολίας (ήλιος) που εκπέμπει προς όλες τις κατευθύνσεις και διασχίζει την ατμόσφαιρα, όπου το εύρος του φάσματος της ακτινοβολίας είναι περιορισμένο. Ένα μέρος της ακτινοβολίας που φτάνει τελικά στη γη αντανακλάται, ένα άλλο μέρος διαχέεται στο περιβάλλον, ένα άλλο μέρος εκπέμπεται και ένα άλλο απορροφάται και επιστρέφει πίσω στο περιβάλλον. Η ανακλώμενη ακτινοβολία που έρχεται πίσω στην ατμόσφαιρα κατευθύνεται προς το διάστημα. Έτσι σε υψόμετρο 300 με 800 χιλιομέτρων, υπάρχουν τεχνητοί δορυφόροι σε τροχιά γύρω από την γη και είναι εξοπλισμένοι με αισθητήρες. Η ακτινοβολία που ανακλάται προς το διάστημα περνά μέσα από πρίσματα, τα οποία διαχωρίζουν την ακτινοβολία σε φασματικές ζώνες προκαθορισμένου εύρους. Έπειτα η ακτινοβολία περνάει από light emitting diodes (LED) ή από charge couple devices (CCD) και μετατρέπεται σε ηλεκτρικό σήμα. Το σήμα ψηφιοποιείται και στέλνεται σε επίγειους σταθμούς, όπου επεξεργάζεται και γίνεται εικόνα. Ο τύπος της εικόνας που παράγεται εξαρτάται από την φασματική ζώνη (band) που λειτουργούσε ο αισθητήρας, για παράδειγμα ραδιοκύματα, μικροκύματα υπέρυθρες, ορατό φως, υπεριώδεις κτλ. Η επιλογή των αισθητήρων που χρησιμοποιούνται εξαρτάται από το αντικείμενο που είναι υπό μελέτη, για παράδειγμα, αν το αντικείμενο υπό μελέτη είναι θαλάσσια οικοσυστήματα, θα πρέπει να εξεταστούν εικόνες στο μπλε τμήμα της ηλεκτρομαγνητικής ακτινοβολίας, διότι σε αυτήν την φασματική περιοχή η ακτινοβολία είναι σε θέση να διαπεράσει το νερό. Εάν ο στόχος όμως είναι να χαρτογραφήσει την ακτογραμμή, θα πρέπει να εξεταστούν οι εικόνες με υπεριώδης ακτινοβολία, επειδή το νερό απορροφά πλήρως την ηλεκτρομαγνητική ακτινοβολία σε αυτό το εύρος ενώ η ακτή αντανακλά. Ως αποτέλεσμα, δημιουργείται μια σημαντική διαφορά μεταξύ νερού και εδάφους, για να οριοθετηθεί και να καταγραφεί η ακτογραμμή [5].



**Εικόνα 2.1 Απλοποιημένο μοντέλο καταγραφής και διαμόρφωσης
τηλεπισκοπικών δεδομένων**



Εικόνα 2.2 Ακτινοβολία που ανακλάται από την γη



Εικόνα 2.3 Φασματικός διαχωρισμός βλάστησης, εδάφους και υδάτινων μαζών

X:Wavelength (μm). Y:Reflectance (%)

Οι ψηφιακές εικόνες που δημιουργούνται από τους αισθητήρες, αποθηκεύονται ως πίνακες τιμών σε έναν υπολογιστή όπου η τιμή του κάθε pixel αντιπροσωπεύει την αντανάκλαση της ηλεκτρομαγνητικής ακτινοβολίας. Αυτές οι ψηφιακές εικόνες έχουν τις ακόλουθες ιδιότητες [2]:

- Χωρική ανάλυση (spatial resolution), αναφέρεται στο μέγεθος των pixel σε πραγματικές διαστάσεις. Στην ουσία, η χωρική ανάλυση καθορίζει το ελάχιστο μέγεθος των αντικειμένων που μπορεί να αποτυπωθεί στην ψηφιακή εικόνα
- Φασματική ανάλυση (spectral resolution), είναι το φασματικό εύρος κάθε φασματικού καναλιού (band).
- Ραδιομετρική ανάλυση (radiometric resolution), είναι ο αριθμός των διαφορετικών εντάσεων της ακτινοβολίας που ο αισθητήρας είναι σε θέση να διακρίνει. Τυπικά αυτό κυμαίνεται από 8 έως 14 bits, που αντιστοιχούν σε 256 επίπεδα της κλίμακας του γκρι και έως 16.384 εντάσεις ή χρωματικές αποχρώσεις, σε κάθε ζώνη. Αυτό εξαρτάται επίσης από το θόρυβο του οργάνου.

- Χρονική ανάλυση (temporal resolution), είναι η συχνότητα των υπέργειων σαρώσεων από το δορυφόρο ή το αεροπλάνο, είναι χρήσιμο σε μελέτες που απαιτούν ανάλυση περιοχών ανά τακτά χρονικά διαστήματα όπως για παράδειγμα η παρακολούθηση αποψίλωσης των δασών.

2.1.3 Τηλεπισκόπηση και πληροφορική

Για περισσότερα από πενήντα χρόνια χρησιμοποιούνται εναέρια μέσα, για την παρατήρηση της γης, και την μελέτη φαινομένων που συμβαίνουν στην επιφάνεια και στην ατμόσφαιρα της. Ο όγκος των δεδομένων που καταγράφονται αυξάνεται ραγδαία, με αποτέλεσμα να δημιουργείται πρόβλημα αποθήκευσης και επεξεργασίας τους. Για παράδειγμα, αν μια συμβατική εικόνα μιας περιοχής είναι 10MB, η αντίστοιχη εικόνα από κάποιον δορυφόρο Landsat που περιέχει ψηφιακά δεδομένα είναι 200-300MB. Εξαιτίας αυτής της αύξησης των δεδομένων, η συμμετοχή των ηλεκτρονικών υπολογιστών στην επεξεργασία των δεδομένων έχει αυξηθεί. Επίσης έχουν αναπτυχθεί προηγμένα πακέτα λογισμικού επεξεργασίας εικόνας, προκειμένου να μειωθεί η οπτική ερμηνεία και γίνεται πιο γρήγορα η επεξεργασία των δεδομένων. Συγκεκριμένα, υπάρχουν πακέτα λογισμικού επεξεργασίας δορυφορικών δεδομένων που μπορούν να κατασκευάσουν χάρτες αυτοματοποιημένα. Χάρτες ειδικού σκοπού μπορούν επίσης να κατασκευαστούν με την ενσωμάτωση ή το συνδυασμό στοιχείων από διαφορετικούς δορυφόρους ή βάσεων δεδομένων με μη δορυφορικά δεδομένα. Τέτοια πακέτα λογισμικού είναι ERDAS, το eCognition, το ERmapper κ.α.

2.1.4 Βιβλιοθήκη GDAL

Οι γεωγραφικές εικόνες συνήθως αποθηκεύονται ως αρχεία δεδομένων raster. Μια εικόνα γραφικών raster είναι μια δομή δεδομένων που αποτελείται από κουκίδες (pixels) και αντιπροσωπεύει ένα ορθογώνιο πλέγμα εικονοστοιχείων. Υπάρχουν πολλές διαφορετικές μορφές αρχείων raster στις οποίες μπορούν να κωδικοποιηθούν τα δεδομένα. Μερικές μορφές αρχείων περιλαμβάνουν arc ascii-grid, arc binary-grid, Erdas Imagine (HFA) και GeoTIFF. Με τόσες πολλές και διαφορετικές μορφές αρχείων, είναι χρήσιμο να υπάρχει μια βιβλιοθήκη που να επιτρέπει την πρόσβαση στα δεδομένα χωρίς να απασχολεί τον χρήστη ποία μορφή αρχείου χρησιμοποιείται.

Η Geospatial Data Abstraction Library (GDAL) είναι μια τέτοια βιβλιοθήκη, που είναι γραμμένη σε C και είναι ανοικτού κώδικα και περιέχει ένα ενιαίο μοντέλο δεδομένων για όλες τις υποστηριζόμενες μορφές. Επίσης αποτελείται από ένα πακέτο χρήσιμων εντολών για τη μετάφραση και την επεξεργασία των δεδομένων. Για μεγαλύτερη ευκολία υπάρχουν διαθέσιμες συνδέσεις της GDAL σε άλλες γλώσσες, όπως JAVA, Python, Perl, Ruby, C # /. Net, Visual Basic 6 και R-Forge [6][7].

Εκτός από τα δεδομένα ανά pixel που περιέχονται στη raster εικόνα, οι γεωγραφικές εικόνες περιέχουν και μερικές πρόσθετες πληροφορίες τα metadata. Για παράδειγμα, εικόνες της επιφάνειας της Γης συνοδεύονται από το γεωγραφικό μήκος και πλάτος ή κάποιες άλλες γεωγραφικές πληροφορίες για να καθορίσουν μέρος της γης που απεικονίζουν, και η GDAL επιτρέπει την εύκολη πρόσβαση σε αυτές.

2.1.5 Εφαρμογές της τηλεπισκόπησης

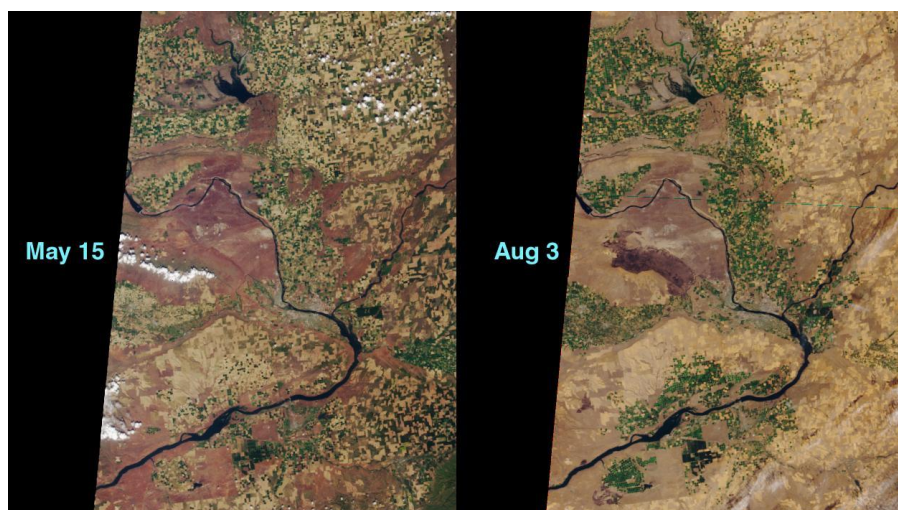
Τα δεδομένα που λαμβάνονται από απομακρυσμένους αισθητήρες, χρησιμοποιούνται σήμερα σε διάφορα επιστημονικά πεδία, όπως στη μετεωρολογία για την παρακολούθηση της κλιματικής αλλαγής και την πρόγνωση καιρού, στη γεωλογία για την ανακάλυψη των φυσικών πόρων και τη μελέτη του ανάγλυφου της γης, στη χωροταξία και στον χωροταξικό σχεδιασμό, στη βιολογία για την μελέτη των οικοσυστημάτων, στη γεωργία για πιο αποτελεσματική χρήση της γεωργικής γης, στη δασοκομία για την παρακολούθηση των δασικών πυρκαγιών, στην ωκεανογραφία στο χάρτη βυθό της θάλασσας, κ.λπ. Μια άλλη χρήση της τηλεπισκόπησης είναι η κατασκοπεία μεταξύ των χωρών.

2.1.6 Χαρτογράφηση καμένων περιοχών

Η χαρτογράφηση καμένων περιοχών (Burn Scar Mapping), είναι μια δημοφιλής εφαρμογή της τηλεπισκόπησης, καθώς η αυτόματη χαρτογράφηση των περιοχών έχει γίνει αρκετά εύκολη και αποτελεσματική, χάρη στους υπολογιστές. Η ανάπτυξη των αλγορίθμων BSM προέκυψε από την ανάγκη για την ακριβή εκτίμηση του μεγέθους των δασικών πυρκαγιών. Η γνώση αυτών των τελευταίων γεγονότων πυρκαγιάς θα βοηθήσει στην εκτίμηση του κινδύνου για τις μελλοντικές πυρκαγιές, και ως αποτέλεσμα θα βοηθήσει την πρόληψη των πυρκαγιών. Επιπλέον, η αναγνώριση των καμένων περιοχών είναι εξαιρετικής σημασίας για της υπηρεσίες

δασοκομίας που κάνουν αναδάσωση και την κάνουν προετοιμασία για την αντιμετώπιση των πυρκαγιών για τα επόμενα χρόνια [8].

Τα δεδομένα που λαμβάνονται από τους δορυφόρους, NOAA / AVHRR, Landsat TM και ETM +, MODIS, MERIS, SPOT και IRS μπορούν να χρησιμοποιηθούν για να εντοπιστεί καμένη έκταση. Στην πράξη, η χαρτογράφηση καμένων εκτάσεων βασίζεται στην φασματική απόκριση της καμένης βλάστησης. Ενώ η υγιής βλάστηση αντανακλά εγγύς υπέρυθρη (NIR) ακτινοβολία και απορροφά το κόκκινο φως στο ορατό (VIS) τμήμα του φάσματος, οι περιοχές καμένων εκτάσεων αντανακλούν συγκριτικά περισσότερη ακτινοβολία στο VIS και στο shortwave infrared (SWIR) τμήμα του φάσματος και απορροφά την ακτινοβολία NIR. Αυτό αποδίδεται στην καταστροφή της δομής των φυτών και φύλλων. Κατά συνέπεια, οι απομακρυσμένοι αισθητήρες που λειτουργούν στις σχετικές φασματικές ζώνες, είναι σε θέση να καταγράψουν την αλλαγή της ακτινοβολίας, που προκύπτει από την κατάργηση της υγιής βλάστησης και την παρουσία του άνθρακα ή του γυμνού εδάφους στην περιοχή της φωτιάς. Αυτές οι φασματικές αποκλίσεις μεταξύ μιας περιοχής με βλάστηση και μιας καμένης περιοχής που εξάγουν οι εικόνες, επιτρέπουν την αναγνώριση των καμένων εκτάσεων.



Εικόνα 2.4 Πριν και μετά την πυρκαγιά, Ιούνιος 2000, Ουάσιγκτον

Η εικόνα 2.4 δείχνει το "πριν και μετά" της περιοχής γύρω από το Hanford Nuclear Reservation κοντά στο Richland της Ουάσιγκτον και τραβήχτηκε από τον δορυφόρο MISR. Στις 27 Ιουνίου του 2000, πυροδοτήθηκε μια πυρκαγιά από την συντριβή αυτοκινήτων. Οι φλόγες εξαπλώθηκαν από το ζεστό καλοκαίρι και τους ανέμους. Μια μέρα μετά από το ατύχημα καταγράφηκαν περίπου 100.000 καμένα στρέμματα, και η εξάπλωση της φωτιάς ανάγκασε το κλείσιμο των αυτοκινητοδρόμων και την απώλεια των σπιτιών.

2.2 Κατανεμημένα Συστήματα

2.2.1 Ορισμός

Στην πληροφορική κατανεμημένα συστήματα ονομάζονται οι υπολογιστές οι οποίοι επιτρέπουν την ταυτόχρονη εκτέλεση πολλαπλών συνεργαζόμενων προγραμμάτων σε μία ή περισσότερες επεξεργαστικές μονάδες [9]. Ένας άλλος ορισμός για τα κατανεμημένα συστήματα είναι, το σύνολο των υπολογιστών που είναι συνδεδεμένοι σε δίκτυο και μοιράζονται τμήματα μίας διεργασίας, όπως ένας υπολογιστής διαμοιράζει τις διεργασίες που κάνει στους πολλαπλούς πυρήνες του. Οι υπολογιστές που αποτελούν ένα κατανεμημένο σύστημα λέγονται κόμβοι (nodes), και μπορούν να είναι προσωπικοί υπολογιστές, servers, clusters, ακόμα και κινητά τηλέφωνα, συνδεδεμένα σε δίκτυο, τοπικό ή ευρύτερο, ανάλογα από την μεταξύ τους γεωγραφική απόσταση. Ένα κατανεμημένο σύστημα μπορεί να αποτελείται από υπολογιστές που τρέχουν διαφορετικά λειτουργικά συστήματα (πχ. Unix, Windows, Linux), και η επικοινωνία μεταξύ τους γίνεται συνήθως με System Network Architecture ή Transmission Control Program/Internet Protocol (TCP/IP).

2.2.2 Χαρακτηριστικά Κατανεμημένων Συστημάτων

Ανεξάρτητα από το μοντέλο κατανεμημένης επεξεργασίας που υποστηρίζει ένα κατανεμημένο σύστημα, υπάρχουν κάποια χαρακτηριστικά που αξιολογούν, αλλά και ορίζουν ένα κατανεμημένο σύστημα [9][10][11]. Τα χαρακτηριστικά αυτά είναι τα εξής:

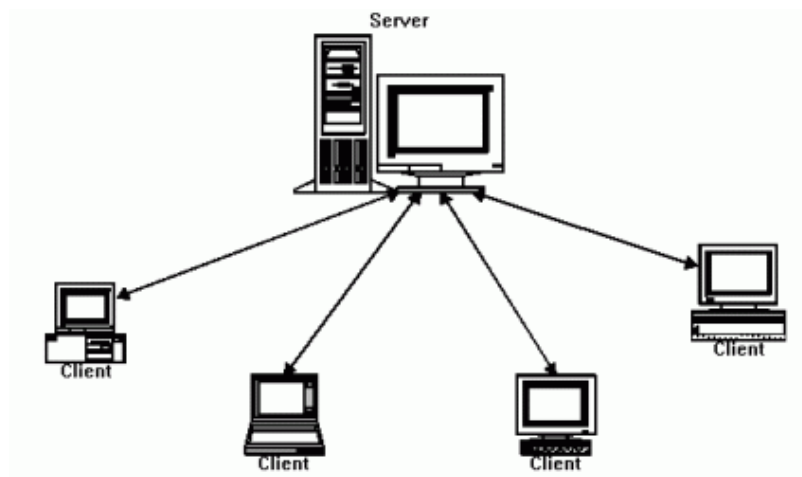
1. Ευελιξία (flexibility), ένα καταναμεμημένο σύστημα επιτρέπει την αλλαγή των κόμβων του και των πόρων του με μεγάλη ευκολία, καθώς και την επέκταση του. Η χρήση ενδιάμεσου λογισμικού για κατασκευή και την επικοινωνία καταναμεμημένων συστημάτων είναι μία κοινή τεχνική για την αύξηση της ευελιξίας.
2. Ανοιχτή υλοποίηση (openness), δηλαδή η δυνατότητα του συστήματος να είναι εύκολα επεκτάσιμο και τροποποιήσιμο.
3. Διαφάνεια (transparency), δηλαδή η δυνατότητα του συστήματος να φαίνεται σαν ένα ενιαίο μηχάνημα προς τον χρήστη.
4. Κλιμάκωση (scalability), το σύστημα πρέπει να είναι εύκολα επεκτάσιμο, δηλαδή να μπορεί να αυξήσει την επεξεργαστική του ισχύ, χωρίς αυτό να γίνεται βάρος την επίδοσης του, καθώς η αύξηση του μεγέθους ενός συστήματος είναι αντιστρόφως ανάλογο με τον συγχρονισμό του και την επικοινωνία μεταξύ των κόμβων του.
5. Ανοχή στις βλάβες (fault tolerance), δηλαδή η δυνατότητα του συστήματος να συνεχίζει να τρέχει τις διεργασίες του, χωρίς να το επηρεάσει κάποιο σφάλμα που θα προκύψει στους κόμβους ή και στην μεταξύ τους επικοινωνία(δίκτυο).

2.2.3 Αρχιτεκτονική Επικοινωνίας

Η αρχιτεκτονική δείχνει τον τρόπο με τον οποίο επικοινωνούν οι κόμβοι σε ένα καταναμεμημένο σύστημα. Υπάρχουν τρεις τρόποι να γίνει αυτό, με το μοντέλο πελάτη-διακομιστή (client-server), των ομότιμων υπολογιστών (peer-to-peer) και ο συνδυασμός αυτών [12][13].

Το μοντέλο πελάτη-διακομιστή (client-server) αποτελείται από δύο τύπους κόμβων, τους διακομιστές και τους πελάτες. Οι διακομιστές (servers) παρέχουν υπηρεσίες ή δεδομένα, και οι πελάτες (clients) κάνουν αιτήματα προς τους διακομιστές για την χρήση αυτών των υπηρεσιών ή των δεδομένων. Πιο αναλυτικά ο πελάτης στέλνει ένα αίτημα (request) στον διακομιστή για μία διεργασία. Ο διακομιστής εκτελεί την διεργασία και στέλνει ένα μήνυμα (response) στον πελάτη που περιέχει τα αποτελέσματα της διεργασίας. Αυτό το μοντέλο επικοινωνίας δεν επιτρέπει στους πελάτες να επικοινωνούν άμεσα μεταξύ τους. Ο διακομιστής

μοιράζεται τους πόρους του μηχανήματός του με τους πελάτες, κατά συνέπεια αν αυξηθούν οι πελάτες, η απόδοση του συστήματος θα μειωθεί. Ένα άλλο χαρακτηριστικό του συστήματος αυτού είναι ότι αν υπάρξει κάποιο σφάλμα στον διακομιστή τότε ολόκληρο το σύστημα βγαίνει εκτός λειτουργίας. Το μοντέλο πελάτης-διακομιστής χρησιμοποιείται ευρέως, με κάποιες διαφοροποιήσεις και προσθήκες για να γίνει πιο αξιόπιστο, ασφαλές και γρήγορο, καθώς είναι εύκολο στον σχεδιασμό του συστήματος και στον διαμερισμό των διεργασιών .



Εικόνα 2.5 Αρχιτεκτονική επικοινωνίας μοντέλου πελάτη-διακομιστή

2.3 Προγραμματιστικό μοντέλο MapReduce

2.3.1 Ορισμός

Το MapReduce είναι ένα προγραμματιστικό μοντέλο που μπορεί να επεξεργάζεται και να παράγει μεγάλο όγκο δεδομένων, είναι βασισμένο σε έναν αλγόριθμο που πετυχαίνει διαμερισμό των δεδομένων ώστε να γίνει παράλληλη επεξεργασία αυτών σε ένα κατανεμημένο σύστημα ή ένα cluster. Αποτελείται από δύο διεργασίες, την Map() και την Reduce(), η Map φιλτράρει και ταξινομεί τα δεδομένα με βάση κριτήρια που επιλέγει ο χρήστης, ενώ η Reduce συνοψίζει τα αποτελέσματα. Ο σκελετός του MapReduce αναπτύχθηκε από την Google για να βοηθήσει τους προγραμματιστές να επικεντρωθούν στο πρόγραμμά τους, ενώ η κατανομή των δεδομένων, η παραλληλοποίηση των εργασιών στον cluster και η διαχείριση των σφαλμάτων γίνεται αυτόματα [14][15][16][17].

2.3.2 Προγραμματισμός με MapReduce

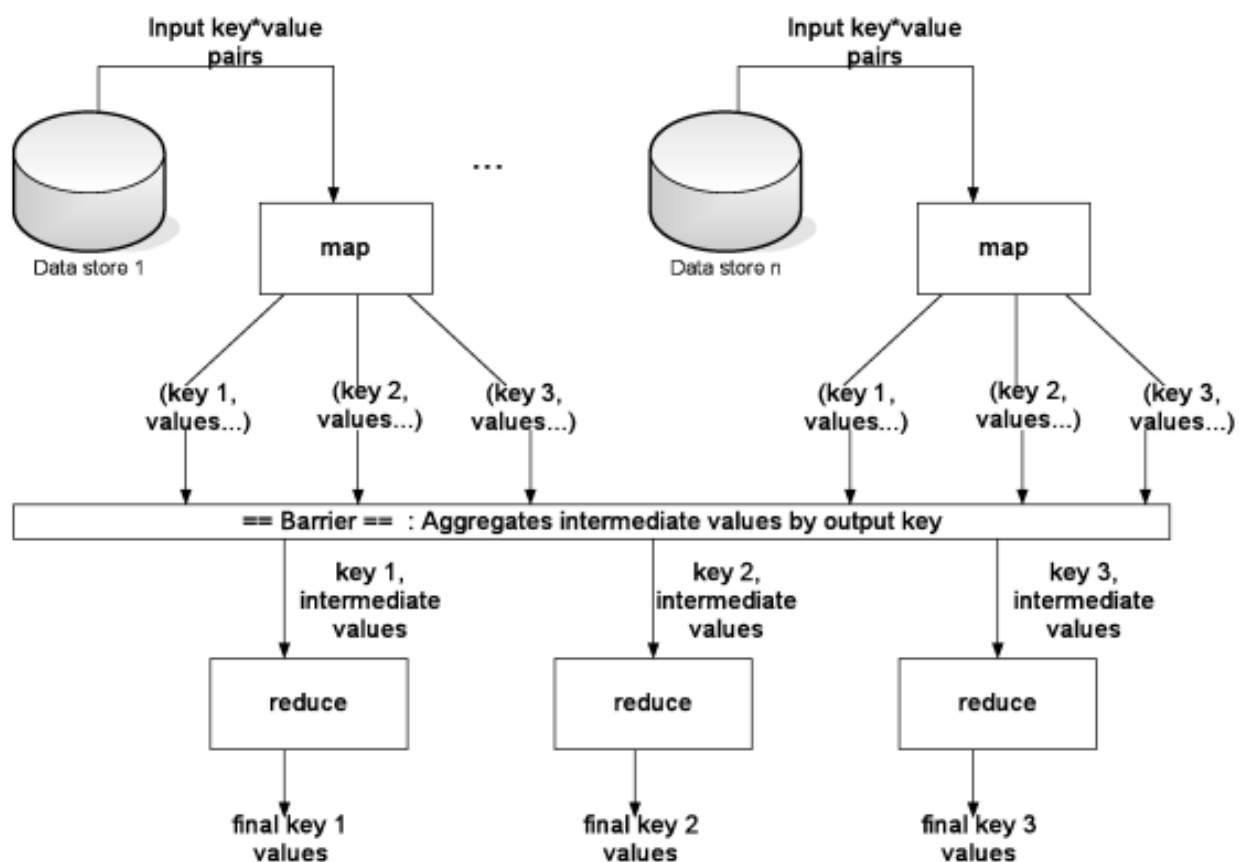
Τα βήματα για να προγραμματίσει κάποιος με το μοντέλο MapReduce είναι τα εξής:

1. Τα δεδομένα πρέπει να αποθηκευτούν στο καταναμημένο σύστημα αποθήκευσης (distributed filesystem).

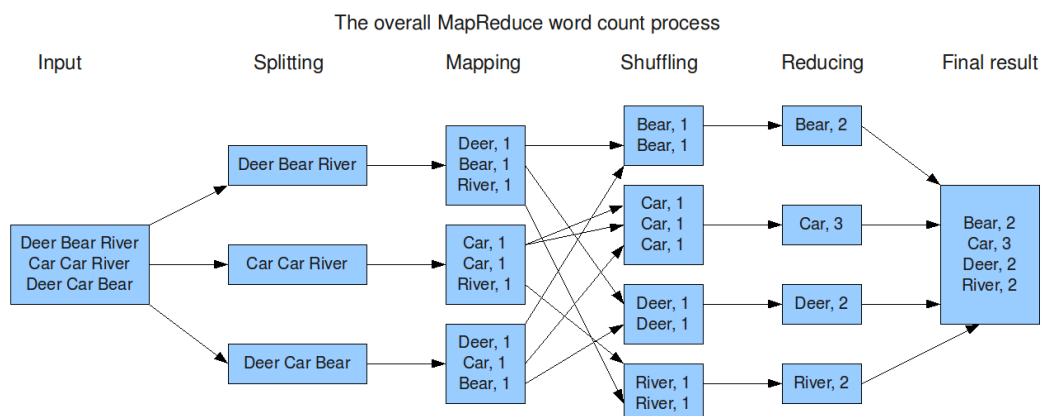
2. Ο μηχανισμός του MapReduce διαχωρίζει τα δεδομένα και τα στέλνει στους Mappers, συνήθως μια γραμμή ανά αρχείο είναι μια εγγραφή. Ο κάθε Mapper τρέχει μια διαδικασία που έχει ορίσει ο προγραμματιστής, με την οποία τα δεδομένα ορίζονται σε <κλειδί, τιμή>.

3. Αυτόματα γίνεται ταξινόμηση με βάση το κλειδί, και οι Reducers λαμβάνουν μια λίστα με όλες τις τιμές ανά κλειδί.

4. Δημιουργία του τελικού ζευγαριού <κλειδί, τιμή>, καταγραφή σε αρχείο και στο distributed filesystem. Συνήθως η έξοδος ενός τέτοιου αρχείου χρησιμοποιείται για την είσοδο ενός άλλου αντίστοιχου προγράμματος.



Εικόνα 2.6 Διάγραμμα της διαδικασίας MapReduce



Εικόνα 2.7 Παράδειγμα μέτρησης λέξεων σε ένα αρχείο (wordcount)

2.4 Apache Hadoop

2.4.1 Τι είναι το Hadoop

Το hadoop είναι ένα ελεύθερο λογισμικό, που αναπτύχθηκε από την Apache Software Foundation, εμπνευσμένο από το Google's MapReduce και το Google File System και υλοποιημένο σε java. Είναι κατάλληλο για την ανάπτυξη προγραμμάτων που διαχειρίζονται μεγάλης κλίμακας δεδομένα. Σχεδιάστηκε για να μπορεί να λειτουργήσει σε μεγάλους servers-clusters, αλλά και σε συμβατικούς υπολογιστές, που προσφέρουν μικρή επεξεργαστική ισχύ, που όμως αν συνδεθούν πολλοί μεταξύ τους προσφέρουν μια αξιόλογη επεξεργαστική δύναμη [18][19][20][21].

2.4.2 Περισσότερα για το Hadoop

Το Hadoop περιλαμβάνει τα ακόλουθα υποπρογράμματα:

- Hadoop Common: Βασικά εργαλεία για υποστήριξη άλλων υποπρογραμμάτων.
- HDFS: Κατανεμημένο σύστημα διαχείρισης αρχείων.
- MapReduce: Framework για κατανεμημένη επεξεργασία μεγάλου όγκου δεδομένων.

Άλλα προγράμματα που σχετίζονται με το Hadoop

- Avro: Σύστημα σειριοποίησης δεδομένων
- Chukwa: Σύστημα συλλογής δεδομένων για την οργάνωση μεγάλων καταναμημένων συστημάτων.
- HBase: Καταναμημένη βάση δεδομένων που υποστηρίζει αποθηκευτικό χώρο για μεγάλους πίνακες.
- Mahout: Ανάπτυξη αλγόριθμων μηχανικής μάθησης και εξόρυξη δεδομένων
- Pig: Μια υψηλού επιπέδου γλώσσα ροής δεδομένων για παράλληλους υπολογισμούς.
- ZooKeeper: Υπηρεσία συντονισμού υψηλής απόδοσης για καταναμημένες εφαρμογές.

2.4.3 Πλεονεκτήματα του Hadoop

- Επεκτασιμότητα (scalability): Δυνατότητα αξιόπιστης αποθήκευσης και επεξεργασίας μέχρι και petabytes δεδομένων
- Οικονομία Πόρων: Κατανομή δεδομένων και επεξεργασίας σε clusters που αποτελούνται από μέχρι και χιλιάδες κοινούς υπολογιστές.
- Αποδοτικότητα: Με την κατανομή των δεδομένων, η επεξεργασία γίνεται παράλληλα σε όλους τους κόμβους, προσφέροντας γρήγορη εκτέλεση των εργασιών.
- Αξιοπιστία (fault tolerance): Επιτυγχάνεται μέσω της αυτόματης διατήρησης πολλαπλών αντιγράφων των δεδομένων, καθώς και αυτόματης ανάθεσης των εργασιών υπολογισμού σε νέους κόμβους σε περίπτωση βλάβης.

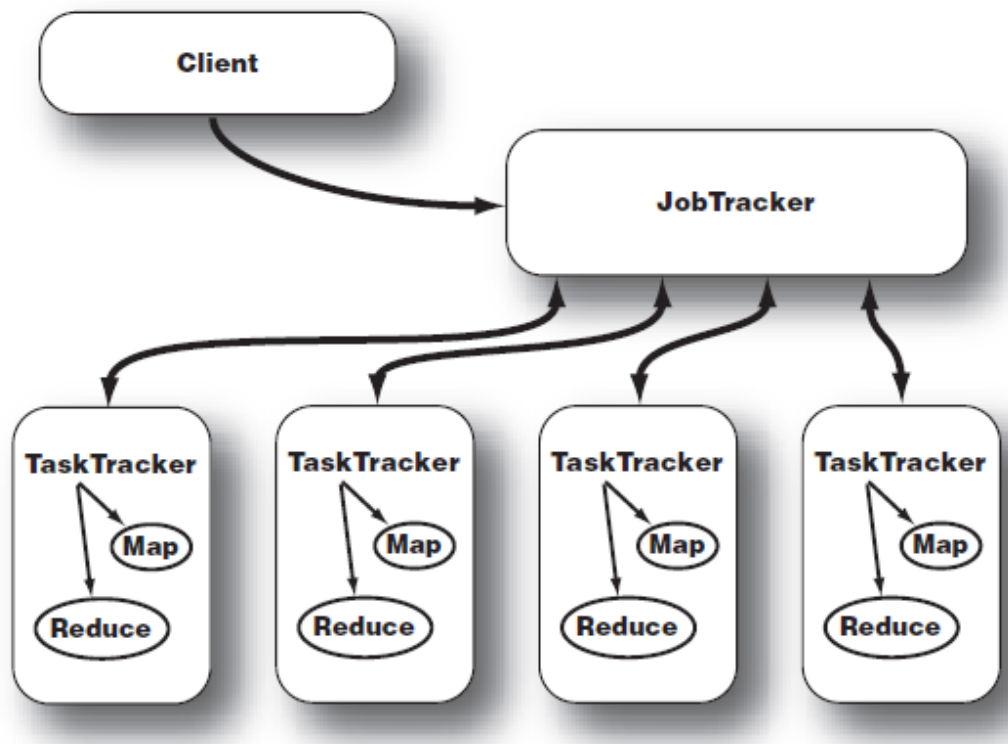
2.4.4 Αρχιτεκτονική του Hadoop

Το Hadoop ακολουθεί το μοντέλο master - slave, όπου ο JobTracker είναι ο master και είναι μοναδικός ανά cluster και οι TaskTrackers οι slaves, οι οποίοι είναι πολλοί [18][20].

JobTracker: όταν ξεκινάει μια εφαρμογή στον cluster που περιέχει διεργασία mapreduce, ο JobTracker ορίζει το πλάνο εκτέλεσης διαλέγοντας ποιο αρχείο θα επεξεργαστεί, έπειτα αναθέτει στους κόμβους (nodes) διαφορετικές διεργασίες και τις παρακολουθεί όσο αυτές τρέχουν. Ο JobTracker προσπαθεί η δουλειά να γίνεται σε κόμβους που είναι κοντά στα δεδομένα, αν ο κόμβος είναι απασχολημένος τότε

κοντινοί κόμβοι αναλαμβάνουν την δουλειά. Με αυτόν τον τρόπο μειώνει την μεταφορά δεδομένων και την υπερφόρτωση του κυρίως δικτύου. Αν μια δουλειά αποτύχει να ολοκληρωθεί, ο TaskTracker την ξεκινάει ξανά, μετά από ένα καθορισμένο όριο αποτυχιών, μπορεί να την αναθέσει σε άλλο κόμβο.

TaskTracker: αναλαμβάνει την διαχείριση και την εκτέλεση των διεργασιών (map-reduce tasks) ανά κόμβο που ο JobTracker έχει ορίσει. Σε κάθε κόμβο υπάρχει ένας TaskTracker που μπορεί όμως να δημιουργήσει εικονικά μηχανήματα (Java Virtual Machines) για να τρέχει διεργασίες map ή reduce παράλληλα. Ο TaskTracker επίσης είναι υπεύθυνος να επικοινωνεί συνεχώς με τον JobTracker και να τον ενημερώνει για την εξέλιξη της διεργασίας, ώστε σε περίπτωση σφάλματος ο JobTracker να αναθέσει την δουλειά σε άλλο κόμβο.



Εικόνα 2.8 Αλληλεπίδραση πελάτη, jobtracker και tasktracker σε ένα cluster

2.4.5 Κατανεμημένο σύστημα αρχείων του Hadoop (HDFS)

Το HDFS είναι ένα κατανεμημένο σύστημα αρχείων που σχεδιάστηκε για να μπορεί να αποθηκεύει και να διαχειρίζεται μεγάλης κλίμακας αρχεία, είναι υλοποιημένο σε java και ακολουθεί το μοντέλο master-slave. Βασικό του

πλεονέκτημα είναι ότι μπορεί να αποθηκεύσει δεδομένα που φτάνουν τα 100TB σε ένα αρχείο, κάτι που τα περισσότερα συστήματα δεν το επιτρέπουν. Την δυνατότητα αυτή την έχει γιατί κάθε αρχείο που φορτώνεται στον cluster διασπάται σε τμήματα (blocks) των 64MB ή 128MB το καθένα, και αυτά αποθηκεύονται σε πολλά μηχανήματα του cluster. Το θέμα της αξιοπιστίας επιτυγχάνεται με το σύστημα να αντιγράφει το κάθε αρχείο, από προεπιλογή τρεις φορές, σε διαφορετικούς κόμβους. Τα είδη των κόμβων που έχει το HDFS είναι οι εξής: ο NameNode και ο DataNode και Secondary NameNode.

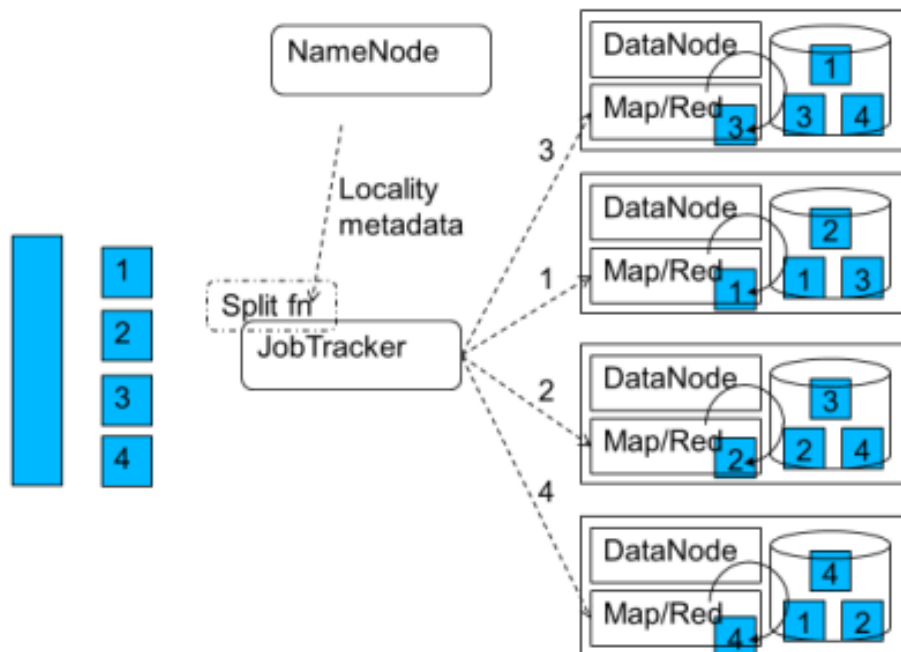
Ο NameNode τρέχει στον master κόμβο του cluster, διαχειρίζεται την πρόσβαση των πελατών στα αρχεία, καταγράφει δημιουργίες αρχείων, διαγραφές αρχείων κλπ, καθώς επίσης αποθηκεύει λίστες με τα αρχεία, τα blocks για κάθε αρχείο, των DataNodes που περιέχουν το κάθε block κτλ. Αν ο NameNode σταματήσει να λειτουργεί, σταματάει και ολόκληρο το σύστημα αρχείων.

Ο DataNode τρέχει σε κάθε slave στον cluster. Αποθηκεύει τα blocks ως ξεχωριστά αρχεία στον τοπικό του σύστημα αρχείων και στέλνει αναφορά στον NameNode με όλα τα blocks. Όλοι οι DataNodes δεν χρειάζεται να έχουν το ίδιο λειτουργικό σύστημα.

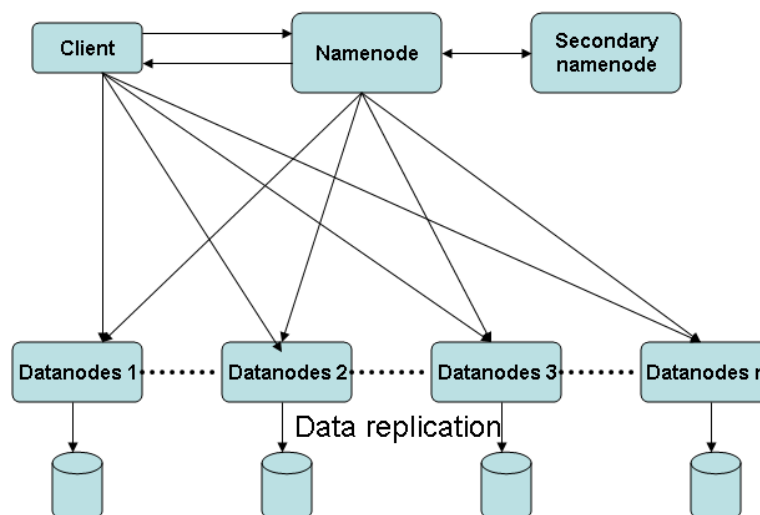
Επειδή, όπως αναφέρεται παραπάνω, αν ο NameNode σταματήσει να λειτουργεί, σταματάει και το HDFS, υπάρχει ένας βοηθητικός κόμβος, ο Secondary NameNode (SNN). Ο SNN επικοινωνεί με τον NameNode ανά τακτά χρονικά διαστήματα και παίρνει στιγμιότυπα των δεδομένων. Τα στιγμιότυπα αυτά βοηθούν τον NameNode μετά την επανεκκίνησή του, να επανέλθει σε σωστή λειτουργία.

Το HDFS είναι ενσωματωμένο με το βασικό πακέτο εγκατάστασης του Hadoop, χωρίς να εμποδίζει αυτό τον χρήστη να κάνει χρήση κάποιου άλλου κατανεμημένου συστήματος αρχείων. Το HDFS όμως προσφέρει ένα βασικό πλεονέκτημα σε σχέση με άλλα συστήματα αρχείων που δεν είναι πάντα διαθέσιμο, το πλεονέκτημα αυτό είναι ότι ο JobTracker γνωρίζει την τοποθεσία των δεδομένων και όταν αναθέτει σε κάποιον TaskTracker να κάνει μια δουλειά map-reduce, επιλέγει κάποιο κόμβο που τα δεδομένα που θα χρησιμοποιήσει είναι κοντά του, ώστε το δίκτυο να αποφεύγει την υπερφόρτωση. Για παράδειγμα αν ένας κόμβος A έχει τα δεδομένα (a,b,c), και ένας κόμβος B έχει τα δεδομένα (d,e,f), τότε ο JobTracker θα αναθέσει στον A να κάνει map-reduce τα δεδομένα (a,b,c) και στον B τα δεδομένα

(d,e,f). Η κατανομή αυτή των εργασιών προσφέρει μεγάλη εξοικονόμηση χρόνου στο σύνολο της δουλειάς.



Εικόνα 2.9 Καταμερισμός εργασιών στους TaskTrackers ανάλογα με τα δεδομένα που έχουν αποθηκευμένα.



Εικόνα 2.10 Αλληλεπίδραση NameNode, DataNode σε ένα Hadoop cluster

Κεφάλαιο 3ο – Αλγόριθμος χαρτογράφησης καμένων εκτάσεων (BSM_NOA)

3.1 Περιγραφή

Ο αλγόριθμος BSM υλοποιήθηκε στο Ινστιτούτο Διαστημικών Εφαρμογών και Τηλεπισκόπησης του Εθνικού Παρατηρητηρίου Αθηνών, και ταξινομεί τα pixel μιας δορυφορικής εικόνας ή μια χρονολογική ακολουθία από εικόνες, σε καμένες και μη. Η διαδικασία ταξινόμησης βασίζεται στη χρήση σταθερών ορίων που εφαρμόζονται σε ειδικούς δείκτες που αναφέρονται παρακάτω (3.2.2) [8].

3.2 Διαδικασία επεξεργασίας εικόνας

Ο αλγόριθμος bsm_noa χωρίζεται σε τρία στάδια όπως φαίνεται στην εικόνα 3.1. Το πρώτο στάδιο περιέχει την προεπεξεργασία της εικόνας, στο δεύτερο στάδιο γίνεται η επεξεργασία της εικόνας και στο τρίτο στάδιο γίνεται έλεγχος ποιότητας και ερμηνεία αποτελεσμάτων.

3.2.1 Προεπεξεργασία εικόνων

Το πρώτο στάδιο του αλγορίθμου περιέχει τα παρακάτω βήματα:

- Δορυφορικά δεδομένα Ραδιομετρικής ομαλοποίησης, δηλαδή την εξάλειψη των σφαλμάτων που προκαλούνται από τις τεχνικές δυσλειτουργίες της διαδικασίας καταγραφής και μετάδοσης, όπως για παράδειγμα την απορρύθμιση του αισθητήρα.
- Διορθώσεις σφαλμάτων των δορυφορικών δεδομένων που προκαλούνται από τη γη και την δορυφορική κίνηση της καθώς και το έντονο ανάγλυφο της περιοχής.
- Γεωγραφική αναφορά της εικόνας, δηλαδή ορισμός εικόνας σε ένα σύστημα γεωγραφικών συντεταγμένων.
- Ένωση των μερικώς επικαλυπτόμενων περιοχών ανά εικόνα.

- Σωστός ορισμός των περιοχών που καλύπτονται από σύννεφα-νερό-σκιά στην εικόνα.
- Υπολογισμός των δεικτών ανά pixel.
- Ορισμός των κατάλληλων τιμών στα κατώφλια χρησιμοποιώντας δείγματα παλαιότερων πυρκαγιών από το ελληνικό έδαφος.

3.2.2 Στάδιο βασικής επεξεργασίας

Κατά την διάρκεια της βασικής επεξεργασίας, οι δείκτες που υπολογίστηκαν στην προεπεξεργασία χρησιμοποιούνται για να γίνει σύγκριση με το σχετικό κατώφλι. Οι δείκτες αυτοί είναι:

- Ο δείκτης ανάκλασης της εγγύς υπέρυθρης (near infrared) ακτινοβολίας (R_{NIR}).
- Ο δείκτης ανάκλασης της μέσης υπέρυθρης (median infrared) ακτινοβολίας (R_{MIR}).
- Ο λόγος κανονικοποίησης καμένου pixel – Normalized Burn Ratio (NBR), όπου υπολογίζεται με τον παρακάτω μαθηματικό τύπο:

$$NBR = (R_{NIR} - R_{MIR}) / (R_{NIR} + R_{MIR}),$$

Ο δείκτης NBR χρησιμοποιείται παγκοσμίως για την χαρτογράφηση καμένων εκτάσεων, λόγω του ότι η τιμές ανάκλασης του κόκκινου χρώματος και της μέσης υπεριώδους ακτινοβολίας παρουσιάζουν μεγαλύτερη αλλαγή σε περίπτωση πυρκαγιάς.

- Ο λόγος διαφοράς βλάστησης – Normalized Difference Vegetation Index (NDVI), ο οποίος υπολογίζεται με τον τύπο:

$$NDVI = (R_{NIR} - R_{RED}) / (R_{NIR} + R_{RED}),$$

όπου RED αντιπροσωπεύει το κόκκινο στο ορατό κομμάτι του φάσματος της ηλεκτρομαγνητικής ακτινοβολίας. Ο τύπος αυτός χρησιμοποιείται σε περίπτωση ανάλυσης δεδομένων πολλαπλών χρονικών στιγμών. Όπου στην περίπτωση αυτή η διαφορά του δείκτη υπολογίζεται από τον τύπο:

$$NDVI = NDVI_{IPIN} - NDVI_{META}$$

- Ο δείκτης ALBEDO, ο οποίος υπολογίζεται με τον τύπο:

$$ALBEDO = (R_{NIR} + R_{RED}) / 2$$

Σε εξαιρετικά διαφοροποιημένα οικοσυστήματα όπως στην Ελλάδα, ο δείκτης NBR και NDVI μπορεί να μην δίνει ακριβή διαφοροποίηση των καμένων και των μη καμένων περιοχών. Λόγω αυτού, ο BSM_NOA χρησιμοποιεί επιπρόσθετα την εμπειρική προσέγγιση του δείκτη ALBEDO, η οποία αποτελεί ένδειξη της φωτεινότητας επιφάνειας.

Η ταξινόμηση κάθε pixel με συντεταγμένες (i,j) γίνεται ως εξής:

- Το pixel (i,j) είναι καμένο όταν η τιμή του δείκτη NBR είναι μικρότερη από το NBR_Threshold - κατώφλι (T1), η τιμή του NIR είναι μικρότερη ή ίση με το NIR_Threshold (T2), η τιμή του ALBEDO είναι μικρότερη ή ίση με το ALBEDO_Threshold (T3) και η τιμή του NDVI είναι μεγαλύτερη του NDVI_Threshold (T4).
- Το pixel (i,j) είναι δεν καμένο όταν η τιμή του δείκτη NBR είναι μεγαλύτερη από το NBR_Threshold+1 - κατώφλι (T1), η τιμή του NIR είναι μεγαλύτερη από το NIR_Threshold (T2), η τιμή του ALBEDO είναι μεγαλύτερη από το ALBEDO_Threshold (T3) και η τιμή του NDVI είναι μικρότερη ή ίση του NDVI_Threshold (T4).
- Το pixel (i,j) δεν θεωρείται καμένο ή μη καμένο (για παράδειγμα, σύννεφο ή νερό) όταν ο NBR είναι ίσο με το NBR_Threshold.

Τα παραπάνω βήματα σε μορφή ψευδοκώδικα:

```
If (NBR [i] [j] <= T1 AND NIR [i] [j] <= T2 AND ALBEDO [i] [j] <= T3 AND NDVI_MULT > T4)
```

```
    Result_pixel [i] [j] = "burned"
```

```
Else if (NBR [i] [j] > (T1+1) AND NIR [i] [j] > T2 AND ALBEDO [i] [j] > T3 AND NDVI_MULT <= T4)
```

```
    Result_pixel [i] [j] = "unburned"
```

Else

Result_pixel [i] [j] = “unknown”

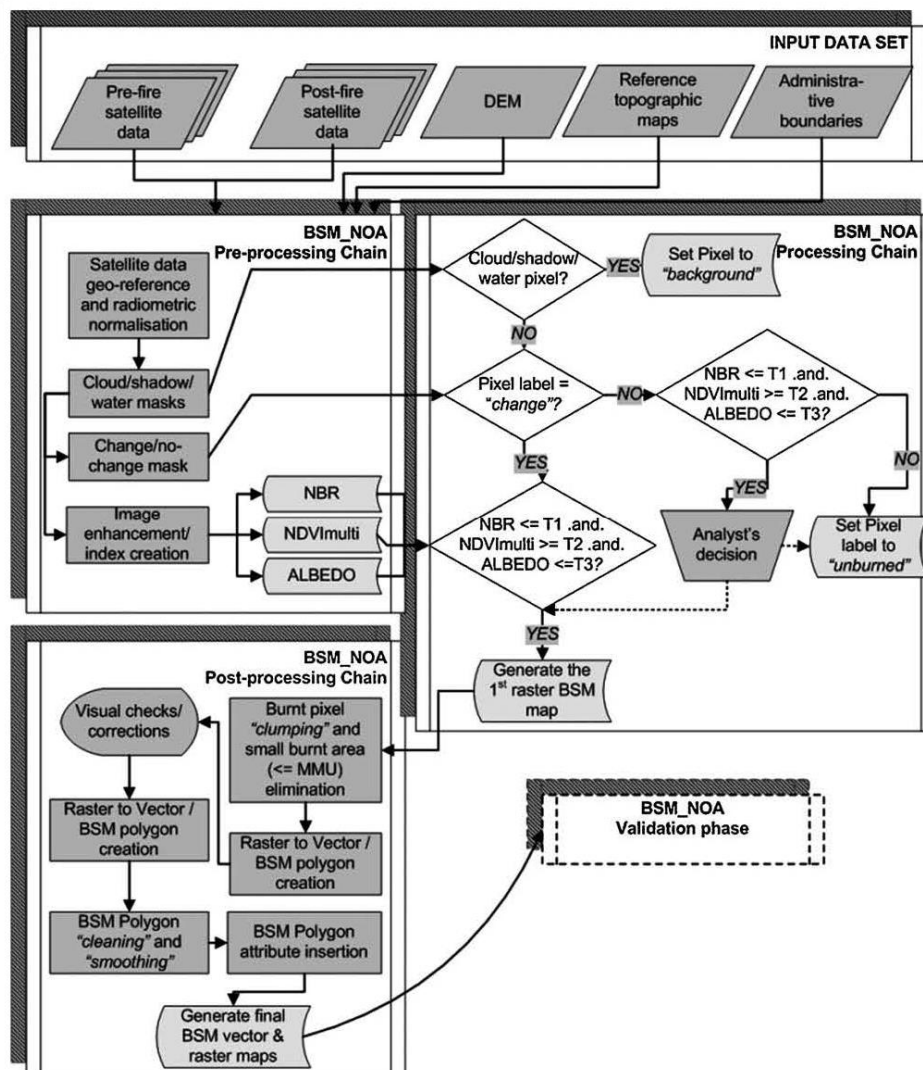
Στην πράξη, το μη καθορισμένο (unknown) pixel ορίζεται ως μη καμένο, και στην έξοδο έχουμε δύο κατηγορίες, καμένο και μη καμένο.

Στη περίπτωση εικόνων με χρονική ακολουθία, μετά την ταξινόμηση των pixel χρησιμοποιώντας τον παραπάνω αλγόριθμο, σε κάθε pixel γίνεται σύγκριση με την τιμή της παλιότερης εικόνας και με τα οκτώ γειτονικά του. Αν η τιμή του pixel και των γειτονικών του είναι μεγαλύτερη από ένα κατώφλι, τότε το pixel ορίζεται ως αλλαγμένο ή είναι στην κρίση του αναλυτή να αποφασίσει αν είναι καμένο ή όχι.

3.2.3 Ποιοτικός έλεγχος και ερμηνεία εικόνων

Η φάση ελέγχου της ποιότητας των εικόνων και της ερμηνείας τους αποτελείται από τα ακόλουθα στάδια:

- Εφαρμογή ενός διάμεσου φίλτρου στην εικόνα που παράγεται για τη μείωση του θορύβου των pixel.
- Καμένα pixel που είναι σε μικρές γειτονιές, εξαλείφονται αν οι γειτονιές είναι μικρότερες από μία προκαθορισμένη ελάχιστη μονάδα χαρτογράφησης - minimum mapping unit (MMU).
- Μετασχηματισμός της raster εικόνας σε πίνακα προκειμένου να δημιουργηθούν τα όρια της καμένης έκτασης.
- Οπτικός έλεγχος των καμένων πολυγώνων που προκύπτουν και σύγκριση με τα αρχικά δεδομένα της εικόνας. Αυτό το βήμα καθαρίζει το χάρτη από περιττά μικρά πολύγωνα και βελτιώνει τη θεματική αξία του.
- Εξομάλυνση πολυγώνων και ενσωμάτωση σημαντικών πληροφοριών για κάθε καμένο πολύγωνο, όπως η επιφάνεια του πολυγώνου (καμένης έκτασης), εκτίμηση των ζημιών από την άποψη της έκτασης, την ημερομηνία έναρξης των πυρκαγιών, ημερομηνία καταστολής πυρκαγιάς, κ.λπ.



Εικόνα 3.1 Διάγραμμα ροής του αλγόριθμου BSM_NOA

Κεφάλαιο 4ο – Υλοποίηση

4.1 Εισαγωγή

Στο παρόν κεφάλαιο περιγράφεται η υλοποίηση του προγράμματος που πραγματοποιήθηκε στο πλαίσιο της παρούσας πτυχιακής ώστε να επιτευχθεί η έξοδος αντίστοιχων σε αριθμό εικόνων με της εισόδου που θα παρουσιάζουν τις καμένες έκτασης των περιοχών που απεικονίζουν. Όπως αναφέρεται και νωρίτερα, ο αλγόριθμος BSM_NOA αποτελείται από βήματα, τα οποία για να υλοποιηθούν ευκολότερα χωρίζονται σε αντίστοιχα υποπρόγραμμα γραμμένα σε Java. Επίσης για τις ανάγκες του Hadoop framework οι εικόνες εισόδου έπρεπε να μετατραπούν σε διαφορετικό τύπου αρχείου και πιο συγκεκριμένα σε txt. Τα υποπρογράμματα είναι τα παρακάτω:

- Φάση ταξινόμησης: Υπολογισμός των δεικτών, που αναφέρονται στην ενότητα 3.2.2, και έπειτα ταξινομεί τα pixel σε καμένα και μη.
- Φάση διάμεσου φίλτρου: Εφαρμογή 3x3 διάμεσου φίλτρου για την μείωση θορύβου, αυτό το βήμα αντιστοιχεί στο κομμάτι του ελέγχου ποιότητας του BSM.
- Φάση ομαδοποίησης και διαγραφής: Εφαρμογή του αλγορίθμου connect_components και διαγραφής των καμένων ομάδων που ο αριθμός των pixel τους είναι μικρότερος από ένα κατώφλι. Επίσης αυτό το βήμα αντιστοιχεί στο κομμάτι του ελέγχου ποιότητας του BSM.

Τέλος τα δεδομένα που εξάγονται, με την βοήθεια της βιβλιοθήκης της GDAL, μετατρέπονται εκ νέου σε εικόνα με τύπο αρχείου tif, για να μπορούν να μελετηθούν από τους αναλυτές και από τα προγράμματα που χρησιμοποιούν για παράδειγμα το GIS.

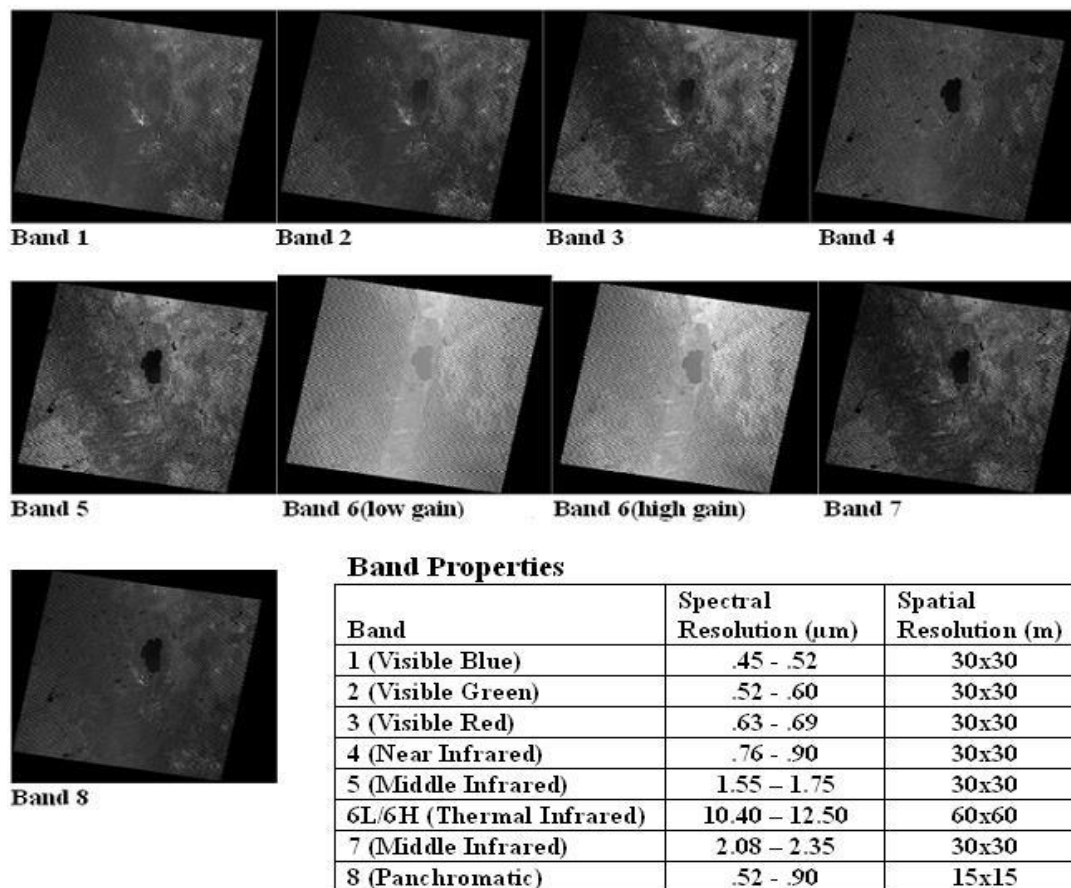
4.2 Είσοδος και μετατροπή δεδομένων

4.2.1 Είσοδος δεδομένων

Για την υλοποίηση του προγράμματος έγινε χρήση εικόνων που τραβήχτηκαν από τον Landsat TM στη Ελλάδα, προσφορά του National Observatory of Athens

(NOA). Οι εικόνες έχουν μέγεθος μεγαλύτερο από 7000x7000 pixels, και έχουν γίνει τα βήματα που αναφέρονται στην ενότητα 3.2.1, δηλαδή γεωμετρικές και ραδιομετρικές διορθώσεις, γεωγραφικές αναφορές κτλ. Επίσης πρέπει να αναφερθεί πως δεν γίνεται εφαρμογή της μάσκας για σύννεφα-νερό-σκιές, καθώς της περισσότερες φορές το βήμα αυτό το παραλείπουν και οι ερευνητές του NOA. Η κάθε εικόνα περιέχει επτά φασματικές μπάντες, ενώ ο BSM χρειάζεται για τους υπολογισμούς του μόνο τις τρεις, την 3,4 και 7 που απεικονίζουν τις ακτινοβολίες ορατού κόκκινου, εγγύς υπέρυθρη και μεσαία υπέρυθρη. Η GDAL προσφέρει εργαλεία για να διαβαστεί κάθε μπάντα άμεσα από το αρχείο και να αποθηκεύει τις τιμές της σε έναν αριθμητικό μονοδιάστατο πίνακα. Η τιμές κυμαίνονται στο διάστημα [0,255] οπότε ο πίνακας περιέχει ακέραιες τιμές. Τέλος, για να γίνει μια εκτίμηση του όγκου των δεδομένων που χρειάζονται, γίνονται οι παρακάτω υπολογισμοί:

3 φασματικές μπάντες * 7000 γραμμές * 7000 στήλες (pixels) χρειάζονται 147.000.000 θέσεις για την αποθήκευση ακέραιων τιμών.



Εικόνα 4.1 Ιδιότητες μπάντας (Lake Tahoe)

4.2.2 Μετατροπή εικόνας σε μορφή κειμένου

Επειδή η τύπος αρχείου της εικόνας είναι μη επεξεργάσιμος από το Hadoop, η εικόνα πρέπει να μετατραπεί σε αρχείο κειμένου. Η βιβλιοθήκη της GDAL περιέχει τις μεθόδους που χρειάζονται για την μετατροπή αυτή. Πιο συγκεκριμένα η εν λόγω βιβλιοθήκη, μετατρέπει τις πληροφορίες της εικόνας (μπάντα) σε ένα μονοδιάστατο πίνακα ακεραίων, έπειτα αυτός ο πίνακας αποτυπώνεται σε ένα αρχείο σε μορφή κειμένου. Όλα αυτά υλοποιούνται στο υποπρόγραμμα ImageToTxt και η μορφή που έχει το παραγόμενο αρχείο είναι:

```
filename,band,xSize,ySize,line,bytes[i],bytes[i+1],...,bytes[7000]
```

```
filename,band,xSize,ySize,line,bytes[7000+i],bytes[7000+i+1],...,bytes[2*7000]
```

```
...
```

```
filename,band,xSize,ySize,line,bytes[6999*7000+i], ...,bytes[7000*7000]
```

Όπου filename είναι το όνομα της εικόνας, band ο αριθμός της μπάντας (3, 4, 7), xSize το μέγεθος της εικόνας ως προς τον άξονα του X, ySize το μέγεθος της εικόνας ως προς τον άξονα του Y, line σε ποία γραμμή είναι τα pixel ως προς το άξονα του X και bytes[] η τιμή του pixel για την αντίστοιχη μπάντα. Επίσης παράγεται και ένα αρχείο στον φάκελο ImageInfo με της γεωγραφικές πληροφορίες - metadata της εικόνας (γεωγραφικές συντεταγμένες κτλ).

4.3 Φάση ταξινόμησης

4.3.1 Mapper

Κατά την διάρκεια της ταξινόμησης, η οποία υλοποιείται στο δεύτερο υποπρόγραμμα, εισάγονται αρχεία με την μορφή που αναφέρεται στην ενότητα 4.2.2. Η μέθοδος TextInputFormat του Hadoop διαβάζει τα αρχεία και διαμοιράζει μια γραμμή από κάθε αρχείο στους mappers [22]. Ο mapper αναλαμβάνει να διαχωρίσει τις πληροφορίες γραμμής που δέχεται ως είσοδο, χρησιμοποιώντας την μέθοδο StringTokenizer, η οποία χωρίζει τα pixel και ο mapper τα εξάγει με κλειδί (key) τις συντεταγμένες του pixel στην εικόνα και το όνομα της εικόνας, και με τιμή (value)

τον αριθμό της μπάντας, το μέγεθος της εικόνας ως προς τον άξονα του X, ySize το μέγεθος της εικόνας ως προς τον άξονα του Y καθώς και την τιμή του pixel.

4.3.2 Reducer

Ο reducer λαμβάνει σαν είσοδο τον συνδυασμό κλειδί-τιμή, ταξινομημένα ως προς την εικόνα και τις συντεταγμένες των pixel. Αναλυτικότερα, ο reducer θα λάβει ως είσοδο τρεις συνδυασμούς (τιμές pixel), έναν για κάθε μπάντα, που αναφέρονται στις ίδιες συντεταγμένες της εικόνας. Έπειτα ορίζονται τα κατώφλια T1, T2, T3 που αναφέρονται στο ενότητα 3.2.2 και υπολογίζονται οι δείκτες NBR, NIR και ALBEDO. Στη συνέχεια γίνεται σύγκριση των δεικτών με τα κατώφλια και στην έξοδο του reducer, που είναι ένα αρχείο txt με τίτλο το όνομα της φωτογραφίας, ορίζεται το pixel αν είναι καμένο ή όχι, γράφοντας στο αρχείο ανά γραμμή, ως κλειδί: τις συντεταγμένες του pixel, το όνομα του αρχείου, και την ανάλυση της φωτογραφίας, όπου θα χρειαστεί για υπολογισμούς παρακάτω, και για τιμή τον αριθμό 1. Με αυτό τον τρόπο μειώνεται ο όγκος των δεδομένων, ενώ στην αρχή υπήρχαν τρεις μπάντες, τώρα στην έξοδο του reducer υπάρχει ένα αρχείο, που περιέχει μόνο τις συντεταγμένες των καμένων pixels.

4.4 Φάση διάμεσου φίλτρου

4.4.1 Median filter

Το διάμεσο φίλτρο είναι μια μη γραμμική ψηφιακή τεχνική φιλτραρίσματος, που χρησιμοποιείται συχνά για την αφαίρεση θορύβου, ιδίως σε ψηφιακή επεξεργασία εικόνας [23]. Η κύρια ιδέα του αλγόριθμου αυτού είναι να σαρώνει το κάθε pixel, αντικαθιστώντας το με τη μέση τιμή των γειτονικών pixel. Στην πράξη, στον αλγόριθμο BSM_NOA, χρησιμοποιείται μια παραλλαγή του κλασικού μεσαίου φίλτρου, το λεγόμενο φίλτρο <<πλειοψηφία>> ή το φίλτρο <<k-πλησιέστερων>>. Σε αυτή την περίπτωση, ελέγχουμε πόσα γειτονικά pixel ανήκουν στην κατηγορία καμένα και πόσα στην κατηγορία μη και αλλάζουμε το pixel σε καμένο αν οι περισσότεροι γείτονες καμένα pixel.

4.4.2 Περιγραφή και λειτουργικότητα MapReducer

Στην φάση του διάμεσου φίλτρου, το οποίο υλοποιείται στο τρίτο υποπρόγραμμα, η είσοδος των mappers είναι η έξοδος της φάσης ταξινόμησης, δηλαδή το αρχείο που περιέχει τις συντεταγμένες με τα καμένα pixel. Το φίλτρο εφαρμόζεται σε γειτονικά pixel 3x3. Πιο συγκεκριμένα, για κάθε καμένο pixel ο mapper παράγει οχτώ γειτονικά με κλειδί τις συντεταγμένες του κάθε pixel και τιμή 1. Στο ίδιο το καμένο pixel δίνεται η τιμή 5. Για παράδειγμα, στο σύνολο των pixel που θα εξάγουν οι mappers, θα υπάρχει ένα ζεύγος <κλειδί, τιμή> με συντεταγμένες X,Y το οποίο αν είναι καμένο θα έχει τιμή 5, επίσης θα υπάρχουν άλλα τόσα ζεύγη με τις ίδιες συντεταγμένες για κάθε γειτονικό pixel του X,Y που είναι καμένο. Η διαδικασία όμως απαιτεί ελέγχους, καθώς αν το pixel βρίσκεται στις γωνίες ή στην πρώτη ή τελευταία γραμμή και στήλη δεν πρέπει ο mapper να εξάγει όλα τα γειτονικά pixel γιατί θα είναι με αρνητικές συντεταγμένες. Στη περίπτωση που το pixel είναι σε περιοχή στη μέση της εικόνας τα γειτονικά pixel έχουν συντεταγμένες που απεικονίζονται στην εικόνα 4.2. Στη συνέχεια ο reducer λαμβάνει σαν είσοδο όλα τα ζεύγη με τις ίδιες συντεταγμένες και προσθέτει τις τιμές τους, αν το σύνολο είναι μεγαλύτερο από 4, δηλαδή της πλειοψηφίας των γειτονικών του pixel, τότε το pixel με συντεταγμένες X, Y θεωρείται καμένο. Ως αποτέλεσμα, τα δεδομένα που θα παράγουν οι reducers και θα αποθηκευτούν σε ένα αρχείο κειμένου, θα είναι περισσότερα από αυτά που έλαβαν οι mappers, καθώς αν ένα pixel δεν θεωρείται καμένο, αλλά η πλειοψηφία των γειτονικών του pixel είναι καμένα, τότε και αυτό ορίζεται ως καμένο.

$x-1$ $y-1$ 1	$x-1$ y 1	$x-1$ 1 $y+1$
x 1 $y-1$	x 5	x 1 $y+1$
$x+1$ $y-1$ 1	$x+1$ y 1	$x+1$ $y+1$ 1

Εικόνα 4.2 Έξοδος mapper με 9 γειτονικά pixel

4.5 Φάση ομαδοποίησης και διαγραφής

4.5.1 Αλγόριθμος σύνδεσης και επισήμανσης στοιχείων

Ο αλγόριθμος connect components labeling είναι μια εφαρμογή της θεωρίας γραφημάτων, όπου υποσύνολα των συνδεδεμένων στοιχείων είναι μοναδικά σημειωμένα. Ο αλγόριθμος σαρώνει δύο φορές τον πίνακα και εκχωρεί προσωρινές ετικέτες (labels) και καταγράφει τις γειτονιές των καμένων pixel [24]. Στη συνέχεια, κατά το δεύτερο πέρασμα αντικαθιστά κάθε προσωρινή ετικέτα με τη μικρότερη ετικέτα της γειτονιάς στην οποία ανήκει.

Οι έλεγχοι διασύνδεσης γειτονιών διενεργείται με τον έλεγχο ετικετών των γειτονικών pixels, (όποιου γειτονικού pixel δεν του έχουν εκχωρηθεί ακόμα ετικέτες αγνοούνται), ή πιο συγκεκριμένα ελέγχεται το Βορειοανατολικό, το Βόρειο, το Βορειοδυτικό και το Δυτικό του τρέχοντος pixel (χρησιμοποιώντας την 8-connectivity έκδοση του αλγορίθμου). Στην 4-connectivity έκδοση του αλγορίθμου χρησιμοποιείται μόνο το Βόρειο και το Δυτικό γειτονικό του τρέχοντος pixel.

Συνθήκες που πρέπει να ελεγχθούν:

Μήπως το pixel προς τα αριστερά (Δυτικά) έχουν την ίδια τιμή με το τρέχον pixel;

Ναι – Είναι στην ίδια περιοχή. Ανάθεση ίδιας ετικέτας για το τρέχον pixel.

Όχι – Ελέγχει την επόμενη κατάσταση.

Μήπως τα δύο pixels στο Βορρά και στη Δύση του τρέχοντος pixel έχουν την ίδια τιμή με το τρέχον pixel, αλλά όχι την ίδια ετικέτα;

Ναι – Είναι γνωστό ότι τα Βόρεια και τα Δυτικά pixels ανήκουν στην ίδια περιοχή και πρέπει να συγχωνευθούν. Αναθέτει στο τρέχον pixel την ελάχιστη ετικέτα μεταξύ του Βόρειου και του Δυτικού pixel, και καταγράφει τη γειτονιά.

Όχι - Ελέγχει την επόμενη κατάσταση.

Μήπως το pixel προς τα αριστερά (Δυτικά) έχει διαφορετική τιμή και το pixel προς το Βορρά την ίδια τιμή με το τρέχον pixel;

Ναι – Αναθέτει την ετικέτα του Βόρειου pixel στο τρέχον pixel.

Όχι – Ελέγχει την επόμενη κατάσταση.

Μήπως το Βόρειο και το Δυτικό, γειτονικά pixel, έχουν διαφορετικές τιμές από το τρέχον pixel;

Ναι – Δημιουργεί ένα νέο id ετικέτα και αντιστοιχεί την τιμή με το τρέχον pixel.

Ο αλγόριθμος συνεχίζει με αυτό το τρόπο, και δημιουργεί νέες ετικέτες ανά περιοχή όποτε είναι αναγκαίο. Το κλειδί για να είναι γρήγορος ο ένας αλγόριθμος, είναι ο τρόπος που πραγματοποιείται η συγχώνευση. Ο αλγόριθμος αυτός χρησιμοποιεί union- find στη δομή δεδομένων, η οποία παρέχει εξαιρετική απόδοση για την παρακολούθηση των σχέσεων γειτονίας. Το union-find αποθηκεύει ουσιαστικά ετικέτες που αντιστοιχούν στο ίδιο block σε δομή δεδομένων disjoint-set, καθιστώντας το εύκολο να θυμούνται την γειτονιά δύο ετικετών με τη χρήση μιας μεθόδου διεπαφής (Π.χ.: findSet(I)). Η μέθοδος findSet(I) επιστρέφει την ελάχιστη τιμή ετικέτας που είναι ισοδύναμη με την παράμετρο «I».

Μόλις ολοκληρωθεί η αρχική καταγραφή και η επισήμανση των στοιχείων, στο δεύτερο πέρασμα γίνεται αντικατάσταση κάθε ετικέτας pixel με την μικρότερη τιμή ετικέτας της γειτονίας που ανήκει το pixel.

Μια ταχύτερη σάρωση αλγόριθμο για την εξαγωγή συνδέεται περιοχής παρουσιάζεται παρακάτω.

Στο πρώτο πέρασμα:

- 1) Γίνεται σάρωση κάθε στοιχείου του πίνακα ανά στήλη, και έπειτα ανά σειρά (σάρωση Raster).
- 2) Εάν το στοιχείο δεν ανήκει στο background.
 - 1) Παίρνει τα γειτονικά στοιχεία του τρέχοντος στοιχείου.
 - 2) Εάν δεν υπάρχουν γείτονες, επισημαίνει το τρέχον στοιχείο μοναδικό τίτλο και να συνεχίζει.
 - 3) Διαφορετικά, βρίσκει το γείτονα με τη μικρότερη ετικέτα και να την αντιστοιχίζει με το τρέχον στοιχείο.
 - 4) Αποθηκεύει την ισοδυναμία μεταξύ γειτονικών ετικετών.

Στο δεύτερο πέρασμα:

- 1) Γίνεται σάρωση κάθε στοιχείου του πίνακα ανά στήλη, και έπειτα ανά σειρά.
- 2) Εάν το στοιχείο δεν ανήκει στο background.
 - 1) Βρίσκει το γείτονα με τη μικρότερη ετικέτα και να την αντιστοιχίζει με το τρέχον στοιχείο.

4.5.2 Λειτουργικότητα MapReducer

Ο mapper στο τρίτο υποπρόγραμμα ταξινομεί τα καμένα pixel που δέχεται σαν είσοδο από το αρχείο που έχει εξάγει το δεύτερο υποπρόγραμμα που υλοποιεί το διάμεσο φίλτρο. Η ταξινόμηση γίνεται ορίζοντας στην έξοδο του mapper ως κλειδί το όνομα της εικόνας, και ως τιμή τις συντεταγμένες του καμένου pixel. Έτσι ο reducer δέχεται σαν είσοδο όλα τα καμένα pixel μιας εικόνας. Στην συνέχεια δημιουργεί ένα δυσδιάστατο πίνακα ίδιων διαστάσεων με της εικόνας και σημειώνει τα καμένα pixel στις αντίστοιχες θέσεις του πίνακα, με αυτές της εικόνας, βάζοντας στον πίνακα την τιμή 1. Τα υπόλοιπα pixel που δεν είναι καμένα παίρνουν την τιμή 0 και στην συνέχεια εφαρμόζεται στον πίνακα ο αλγόριθμος connect components labeling για να

οριστούν οι καμένες γειτονίες και το πλήθος των pixel που έχει η κάθε μια, ώστε να διαγραφούν οι γειτονίες που έχουν πλήθος καμένων μικρότερο από ένα κατώφλι (στην παρούσα εργασία το κατώφλι έχει τιμή 10). Μια διαφορά που έχει η υλοποίηση του τρίτου υποπρογράμματος από τα άλλα δύο, είναι ότι χρησιμοποιεί την κλάση MultiFileOutput [27] και παρέχει στην έξοδο διαφορετικό αρχείο ανά εικόνα, ενώ τα δύο παραπάνω υποπρογράμματα κατέγραφαν όλα τα δεδομένα εξόδου σε ένα αρχείο. Η υλοποίηση αυτή είναι πολύ χρήσιμη καθώς τα αρχεία που παράγονται, περιέχουν σε μορφή κειμένου τις τιμές όλων των pixel της εικόνας. Έτσι πολύ εύκολα με τα εργαλεία που παρέχει η GDAL, τα αρχεία κειμένου θα μετατραπούν σε εικόνες με τύπο αρχείου .tif, για να μπορούν να διαβαστούν από τα προγράμματα ανάλυσης δορυφορικών εικόνων (για παράδειγμα, GIS) και γίνει περεταίρω ανάλυση από τους ερευνητές.

4.6 Μετατροπή αρχείων κειμένου σε εικόνα

Στο τελευταίο υποπρόγραμμα που υλοποιείται ο χρήστης εισάγει την διεύθυνση του φακέλου που είναι αποθηκευμένες οι εικόνες σε μορφή κειμένου, και την διεύθυνση που θα αποθηκευτούν οι παραγόμενες εικόνες σε μορφή tif. Στη συνέχεια τρέχει μία επανάληψη ανάλογα με τον αριθμό των αρχείων-εικόνων που υπάρχουν, και για κάθε εικόνα διαβάζει επίσης τα metadata που έχουν αποθηκευτεί στο φάκελο ImageInfo για την αντίστοιχη εικόνα. Έπειτα διαβάζει τα pixel από το αρχείο και τα αποθηκεύει σε έναν δυσδιάστατο πίνακα και τα metadata σε μια μεταβλητή τύπου κειμένου. Η μέθοδος WriteRaster της GDAL, παίρνει ως παράμετρο τις διαστάσεις της εικόνας και τον παραπάνω πίνακα και εξάγει την εικόνα σε μορφή tif μαζί με τα metadata της.

Κεφάλαιο 5^ο – Λειτουργία και αποτελέσματα

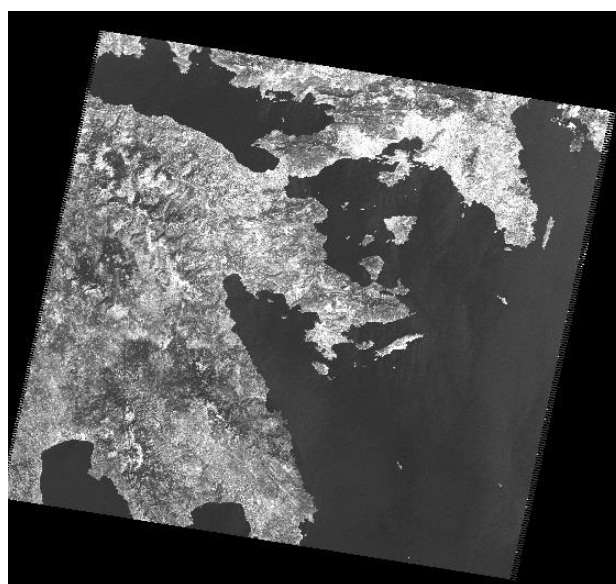
5.1 Εισαγωγή

Σε αυτό το κεφάλαιο αναφέρονται παραδείγματα εκτέλεσης του προγράμματος, σε έναν συμβατό υπολογιστή που έτρεχε σε ψευδο-κατανεμημένη λειτουργία και σε ένα cluster που παρέχει η πλατφόρμα azure της Microsoft. Στη συνέχεια αναλύονται οι χρόνοι εκτέλεσης του προγράμματος στις παραπάνω δυο περιπτώσεις και τα αποτελέσματα που παράγονται.

5.2 Δοκιμή λειτουργίας προγράμματος

Για την ανάπτυξη του προγράμματος, του οποίου η λειτουργικότητα αναλύθηκε στο προηγούμενο κεφάλαιο, κρίθηκε αναγκαία η χρήση ενός κατανεμημένου συστήματος υπολογιστικών μονάδων, μιας και το πρόγραμμα επεξεργάζεται παράλληλα μεγάλο όγκο δεδομένων. Για την υλοποίηση του λοιπόν, και για μεγαλύτερη διευκόλυνση στην αποσφαλμάτωση, χρησιμοποιήθηκε το Hadoop σε ψευδο-κατανεμημένη λειτουργία (Pseudo-Distributed Operation). Το Hadoop μπορεί επίσης να τρέξει σε τοπική λειτουργία (Standalone Operation), και σε πλήρη κατανεμημένη λειτουργία (Fully-Distributed Operation) [].

Στη συνέχεια παρουσιάζονται στιγμιότυπα εκτέλεσης σε προσωπικό υπολογιστή με μνήμη ram 4GB και cpu dual-core 2.4Ghz.



Εικόνα 5.1 Εικόνα εισόδου στο πρόγραμμα

[illegible]

Εικόνα 5.2 Μορφή αρχείου κειμένου που εισάγεται στο Hadoop

Αφού ολοκληρωθεί η μετατροπή της εικόνας σε αρχείο κειμένου (εικόνα 5.2), το πρόγραμμα ξεκινάει να εκτελεί με σειρά της τρεις φάσεις. Ο jobtracker δείχνει την κατάσταση της εκτέλεσης όπως εμφανίζεται στις παρακάτω εικόνες.

Running Jobs

none

Completed Jobs

Jobid	Started	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total	Reduces Completed	Job Scheduling Information	Diagnostic Info
job_201409211737_0002	Sun Sep 21 18:31:01 EEST 2014	NORMAL	emanon	MainClass	<div>100.00%</div>	8	8	<div>100.00%</div>	1	1	NA	NA

Retired Jobs

none

Εικόνα 5.3 Jobtracker

Hadoop job_201409211737_0002 on localhost

```
User: emanon
Job Name: MainClass
Job File: hdfs://localhost:9000/tmp/hadoop-emanon/mapred/staging/emanon/.staging/job\_201409211737\_0002/job.xml
Submit Host: emanon-K55VJ
Submit Host Address: 127.0.1.1
Job-ACLs: All users are allowed
Job Setup: Successful
Status: Succeeded
Started at: Sun Sep 21 18:31:01 EEST 2014
Finished at: Sun Sep 21 19:02:42 EEST 2014
Finished in: 31mins, 40sec
Job Cleanup: Successful
```

Kind	% Complete	Num Tasks	Pending	Running	Complete	Killed	Failed/Killed Task Attempts
map	100.00% <div><div></div></div>	8	0	0	8	0	0 / 0
reduce	100.00% <div><div></div></div>	1	0	0	1	0	0 / 0

Εικόνα 5.4 Λεπτομέρειες εκτέλεσης MainClass-ClassificationPhase (1)

	Counter	Map	Reduce	Total
File Input Format Counters	Bytes Read	445,372,509	0	445,372,509
Job Counters	SLOTS_MILLIS_MAPS	0	0	2,887,546
	Launched reduce tasks	0	0	1
	Total time spent by all reduces waiting after reserving slots (ms)	0	0	0
	Total time spent by all maps waiting after reserving slots (ms)	0	0	0
	Launched map tasks	0	0	8
	Data-local map tasks	0	0	8
	SLOTS_MILLIS_REDUCE	0	0	1,506,642
File Output Format Counters	Bytes Written	0	181,372,250	181,372,250
FileSystemCounters	FILE_BYTES_READ	15,957,942,824	7,916,061,646	23,874,004,470
	HDFS_BYTES_READ	445,373,525	0	445,373,525
	FILE_BYTES_WRITTEN	23,874,418,686	7,916,113,310	31,790,531,996
	HDFS_BYTES_WRITTEN	0	181,372,250	181,372,250
Map-Reduce Framework	Map output materialized bytes	7,916,061,646	0	7,916,061,646
	Map input records	23,721	0	23,721
	Reduce shuffle bytes	0	7,916,061,646	7,916,061,646
	Spilled Records	527,709,214	174,942,375	702,651,589
	Map output bytes	7,566,176,848	0	7,566,176,848
	Total committed heap usage (bytes)	1,909,391,360	985,858,048	2,895,249,408
	CPU time spent (ms)	708,170	264,600	972,770
	Map input bytes	445,302,877	0	445,302,877
	SPLIT_RAW_BYTES	1,016	0	1,016
	Combine input records	0	0	0
	Reduce input records	0	174,942,375	174,942,375
	Reduce input groups	0	116,628,250	116,628,250
	Combine output records	0	0	0
	Physical memory (bytes) snapshot	4,135,890,944	1,028,292,608	5,164,183,552
	Reduce output records	0	4,555,984	4,555,984
	Virtual memory (bytes) snapshot	42,866,692,096	5,364,699,136	48,231,391,232
	Map output records	174,942,375	0	174,942,375

Εικόνα 5.5 Λεπτομέρειες εκτέλεσης MainClass-ClassificationPhase (2)

File: [/main/output/part-00000](#)

```

1000,5059,LT51830342011259,7907,7375, 1
1000,5095,LT51830342011259,7907,7375, 1
1000,5103,LT51830342011259,7907,7375, 1
1000,5109,LT51830342011259,7907,7375, 1
1000,5110,LT51830342011259,7907,7375, 1
1000,5112,LT51830342011259,7907,7375, 1
1000,5114,LT51830342011259,7907,7375, 1
1000,5116,LT51830342011259,7907,7375, 1
1000,5118,LT51830342011259,7907,7375, 1
1000,5121,LT51830342011259,7907,7375, 1
1000,5123,LT51830342011259,7907,7375, 1
1000,5125,LT51830342011259,7907,7375, 1
1000,5127,LT51830342011259,7907,7375, 1
1000,5129,LT51830342011259,7907,7375, 1

```

Εικόνα 5.6 Μορφή αρχείου που εξάγεται από την MainClass

Hadoop job_201409211737_0003 on localhost

User: emanon
Job Name: MedianFilterPhase
Job File: hdfs://localhost:9000/tmp/hadoop-emanon/mapred/staging/emanon/staging/job_201409211737_0003/job.xml
Submit Host: emanon-K55VJ
Submit Host Address: 127.0.1.1
Job-ACLs: All users are allowed
Job Setup: [Successful](#)
Status: Succeeded
Started at: Sun Sep 21 19:09:39 EEST 2014
Finished at: Sun Sep 21 19:14:12 EEST 2014
Finished in: 4mins, 33sec
Job Cleanup: [Successful](#)

Kind	% Complete	Num Tasks	Pending	Running	Complete	Killed	Failed/Killed Task Attempts
map	<div><div></div></div> 100.00%	3	0	0	3	0	0 / 0
reduce	<div><div></div></div> 100.00%	1	0	0	1	0	0 / 0

	Counter	Map	Reduce	Total
File Input Format Counters	Bytes Read	181,380,442	0	181,380,442
Job Counters	SLOTS_MILLIS_MAPS	0	0	379,242
	Launched reduce tasks	0	0	1
	Total time spent by all reduces waiting after reserving slots (ms)	0	0	0
	Total time spent by all maps waiting after reserving slots (ms)	0	0	0
	Launched map tasks	0	0	3
	Data-local map tasks	0	0	3
	SLOTS_MILLIS_REDUCE	0	0	125,154
File Output Format Counters	Bytes Written	0	187,857,773	187,857,773
FileSystemCounters	FILE_BYTES_READ	3,353,305,649	1,755,290,163	5,108,595,812
	HDFS_BYTES_READ	181,380,733	0	181,380,733
	FILE_BYTES_WRITTEN	5,108,751,221	1,755,341,853	6,864,093,074
	HDFS_BYTES_WRITTEN	0	187,857,773	187,857,773
Map-Reduce Framework	Map output materialized bytes	1,755,290,163	0	1,755,290,163
	Map input records	4,555,984	0	4,555,984
	Reduce shuffle bytes	0	1,755,290,163	1,755,290,163
	Spilled Records	119,336,219	41,002,074	160,338,293
	Map output bytes	1,673,285,997	0	1,673,285,997
	Total committed heap usage (bytes)	653,656,064	81,068,032	734,724,096
	CPU time spent (ms)	198,720	40,740	239,460
	Map input bytes	181,372,250	0	181,372,250
	SPLIT_RAW_BYTES	291	0	291
	Combine input records	0	0	0
	Reduce input records	0	41,002,074	41,002,074
	Reduce input groups	0	22,180,012	22,180,012
	Combine output records	0	0	0
	Physical memory (bytes) snapshot	1,241,296,896	113,733,632	1,355,030,528
	Reduce output records	0	4,718,026	4,718,026

Εικόνα 5.7 Λεπτομέρειες εκτέλεσης MedianFilterPhase

File: [/median/output/part-00000](#)

```
LT51830342011259:1000:5059:7907:7375: 6
LT51830342011259:1000:5095:7907:7375: 6
LT51830342011259:1000:5103:7907:7375: 6
LT51830342011259:1000:5109:7907:7375: 7
LT51830342011259:1000:5110:7907:7375: 8
LT51830342011259:1000:5112:7907:7375: 6
LT51830342011259:1000:5114:7907:7375: 6
LT51830342011259:1000:5116:7907:7375: 6
LT51830342011259:1000:5118:7907:7375: 6
LT51830342011259:1000:5121:7907:7375: 6
LT51830342011259:1000:5123:7907:7375: 6
LT51830342011259:1000:5125:7907:7375: 6
LT51830342011259:1000:5127:7907:7375: 6
LT51830342011259:1000:5129:7907:7375: 8
```

Εικόνα 5.8 Μορφή αρχείου που εξάγεται από το MediaFilter

Hadoop job_201409211737_0006 on localhost

User: emanon

Job Name: PixelSortByImage

Job File: hdfs://localhost:9000/tmp/hadoop-emanon/mapred/staging/emanon/.staging/job_201409211737_0006/job.xml

Submit Host: emanon-K55VJ

Submit Host Address: 127.0.1.1

Job-ACLs: All users are allowed

Job Setup: [Successful](#)



Status: Succeeded

Started at: Sun Sep 21 19:32:12 EEST 2014

Finished at: Sun Sep 21 19:32:55 EEST 2014

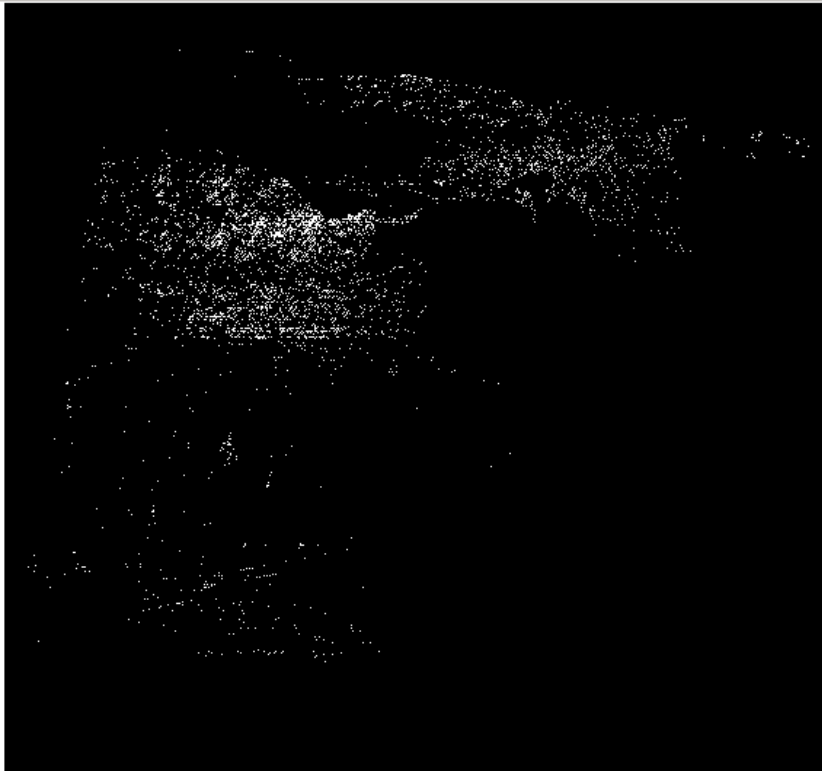
Finished in: 42sec

Job Cleanup: [Successful](#)

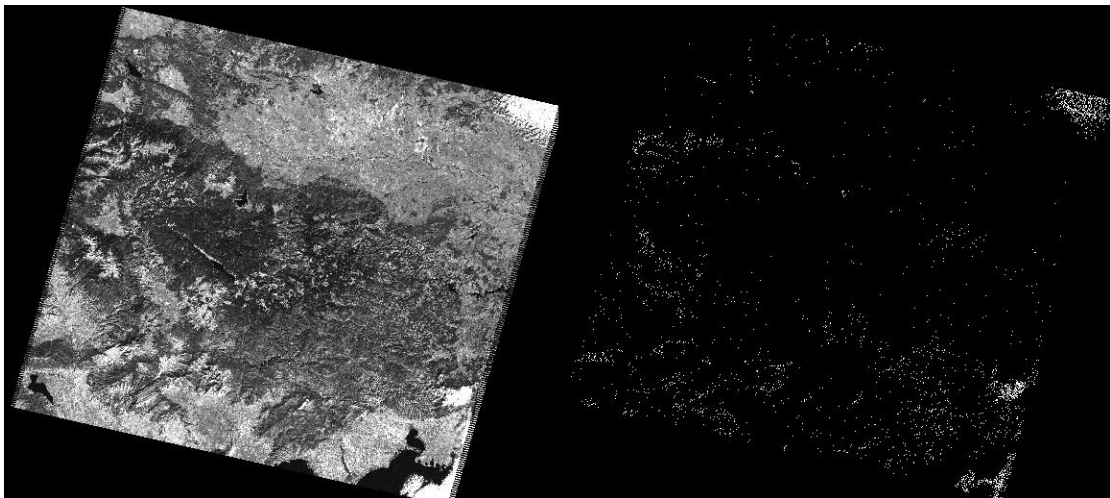
Kind	% Complete	Num Tasks	Pending	Running	Complete	Killed	Failed/Killed Task Attempts
map	100.00% 	3	0	0	3	0	0 / 0
reduce	100.00% 	1	0	0	1	0	0 / 0

	Counter	Map	Reduce	Total
File Input Format Counters	Bytes Read	187,865,965	0	187,865,965
Job Counters	SLOTS_MILLIS_MAPS	0	0	35,978
	Launched reduce tasks	0	0	1
	Total time spent by all reduces waiting after reserving slots (ms)	0	0	0
	Total time spent by all maps waiting after reserving slots (ms)	0	0	0
	Launched map tasks	0	0	3
	Data-local map tasks	0	0	3
	SLOTS_MILLIS_REDUCEs	0	0	27,173
File Output Format Counters	Bytes Written	0	21,618,725	21,618,725
FileSystemCounters	FILE_BYTES_READ	183,109,347	183,109,233	366,218,580
	HDFS_BYTES_READ	187,866,262	0	187,866,262
	FILE_BYTES_WRITTEN	366,374,085	183,160,951	549,535,036
	HDFS_BYTES_WRITTEN	0	21,618,725	21,618,725
Map-Reduce Framework	Map output materialized bytes	183,109,245	0	183,109,245
	Map input records	4,718,026	0	4,718,026
	Reduce shuffle bytes	0	183,109,245	183,109,245
	Spilled Records	9,436,052	4,718,026	14,154,078
	Map output bytes	173,673,175	0	173,673,175
	Total committed heap usage (bytes)	1,403,125,760	614,662,144	2,017,787,904
	CPU time spent (ms)	30,710	14,880	45,590
	Map input bytes	187,857,773	0	187,857,773
	SPLIT_RAW_BYTES	297	0	297
	Combine input records	0	0	0
	Reduce input records	0	4,718,026	4,718,026
	Reduce input groups	0	1	1
	Combine output records	0	0	0
	Physical memory (bytes) snapshot	1,741,205,504	743,030,784	2,484,236,288
	Reduce output records	0	807,009	807,009
	Virtual memory (bytes) snapshot	16,074,743,808	5,365,616,640	21,440,360,448
	Map output records	4,718,026	0	4,718,026

Εικόνα 5.9 Λεπτομέρειες εκτέλεσης της ταξινόμησης των pixel ανά εικόνα



Εικόνα 5.10 Τελική εικόνα εξόδου, χαρτογράφηση καμένης έκτασης



Εικόνα 5.11 Εικόνα εισόδου (αριστερά), εικόνα εξόδου (δεξιά).

5.3 HDINSIGHT και υπηρεσίες cloud

5.3.1 Εισαγωγή στο HDInsight

Το HDInsight είναι μια υπηρεσία cloud της Microsoft που στηρίζεται στο Apache Hadoop. Είναι δηλαδή μια σύγχρονη, στηριζόμενη στο cloud, πλατφόρμα

δεδομένων που διαχειρίζεται τα δεδομένα οποιουδήποτε τύπου, δομημένα ή αδόμητα, και οποιουδήποτε μεγέθους. Ουσιαστικά με αυτή την υπηρεσία ο χρήστης δημιουργεί ένα cluster το οποίο διαχειρίζεται δεδομένα με το Hadoop. Τα απαραίτητα στοιχεία για τη δημιουργία του HDInsight είναι: το όνομα του cluster, το μέγεθος του cluster σε κόμβους δεδομένων (πχ 4 data nodes) , ο κωδικός πρόσβασης στο cluster (το όνομα χρήστη παρέχεται ως admin) και ο δεσμευμένος αποθηκευτικός χώρος που είναι τοποθετημένα τα αρχεία του χρήστη. Όταν δημιουργηθεί η υπηρεσία, παρέχεται η δυνατότητα στο χρήστη να παρακολουθεί τους δεσμευμένους πυρήνες, το σύνολο των εφαρμογών που τρέχουν και το δεσμευμένο αποθηκευτικό χώρο από το cluster. Επίσης δίνεται η δυνατότητα στο χρήστη να συνδεθεί σε μια πλατφόρμα που μπορεί να εκτελεί εργασίες και να θέτει ερωτήματα (queries) που εκτελούνται με MapReduce. Τέλος παρέχεται η δυνατότητα απομακρυσμένης σύνδεσης στο cluster και χειρισμός του κατανεμημένου συστήματος του Hadoop από εκεί. Αντίστοιχες υπηρεσίες για να μπορεί κάποιος να τρέξει δουλειές map-reduce σε περιβάλλον cloud προσφέρουν πολλές εταιρίες στο χώρο, για παράδειγμα: oracle, amazon, google κτλ.

5.3.2 Εκτέλεση στο HDInsight

Μετά την ολοκλήρωση της υλοποίησης του προγράμματος, για να μελετηθούν οι χρόνοι εκτέλεσης του προγράμματος, σε κατανεμημένο περιβάλλον, επιλέχθηκε το HDInsight της Microsoft. Στη συνέχεια παρουσιάζονται στιγμιότυπα εκτέλεσης του προγράμματος στο cluster με συνολική μνήμη ram 6GB, 12 πυρήνες διαθέσιμους χρονισμού 2.2Ghz ο καθένας διαμοιρασμένα σε 2 κόμβους.

← Create Job

Job Name and JAR File

Job Name: WordCount

JAR File: hadoop-examples.jar

Replace file

Delete existing JAR

Parameters

Parameter 0: wordcount

Parameter 1: /user/admin/DaVinci.txt

Add parameter

Final Command

hadoop jar hadoop-examples.jar wordcount /user/admin/DaVinci.txt /user/admin/outcount

Delete job Save draft Execute job

Εικόνα 5.12 Δημιουργία Map-Reduce δουλειάς στο HDInsight



All Applications

Logged in as: gopher

Cluster

About Nodes Applications

NEW NEW SAVING SUBMITTED ACCEPTED RUNNING FINISHED FAILED KILLED

Scheduler

Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
1	0	1	0	11	6 GB	6 GB	0 B	2	0	0	0	0

Show 20 entries

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI
application_1411583032854_0001	manos	MainClass	MAPREDUCE	default	Fri, 26 Sep 2014 15:05:11 UTC	N/A	RUNNING	UNDEFINED		ApplicationMaster

Showing 1 to 1 of 1 entries

First Previous 1 Next Last

Application Overview

User: manos
Name: MainClass
Application Type: MAPREDUCE
Application Tags:
State: FINISHED
FinalStatus: SUCCEEDED
Started: 26-Sep-2014 15:05:11
Elapsed: 3hrs, 5mins, 59sec
Tracking URL: History
Diagnostics:

Job Overview

Job Name: MainClass
User Name: manos
Queue: default
State: SUCCEEDED
Uberized: false
Submitted: Fri Sep 26 15:05:11 GMT 2014
Started: Fri Sep 26 15:05:20 GMT 2014
Finished: Fri Sep 26 18:11:09 GMT 2014
Elapsed: 3hrs, 5mins, 48sec
Diagnostics:
Average Map Time 29mins, 13sec
Average Reduce Time 1hrs, 38mins, 58sec
Average Shuffle Time 54mins, 12sec
Average Merge Time 1sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Fri Sep 26 15:05:16 GMT 2014	workernode030060	logs

Task Type	Total	Complete	
Map	24	24	
Reduce	1	1	
Attempt Type	Failed	Killed	Successful
Maps	0	0	24
Reduces	0	0	1

Εικόνα 5.13 Πληροφορίες για την φάση ταξινόμησης

MapReduce Job job_1411583032854_0002

Job Overview

Job Name: MedianFilterPhase
User Name: manos
Queue: default
State: SUCCEEDED
Uberized: false
Submitted: Sat Sep 27 05:35:23 GMT 2014
Started: Sat Sep 27 05:35:32 GMT 2014
Finished: Sat Sep 27 06:22:17 GMT 2014
Elapsed: 46mins, 44sec
Diagnostics:
Average Map Time 25mins, 50sec
Average Reduce Time 16mins, 38sec
Average Shuffle Time 5mins, 19sec
Average Merge Time 0sec

Task Type	Total	Complete	
Map	7	7	
Reduce	1	1	
Attempt Type	Failed	Killed	Successful
Maps	0	0	7
Reduces	0	0	1

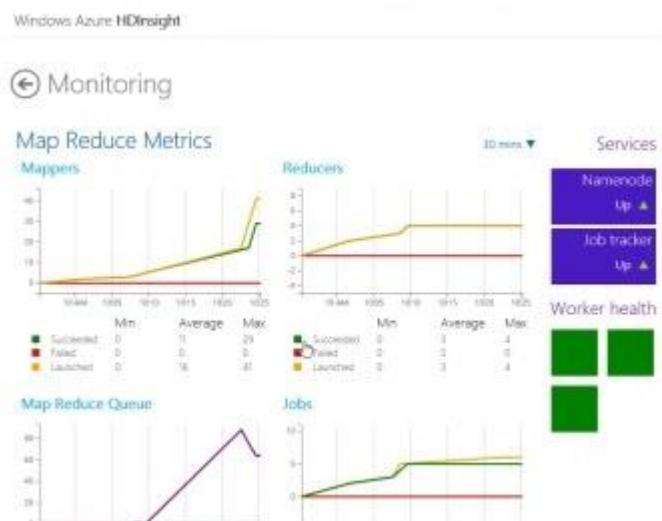
Εικόνα 5.14 Πληροφορίες εφαρμογής ενδιάμεσου φίλτρου

MapReduce Job job_1411583032854_0003

Job Overview			
Job Name:	PixelSortByImage		
User Name:	manos		
Queue:	default		
State:	SUCCEEDED		
Uberized:	false		
Submitted:	Sat Sep 27 08:19:03 GMT 2014		
Started:	Sat Sep 27 08:19:13 GMT 2014		
Finished:	Sat Sep 27 08:34:00 GMT 2014		
Elapsed:	14mins, 47sec		
Diagnostics:			
Average Map Time	3mins, 14sec		

Task Type	Total		Complete
Map	8		8
Reduce	4		4
Attempt Type	Failed	Killed	Successful
Maps	0	0	8
Reduces	0	0	4

Εικόνα 5.15 Πληροφορίες εφαρμογής φίλτρου ελάττωσης θορύβου και ταξινόμησης των pixel ανά εικόνα



Εικόνα 5.16 Παρακολούθηση εξέλιξης της διεργασίας στο HDInsight

5.4 Αναφορά χρόνων εκτέλεσης

Η εκτέλεση του προγράμματος σε συμβατό υπολογιστή, είχε είσοδο 1εικόνα και οι χρόνοι εκτέλεσης ανά φάση είναι οι παρακάτω.

Υποπρόγραμμα	Χρόνος εκτέλεσης
Φάση ταξινόμησης	31m 40s
Ενδιάμεσο φίλτρο	4m 33s

Φίλτρο θορύβου και ταξινόμηση pixel	42s
-------------------------------------	-----

Το πρόγραμμα εκτελέστηκε 8 φορές με είσοδο μία εικόνα, αλλά διαφορετική σε κάθε επανάληψη για καλύτερη εκτίμηση των χρόνων εκτέλεσης. Τα αποτελέσματα παρουσιάζονται στον παρακάτω πίνακα.

Υποπρογράμματα	ΜΟ Χρόνων εκτέλεσης
Φάση ταξινόμησης	33m
Ενδιάμεσο φίλτρο	6m
Φίλτρο θορύβου και ταξινόμηση pixel	1,5m

Για την εκτέλεση του προγράμματος στο cluster, χρησιμοποιήθηκαν ως είσοδο 8 εικόνες και οι χρόνοι εκτέλεσης ανά φάση είναι οι παρακάτω.

Υποπρογράμματα	Χρόνος εκτέλεσης
Φάση ταξινόμησης	3h 5m
Ενδιάμεσο φίλτρο	8m 23s
Φίλτρο θορύβου και ταξινόμηση pixel	5m 43s

Ο μέσος όρος εκτέλεσης των τριών φάσεων στο cluster, με είσοδο τις ίδιες εικόνες, για 10 επαναλήψεις παρουσιάζεται στον πίνακα παρακάτω.

Υποπρογράμματα	ΜΟ Χρόνων εκτέλεσης
Φάση ταξινόμησης	3h 4m
Ενδιάμεσο φίλτρο	8m 53s
Φίλτρο θορύβου και ταξινόμηση pixel	5m 25s

Αναλύοντας τα παραπάνω αποτελέσματα παρατηρείται η διαφορά των χρόνων εκτέλεσης, όταν το πρόγραμμα τρέχει σε κάποιο συμβατό υπολογιστή, και όταν το πρόγραμμα τρέχει σε κατανεμημένο περιβάλλον. Όπως είναι φυσικό, με την αύξηση των κόμβων (cpu-ram) θα επιτευχθούν ακόμα μικρότεροι χρόνοι.

Στις σύγχρονες πλατφόρμες cloud που υποστηρίζουν clusters και hadoop, η κλιμάκωση της επεξεργαστικής ισχύς γίνεται εύκολα, καθώς ο χρήστης αρκεί να ορίσει το μέγεθος της συνολικής μνήμης και των επεξεργαστών που θέλει. Στο τέλος,

ανάλογα με την επεξεργαστική ισχύ που χρειάστηκε για την εκτέλεση της διεργασίας, θα χρεωθεί με το αντίστοιχο αντίτιμο.

5.5 Συμπεράσματα

Είναι εμφανές ότι η ανάγκη για την επεξεργασία μεγάλου όγκου δεδομένων και σε μικρό χρονικό διάστημα γίνεται ολοένα και μεγαλύτερη. Επιστήμες όπως η τηλεπισκόπηση, η βιολογία, η φυσική κτλ., εφαρμογές τύπου social network, μηχανές αναζήτησης, κατανεμημένη ταξινόμηση κτλ., κάνουν χρήση του μοντέλου MapReduce για τις απαιτήσεις του τεράστιου όγκου των δεδομένων που χρειάζεται να επεξεργαστούν οι εφαρμογές ή οι μηχανισμοί που χρησιμοποιούν. Η εφαρμογή που αναπτύχθηκε στην παρούσα πτυχιακή, για την χαρτογράφηση των καμένων εκτάσεων, απαιτούσε την επεξεργασία δορυφορικών εικόνων. Με το μοντέλο MapReduce και τα εργαλεία της GDAL βιβλιοθήκης, επετεύχθη γρήγορη ταξινόμηση των στοιχείων και μια εύκολη στην διαχείριση δομή για να γίνουν οι υπολογισμοί των τιμών από τις τρεις μπάντες της εικόνας, ώστε να καταγραφούν τα καμένα pixel. Τα συμπεράσματα αυτής της πτυχιακής χωρίζονται σε τρεις κατηγορίες, α) στην διαχείριση και επεξεργασία μεγάλου όγκου δεδομένων και την ανάπτυξη του προγράμματος, β) στα αποτελέσματα των εικόνων και την εγκυρότητα τους, γ) στους χρόνους εκτέλεσης του προγράμματος. Στο πρώτο σκέλος αντιμετωπίστηκαν μερικά προβλήματα στην υλοποίηση της εφαρμογής, καθώς το hadoop είναι ανοικτού κώδικα και επίσης έχει πολλές εκδόσεις, με διαφορετικό documentation ανά έκδοση, και με ελάχιστα παραδείγματα χρήσης. Παρά τις δυσκολίες όμως, μόλις καταλάβει κάποιος τον τρόπο με τον οποίο λειτουργεί το μοντέλο MapReduce, μπορεί να αναπτύξει εύκολα ένα πρόγραμμα, που επεξεργάζεται μεγάλο όγκο δεδομένων. Στο δεύτερο σκέλος, τα αποτελέσματα των εικόνων συγκρίθηκαν με τις χαρτογραφημένες πυρκαγιές στο Diachronic Inventory of Forest Fires based on LANDSAT [26] και συμφωνούν. Τέλος, στο τρίτο σκέλος έγινε εκτέλεση του προγράμματος σε περιβάλλον cloud, και υπήρξε μεγάλη εξοικείωση στους σύγχρονους τρόπους ανάθεσης δουλειών MapReduce σε πλατφόρμες, όπου η κλιμάκωση των πόρων γίνεται με εύκολο και γρήγορο τρόπο.

5.6 Μελλοντικές κατευθύνσεις

Στην εφαρμογή που αναπτύχθηκε μπορούν να γίνουν μερικές βελτιώσεις, για την μείωση του χρόνου εκτέλεσης του προγράμματος. Μια τέτοια βελτίωση είναι, να μπορούν οι εικόνες να αποθηκευτούν στο κατανεμημένο σύστημα αρχείων του Hadoop, και κάθε Mapper να αναλαμβάνει να διαβάσει την δορυφορική εικόνα και να περάσει σε μια δομή δεδομένων (π.χ. πίνακας, λίστα) τις τιμές των pixel. Με αυτό τον τρόπο η διαδικασία μετατροπής των πληροφοριών της εικόνας σε αρχείο με μορφή κειμένου, δεν θα είναι απαραίτητη, και με κατανεμημένο τρόπο οι εικόνες θα διαβάζονται πιο γρήγορα. Αντίστοιχα και η έξοδος του προγράμματος θα μπορούσε να είναι απευθείας αρχείο σε μορφή εικόνα, για να μην χρειάζεται η διαδικασία μετατροπής αρχείου κειμένου σε εικόνα. Μία ακόμα βελτίωση που θα μπορούσε να γίνει είναι η έξοδος της πρώτης και της δεύτερης φάσης να γίνονται σε ξεχωριστά αρχεία ανά εικόνα, με τον τρόπο αυτό η πληροφορία που συνοδεύει το pixel (όνομα εικόνας, μέγεθος εικόνας) δεν θα είναι απαραίτητη, οπότε θα έχουμε μείωση του όγκου δεδομένων, και καλύτερη ταξινόμηση στην έξοδο των αρχείων εξόδου.

Συντομογραφίες

NOA	National Observatory of Athens
BSM	Burn Scar Mapping
WWF	World Wide Fund for Nature
TCP/IP	Transmission Control Program / Internet Protocol
CPU	Central Processing Unit
GFS	Goolge File System
HDFS	Hadoop Distributed File System
I/O	Input / Output
SNN	Secondary NameNode
LED	Light-Emitting Diode
GDAL	Geospatial Data Abstraction Library
NIR	Near-Infrared
VIS	Visible
NBR	Normalized Burn Ratio Index
SWIR	Shortwave Infrared
MIR	Mid-Infrared
NDVI	Normalized Difference Vegetation Index
MMU	Minimum Mapping Unit

Βιβλιογραφία

1. Enet, Ελευθεροτυπία, 2011, <http://www.enet.gr/?i=news.el.article&id=295441>, τελευταία πρόσβαση: [20/6/2014]
2. Μαρία Γαζάκη – Χρήστος Λόντος, Επεξεργασία Δορυφορικών Εικόνων για την Χαρτογράφηση Καμένων Περιοχών με χρήση MapReduce, Αθήνα, Νοέμβριος 2011
3. Wikipedia encyclopedia, "Remote Sensing", http://en.wikipedia.org/wiki/Remote_sensing, τελευταία πρόσβαση: [20/6/2014]
4. Σ.Μετρίκας, 1999, "Τηλεπισκόπηση και Ψηφιακή Ανάλυση Εικόνας", Εκδόσεις Ίων
5. Αθανάσιος Αργυρίου, "Τηλεπισκόπηση – Περιβαλλοντικές Εφαρμογές", http://www.hep.upatras.gr/class/download/geo_sim_til/Environmental_Remote_Sensing_Course.pdf, τελευταία πρόσβαση: [21/6/2014]
6. GDAL official web site: www.gdal.org, τελευταία πρόσβαση: [23/6/2014]
7. Wikipedia encyclopedia, "GDAL", <http://en.wikipedia.org/wiki/GDAL>, τελευταία πρόσβαση: [23/6/2014]
8. Kontoes, C., Poilvé, H., Florsch, G., Keramitsoglou, I., and Paralikidis, S., 2009. A comparative analysis of a fixed thresholding vs. a classification tree approach for operational burn scar detection and mapping. International Journal of Applied Earth Observation and Geoinformation, 11, 5, 299
9. Ι.Κ. Κάβουρας, Ι.Ζ. Μήλης, Γ.Β. Ξυλωμένος, Α.Α. Ρουκουνάκη, Αθήνα 2011, "Κατανεμημένα Συστήματα με Java", Συστήματα Υπολογιστών - Τόμος ΙΙΙ, 3η Έκδοση, Εκδόσεις Κλειδάριθμος
10. Andrew S.Tanenbaum, Maarten Van Steen (2007). "Distributed Systems: Principles and Paradigms", Second Edition, Prentice Hall, Upper Saddle River, New Jersey.
11. Thiha Kyaw Zaw, Distributed Systems Definitions, <http://students.depaul.edu/~tkyawzaw/ds-definations.html>, τελευταία πρόσβαση: [25/6/2014]
12. Wikipedia encyclopedia. "Client-Server Model", http://en.wikipedia.org/wiki/Client-server_model, τελευταία πρόσβαση: [25/6/2014]

13. Wikipedia encyclopedia. "Peer-to-Peer Model",
http://en.wikipedia.org/wiki/Peer_to_peer , τελευταία πρόσβαση: [25/6/2014]
14. Jeffrey Dean , Sanjay Ghemawat , 2004 , "MapReduce: Simplified Data Processing on Large Clusters" ,
https://www.usenix.org/legacy/event/osdi04/tech/full_papers/dean/dean.pdf ,
τελευταία πρόσβαση: [25/6/2014]
15. Wikipedia encyclopedia, “ MapReduce” ,
<http://en.wikipedia.org/wiki/MapReduce> , τελευταία πρόσβαση: [25/6/2014]
16. Jimmy Lin , Chris Dyer , 2010, “Data-Intensive Text Processing with MapReduce” , University of Maryland, College Park ,
<http://lntool.github.io/MapReduceAlgorithms/MapReduce-book-final.pdf> ,
τελευταία πρόσβαση: [25/6/2014]
17. Jairam Chandar, 2010, “Join Algorithms using Map/Reduce”, Master’s Thesis, School of Informatics, University of Edinburgh ,
<http://www.inf.ed.ac.uk/publications/thesis/online/IM100859.pdf> , τελευταία πρόσβαση: [26/6/2014]
18. Andreas Kamilaris , “Introduction to Hadoop”, Πανεπιστήμιο Κύπρου,
<http://www.cs.ucy.ac.cy/courses/EPL660/lectures/lab4.pdf> , τελευταία πρόσβαση: [24/6/2014]
19. Wikipedia encyclopedia , “Apache Hadoop”,
http://en.wikipedia.org/wiki/Apache_Hadoop , τελευταία πρόσβαση: [26/6/2014]
20. Chuck Lam , 2010 , “Hadoop in Action” , Manning Publications ,
<http://www.manning.com/lam/SampleCh1.pdf> , τελευταία πρόσβαση: [26/6/2014]
21. Apache Hadoop official web site , <http://hadoop.apache.org/> , τελευταία πρόσβαση: [26/6/2014]
22. Apache Hadoop Api , Class TextInputFormat ,
<https://hadoop.apache.org/docs/current/api/org/apache/hadoop/mapreduce/lib/Input/TextInputFormat.html> , τελευταία πρόσβαση: [28/6/2014]
23. Wikipedia encyclopedia, “Median filter”,
http://en.wikipedia.org/wiki/Median_filter , τελευταία πρόσβαση: [28/6/2014]

24. Wikipedia encyclopedia, "Connected component labeling",
http://en.wikipedia.org/wiki/Connected-component_labeling , τελευταία πρόσβαση: [30/6/2014]
25. Κ.Τσαγκάρη, Γ.Καρέτσος, Ν.Προύτσος (2011). Ινστιτούτο Μεσογειακών Δασικών Οικοσυστημάτων και Τεχνολογίας Δασικών Προϊόντων, WWF Greece, "Δασικές Πυρκαγές Ελλάδας 1983-2008", Αθήνα
26. Diachronic Inventory of Forest Fires based on LANDSAT,
http://ocean.space.noa.gr/diachronic_bsm/index.php , τελευταία πρόσβαση: [19/9/2014]
27. How to write output to multiple named files in Hadoop using MultipleTextOutputFormat ,
<https://sites.google.com/site/hadoopandhive/home/how-to-write-output-to-multiple-named-files-in-hadoop-using-multipletextoutputformat>

Ακολουθούν μερικά βοηθητικά links:

1. Getting Started With Hadoop , <https://docs.google.com/document/d/1v-J19xwJn-Pw9F8OCgLn04dqKwYkGIOxyqRAIGpmHk0/edit?pli=1> , τελευταία πρόσβαση: [25/9/2014]
2. Running GDAL Java , <http://geoexamples.blogspot.gr/2012/05/running-gdal-java.html> , τελευταία πρόσβαση: [25/9/2014]
3. Compiling hadoop-eclipse-plugin , <http://compiling-hadoop-eclipse-plugin.blogspot.gr/> , τελευταία πρόσβαση: [25/9/2014]
4. Module 3: Getting Started With Hadoop ,
<https://developer.yahoo.com/hadoop/tutorial/module3.html#eclipse> , τελευταία πρόσβαση: [25/9/2014]
5. Running Hadoop on Ubuntu Linux (Single-Node Cluster) ,
<http://www.michael-noll.com/tutorials/running-hadoop-on-ubuntu-linux-single-node-cluster/#update-homebashrc> , τελευταία πρόσβαση: [25/9/2014]
6. Hadoop Core Installation on Ubuntu 12.10 Video ,
<https://www.youtube.com/watch?v=cxXBpcOjJWk> , τελευταία πρόσβαση: [25/9/2014]
7. Configure eclipse for map reduce and writing sample word count program ,
<https://www.youtube.com/watch?v=TavehEdfNDk> , τελευταία πρόσβαση: [25/9/2014]

8. Hadoop: How to read a file from HDFS in Hadoop classes in Java ,
<https://sites.google.com/site/hadoopandhive/home/hadoop-how-to-read-a-file-from-hdfs> , τελευταία πρόσβαση: [25/9/2014]
9. Hadoop Distributed File System (HDFS) Java tutorial,
<http://myjavanotebook.blogspot.gr/2008/05/hadoop-file-system-tutorial.html> ,
τελευταία πρόσβαση: [25/9/2014]

Εικόνες

- 2.1 http://el.wikipedia.org/wiki/%CE%91%CF%81%CF%87%CE%B5%CE%AF%CE%BF:Remote_model.jpg
- 2.3 <http://sundar5.wordpress.com/2010/03/19/hadoop-basic/>
- 2.4 <http://www.jpl.nasa.gov/spaceimages/details.php?id=PIA02622>
- 2.9 <http://www.cac.cornell.edu/Ranger/MapReduce/locality.aspx>
- 2.10 <http://www.gisfun.50megs.com/Tilepiskopisi.html>
- 4.1 <http://academic.emporia.edu/aberjame/student/alvarez2/771proj.html>