

Δυναμική Απεικόνιση Φράκταλ με χρήση MapReduce

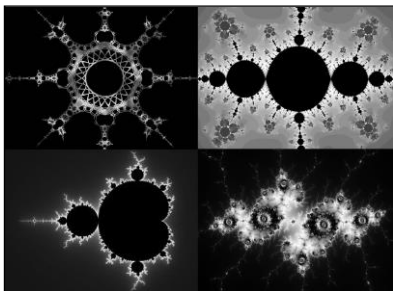
Ευαγγελία Παπουτσόγλου, Αριστοτέλης Φαρμάκης
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Πολυτεχνική Σχολή, Δημοκρίτειο Πανεπιστήμιο Θράκης
Ξάνθη 67 100, Ελλάδα
evanpapo1@ee.duth.gr, arisfarm@ee.duth.gr

Περίληψη: Στην παρούσα εργασία μελετάται το θέμα της κατασκευής μιας δυναμικής απεικόνισης φράκταλ (fractal animation) με χρήση της προγραμματιστικής τεχνικής MapReduce του Hadoop. Αυτή η τεχνική μας επιτρέπει να κατανέμουμε, σε περισσότερα συστήματα, τόσο το υπολογιστικό φορτίο όσο και το κόστος εγγραφής στο δίσκο. Δίνεται, λοιπόν, έμφαση στο θέμα της απόδοσης, ιδίως από πλευράς χρόνου. Εξετάζουμε αν η επιλεγμένη τεχνολογία επιφέρει τις αναμενόμενες επιδόσεις, και καταλήγουμε στο συμπέρασμα ότι η μία από τις δύο μεθόδους που προτείνουμε οδηγεί σε δραματική βελτίωση της διαδικασίας.

[λέξεις-κλειδιά: `mapreduce`, φράκταλ, δυναμική απεικόνιση φράκταλ, `hadoop`, κατανεμημένα συστήματα]

I. ΕΙΣΑΓΩΓΗ

Μια εικόνα fractal (ελλ. φράκταλ, μορφόκλασμα ή μορφοκλασματικό σύνολο) είναι παράγωγο μαθηματικών πράξεων, και παρουσιάζει αυτό-περιεχόμενες δομές σε μια ποικιλία κλιμάκων. Με λιγότερο αυστηρή ορολογία, φράκταλ θα ονομάζαμε ένα γεωμετρικό σχήμα, του οποίου η δομή



Σχήμα 1: Δισδιάστατα fractals

επαναλαμβάνεται σε ολοένα και μικρότερες ή μεγαλύτερες κλίμακες, δηλαδή μια γεωμετρική δομή που εμφανίζει αυτό-ομοιότητα υπό αλλαγή κλίμακας. Αρκετές είναι μάλιστα οι περιπτώσεις που ένα φράκταλ μπορεί να δημιουργηθεί από ένα πρότυπο, εάν αυτό επαναλαμβάνεται επ' άπειρο μέσω κάποιας επαναληπτικής διαδικασίας [1,2]. Μερικά δημοφιλή δισδιάστατα φράκταλς απεικονίζονται στο Σχήμα 1.

Στη σύγχρονη εποχή τα φράκταλ έχουν πολλές εφαρμογές σε επιστημονικούς τομείς, όπως ο τομέας ανάλυσης δεδομένων, καθώς και στην τέχνη. Παρ' όλα αυτά, οι παραδοσιακές μέθοδοι είναι συχνά υπερβολικά χρονοβόρες για τον υπολογισμό ενός ικανοποιητικού όγκου αυτών των εικόνων. Συνεπώς, σε αυτήν την εργασία επιλέγουμε τη MapReduce [3], ένα μοντέλο κατανεμημένης επεξεργασίας μεγάλου όγκου δεδομένων. Έτσι, επιχειρούμε να

διαμοιράσουμε το φόρτο που προκύπτει, να επιταχύνουμε τη διαδικασία παραγωγής πολλών εικόνων φράκταλ, και εν τέλει να επιτύχουμε τη συνένωσή τους σε ένα βίντεο.

Το θέμα της κατασκευής φράκταλ έχει προσεγγιστεί κατά καιρούς με διαφορετικούς τρόπους, καθώς προκαλούν έντονο ενδιαφέρον ως ελκυστικές οπτικοποιήσεις μαθηματικών συμπεριφορών. Η παρούσα εργασία επικεντρώνεται στο θέμα της απόδοσης. Ο υπολογισμός των εικονοστοιχείων που συγκροτούν μια εικόνα φράκταλ εμπλέκει στο σύνολό του ένα μεγάλο αριθμό πράξεων, ωστόσο αυτές είναι στην πλειοψηφία τους απλές, και ο χρόνος που τους αφιερώνεται από την υπολογιστική μονάδα μάλλον είναι κατά πολύ ελαττωμένος σε σχέση με αυτόν που απαιτείται για την εγγραφή των νέων πληροφοριών στο μέσο αποθήκευσης.

Στο σενάριο που μελετάται, όπου είναι επιθυμητή η παραγωγή μιας δυναμικής απεικόνισης φράκταλ, είναι προφανές ότι η επιβάρυνση στους υπολογιστικούς πόρους πολλαπλασιάζεται. Θα διερευνήσουμε εάν είναι δυνατή η επίτευξη κάποιας επιτάχυνσης στη διαδικασία αυτή. Το MapReduce ενδείκνυται για εφαρμογές η απόδοση των οποίων περιορίζεται από τις ταχύτητες εγγραφής και ανάγνωσης της δευτερεύουσας αποθηκευτικής μνήμης (πχ σκληρός δίσκος) όταν αυτές εκτελούνται σε ένα μόνο σύστημα. Επιδιώκουμε το διαμοιρασμό του φόρτου για την εγγραφή σε περισσότερες μονάδες, και έτσι τη μείωση του καθολικού χρόνου παραγωγής της δυναμικής απεικόνισης φράκταλ.

Στη συνέχεια θα παρουσιαστεί η μεθοδολογία που ακολουθείται για το σκοπό μας. Προτείνονται δύο μέθοδοι με τις οποίες επιδιώκουμε να περιορίσουμε τη χρονική διάρκεια παραγωγής ενός βίντεο –δυναμικής απεικόνισης– φράκταλ. Περιλαμβάνονται ο αλγόριθμος παραγωγής φράκταλ, οι προγραμματιστικές ιδιαιτερότητες και επιπλέον ρυθμίσεις απαραίτητες για την προσαρμογή των αλγορίθμων στο πλαίσιο της MapReduce, και τέλος τα αποτελέσματα του εγχειρήματός μας και τα σχετικά συμπεράσματα.

II. ΣΧΕΤΙΚΕΣ ΛΥΣΕΙΣ/ΜΕΘΟΔΟΙ

Αναζητώντας αρχικά το πλαίσιο της εργασίας αυτής, δεν καταφέραμε να βρούμε βιβλιογραφία απόλυτα σχετική με το θέμα. Εάν όμως κοιτάξουμε την προσπάθειά μας από την πλευρά του μηχανικού και σπάσουμε το θέμα μας σε μικρότερα τμήματα, διαπιστώνουμε πως παρόμοιας φιλοσοφίας ερωτήματα έχουν ερευνηθεί.

Ενδεικτικά, ας δούμε το αντικείμενο της παραγωγής εικόνων φράκταλ με παραλληλία. Η δημοσίευση [4], παρόλο

που ασχολείται με παράλληλη κατασκευή εικόνων φράκταλ, δεν άπτεται του ίδιου ακριβώς αντικειμένου μελέτης, καθώς ασχολείται γενικότερα με το ζήτημα της αύξησης της ταχύτητας του υπολογισμού σειριακών δεδομένων εφαρμόζοντας παράλληλους αλγορίθμους. Προσπαθεί δηλαδή να εφαρμόσει ό,τι χρειάζεται για να παραχθεί τελικά κάποιος αποδοτικός, παράλληλος αλγόριθμος, ο οποίος θα κατανέμει ίσα το φορτίο στους επεξεργαστές και θα παράγει στο τέλος ένα σειριακό αποτέλεσμα. Για τη μελέτη αυτή, ως παράδειγμα, χρησιμοποιείται η δυναμική απεικόνιση φράκταλ. Έμφαση ωστόσο δίνεται κυρίως στη μέθοδο σχεδιασμού του αλγορίθμου και στην προσπάθεια επίτευξης παραλληλίας, αντίθετα με την παρούσα εργασία που αναλύει το αντικείμενο της παραγωγής μιας δυναμικής απεικόνισης φράκταλ, εφαρμόζοντας τις μεθόδους που προτείνονται σε μια Hadoop συστάδα υπολογιστών, και όχι τοπικά. Ακολουθεί η σύγκριση και τελικά ο σχολιασμός και τα συμπεράσματα των νέων μεθόδων.

Επιπρόσθετα, αναζητώντας τους τομείς όπου έχει σημειωθεί χρήση μεθόδων παραλληλίας και μάλιστα της πλατφόρμας του Hadoop, φαίνεται ότι ο χώρος της συμπίεσης εικόνων φράκταλ έχει βρει απήχηση, όπως φαίνεται στη δημοσίευση [5]. Μάλιστα η *Μελέτη Άρτιας Απόδοσης Εικόνων Φράκταλ και Επίτευξη Συμπίεσης* φαίνεται να συνδέονται στενά. Ωστόσο δε σχετίζεται άμεσα με την έρευνα της παρούσας εργασίας. Τα παραδείγματα της συμπίεσης ασχολούνται σχεδόν αποκλειστικά με τη φωτορεαλιστική απεικόνιση εικόνων υψηλής ευκρίνειας σε περιορισμένο χρόνο, ενώ στόχος μας είναι η πιο αποδοτική παραγωγή μιας δυναμικής απεικόνισης φράκταλ.

III. ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΜΕΘΟΔΟΙ

A. Fractals

Αρχικά, για λόγους πληρότητας, αναφέρουμε ότι διακρίνονται τρεις κύριες κατηγορίες φράκταλ [6]. Η πρώτη είναι τα φράκταλ τύπου μιγαδικών αριθμών, όπως τα φράκταλ τύπου Μάντελμπροτ που αναλύουμε στην εργασία αυτή. Οι μέθοδοί μας καλύπτουν οτιδήποτε παράγεται με αυτή τη λογική, όπως τα παραπλήσια φράκταλ τύπου Julia ή Newton. Στη δεύτερη κατηγορία εντάσσονται τα φράκταλ που προκύπτουν από επαναληπτικά συστήματα συναρτήσεων (iterated function systems), ενώ η τρίτη περιλαμβάνει τα τροχιακά (orbit) φράκταλ, που σχετίζονται με τη θεωρία του χάους.

Στα πλαίσια της παρούσας εργασίας ασχοληθήκαμε μόνο με φράκταλ από το σετ του Μάντελμπροτ. Σε αυτά, το κάθε εικονοστοιχείο υπολογίζεται ανεξάρτητα από άλλα, βάσει του αναδρομικού τύπου $z_{n+1} = z_n^2 + c$. Ο μιγαδικός c είναι σταθερός για κάθε εικονοστοιχείο, και προκύπτει από τις συντεταγμένες του στην εικόνα. Εξετάζουμε έτσι αν το μέτρο του μιγαδικού z ξεπερνά το 2, μετά από N επαναλήψεις. Με το κριτήριο αυτό αποφασίζουμε αν υπάρχει ή όχι σύγκλιση για το τρέχον εικονοστοιχείο.

B. MapReduce

Το μοντέλο της MapReduce βασίζεται στη διάσπαση μεγάλων όγκων δεδομένων, και παράλληλη επεξεργασία των τμημάτων τους σε ένα καταναμημένο σύστημα. Περιλαμβάνει πρώτα τη φάση της Map και στη συνέχεια τη φάση της Reduce. Στη Map τα αρχικά δεδομένα (η είσοδος) διασπώνται σε κομμάτια και στη συνέχεια υποβάλλονται σε επεξεργασία, προκειμένου να αντλήσουμε πληροφορία από αυτά. Είναι σημαντικό εδώ να σημειωθεί πως το κάθε κομμάτι υπόκειται σε όμοια, αλλά ανεξάρτητη επεξεργασία από τα υπόλοιπα, άρα πληροφορία ακόμη δεν ανταλλάσσεται. Η προκύπτουσα πληροφορία αυτή εκτέμνεται με τη μορφή ζευγών κλειδιού-τιμής. Το κλειδί αποτελεί το χαρακτηριστικό βάσει του οποίου τα ζεύγη ομαδοποιούνται και συλλαμβάνονται από τη φάση της Reduce, κατά την οποία πολλοί Reducers τρέχουν ταυτόχρονα σε διαφορετικούς κόμβους της συστάδας. Εφόσον το περιβάλλον μας εγγυάται ότι όλα τα ζεύγη με ένα συγκεκριμένο κλειδί θα κατευθυνθούν προς ένα συγκεκριμένο Reducer, μπορούμε να προβούμε σε ανάλογη επεξεργασία της πληροφορίας (τιμών) που συνοδεύουν, για να παράξουμε την τελική έξοδο.

Η φάση της Map εκτελείται πάντα, όμως η φάση της Reduce είναι προαιρετική. Η επιβάρυνση που προκύπτει στο δίκτυο οφείλεται στην μεταφορά και διανομή των ζευγών κλειδιού-τιμής από τους Mapper στους Reducer. Δίνεται επίσης η δυνατότητα ένταξης Combiner στη διαδικασία. Ένας Combiner μπορεί να εκτελεστεί μετά την ολοκλήρωση ενός Mapper, στον ίδιο κόμβο με αυτόν, λαμβάνοντας όλα τα ζεύγη που έχει εκτέμψει, για να παρέχει κάποια πρόσθετη επεξεργασία πάνω τους πριν διοχετευτούν τελικά στο δίκτυο. Η επεξεργασία αυτή μπορεί να συρρικνώσει τον όγκο δεδομένων που τελικά πρέπει να μεταφερθούν προς άλλους κόμβους, περιορίζοντας τον αντίκτυπο στο δίκτυο.

C. Μέθοδοι

Στην εργασία αυτή προτείνουμε δύο μεθόδους, οι οποίες επιδιώκουμε να αποτελέσουν έγκυρες εναλλακτικές για την παραδοσιακή σειριακή: Η πρώτη περιλαμβάνει τις φάσεις της Map και της Reduce, ενώ η δεύτερη προσθέτει έναν Combiner και παραλείπει τη φάση της Reduce.

Ως πιο φυσικός τρόπος για την αρχική κατανομή φόρτου, όσον αφορά την παραγωγή του υλικού, θεωρήθηκε η διαίρεση με τα καρέ (frames) ως βασική μονάδα: σε κάθε Mapper ανατίθεται η κατασκευή ενός συνόλου από αυτά. Πίσω από την επιλογή αυτή κρύβεται η προσδοκία πως μια δυναμική απεικόνιση φράκταλ θα αποτελείται από έναν υψηλό αριθμό καρέ, οπότε η χρήση υποδιαίρεσών τους σε επίπεδο εικονοστοιχείων δεν θα μπορούσε να συνεισφέρει περαιτέρω. Σε ένα τέτοιο σενάριο θα έπρεπε να συνυπολογιστεί ο φόρτος ανασυγκρότησης της κάθε εικόνας, αφού οι σχετικές Map ολοκληρωθούν, αλλά και η επιβάρυνση στο δίκτυο καθώς τα υποσύνολα μιας εικόνας θα συμπίεζονταν λιγότερο αποδοτικά από ότι ένα καρέ ολόκληρο.

Με στόχο την αποφυγή των πρώιμων εγγραφών σε αρχεία, τα παραγόμενα καρέ μεταβιβάζονται από τη φάση της Map στη φάση της Reduce μόνο ως μια ακολουθία χαρακτήρων. Για την ευκολότερη διασφάλιση της ακεραιότητας της πληροφορίας σε επίπεδο byte, τη διακίνηση πληροφορίας

υποστηρίζει η κωδικοποίηση Base64, η οποία μπορεί επίσης να προσφέρει περαιτέρω μείωση του όγκου δεδομένων που διακινείται στο δίκτυο. Εκτός αυτού, καθίσταται ευκολότερη η προσθήκη και άλλων στοιχείων στο αλφαριθμητικό string που εκπέμπεται από το Mapper, και στη συνέχεια η απομόνωση του κάθε μέρους, αφού μπορούμε να τα διαχωρίσουμε βάσει χαρακτήρων που δεν ανήκουν στο σύνολο τιμών της κωδικοποίησης Base64.

Ένα τέτοιο στοιχείο είναι το ID του κάθε καρτέ, που προσδιορίζει μοναδικά τη χρονική θέση του σε σχέση με τα άλλα καρτέ στην τελική δυναμική απεικόνιση. Πρόκειται για έναν αύξοντα αριθμό N ψηφίων, όπου το N προσδιορίζεται από το αρχείο εισόδου που θα παρέχεται ως είσοδος στη διαδικασία. Και οι δυο υλοποιήσεις δέχονται ένα αρχείο κειμένου, κάθε γραμμή του οποίου παρέχει όλη την πληροφορία για την κατασκευή του κάθε καρτέ. Με τη σειρά εμφάνισής τους, οι παράμετροι είναι:

- το ID του καρτέ με τη μορφή frameXXXXX
- οι διαστάσεις της εικόνας (width, height)
- παράγοντας μεγέθυνσης
- μέγιστος αριθμός επαναλήψεων για τον αναδρομικό τύπο
- οριζόντια και κάθετη μετατόπιση της εικόνας

Για παράδειγμα, μια έγκυρη σειρά θα ήταν:
frame00001:800,600,200,570,400,300

Έτσι, μετατοπίζοντας και μεγεθύνοντας με μικρά βήματα μπορούμε να «περιηγηθούμε» κατά βούληση σε ένα φράκταλ, παράγοντας μια δυναμική απεικόνιση.

Την κατασκευή των καρτέ ακολουθεί, μετά τη Map, η διαδικασία Reduce. Εφόσον τα αρχεία εξόδου θα είναι ισάριθμα με τους Reducers που θα εκτελεστούν και επίσης δεν επιθυμούμε να λάβουμε ως έξοδο την τελική δυναμική απεικόνιση τμηματικά διαχωρισμένη σε διαφορετικά αρχεία, ορίζουμε πως θα τρέξει μόνο ένας Reducer. Ορίζουμε ακόμα παντού το ίδιο κλειδί, ώστε όλα τα καρτέ να υποστούν επεξεργασία κατά την ίδια διαδικασία. Θα μπορούσε να παραλειφθεί επομένως ο ρητός προσδιορισμός του αριθμού των Reducers, αφού σε κάθε περίπτωση, τα key-value pairs με κοινό κλειδί κατευθύνονται μόνο προς έναν. Η εκλογή του κλειδιού με την παραπάνω φιλοσοφία είναι που καθιστά επιτακτική την ανάγκη ενσωμάτωσης πληροφορίας όπως αυτή του ID του καρτέ στην πλευρά του value.

Στη φάση της Reduce, τώρα, γίνεται η ταξινόμηση των καρτέ με τη βοήθεια του ID τους, ο διαχωρισμός και η αποκωδικοποίηση των δεδομένων εικόνας. Ακολουθεί η ένταξή τους σε ένα βίντεο. Με κύρια έννοια τη διατήρηση όλης της παραχθείσας πληροφορίας, η δυναμική απεικόνιση υποβάλλεται σε συμπίεση χωρίς απώλειες. Επιλέχθηκε η συμπίεση σε μορφή H.264 για λόγους συμβατότητας, ευχρηστίας, και εύκολης παραμετροποίησης της κωδικοποίησης. Τελικά, το βίντεο μεταφέρεται στο κατανεμημένο σύστημα, όπου είναι εύκολα προσβάσιμο από όλους τους κόμβους της συστάδας.

Η δεύτερη εκδοχή της διαδικασίας, η οποία περιλαμβάνει και το στάδιο του Combiner, αποτελεί προσπάθεια μείωσης, τόσο της επιβάρυνσης μεταφοράς δεδομένων μέσω του δικτύου, όσο και του μη κατανεμημένου υπολογιστικού φόρτου. Ο φόρτος αυτός διαφορετικά περιορίζεται αποκλειστικά στη μονάδα που αναλαμβάνει τον ένα Reducer

στην αντίστοιχη φάση. Τα προϊόντα του κάθε Mapper απορροφώνται από έναν Combiner, που συγκεντρώνει τα καρτέ σε ένα σύντομο βίντεο, που θα αποτελέσει ένα κομμάτι, μόνο, του τελικού. Η κλάση που αναλαμβάνει αυτή τη διεργασία λειτουργεί όμοια με το Reducer της πρώτης εκδοχής. Αυτό συντελεί και στη βελτίωση από άποψης χρόνου, αφού δεν είναι απαραίτητη η λήξη όλων των διαδικασιών Mapper προτού αρχίσει οποιαδήποτε διαδικασία συμπίεσης σε βίντεο.

Οι παραπάνω προσεγγίσεις επιχειρούν, αμφότερες, να βελτιώσουν την απόδοση των διεργασιών, μειώνοντας τους χρόνους εγγραφής και ανάγνωσης στο δίσκο. Ξεκινήσαμε εξάλλου με την υπόθεση ότι ο χρόνος παραγωγής και επεξεργασίας των πληροφοριών που επιθυμούμε είναι σημαντικά λιγότερος από το χρόνο πρόσβασης στο δίσκο.

Συνοψίζοντας, στις μεθόδους μας κερδίσαμε υπολογιστικό χρόνο μειώνοντας την ανταλλαγή δεδομένων με τα πιο αργά αποθηκευτικά μέσα. Στην πρώτη περίπτωση, οι Mapper κατασκευάζουν τα καρτέ, που στο σύνολό τους κατευθύνονται και συνενώνονται στο Reducer. Στη δεύτερη το ρόλο του Reducer αναλαμβάνουν οι Combiners, καθένας από τους οποίους κατασκευάζει ένα μικρό βίντεο (μέρος του τελικού).

IV. ΘΕΩΡΗΤΙΚΗ ΣΥΓΚΡΙΣΗ ΜΕΘΟΔΩΝ

Συμβολισμοί:

Θεωρούμε ότι έχουμε σχετική ομοιογένεια στο κάθε μέγεθος, και οι τιμές δεν διαφέρουν δραματικά. Ο υπολογισμός του κόστους γίνεται παντού σε μονάδες χρόνου.

- X** ο αριθμός των καρτέ προς παραγωγή
- Y** το μέσο μέγεθος ενός καρτέ
- P** ο χρόνος υπολογισμού του ενός καρτέ
- C** το μέσο κόστος ενσωμάτωσης ενός καρτέ στο βίντεο
- W** η ταχύτητα εγγραφής στο τοπικό σύστημα
- Bw** η χωρητικότητα του δικτύου της συστάδας
- N** ο αριθμός των κόμβων στη συστάδα
- cf** ο μέσος παράγοντας συμπίεσης (compression factor): εδώ λαμβάνεται ως ο σχετικά σταθερός λόγος του μεγέθους του συνόλου των καρτέ, προς το μέγεθος του τελικού (συμπίεσμένου) βίντεο, χάριν απλότητας.

Σύγκριση μεθόδων

A. Μέθοδος #1: χωρίς καμία παραλληλία, τοπικά

Θεωρείται πως έχουμε μια διαδικασία παραγωγής μιας δυναμικής απεικόνισης φράκταλ σε έναν επιτραπέζιο υπολογιστή. Ένας κόμβος παράγει όλα τα καρτέ και τα ενώνει σε ένα βίντεο, δηλαδή έχουμε την απλή διαδικασία του **Διαγράμματος 1**.

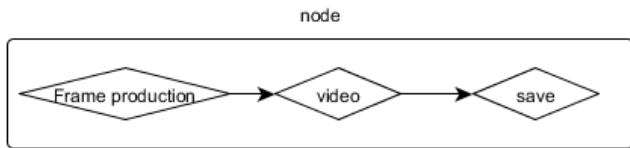
Παραγωγή όλων των καρτέ: $P \cdot X$

όπου X ο συνολικός αριθμός των καρτέ που παράγονται, και P το μέσο υπολογιστικό κόστος για το κάθε καρτέ. Εδώ, ελλείψει οποιασδήποτε παραλληλίας, τα καρτέ όλα υπολογίζονται σειριακά, το ένα μετά το άλλο στην ίδια επεξεργαστική μονάδα.

Σύνθεση όλων των καρτέ σε ένα βίντεο: $X \cdot C$,

όπου τα καρέ αριθμού X ενώνονται σε ένα βίντεο με υπολογιστικό κόστος C για την επεξεργασία-ενσωμάτωση στο βίντεο ανά καρέ. Το αποτέλεσμα είναι ένα βίντεο μεγέθους $X \cdot Y / cf$.

Αποθήκευση του βίντεο στο τοπικό σύστημα: $(X \cdot Y) / (cf \cdot W)$, όπου δεδομένα όγκου $X \cdot Y / cf$ περνούν στο αποθηκευτικό μέσο με ταχύτητα εγγραφής W.



Διάγραμμα 1: διαγραμματική περιγραφή της μεθόδου 1

Δηλαδή:

Το κόστος της μεθόδου #1 (αναφέρεται ως Κόστος #1 για συντομία στο εξής, και ομοίως για τις ακόλουθες μεθόδους) θα είναι:

$$\text{Κόστος \#1} = P \cdot X + X \cdot C + \frac{X \cdot Y}{cf \cdot W} = X \cdot \left(P + C + \frac{Y}{cf \cdot W} \right)$$

B. Μέθοδος #2

Η παραπάνω MapReduce τρέχει τώρα σε μια συστάδα των N κόμβων: Ο κάθε κόμβος παράγει έναν αριθμό καρέ, καθένα από τα οποία μεταφέρεται ως ένα ζεύγος κλειδιού-τιμής στο Reducer. Εκεί ενώνονται στο τελικό βίντεο. Αυτό απεικονίζεται στο Διάγραμμα 2.

Παραγωγή καρέ σε έναν κόμβο: $X \cdot P / N$

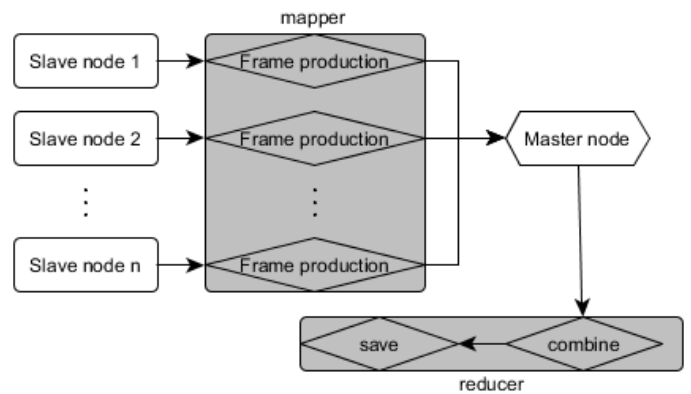
αφού για X καρέ και N κόμβους, το μερίδιο του κάθε κόμβου θα είναι X/N .

Ακολουθεί η εγγραφή των παραχθέντων ζευγών κλειδιού-τιμής (δηλαδή των δεδομένων των καρέ) στο τοπικό σύστημα: $(X \cdot Y) / (N \cdot W)$,

και η μεταφορά των δεδομένων των καρέ στο Reducer: $X \cdot Y / Bw$, αφού όλα τα δεδομένα των καρέ πρέπει να μεταφερθούν μέσω του δικτύου από τους Mappers στο Reducer. Τα δεδομένα θα είναι όγκου $X \cdot Y$, για X καρέ, Y μέσο όγκο δεδομένων ανά καρέ, και Bw εύρος καναλιού.

Σύνθεση όλων των καρέ σε ένα βίντεο: $X \cdot C$.

Αποθήκευση του βίντεο στο τοπικό σύστημα: $(X \cdot Y) / (cf \cdot W)$ όπου δεδομένα όγκου $X \cdot Y / cf$ περνούν στο αποθηκευτικό μέσο με ταχύτητα εγγραφής W. Το cf πρόκειται για τον παράγοντα συμπίεσης.



Διάγραμμα 2: διαγραμματική περιγραφή της μεθόδου 2

Δηλαδή Κόστος #2 =

$$\frac{P \cdot X}{N} + \frac{X \cdot Y}{N \cdot W} + \frac{X \cdot Y}{Bw} + X \cdot C + \frac{X \cdot Y}{cf \cdot W} = X \cdot \left(\frac{P}{N} + \frac{Y}{N \cdot W} + \frac{Y}{Bw} + C + \frac{Y}{cf \cdot W} \right)$$

C. Μέθοδος #3

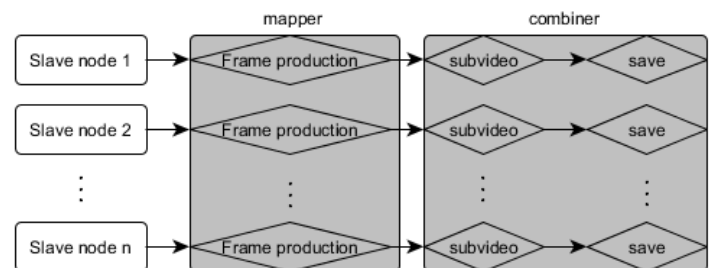
Τώρα εξετάζεται η περίπτωση μιας MapReduce με Combiners: Ο κάθε κόμβος παράγει ένα σύνολο καρέ στο Mapper, και ο Combiner τα ενώνει σε ένα βίντεο που θα αποτελέσει μέρος του τελικού. Όλα τα υποβίντεο περνούν στο Reducer, όπου συγχωνεύονται σε ένα ολοκληρωμένο βίντεο. Αυτό φαίνεται στο Διάγραμμα 3.

Παραγωγή καρέ σε έναν κόμβο: $X \cdot P / N$.

Σύνθεση των καρέ κάθε κόμβο σε ένα υπο-βίντεο: $X \cdot C / N$.

Τα υπο-βίντεο θα έχουν συνολικό όγκο $(X \cdot Y) / (N \cdot cf)$.

Το καθένα εγγράφεται στο τοπικό σύστημα: $(X \cdot Y) / (N \cdot cf \cdot W)$.



Διάγραμμα 3: διαγραμματική περιγραφή της μεθόδου 3

Σύγκριση κόστους ανά καρέ:

$$\text{Κόστος \#1 ανά καρέ} = P + C + \frac{Y}{cf \cdot W}$$

$$\text{Κόστος \#2 ανά καρέ} = \frac{P}{N} + \frac{Y}{N \cdot W} + \frac{Y}{Bw} + C + \frac{Y}{cf \cdot W}$$

$$\text{Κόστος \#3 ανά καρέ} = \frac{1}{N} \cdot \left(P + C + \frac{Y}{cf \cdot W} \right)$$

Γίνεται αντικατάσταση στους παραπάνω τύπους με τα παρακάτω (συντηρητικά) μεγέθη:
 $X = 1000$ καρτέ, $Y = 100KB$, $P = 0.5ms$, $C = 0.02ms$,
 $W = 100 MB/s$, $Bw = 1Gb/s$, $N = 100$, $cf = 2$.

Κόστος #1 ανά καρτέ = $1.02ms$

Κόστος #2 ανά καρτέ = $1.365ms$

Κόστος #3 ανά καρτέ = $0.0102ms$

Τα κόστη #1 και #2 έχουν κοινούς τους δύο τελευταίους τους όρους, άρα τους παραλείπουμε. Επομένως θα παρατηρούσαμε βελτίωση όταν:

$$P > \frac{P}{N} + \frac{Y}{N \cdot W} + \frac{Y}{Bw}$$

Θεωρώντας τους όρους που περιλαμβάνουν το N στον παρονομαστή αμελητέους, φαίνεται ότι δεν έχουμε επιτύχει ακόμα βελτίωση. Η ταχύτητα μεταφοράς ενός καρτέ στο δίκτυο θα είναι πάντα κατά τάξεις πιο αργή από το χρόνο που χρειάζεται ο επεξεργαστής για να παράξει ένα καρτέ.

Ακριβώς σε αντίθετο αποτέλεσμα καταλήγουμε όμως συγκρίνοντας τα κόστη 1 και 3. Είναι ξεκάθαρο ότι

$$\text{Κόστος \#3} = \frac{\text{Κόστος \#1}}{N},$$

πράγμα που σημαίνει ότι σίγουρα έχουμε επιτύχει βελτίωση, και μάλιστα γραμμική προς τον αριθμό των κόμβων στο cluster.

Η μέθοδος 3 ελαχιστοποιεί το φόρτο στο σύστημα, και κατανέμει το υπολογιστικό φορτίο. Θεωρείται πως πρόκειται για μια κατά πολύ βελτιωμένη έκδοση της πρώτης μεθόδου, καθώς διατηρεί το πλεονέκτημα του διαμοιρασμού των υπολογισμών που απαιτούνται για την παραγωγή των καρτέ, και το επανξάνει κατανέμοντας επίσης τη συμπίεσή τους σε βίντεο.

V. ΣΥΜΠΕΡΑΣΜΑΤΑ

Κατόπιν έρευνας και πειραματισμού, φαίνεται πως είναι εφικτή μια αναμφισβήτητα αποδοτική επιτάχυνση της διαδικασίας παραγωγής μιας δυναμικής απεικόνισης φράκταλ, με τη χρήση μεθόδων της φιλοσοφίας της MapReduce. Η πρώτη πρότασή μας αποδείχθηκε λιγότερο αποδοτική, όμως μας παρείχε πολύτιμες πληροφορίες για τα μειονεκτήματά της. Επακολούθως, τροποποιώντας τα αδύναμά της σημεία, σχεδιάσαμε μια πραγματικά βελτιωμένη επόμενη μέθοδο, την οποία και ονομάσαμε Μέθοδο 3. Τελικά πετύχαμε τον αρχικό μας στόχο, οδηγούμενοι στο συμπέρασμα ότι η παραγωγή δυναμικών απεικονίσεων φράκταλ είναι συχνά μια, αναμφισβήτητα, απόλυτα και εύκολα παραλληλοποιήσιμη διαδικασία. Η συγκριτική ανάλυση των δύο μεθόδων και ο σχολιασμός των συμπερασμάτων που απορρέουν, βοηθούν στην ενίσχυση αυτού του συμπεράσματος.

Η πειραματική εφαρμογή των τελικών μεθόδων που αναπτύξαμε πραγματοποιείται σε μια Hadoop συστάδα

υπολογιστών, επιτυγχάνοντας έτσι πραγματική παραλληλία. Αντιμετωπίζει ταυτόχρονα ένα πρόβλημα που –αν και αρχικά φαίνεται αρκετά εξειδικευμένο– απαιτεί τις τελευταίες εξελίξεις σε εφαρμογές που αναζητούν ολοένα και αποδοτικότερες υλοποιήσεις. Είναι εξάλλου ένα πραγματικά από το και οπτικά συναρπαστικό παράδειγμα χρήσης μιας ανερχόμενης λογικής για τη βελτιστοποίηση προγενέστερων υλοποιήσεων.

Η συγκεκριμένη εργασία ασχολείται συγκεκριμένα με την παραγωγή μιας δυναμικής απεικόνισης. Είναι σαφές ότι υλοποίηση κατά τον τρόπο που προτείνουμε, εντούτοις, μπορεί να χρησιμοποιηθεί για την παραγωγή ενός υπερμεγέθους συνόλου εικόνων φράκταλ, ακόμα και χωρίς την ύπαρξη συνοχής μεταξύ τους για τη συγκρότηση ενός συμβατικού animation. Οι ανεξάρτητες και σαφώς ορισμένες παράμετροι προσδιορίζουν άλλωστε μοναδικά την κάθε εικόνα.

Ο απόλυτος ορισμός των εικόνων από τις παραμέτρους αυτές δρα ως περιοριστικός παράγοντας, αφού οι παράμετροι πρέπει να είναι ορισμένες εξ' αρχής, όμως προσφέρει ταυτόχρονα περισσότερες ιδέες. Σε επόμενα βήματα ανάπτυξης της μεθοδολογίας μας θα μπορούσαν να εξεταστούν τρόποι γενίκευσής της για άλλων ειδών φράκταλ, όπου, για παράδειγμα, η κάθε εικόνα προκύπτει ως εξέλιξη της αμέσως προηγούμενης της.

Είναι βέβαια φανερό πως και τα ίδια τα φράκταλ έχουν πολυποικίλες εφαρμογές. Άρα μια μέθοδος υπολογισμού εικονοστοιχειών φράκταλ και τελικά παραγωγής τους με τρόπο αποδοτικότερο από άλλες, σίγουρα μπορεί να βρει εφαρμογές σε τομείς όπως, ενδεικτικά, αυτός της ιατρικής απεικόνισης [7].

VI. ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα εργασία πραγματοποιήθηκε στα πλαίσια του μαθήματος «Προχωρημένα Θέματα Βάσεων Δεδομένων» και υλοποιήθηκε υπό την επίβλεψη του Αυγερινού Αραμπατζή, επ. Καθηγητή στο Δημοκρίτειο Πανεπιστήμιο Θράκης [avi@ee.duth.gr].

VII. ΑΝΑΦΟΡΕΣ

- [1] Briggs J. και Peat F.D., Ο ταραγμένος καθρέφτης, Αθήνα, Εκδόσεις Κάτοπτρο, 1996.
- [2] Feder J., Fractals, New York, Plenum Press, 1988.
- [3] Rajaraman A. και Ullman JD, Mining of Massive Datasets, Cambridge University Press, 2011
- [4] Book M., Parallel Fractal Image Generation: A Study of Generating Sequential Data With Parallel Algorithms, The University of Montana, Missoula, 2001
- [5] Li Wei, Fractal Image Compression Implemented on Hadoop Platform, Automation, Xi'an University of Posts and Telecommunications, 2014 (αγγλική μετάφραση)
- [6] Ibrahim M. and Krawczyk R., "Exploring the effect of direction on vector-based fractals." Bridges: Mathematical Connections in Art, Music, and Science, Bridges Conference, 2002
- [7] Losa, G. A., "Fractals and their contribution to biology and medicine." Medicographia 34.3 (2012): 364-74.