

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

***ΑΞΙΟΛΟΓΗΣΗ ΤΗΣ ΕΠΙΔΟΣΗΣ ΜΙΑΣ
ΠΟΛΥΠΥΡΗΝΗΣ ΜΗΧΑΝΗΣ ΜΕΣΩ ΤΗΣ
ΡΥΘΜΟΑΠΟΔΟΣΗΣ***

Ατομική Διπλωματική Εργασία

***ΑΞΙΟΛΟΓΗΣΗ ΤΗΣ ΕΠΙΔΟΣΗΣ ΜΙΑΣ
ΠΟΛΥΠΥΡΗΝΗΣ ΜΗΧΑΝΗΣ ΜΕΣΩ ΤΗΣ
ΡΥΘΜΟΑΠΟΔΟΣΗΣ***

Παρασκευή Θεοδώρου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΞΙΟΛΟΓΗΣΗ ΤΗΣ ΕΠΙΔΟΣΗΣ ΜΙΑΣ ΠΟΛΥΠΥΡΗΝΗΣ
ΜΗΧΑΝΗΣ ΜΕΣΩ ΤΗΣ ΡΥΘΜΟΑΠΟΔΟΣΗΣ

Παρασκευή Θεοδώρου

Επιβλέπων Καθηγητής
Pedro Trancoso

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Δεκέμβριος 2012

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Pedro Trancoso για την υποστήριξη και καθοδήγηση που μου προσέφερε κατά την διάρκεια της υλοποίησης της παρούσας εργασίας, καθώς και την επίτευξη μιας άψογης συνεργασίας. Επίσης θερμές ευχαριστίες προς τον διδάκτορα Παναγιώτη Πετρίδη και την ομάδα με τον Κωνσταντίνο Χριστοφή και Αντρέα Διαβαστό, για την υπερπολύτιμη βοήθεια τους καθ' όλη την διάρκεια της εκπόνησης αυτής της εργασίας.

Τέλος στην οικογένεια μου και στους φίλους και συμφοιτητές μου για την στήριξη που μου έδιναν καθ' όλη τη διάρκεια αυτής της χρονιάς.

Περίληψη

Η επίδοση της μηχανής είναι πολύ σημαντική στη σύγχρονη εποχή ειδικότερα για τις επιχειρήσεις εταιρίες που αναζητούν την μέγιστη επίδοση για τις δικές τους μηχανές server. Θέλουμε τις βάσεις δεδομένων να ανταποκρίνονται στις απαιτητικές ανάγκες των εφαρμογών με ογκώδη σύνολα δεδομένων, στην υψηλή υπολογιστική και I/O απαιτήσεις και για ένα μεγάλο αριθμό χρηστών. Στην εργασία αυτή μελετήθηκαν το σύστημα διαχείρισης βάσεων δεδομένων PostgreSQL και TPC-H Benchmark ο οποίος είναι ένας δείκτης υποστήριξης αποφάσεων που αποτελείται από ένα σύνολο επιχειρησιακού προσανατολισμού ad-hoc επερωτήσεις. Επίσης δημιουργία δύο μεγεθών βάσεων δεδομένων του ενός και των δέκα Gigabyte όπου θα πραγματοποιήσαμε τέσσερα πειράματα στις δύο αυτές βάσεις, με βασικό σενάριο για ένα πυρήνα και για τα υπόλοιπα πειράματα εισάγαμε τη χρήση του παραλληλισμού για δύο, τέσσερις και οχτώ πυρήνες. Στόχος της εργασίας αυτής ήταν ο συνδυασμός του συστήματος διαχείρισης βάσεων δεδομένων PostgreSQL με επερωτήσεις των TPC-H Benchmark που προσφέρουν αξιοπιστία στα αποτελέσματα και επίσης να αναλύσουμε και να αξιολογήσουμε τα αποτελέσματα από τα πειράματα στις δύο βάσεις και να δείξουμε τη σημασία του παραλληλισμού και τις τιμές της ρυθμοαπόδοσης για κάθε πείραμα ξεχωριστά και γενικότερα τους παράγοντες που επηρεάζουν την ρυθμοαπόδοση της μηχανής που εξαρτάται η επίδοση.

Αφού πραγματοποιήσαμε τα πειράματα, χρησιμοποιήσαμε τις μετρήσεις αυτές για να δημιουργήσουμε αντίστοιχες γραφικές παραστάσεις οι οποίες βοήθησαν στην αξιολόγηση της ρυθμοαπόδοσης και συγκρίνοντας τα αποτελέσματα μεταξύ των δύο βάσεων παρουσιάζουμε τους λόγους της συμπεριφοράς των τιμών στις γραφικές που επηρεάζουν τη ρυθμοαπόδοση. Οι λόγοι αυτοί είναι ο όγκος των δεδομένων στην βάση, ο αριθμός των πυρήνων που χρησιμοποιείται, το μέγεθος της κρυφής μνήμης των πυρήνων, ο συνδυασμός του σετ επερωτήσεων που εκτελείται σε κάθε πυρήνα και η κοινή μνήμη όπου μέσω του buffer μοιράζονται οι πληροφορίες μεταξύ των πυρήνων. Τέλος, με τις μετρήσεις της ρυθμοαπόδοσης από τις δύο βάσεις δεδομένων που δημιουργήθηκαν, δείξαμε ότι για τους οχτώ πυρήνες έχουμε τη μεγαλύτερη ρυθμοαπόδοση όμως έχουμε μεγάλη διαφορά στις τιμές της μεταξύ των πειραμάτων των δύο βάσεων, και η ρυθμοαπόδοση στους τέσσερις πυρήνες αρχίζει να μειώνεται σημαντικά έναντι της αναμενόμενης και στους οχτώ πυρήνες μειώνεται ακόμη περισσότερο.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	1
	1.1 Κίνητρο και Στόχος Διπλωματικής Εργασίας	1
	1.2 Μεθοδολογία	3
	1.3 Οργάνωση της Εργασίας	5
Κεφάλαιο 2	Υπόβαθρο.....	6
	2.1 PostgreSQL	6
	2.2 TPC-H Benchmark	7
	2.3 Ρυθμοαπόδοση	8
	2.4 Μηχανή Εκτέλεσης Πειραμάτων	9
Κεφάλαιο 3	Performance.....	6
	3.1 Performance	6
Κεφάλαιο 4	Βάση Δεδομένων Μεγέθους 1GB.....	13
	4.1 Περιγραφή Πειράματος	13
	4.2 Υλοποίηση Πειράματος	15
	4.3 Πειραματική Αξιολόγηση Πειράματος	17
	4.4 Συμπεράσματα	18
Κεφάλαιο 5	Βάση Δεδομένων Μεγέθους 10GB	20
	5.1 Περιγραφή Πειράματος	20
	5.2 Υλοποίηση Πειράματος	22
	5.3 Πειραματική Αξιολόγηση Πειράματος	24
	5.4 Συμπεράσματα	26

Κεφάλαιο 6	Σύγκριση Πειραμάτων.....	28
	6.1 Σύγκριση Πειραμάτων.....	28
	6.2 Συμπεράσματα.....	34
Κεφάλαιο 7	Επίλογος.....	28
	7.1 Γενικά Συμπεράσματα.....	36
	7.2 Μελλοντική Εργασία.....	38
	7.3 Οφέλη από τη Διπλωματική Εργασία.....	40
Βιβλιογραφία		80
Παράρτημα Α.....		A-1
Παράρτημα Β.....		B-3
Παράρτημα Γ.....		Γ-9

Κεφάλαιο 1

Εισαγωγή

1.1 Κίνητρο και Στόχος Διπλωματικής Εργασίας	1
1.2 Μεθοδολογία	3
1.3 Οργάνωση της Εργασίας	5

1.1 Κίνητρο και Στόχος Διπλωματικής Εργασίας

Όσο περνάνε τα χρόνια ο κόσμος χρειάζεται να γίνει πιο γρήγορος και πιο αποδοτικός σε οτιδήποτε καταπιάνεται, γιατί οι απαιτήσεις στην ζωή τους μεγαλώνουν και πρέπει να είναι σε διάφορους χώρους ταυτόχρονα ή γενικότερα να κάνουν διάφορες εργασίες ταυτόχρονα. Στον κόσμο της πληροφορικής συμβαίνει κάτι παρόμοιο. Όλοι αυτοί που ασχολούνται καθημερινά με τον δικό τους προσωπικό ή εταιρικό υπολογιστή, δουλεύουν ή απλά χρησιμοποιούν πολλές εφαρμογές ταυτόχρονα. Επίσης, έναν πολύ σημαντικό ρόλο διαδραματίζει και ο χρόνος ανταπόκρισης της κάθε εφαρμογής στα διάφορα ερωτήματα που μπορεί να τεθούν από τον χρήστη της εφαρμογής άμεσα ή έμμεσα.

Πλείστες εφαρμογές χρησιμοποιούν τις βάσεις δεδομένων. Οι βάσεις δεδομένων υπάρχουν και χρησιμοποιούνται ευρέως στη καθημερινότητα του κάθε ατόμου στην σύγχρονη κοινωνία. Υπάρχουν πολλά παραδείγματα παραδοσιακών εφαρμογών των βάσεων δεδομένων που μπορεί να το αποδεικνύουν αυτό, όπως όταν θέλουμε να αγοράσουμε ένα ρούχο είτε από ένα κατάστημα ρούχων είτε μέσω κάποιου προμηθευτή της web σελίδας στο διαδίκτυο που προμηθεύει ρούχα, είτε με την αγορά προϊόντων από ένα σούπερ μάρκετ που στο ταμείο γίνεται ενημέρωση της βάσης δεδομένων, όπου εκεί καταχωρούνται τα αποθέματα των προϊόντων του σούπερ μάρκετ αμέσως μετά την πληρωμή από τον πελάτη.

Οι πληροφορίες που αποθηκεύονται σε μία βάση δεδομένων μπορεί να είναι είτε σε μορφή κειμένου, είτε σε μορφή αριθμών, εικόνας, video και μηνυμάτων ήχου. Γενικότερα μία βάση δεδομένων είναι μία συλλογή από σχετιζόμενα δεδομένα παράδειγμα μία βάση δεδομένων

για ένα πανεπιστήμιο θα περιέχει αρχεία που στο κάθε αρχείο θα αποθηκευτούν δεδομένα που αφορούν το αρχείο αυτό. Το αρχείο φοιτητής θα περιέχει δεδομένα που αφορούν τον φοιτητή και το αρχείο βαθμολογία θα περιέχει τους βαθμούς των φοιτητών σε κάποιο μάθημα και ούτω καθεξής.

Στις εφαρμογές αυτές που χρησιμοποιούν βάσεις δεδομένων, αποθηκεύεται μεγάλος όγκος δεδομένων. Για να ανακτήσουμε συγκεκριμένα δεδομένα από τη βάση, οι εφαρμογές χρειάζεται να εκτελέσουν διάφορες επερωτήσεις (queries) προς τη βάση. Η απόδοση των επερωτήσεων επηρεάζει γενικότερα την απόδοση της εφαρμογής. Μερικοί παράγοντες που επηρεάζεται η απόδοση της εφαρμογής είναι ο όγκος των αποθηκευμένων δεδομένων στην βάση, ο αριθμός των πυρήνων (core) που χρησιμοποιούνται για την εκτέλεση των επερωτήσεων και τη σειρά της εκτέλεσης των επερωτήσεων στον κάθε πυρήνα.

Κίνητρο αυτής της διπλωματικής εργασίας είναι να μελετήσουμε σε ποιο βαθμό αυτοί οι παράγοντες που έχουν προαναφερθεί, κατά πόσο επηρεάζουν την ρυθμοαπόδοση μίας εφαρμογής.

Η διπλωματική αυτή εργασία στοχεύει στην πειραματική αξιολόγηση δύο βάσεων δεδομένων με όγκο πληροφοριών του ενός Gigabyte και δέκα Gigabyte, που με την εκτέλεση των επερωτήσεων σε διάφορο συγκεκριμένο καθορισμένο αριθμό πυρήνων, θα δειχθεί ο βαθμός που επηρεάζουν κάποιοι παράγοντες την απόδοση μίας εφαρμογής. Οι παράγοντες που θα μελετηθούν για αυτό το σκοπό είναι ο όγκος των δεδομένων που είναι τα αποθηκευμένα δεδομένα στην βάση, ο αριθμός των πυρήνων που θα χρησιμοποιηθούν για την εκτέλεση των επερωτήσεων και τέλος θα εξετασθεί και κατά πόσο η σειρά της εκτέλεσης των επερωτήσεων που εκτελούνται σε κάθε πυρήνα αλλάζει την απόδοση του χρόνου της εφαρμογής.

Το πρώτο πείραμα που θα υλοποιηθεί, θα είναι σε μία βάση δεδομένων με όγκο δεδομένων μεγέθους του ενός Gigabyte (1Gb). Αυτά τα δεδομένα θα χρησιμοποιηθούν από τις επερωτήσεις που θα εκτελούνται με προκαθορισμένη σειρά στους πυρήνες μίας μηχανής. Από την εκτέλεση των επερωτήσεων αυτών, θα έχουμε ως αποτελέσματα τους χρόνους που διάνυσαν οι επερωτήσεις να εκτελεστούν, δηλαδή τον συνολικό χρόνο που πήρε για να εκτελεστεί, η κάθε μία επερώτηση για να ανακτήσει τα συγκεκριμένα δεδομένα που ζητούσε από τη βάση αυτή.

Για το δεύτερο πείραμα θα δημιουργήσουμε μία άλλη βάση δεδομένων με όγκο δεδομένων μεγέθους των δέκα Gigabyte (10Gb). Τα δεδομένα της βάσης αυτής θα χρησιμοποιηθούν από τις ίδιες επερωτήσεις που είχαμε και για την βάση δεδομένων του ενός Gigabyte (1Gb). Οι επερωτήσεις θα εκτελεσθούν με την ίδια προκαθορισμένη σειρά στους πυρήνες όπως είχαν εκτελεσθεί και στο πρώτο πείραμα με την χρήση της ίδιας μηχανής. Με την ολοκλήρωση των επερωτήσεων αυτών, θα έχουμε τους χρόνους που ήθελε για να ολοκληρωθεί η εκτέλεση για την κάθε μία επερώτηση για να ανακτήσει τα συγκεκριμένα δεδομένα που ζητούσε από τη καινούργια βάση.

Ως στόχος λοιπόν αυτής της διπλωματικής εργασίας είναι αρχικά η μελέτη της γλώσσας που θα είναι γραμμένες οι επερωτήσεις που θα χρησιμοποιήσουμε, στη συνέχεια ο τρόπος πλοήγησης και τρόπος εκτέλεσης των επερωτήσεων στο περιβάλλον που χρησιμοποιεί η μηχανή και εν συνεχεία η μελέτη και αξιολόγηση των αποτελεσμάτων από τα δύο πειράματα. Σημαντικό μέρος της εργασίας είναι η κατανόηση και η εκμάθηση μίας νέας γλώσσας προγραμματισμού της PostgreSQL και τρόπου δημιουργίας βάσης δεδομένων, εκτέλεσης των επερωτήσεων και πλοήγησης εξ' αποστάσεως σε μία μηχανή με περιβάλλον Linux με τη χρήση της PostgreSQL και των επερωτήσεων των TPC-H Benchmarks. Μέσω των πειραμάτων στις βάσεις δεδομένων θα δείξουμε πώς ανταποκρίνονται στις απαιτητικές ανάγκες των εφαρμογών με ογκώδη σύνολα δεδομένων, στην υψηλή υπολογιστική και I/O απαιτήσεις.

1.2 Μεθοδολογία

Για την επίτευξη των στόχων της διπλωματικής αυτής εργασίας ακλούθησα την πιο κάτω μεθοδολογία :

Αρχικά μελέτησα διάφορα επιστημονικά άρθρα, κώδικες και οδηγούς χρήσης που είχαν μερικώς κάποιες αναφορές για το δικό μου θέμα της διπλωματικής εργασίας. Ακολούθως θα έπρεπε να επιλέξουμε μία κατάλληλη μηχανή για να πραγματοποιήσω τα πειράματα με τις δύο βάσεις δεδομένων με μεγέθη του ενός Gigabyte (1Gb) και δέκα Gigabyte (10Gb). Συνεπώς θα έπρεπε να επιλέξω μία μηχανή με αρκετό χώρο στη μνήμη για τα δεδομένα που θα είχα στις βάσεις δεδομένων και για τα δεδομένα που θα αποκόμιζα για τα αποτελέσματά μου, επίσης η μηχανή αυτή θα έπρεπε να αποτελείται τουλάχιστον από οχτώ πυρήνες (8 cores). Μετά από συζήτηση με τον επιβλέπων καθηγητή μου κ. Pedro Trancoso, συμφωνήσαμε πόσους πυρήνες κάποιας μηχανής θα χρησιμοποιήσω και μου δόθηκε η

κατάλληλη μηχανή. Ακολούθως έγινε η επιλογή της γλώσσας PostgreSQL που θα χρησιμοποιούσα αργότερα για την δημιουργία βάσεων δεδομένων και εκτέλεσης επερωτήσεων με την χρήση των βάσεων αυτών για να εξάγω τα αποτελέσματα που θα ήθελα για την ολοκλήρωση της διπλωματικής αυτής εργασίας. Επίσης υπάρχουν ακόμη δύο σημαντικοί λόγοι που επιλέχθηκε η γλώσσα PostgreSQL, είναι επειδή είναι γλώσσα ανοιχτού κώδικα, δηλαδή είναι δωρεάν η πρόσβαση και απόκτηση της και επίσης έπρεπε να είναι συμβατή στο περιβάλλον της μηχανής που συγκεκριμένα είναι για περιβάλλον των Linux. Αφού εγκαταστήσαμε την PostgreSQL με έκδοση της 8.4 στην μηχανή, χρησιμοποίησα έτοιμα δεδομένα με περιεχόμενα σε μορφή κειμένου και σε μορφή αριθμών. Τα δεδομένα αυτά είναι κατανεμημένα σε οχτώ πίνακες, συνολικά μεγέθους του ενός Gigabyte (1Gb). Ακολούθως δημιούργησα μία βάση δεδομένων μεγέθους του ενός Gigabyte (1Gb) στην μηχανή που μου δόθηκε. Και πέρασα τα δεδομένα των πινάκων αυτών μέσα στους πίνακες της βάσης δεδομένων που δημιούργησα. Οι επερωτήσεις που χρησιμοποίησα, εκτελέστηκαν σύμφωνα με καθορισμένη σειρά που αναφέρεται στον οδηγό στην σελίδα των TPC-H Benchmarks για την αξιολόγηση του throughput μιας μηχανής. Από τις είκοσι δύο επερωτήσεις που υπήρχαν, μερικές επερωτήσεις τις έχω αφαιρέσει, γιατί μετά από τα πρώτα πειράματα που είχα κάνει οι τιμές των χρόνων εκτέλεσης τους διέφεραν κατά πολύ από τις υπόλοιπες έτσι αποφασίσαμε να τις αφαιρέσω και να εκτελέσω τα πειράματα μου με τις υπόλοιπες επερωτήσεις σύνολο δεκαεπτά. Έκανα την πειραματική αξιολόγηση με τις καθορισμένες σειρές επερωτήσεων που αναφέρονται στον οδηγό των TPC-H Benchmarks, σε ένα πυρήνα, για δύο πυρήνες, για τέσσερις και οχτώ. Δηλαδή τέσσερα συνολικά πειράματα για την βάση δεδομένων μεγέθους του ενός Gigabyte (1Gb). Το πρώτο πείραμα ήταν εκτέλεση των δεκαεπτά επερωτήσεων με την καθορισμένη σειρά μέσω ενός script που είχα δημιουργήσει. Ακολούθως για τους δύο πυρήνες έχουμε στον πρώτο πυρήνα την ίδια σειρά που είχα και στο πρώτο πείραμα με ένα πυρήνα και στον δεύτερο πυρήνα έχουμε διαφορετική σειρά των δεκαεπτά επερωτήσεων πάλι όπως αναγράφεται στον πίνακα (Tables of Random Permutations) του οδηγού των TPC-H Benchmarks. Η εκτέλεση των επερωτήσεων και για τους δύο πυρήνες ξεκινά ταυτόχρονα. Επίσης με παρόμοιο τρόπο και για τα πειράματα με τέσσερις και οχτώ πυρήνες. Όταν ολοκληρώθηκε η εκτέλεση των επερωτήσεων σε όλους τους πυρήνες και για τα τέσσερα πειράματα, άλλαξα τους πίνακες της βάσης με δεδομένα μεγέθους των δέκα Gigabyte (1Gb). Ακολούθως πραγματοποίησα τα ίδια πειράματα με τον ίδιο ακριβώς τρόπο όπως και για την βάση του ενός Gigabyte (1Gb). Και ως αποτελέσματα για όλα τα πειράματα και για τις δύο βάσεις, είχα τα διάφορα αρχεία με τους χρόνους που έκαναν για να ολοκληρωθούν οι δεκαεπτά επερωτήσεις (17 queries) για κάθε πυρήνα και τα διάφορα αρχεία με τις εξόδους των δεδομένων που ζητήθηκαν από τις επερωτήσεις από τους πίνακες της βάσης. Ακολούθησε η ανάλυση των τιμών των χρόνων

και δημιουργία διάφορων γραφικών παραστάσεων που με οδήγησαν στα αποτελέσματα να αποφανθώ για τον βαθμό που οι παράγοντες επηρεάζουν την απόδοση της εφαρμογής συγκεκριμένα οι παράγοντες αυτοί είναι ο όγκος των αποθηκευμένων δεδομένων στην βάση, ο αριθμός των πυρήνων που χρησιμοποιούνται για την εκτέλεση των επερωτήσεων και η σειρά της εκτέλεσης των επερωτήσεων στον κάθε πυρήνα. Διαχωρισμός των επερωτήσεων σε κατηγορίες κατά πόσο επηρεάζονται ή όχι. Τέλος σύγκριση των πειραμάτων μεταξύ των, για τις δύο βάσεις δεδομένων και παρουσίαση των γενικών συμπερασμάτων.

1.3 Οργάνωση της Εργασίας

Η εργασία αυτή αποτελείται από επτά κεφάλαια. Το επόμενο κεφάλαιο, Κεφάλαιο 2, ασχολείται με το υπόβαθρο της εργασίας δηλαδή με τη γενική περιγραφή της PostgreSQL, τα TPC-H Benchmarks, λίγα λόγια για τον πυρήνα και τους επεξεργαστές με περισσότερους από ένα πυρήνα και εν συνεχεία τα χαρακτηριστικά της μηχανής που χρησιμοποίησα για την ολοκλήρωση της διπλωματικής εργασίας αυτής. Στο Κεφάλαιο 3 αναφέρουμε λίγα λόγια για τον παραλληλισμό που βοηθάει στην καλύτερη επίδοση. Στο Κεφάλαιο 4 αρχίζει η υλοποίηση πειραμάτων και πειραματική αξιολόγηση που καταπιάνεται με μία βάση δεδομένων μεγέθους του ενός Gigabyte (1Gb). Ακολουθεί η περιγραφή, η υλοποίηση και η πειραματική αξιολόγηση με τα συμπεράσματα για την βάση των δέκα Gigabyte (10Gb), στο Κεφάλαιο 5. Σύγκριση μεταξύ των αποτελεσμάτων των δύο βάσεων στο Κεφάλαιο 6 και η εργασία κλείνει με το Κεφάλαιο 7 όπου περιγράφονται τα γενικά συμπεράσματα της εργασίας αυτής καθώς και μελλοντικές εργασίες.

Κεφάλαιο 2

Υπόβαθρο

2.1 PostgreSQL	6
2.2 TPC-H Benchmark	7
2.3 Ρυθμοαπόδοση	8
2.4 Μηχανή Εκτέλεσης Πειραμάτων	9

2.1 PostgreSQL

Η PostgreSQL είναι ένα αντικειμενοστραφές σύστημα διαχείρισης βάσεων δεδομένων, χαρακτηρίζεται από την αξιοπιστία, την ακεραιότητα των δεδομένων και την ορθότητα. Μπορεί να τρέξει σε όλα τα σημαντικά λειτουργικά συστήματα όπως Linux, Unix και Windows, εμείς για τη έρευνα αυτή χρησιμοποιήσαμε το σύστημα διαχείρισης βάσεων δεδομένων PostgreSQL σε λειτουργικό Linux. Είναι πλήρως συμβατή με την ACID, με τον όρο αυτό εννοούμε την ατομικότητα, τη συνέπεια, την απομόνωση και τη διάρκεια που αποτελούν ένα σύνολο ιδιοτήτων που εγγυώνται ότι οι συναλλαγές στη βάση δεδομένων επεξεργάζονται αξιόπιστα. Στην PostgreSQL εμπεριέχεται ένα μεγάλο μέρος τύπων δεδομένων της SQL. Για αυτό το λόγο θα χρησιμοποιήσουμε μαζί με τη PostgreSQL, τις επερωτήσεις των TPC-H Benchmark που είναι γραμμένες σε γλώσσα SQL, έτσι ώστε να μην κάνουμε πολλές αλλαγές στις επερωτήσεις και για άλλους λόγους που θα επεξηγηθούν αργότερα στο υποκεφάλαιο 2.2. Χρησιμοποιήσαμε για τη μελέτη αυτή, βασισμένοι στην τελευταία έκδοση της PostgreS, την έκδοση 4.2, θα χρησιμοποιήσουμε την έκδοση PostgreSQL 8.4, [1]. Η PostgreSQL είναι ανοιχτού κώδικα, επεκτάσιμη και διαθέτει εξελιγμένα χαρακτηριστικά όπως πολλές εκδοχές για τον έλεγχο ταυτοχρονισμού, που μπορεί να φιλοξενήσει και να διαχειριστεί αυτό τον αριθμό των ταυτόχρονων χρηστών. Υπάρχουν ενεργά συστήματα PostgreSQL στις επιχειρήσεις που διαχειρίζονται όγκο δεδομένων που ξεπερνάει και τα 4 Terabytes. Ορισμένα γενικά όρια PostgreSQL φαίνονται στον πιο κάτω πίνακα 2.1.

Limit	Value
Maximum Database Size	Unlimited
Maximum Table Size	32 TB
Maximum Row Size	1.6 TB
Maximum Field Size	1 GB
Maximum Rows per Table	Unlimited
Maximum Columns per Table	250 - 1600 depending on column types
Maximum Indexes per Table	Unlimited

Πίνακας 2.1 Γενικά όρια ενός συστήματος PostgreSQL

2.2 TPC-H Benchmark

TPC-H Benchmark είναι ένας δείκτης υποστήριξης αποφάσεων, αποτελείται από ένα σύνολο επιχειρησιακού προσανατολισμού ad-hoc επερωτήσεις και ταυτόχρονη τροποποίηση των δεδομένων. Με τον όρο ad-hoc εννοούμε ότι το σύστημα επιτρέπει στους χρήστες του να δημιουργούν συγκεκριμένες προσαρμοσμένες επερωτήσεις. Ο δείκτης παρουσιάζει τα συστήματα υποστήριξης αποφάσεων που εξετάζουν μεγάλο όγκο δεδομένων, εκτελούν επερωτήσεις με υψηλό βαθμό πολυπλοκότητας και δίνει απαντήσεις σε κρίσιμα ερωτήματα των επιχειρήσεων. Για τη δική μας μελέτη χρησιμοποιήσαμε δεκαεπτά επερωτήσεις (queries), τις q1, q3, q4, q5, q6, q7, q8, q9, q10, q11, q12, q13, q14, q15, q16, q18 και q19 από το TPC-H Benchmark suite [3]. Οι επερωτήσεις αυτές ανήκουν σε μια μεγάλη οικογένεια δοκιμαστικών βάσεων δεδομένων που χρησιμοποιούν τις επερωτήσεις. Οι βάσεις αυτές χρησιμοποιούνται από διάφορες εταιρίες και οργανισμούς για την αξιολόγηση υπολογιστικών συστημάτων. Επειδή αποτελεί ένα έγκυρο και αξιόπιστο μέσο για να έχουμε σωστά και αξιόπιστα αποτελέσματα, το χρησιμοποιήσαμε στην μελέτη αυτή.

Ένα παράδειγμα μίας επερώτησης από το TPC-H Benchmark suite που είναι στην αρχική της μορφή σε SQL δίνεται πιο κάτω στο σχήμα 2.1.

```
select
    l_returnflag,
    l_linestatus,
    sum(l_quantity) as sum_qty,
    sum(l_extendedprice) as sum_base_price,
    sum(l_extendedprice * (1 - l_discount)) as sum_disc_price,
    sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) as
sum_charge,
    avg(l_quantity) as avg_qty,
```

```

        avg(l_extendedprice) as avg_price,
        avg(l_discount) as avg_disc,
        count(*) as count_order
from
    lineitem
where
    l_shipdate <= date '1998-12-01' - interval ':1' day (3)
group by
    l_returnflag,
    l_linestatus
order by
    l_returnflag,
    l_linestatus;

```

Σχήμα 2.1: TPC-H Query 1 σε SQL

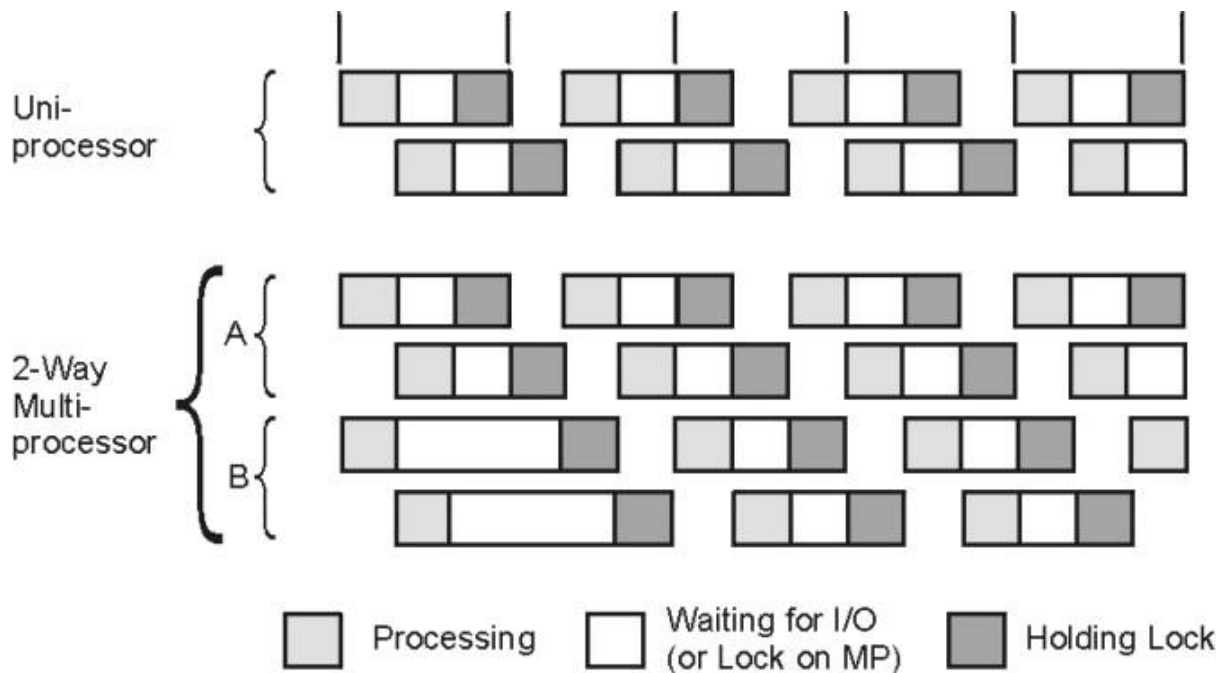
Για τη δική μας μελέτη θα έπρεπε να κάνουμε κάποιες μικρές αλλαγές ώστε να μπορούμε να τρέξουμε τις επερωτήσεις αυτές, αργότερα στην βάση δεδομένων χρησιμοποιώντας PostgreSQL γλώσσα προγραμματισμού. Η βάση δεδομένων με τις εγγραφές (records) που χρειάζονται οι επερωτήσεις για να εκτελεστούν είναι αποθηκευμένη σε αρχεία κειμένου και σε αναπαράσταση χαρακτήρων.

2.3 Ρυθμοαπόδοση

Η επεκτασιμότητα της ρυθμοαπόδοσης (throughput) πολλών επεξεργαστών εξαρτάται από πολλούς παράγοντες. Για ένα σύστημα SMP (symmetric multiprocessing εννοούμε ένα υπολογιστή με δύο ή περισσότερους πανομοιότυπους επεξεργαστές που είναι ενωμένοι σε μια κοινή διαμοιραζόμενη μνήμη και πως ελέγχονται από ένα λειτουργικό σύστημα) σε πραγματικό φόρτο εργασίας του συστήματος δεν έχουμε την τέλεια κλιμάκωση. Μερικοί παράγοντες για αυτό που συμβαίνει είναι η αυξημένη σύγκρουση της πρόσβασης σε κοινόχρηστους πόρους στην μνήμη διότι είναι κοινή η μνήμη για όλους τους επεξεργαστές, ένας άλλος παράγοντας είναι η αύξηση του Bus/switch contention ενόσω αυξάνεται ο αριθμός των επεξεργαστών που χρησιμοποιούνται, επίσης ενόσω τα δεδομένα στη μνήμη αυξάνονται έχουμε μεγαλύτερο κόστος για cache misses. Έχουμε περισσότερα cache misses διότι έχουμε περισσότερες διεργασίες για να αποσταλούν στο σύστημα και λόγω του μεγαλύτερου λειτουργικού συστήματος και των δομών δεδομένων της εφαρμογής. Επίσης επειδή μεγαλώνει το κόστος για τις οδηγίες να συγχρονιστούν.

Ένα παράδειγμα έχουμε στο σχήμα 2.2 [4] για την ίδια χρονική στιγμή έχουμε για τον μονό επεξεργαστή περίπου 8 οδηγίες (7,67) να εκτελούνται ενώ για δύο επεξεργαστές έχουμε 14 οδηγίες να εκτελούνται. Αυτό συμβαίνει επειδή στο μονό επεξεργαστή (uni-processor) γίνεται επεξεργασία μίας οδηγίας (request), ενώ για τους δύο επεξεργαστές (2 way multi-processor) έχουμε παράλληλη εκτέλεση δύο ερωτημάτων κάθε φορά. Και η ρυθμοαπόδοση (throughput factor) θα είναι ίση με το πηλίκο του αριθμού οδηγιών στους δύο

πυρήνες με τον αριθμό οδηγιών για τον ένα πυρήνα για την ίδια χρονική στιγμή (δηλαδή $14/7,67=1,82$)



Σχήμα 2.2: Υποθετικές οδηγίες προς εκτέλεση σε ένα πρόγραμμα

Αναμένεται να έχουμε μεγαλύτερη ρυθμοαπόδοση(throughput) από το 1,82 για πολυεπεξεργαστές με τέσσερις είτε οχτώ πυρήνες για την ίδια χρονική στιγμή.

Στη μελέτη μας χρησιμοποιώντας τα TPC-H Benchmark θα δείξουμε τα αποτελέσματα της ρυθμοαπόδοσης σε πραγματική βάση δεδομένων.

2.4 Μηχανή Εκτέλεσης Πειραμάτων

Η μηχανή που θα χρησιμοποιηθεί για την εκτέλεση των πειραμάτων στη μελέτη αυτή, αποτελείται από δώδεκα επεξεργαστές. Με την χρήση των TPC-H Benchmarks μέσω του συστήματος διαχείρισης βάσης δεδομένων PostgreSQL, θα εκτελούμε τις ερωτήσεις (queries) με μια συγκεκριμένη καθορισμένη σειρά σύμφωνα με το TPC-H Benchmark suite [3], στους πυρήνες της μηχανής αυτής και θα λαμβάνουμε τις διάφορες μετρήσεις του χρόνου. Η επίδοση (performance) θα είναι η επίδοση που έχω από τα ολοκληρωμένα ερωτήματα (query requests) στην βάση σε σχέση με τον χρόνο (query / t). Η μηχανή είναι σε Linux περιβάλλον έτσι θα μπορούμε να εγκαταστήσουμε τη PostgreSQL και να δημιουργήσουμε βάσεις δεδομένων για την εκτέλεση των ερωτήσεων.

Χαρακτηριστικά της μηχανής αναφέρονται πιο αναλυτικά στο παράρτημα Β, μερικά από αυτά είναι τα πιο κάτω:

Για ένα επεξεργαστή (ή αλλιώς πυρήνας) έχουμε μέγεθος της κρυφής μνήμης (cache size) 512 KB, κεντρική μονάδα επεξεργασίας ίση με 800.000 MHz. Συνολικά για ολόκληρη τη μηχανή (όλους τους επεξεργαστές και τους δώδεκα) έχουμε μέγεθος της κρυφής μνήμης ίσο με 6144 KB και συνολική μνήμη 33015724 KB.

Κεφάλαιο 3

Performance

3.1 Performance

10

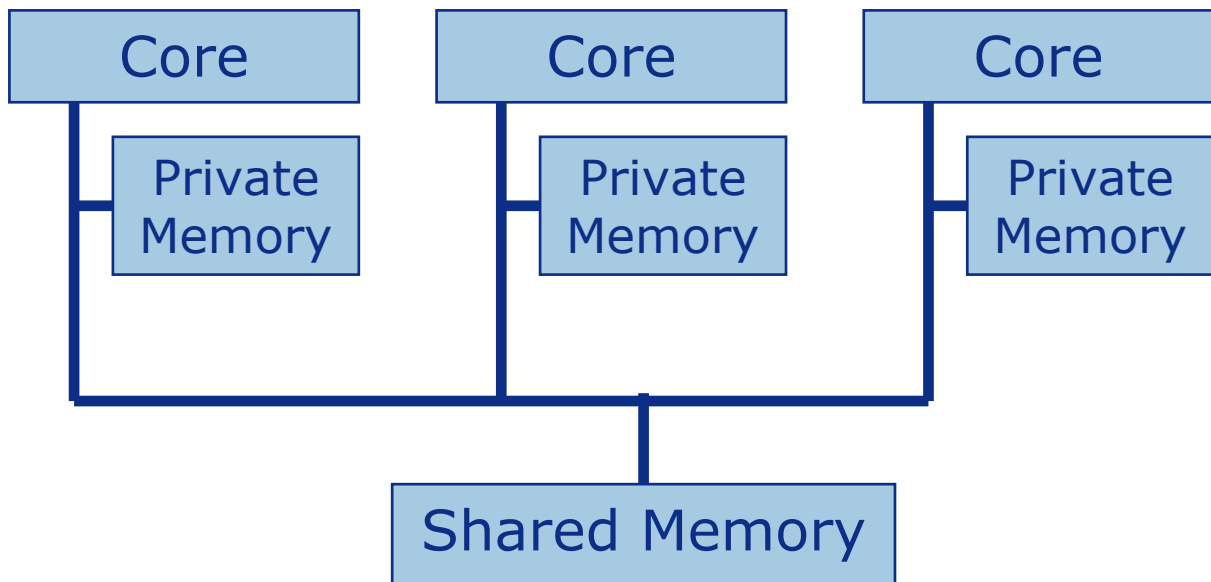
3.1 Performance

Η επίδοση (performance) μιας μηχανής είναι τα ολοκληρωμένα ερωτήματα (query requests) στην βάση σε σχέση με τον χρόνο (query / t). Ταυτοχρονισμός είναι όταν έχω δύο ή περισσότερες διεργασίες σε εξέλιξη την ίδια στιγμή. Ενώ παραλληλισμός είναι όταν έχω δύο ή περισσότερες διεργασίες να εκτελούνται την ίδια στιγμή. Επίσης για να υπάρχει παραλληλισμός θα πρέπει να έχουμε περισσότερους από ένα πυρήνες.

Για να βελτιώσω την επίδοση[5] θα πρέπει να χρησιμοποιήσω τον παραλληλισμό έτσι ώστε να βελτιώσω τη ρυθμοαπόδοση και την ολοκλήρωση μίας διεργασίας σε πολύ μικρό χρόνο. Στην έρευνα αυτή θα ασχοληθούμε με τη ρυθμοαπόδοση που είναι η ολοκλήρωση πολλών διεργασιών σε μια συγκεκριμένη σταθερή χρονική στιγμή.

Για το παραλληλισμό θα πρέπει να κοιτάξουμε αρχικά δύο τομείς, την αποσύνθεση πεδίου και την αποσύνθεση διεργασιών. Για την αποσύνθεση πεδίου αποφασίζουμε ποιες διεργασίες θα εκτελεστούν και από ποιους επεξεργαστές. Και αργότερα στην αποσύνθεση διεργασιών αποφασίζουμε ποιοι πυρήνες θα εκτελέσουν σε προκαθορισμένη σειρά των διεργασιών.

Στην έρευνα μας που θα ασχοληθούμε μέχρι και τους οχτώ πυρήνες, για κάθε πυρήνα γνωρίζουμε ότι έχει την δική του κρυφή μνήμη και μία κοινή διαμοιραζόμενη μνήμη με τους υπόλοιπους πυρήνες που θα εκτελείται, όπως φαίνεται στο πιο κάτω σχήμα 3.1.



Σχήμα 3.1: Διαμοιραζόμενη κοινή μνήμη

Οι διεργασίες που εκτελούνται παράλληλα συνεργάζονται έτσι ώστε να έχουμε καλύτερο συνολικό χρόνο εκτέλεσης, με συγχρονισμό και μοιράζονται πληροφορίες. Μια διαδικασία εκτελείται από το λειτουργικό σύστημα ως νήμα (thread).

Κεφάλαιο 4

Βάση Δεδομένων Μεγέθους 1GB

4.1 Περιγραφή Πειράματος	14
4.2 Υλοποίηση Πειράματος	15
4.3 Πειραματική Αξιολόγηση Πειράματος	17
4.4 Συμπεράσματα	25

4.1 Περιγραφή Πειράματος

Για την βάση του ενός Gigabyte (1Gb) θα πραγματοποιήσουμε τέσσερα πειράματα συνολικά. Τέσσερα πειράματα διότι θα ασχοληθούμε για το πρώτο πείραμα με ένα πυρήνα που θα είναι και το βασικό μας σενάριο, στο δεύτερο πείραμα με δύο πυρήνες, στο τρίτο πείραμα με τέσσερις πυρήνες και στο τέταρτο και τελευταίο πείραμα για αυτή την βάση δεδομένων μεγέθους του ενός Gigabyte (1Gb) θα ασχοληθούμε με τους οχτώ πυρήνες της μηχανής. Με την βοήθεια των επερωτήσεων (queries) των TPC-H Benchmarks που θα εκτελεστούν στους πυρήνες της μηχανής που έχει επιλεγεί για την διπλωματική εργασία αυτή, θα μας δώσουν σαν αποτελέσματα στην έξοδο δύο αρχεία που έχουμε ορίσει για να έχουμε για την κάθε επερώτηση, το ένα αρχείο θα περιέχει την απάντηση της επερώτησης (query), δηλαδή πληροφορίες που ικανοποιούν την κάθε επερώτηση που θα ανακτήσει από τη βάση μέσω των πινάκων της βάσης, και το δεύτερο αρχείο θα είναι ο χρόνος εκτέλεσης της κάθε επερώτησης που έχει ολοκληρώσει την ανάκτηση των πληροφοριών που ζητούσε από τη βάση δεδομένων. Με τα αποτελέσματα αυτά που θα μας δοθούν από τις εκτελέσεις των πειραμάτων θα δημιουργήσουμε γραφικές παραστάσεις που να δείχνουν τον χρόνο εκτέλεσης των επερωτήσεων στους πυρήνες (cores). Ακολούθως θα αναλύσουμε και θα μελετήσουμε την συμπεριφορά των χρόνων αυτών. Ανάλογα με την συμπεριφορά των χρόνων αυτών θα χωριστούν σε κατηγορίες. Οι κατηγορίες αυτές θα αναφέρονται για τον βαθμό που οι παράγοντες επηρεάζουν την απόδοση της εφαρμογής συγκεκριμένα οι παράγοντες αυτοί είναι ο όγκος των αποθηκευμένων δεδομένων στην βάση, ο αριθμός των πυρήνων που χρησιμοποιούνται για την εκτέλεση των επερωτήσεων και η σειρά της

εκτέλεσης των επερωτήσεων στον κάθε πυρήνα. Τέλος θα δώσουμε γενικά συμπεράσματα για τις εκτελέσεις των επερωτήσεων και για τα τέσσερα πειράματα που έχουν διεξαχθεί.

4.2 Υλοποίηση Πειράματος

Αφού έχουμε δημιουργήσει την καινούργια βάση με οχτώ πίνακες στην μηχανή που δουλεύουμε εξ' αποστάσεως, συνεχίζουμε με το να γεμίσουμε τους πίνακες αυτούς με τα δεδομένα με τους πιο κάτω τρόπους και εντολές:

Για να διαγράψουμε τυχόν δεδομένα που μπορεί να υπήρχαν στους πίνακες εκτελούμε αρχικά την εντολή `psql`, για να μπούμε στην βάση, μετά αφού είμαστε στην βάση έχουμε την εντολή:

```
<όνομα της βάσης> => SELECT * from <όνομα του πίνακα> ;
```

Με αυτή την εντολή βλέπω εάν υπάρχουν τυχόν εγγραφές στον πίνακα που έχουμε επιλέξει να δούμε τα περιεχόμενά του, εάν υπάρχει τότε εκτελούμε μία άλλη εντολή την:

```
<όνομα της βάσης> =>DELETE FROM <όνομα του πίνακα> ;
```

όπου θα διαγραφούν οι οποιεσδήποτε τυχόν εγγραφές υπήρχαν μέσα στον πίνακα που έχουμε επιλέξει για να διαγράψουμε τα περιεχόμενα του. Ακολούθως βγαίνουμε εκτός της βάσης με το να γράψουμε

```
<όνομα της βάσης> => \q
```

Αυτή την εντολή την βάζουμε επειδή οι πίνακες που έχουμε μεταφέρει στην μηχανή, βρίσκονται μέσα στον φάκελο που έχουμε πρόσβαση ως ιδιοκτήτες του φακέλου αυτού. Μετά εκτελούμε την εντολή :

```
<όνομα ιδιοκτήτη>@<όνομα μηχανής>::~$ cat <όνομα πίνακα>.tbl | psql -c "COPY <όνομα πίνακα> FROM STDIN DELIMITER AS '|' "
```

Με αυτή την εντολή έχουμε αντιγράψει όλα τα περιεχόμενα από τον πίνακα που έχουμε στον φάκελο εκτός της βάσης μέσα στον πίνακα της βάσης έχοντας τα περιεχόμενα ως τις εγγραφές στον πίνακα της βάσης που επιλέξαμε. Η πρώτη εγγραφή του πίνακα της βάσης αντιστοιχεί με τα περιεχόμενα των πρώτων γραμμών μέχρι τον πρώτο χαρακτήρα '|' του πίνακα που βρισκόταν εκτός της βάσης. Επαναλαμβάνουμε τις εντολές αυτές για όλους τους πίνακες που θέλουμε να μεταφέρουμε τα δεδομένα τους, ως τις εγγραφές των πινάκων της βάσης μας. Όταν έχουμε τελειώσει με το γέμισμα των πινάκων της βάσης, τότε συνεχίζουμε με την ετοιμασία αρχείων τύπου shell script (SH), δηλαδή script αρχεία που το κάθε ένα θα περιέχει δεκαεπτά επερωτήσεις προς εκτέλεση. Παράδειγμα ενός τέτοιου αρχείου είναι :

```
#!/bin/bash
```

```

for i in 1 2 3 4 5
do
(time psql name_of_database < /home/.../< file_name_of_owner>/queries_old/q3.sql) 1>>
output_q3.txt 2>> time_q3.txt
done

```

Με τον πιο πάνω κώδικα ως παράδειγμα, θα εκτελεστεί η επερώτηση τρία (q3.sql) από τον φάκελο με όνομα queries_old, όπου υπάρχουν αποθηκευμένες όλες οι επερωτήσεις. Η βάση που χρησιμοποιείται είναι η name_of_database. Οι απαντήσεις στην επερώτηση τρία που ανακτούνται με τη χρήση των πινάκων της βάσης name_of_database, αποθηκεύονται στο αρχείο κειμένου output_q3.txt και ο χρόνος εκτέλεσης της επερώτησης αποθηκεύεται στο αρχείο κειμένου time_q3.txt, αυτή η διαδικασία γίνεται πέντε επαναλαμβανόμενες φορές. Δηλαδή στο αρχείο κειμένου output_q3.txt θα έχω πέντε φορές τις ίδιες απαντήσεις αλλά στο αρχείο κειμένου time_q3.txt θα έχω πέντε χρόνους που μπορεί να είναι διαφορετικοί μεταξύ τους. Για παράδειγμα οι χρόνοι πιο κάτω:

```

real  0m9.420s
user  0m0.100s
sys   0m0.050s

```

```

real  0m9.497s
user  0m0.140s
sys   0m0.000s

```

```

real  0m9.519s
user  0m0.120s
sys   0m0.020s

```

```

real  0m9.514s
user  0m0.090s
sys   0m0.040s

```

```

real  0m9.514s
user  0m0.120s
sys   0m0.030s

```

Με τους χρόνους real εννοούμε τους πραγματικούς χρόνους που έκανε μία επερώτηση να ολοκληρώσει την ανάκτηση των δεδομένων που ζητούσε από τη βάση. Με τους χρόνους user έχουμε τους χρόνους που εργάζονταν οι χρήστες της μηχανής την δεδομένη στιγμή που η συγκεκριμένη επερώτηση προσπαθούσε να ολοκληρωθεί. Και με τους χρόνους sys εννοούμε τους χρόνους που δούλεψε η μηχανή τη συγκεκριμένη δεδομένη στιγμή. Και ακολούθως τρέχω το nameOfscript.sh με την εντολή :

```
screen bash ./nameOfscript.sh
```

ο λόγος που έχουμε τη λέξη screen είναι επειδή τρέχουμε εξ' αποστάσεως οτιδήποτε, θέλουμε να συνεχίσει να εκτελείται το αρχείο τύπου shell script με όνομα nameOfscript.sh, ακόμη και όταν εμείς θα έχουμε απενεργοποιήσει τον προσωπικό μας υπολογιστή.

Για το πρώτο πείραμα της βάσης αυτής για ένα πυρήνα, εμείς θα έχουμε επαναλήψεις των είκοσι για καλύτερα αποτελέσματα και θα εκτελούνται οι δεκαεπτά επερωτήσεις σε κάθε επανάληψη, σύμφωνα με την σειρά που έχει καθοριστεί από τα TPC-H Benchmarks (δες Appendix A). Μετά για το δεύτερο πείραμα θα έχω δύο αρχεία τύπου shell script για τον πρώτο πυρήνα θα έχω ότι είχα και για το αρχείο τύπου shell script που είχα χρησιμοποιήσει στο πρώτο πείραμα και για τον δεύτερο πυρήνα θα δημιουργήσω καινούργιο αρχείο τύπου shell script με διαφορετική σειρά εκτέλεσης των επερωτήσεων όπως επίσης έχει καθοριστεί από τα TPC-H Benchmarks. Εκτελώ δύο φορές την εντολή για να εκτελεστούν τα αρχεία τύπου shell script για δύο πυρήνες ταυτόχρονα. Παρόμοια για το τρίτο πείραμα με τέσσερις πυρήνες και για το τέταρτο πείραμα με του οχτώ πυρήνες.

4.3 Πειραματική Αξιολόγηση Πειράματος

Μεθοδολογία Πειραματικής Αξιολόγησης

Στο πρώτο πείραμα που πραγματοποιείται μόνο για έναν πυρήνα θα εξαχθούν συνολικά δεκαεπτά αρχεία κειμένου με τα αποτελέσματα των επερωτήσεων, δηλαδή για κάθε τέτοιο αρχείο έχουμε τις εγγραφές που απαντούν την επερώτηση μέσω των πινάκων της βάσης δεδομένων που έχουμε αρχικά δημιουργήσει. Επίσης θα εξαχθούν και δεκαεπτά αρχεία κειμένου που θα υπάρχουν οι χρόνοι που έκαναν να ολοκληρωθούν οι δεκαεπτά επερωτήσεις και σε κάθε τέτοιο αρχείο έχουμε είκοσι χρόνους της εκτέλεσης της επερώτησης. Ο λόγος που η επερώτηση έχει εκτελεσθεί είκοσι φορές είναι επειδή στην μηχανή μπορούν να δουλεύουν και άλλοι χρήστες και επίσης για να έχουμε πιο σωστά αποτελέσματα. Παρομοίως και για τα άλλα τρία πειράματα που πραγματοποιήθηκαν στη βάση.

Αποτελέσματα Πειραματικής Αξιολόγησης

Με τις μετρήσεις των χρόνων που έχουμε στα αρχεία κειμένου με τους χρόνους της κάθε επερώτησης αναλύουμε για να φτιάξουμε τις γραφικές παραστάσεις με τα τελικά μας αποτελέσματα. Πιο κάτω στον πίνακα 4.1 βλέπουμε τις τελικές μετρήσεις για όλα τα

πειράματα που έχουν πραγματοποιηθεί στην βάση δεδομένων του ενός Gigabyte (1Gb). Ο χρόνος εκτέλεσης είναι σε λεπτά. Ο μέσος χρόνος (Average) είναι ο μέσος όρος από τους ολικούς μέγιστους χρόνους που έχουμε από τις είκοσι εκτελέσεις όλων των επερωτήσεων. Με τον ολικό μέγιστο χρόνο σε μία εκτέλεση του πρώτου πειράματος σε ένα πυρήνα(1core) εννοούμε τον μέγιστο συνολικά χρόνο που είναι ο ίδιος με το άθροισμα των δεκαεπτά τελικών χρόνων εκτέλεσης των δεκαεπτά επερωτήσεων. Για το δεύτερο πείραμα με δύο πυρήνες έχουμε τα αποτελέσματα των είκοσι εκτελέσεων των δεκαεπτά επερωτήσεων για τον πρώτο πυρήνα και τα αποτελέσματα των είκοσι εκτελέσεων των δεκαεπτά επερωτήσεων για τον δεύτερο πυρήνα. Αθροίζουμε για κάθε εκτέλεση των δεκαεπτά επερωτήσεων τους χρόνους εκτέλεσής τους για τη συγκεκριμένη εκτέλεση. Παρομοίως πράττουμε και για τη κάθε εκτέλεση των δεκαεπτά επερωτήσεων στο δεύτερο πυρήνα. Ακολουθώς παίρνουμε το μέγιστο χρόνο από τα αθροίσματα των δύο πυρήνων για την συγκεκριμένη εκτέλεση και τέλος έχουμε τον χρόνο με τον μέσο όρο από τους μέγιστους χρόνους και των είκοσι εκτελέσεων. Για το τρίτο πείραμα με τέσσερις πυρήνες και για το τέταρτο πείραμα με οχτώ πυρήνες πράττουμε παρομοίως με το δεύτερο πείραμα. Με τη γραμμή plus στον πίνακα 4.1 εννοούμε τον επιπρόσθετο χρόνο που θα μπορούσε να έχει ο συνολικός χρόνος εκτέλεσης του όλου πειράματος. Αυτό ισχύει και για τα τέσσερα πειράματα που έγιναν στη βάση. Ο χρόνος αυτός είναι ο μέγιστος χρόνος από όλους τους μέγιστους χρόνους που έχουμε από όλες τις εκτελέσεις αφαιρώντας τον μέσο όρων των μέγιστων χρόνων αυτών. Πιο αναλυτικά, αθροίζουμε για κάθε εκτέλεση για οποιοδήποτε πείραμα, τους χρόνους εκτέλεσης των επερωτήσεων για τον κάθε πυρήνα ξεχωριστά και ακολουθώς παίρνουμε τον μέγιστο χρόνο από αυτά τα αθροίσματα. Ακολουθώς παίρνουμε τον μέγιστο χρόνο από αυτούς τους είκοσι μέγιστους χρόνους των είκοσι εκτελέσεων και αφαιρούμε το μέσο όρο από είκοσι μέγιστους χρόνους αυτούς και έτσι έχουμε τελικά τον επιπρόσθετο μέγιστο χρόνο που θα μπορούσε να έχει συνολικά η εκτέλεση ολόκληρου του πειράματος.

Με τη γραμμή minus στον πίνακα 4.1 εννοούμε τον ελάχιστο χρόνο που θα μπορούσε να έχει η όλη εκτέλεση του πειράματος. Αυτό ισχύει και για τα τέσσερα πειράματα που πραγματοποιήθηκαν για τη βάση δεδομένων αυτή. Ο χρόνος αυτός μπορεί να βρεθεί με το να αφαιρέσουμε από το μέσο όρο των μέγιστων χρόνων που είχαμε στη γραμμή average του πίνακα 4.1, τον ελάχιστο χρόνο που έχουμε βρει από όλους τους μέγιστους χρόνους των είκοσι εκτελέσεων.

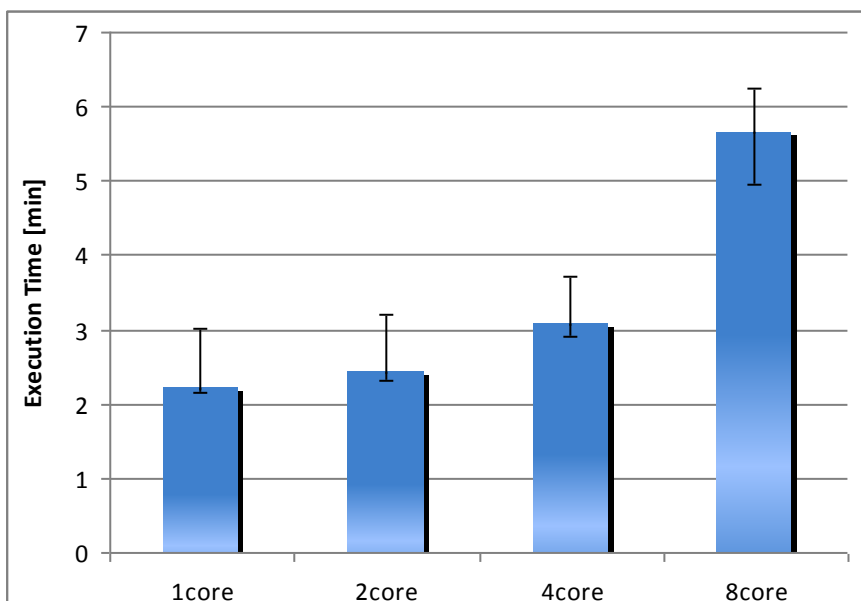
	1core	2core	4core	8core
Average	2,2035075	2,4232245	3,064202	5,6634107
Plus	0,8130725	0,7913955	0,661598	0,5917893
Minus	0,0596775	0,0957545	0,153042	0,7170907

Πίνακας 4.1 Χρόνοι για την βάση δεδομένων του ενός Gigabyte (1Gb)

Μπορούμε να δούμε πιο καθαρά τους χρόνους του πίνακα 4.1 πώς φαίνονται στην πιο κάτω γραφική παράσταση 4.2.

Παρατηρήσεις

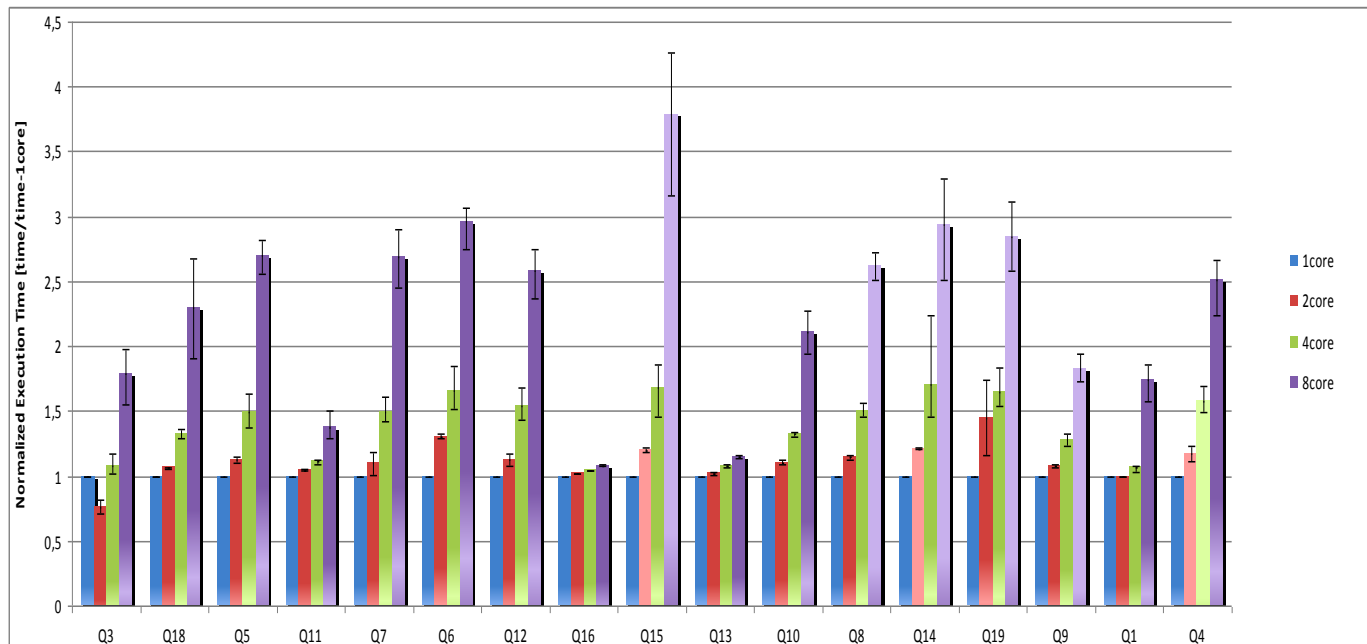
Στη γραφική παράσταση 4.2 οι χρόνοι της εκτέλεσης για το πρώτο πείραμα σε ένα πυρήνα έχει χρόνο δύο λεπτά και είκοσι δευτερόλεπτα και για τις διπλάσιες επερωτήσεις στο δεύτερο πείραμα για δύο πυρήνες χρόνος εκτέλεσης είναι δύο λεπτά και σαράντα δύο δευτερόλεπτα. Ενώ θα αναμέναμε να ήταν ο διπλάσιος χρόνος με βάση το πρώτο πείραμα έχουμε σχεδόν τον ίδιο χρόνο εκτέλεσης. Αυτό συμβαίνει επειδή οι επερωτήσεις ξεκινούν ταυτόχρονα και τρέχουν παράλληλα στους δύο πυρήνες δεκαεπτά επερωτήσεις στον πρώτο πυρήνα και δεκαεπτά επερωτήσεις στο δεύτερο πυρήνα. Μοιράζονται το φόρτο εργασίας οι δύο πυρήνες ενώ αν ήταν σε ένα πυρήνα και οι 34 επερωτήσεις θα είχαμε διπλάσιο χρόνο. Η μνήμη του cache δεν επιβαρύνεται με το να τρέχουμε σε δύο πυρήνες ταυτόχρονα και παράλληλα. Επίσης θα αναμέναμε να τελείωναν τον ίδιο χρόνο με το πρώτο πείραμα αφού τρέχουν οι ίδιες επερωτήσεις και στον πρώτο πυρήνα και στο δεύτερο. Έχουμε μία μικρή καθυστέρηση των είκοσι δύο δευτερολέπτων, αυτή τη καθυστέρηση θα την μελετήσουμε αργότερα για κάθε επερωτηση ξεχωριστά για να δούμε τον ρόλο του cache και το ρόλο της σειράς εκτέλεσης των επερωτήσεων στα πειράματα αυτά. Παρομοίως για το τρίτο πείραμα έχουμε καθυστέρηση ογδόντα έξι δευτερολέπτων περίπου και για το τέταρτο πείραμα έχουμε τη μεγαλύτερη καθυστέρηση στα τρία λεπτά και σαράντα έξι δευτερόλεπτα.



Γραφική παράσταση 4.2 Χρόνος εκτέλεσης σε λεπτά των τεσσάρων πειραμάτων

Στη γραφική παράσταση 4.2 φαίνονται με μικρές γραμμές κατακόρυφες, οι ελάχιστοι (minus) χρόνοι και οι μέγιστοι (plus) χρόνοι που θα μπορούσαν οι εκτελέσεις να ολοκληρωθούν στα πειράματα για ένα πυρήνα για δύο, τέσσερις και οχτώ πυρήνες.

Πιο κάτω στη γραφική παράσταση 4.3 έχουμε για όλες τις επρωτήσεις που εκτελέστηκαν τη συμπεριφορά του κανονικοποιημένου χρόνου εκτέλεσής τους ξεχωριστά.



Γραφική παράσταση 4.3 Κανονικοποιημένος Χρόνος εκτέλεσης

Παρατηρήσεις

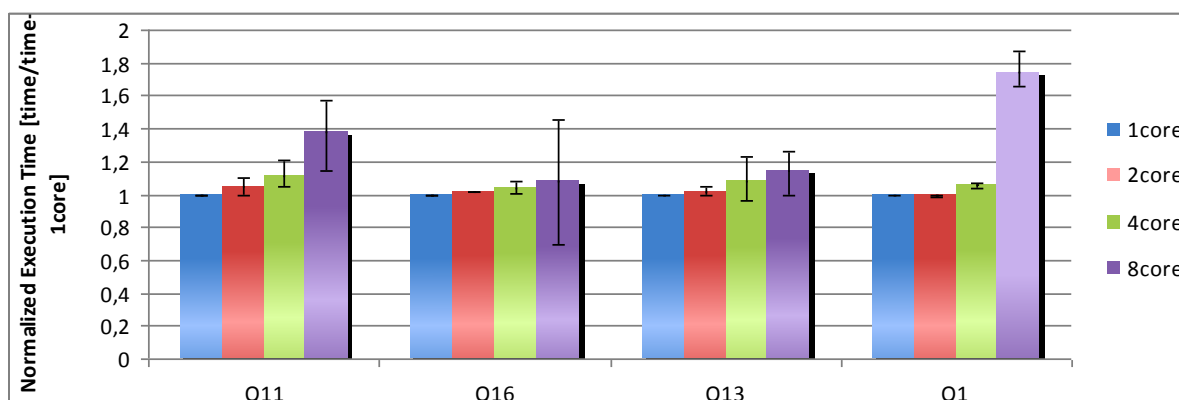
Στη γραφική παράσταση 4.3 βλέπουμε όλες τις επρωτήσεις που χρησιμοποιήθηκαν από τα TPC-H Benchmarks και τα λάθη (error bars) για την κάθε επρωτήση ξεχωριστά και για τα τέσσερα πειράματα. Στη γραφική παράσταση ξεχωρίζει με μεγάλη διαφορά ο κανονικοποιημένος χρόνος εκτέλεσης σχεδόν για όλες τις επρωτήσεις για το πείραμα σε οχτώ πυρήνες. Επίσης διαφορά αλλά όχι και τόσο μεγάλη από τον κανονικοποιημένο χρόνο για ένα πυρήνα έχει και ο κανονικοποιημένος χρόνος στο τρίτο πείραμα σχεδόν σε όλες τις επρωτήσεις με τους τέσσερις πυρήνες. Την πιο μικρή διαφορά κάνει ο κανονικοποιημένος χρόνος των επρωτήσεων για το πείραμα με τους δύο πυρήνες. Για να δείξουμε τους λόγους που συμβαίνει αυτό θα πρέπει να χωρίσουμε τις επρωτήσεις μας σε κατηγορίες για να μπορούμε να μελετήσουμε καλύτερα τη συμπεριφορά αυτή. Πιο κάτω θα αναφέρουμε τις κατηγορίες αυτές.

Κατηγορίες των επερωτήσεων

Για να δείξουμε τους λόγους που συμβαίνουν αυτές οι συμπεριφορές των επερωτήσεων, θα πρέπει να χωρίσουμε τις επερωτήσεις μας σε κατηγορίες. Με τις κατηγορίες αυτές θα μπορούμε πιο εύκολα να αναλύσουμε την συμπεριφορά των επερωτήσεων που εκτελέστηκαν στα πειράματα της βάσης δεδομένων του ενός Gigabyte (1Gb). Οι κατηγορίες αυτές θα είναι σε σχέση με την αλλαγή του αριθμού πυρήνων και την εκτέλεση με άλλες επερωτήσεις μαζί έχοντας ως βασικό σενάριο το πείραμα σε έναν πυρήνα.

Κατηγορία 1

Στην πρώτη κατηγορία θα έχουμε τις επερωτήσεις αυτές, όπου με την αλλαγή του αριθμού των πυρήνων δεν έχουμε σημαντική αλλαγή στα αποτελέσματα, δηλαδή ο κανονικοποιημένος χρόνος εκτέλεσης των επερωτήσεων δεν επηρεάζεται με την εκτέλεση της κάθε επερωτήσης με τις άλλες επερωτήσεις. Οι επερωτήσεις που ανταποκρίνονται στην κατηγορία αυτή είναι οι επερωτήσεις q1, q11, q13 και q16 όπως φαίνονται στη γραφική παράσταση 4.4.

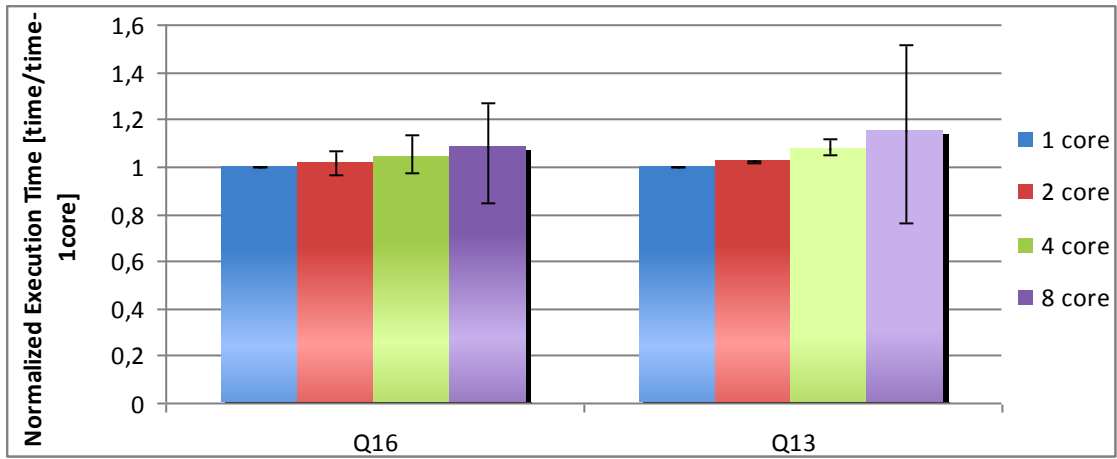


Γραφική παράσταση 4.4 Κανονικοποιημένος Χρόνος Εκτέλεσης-Πρώτη Κατηγορία

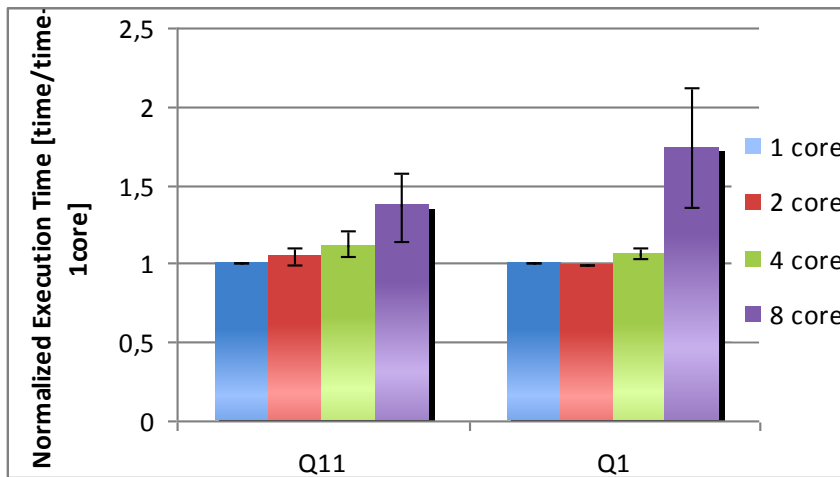
Παρατηρήσεις

Στη γραφική παράσταση 4.4 φαίνονται οι επερωτήσεις για την πρώτη κατηγορία. Οι επερωτήσεις q13 και q16 έχουν την ίδια συμπεριφορά και στα τέσσερα πειράματα, συνεπώς με την αλλαγή του αριθμού των πυρήνων δεν έχουμε αλλαγή στα αποτελέσματα, και ο κανονικοποιημένος χρόνος εκτέλεσης των επερωτήσεων δεν επηρεάζεται με την εκτέλεση με τις άλλες επερωτήσεις. Η επερωτήση q11 έχει πολύ μικρή διαφορά και έτσι το κατατάσσουμε στην κατηγορία αυτή. Η επερωτήση q1 έχει την ίδια συμπεριφορά για τον κανονικοποιημένο χρόνο και στα τρία πρώτα πειράματα, δηλαδή για τα πειράματα σε ένα

πυρήνα, δύο και τέσσερις πυρήνες, αλλά στο τέταρτο πείραμα με οχτώ πυρήνες έχουμε μεγάλη αλλαγή στη συμπεριφορά, όμως παρόλα αυτά την κατατάσσουμε στην κατηγορία αυτή. Μετά από μελέτη της συμπεριφοράς της συγκεκριμένης q1 επερώτησης σε όλους τους πυρήνες του πειράματος τέσσερα, με τους οχτώ πυρήνες, έχουμε παρατηρήσει ότι είναι πάρα πολύ μικρή η μεταβολή του χρόνου, δηλαδή η αυξομείωση του χρόνου που έχει σε όλες τις εκτελέσεις της στους οχτώ πυρήνες είναι σχεδόν μηδαμινή. Από αυτό καταλαβαίνουμε πώς δεν επηρεάζεται με την εκτέλεση με άλλες επερωτήσεις, για αυτό το λόγο το έχουμε κατατάξει σε αυτή την γενική κατηγορία. Συνεπώς ο λόγος αυτής της συμπεριφοράς θα είναι η κρυφή μνήμη (cache). Λόγο του χώρου στην κρυφή μνήμη που πρέπει να μεταφερθούν δεδομένα και από τους οχτώ πυρήνες, δημιουργείται αυτή η καθυστέρηση και αυξάνει τον κανονικοποιημένο χρόνο εκτέλεσης μέχρι και το ένα λεπτό και εβδομήντα τέσσερα δευτερόλεπτα (1,7436 minutes) από 1,06 λεπτά σχεδόν, που είχαμε για τους προηγούμενους στα υπόλοιπα τρία πειράματα. Επειδή όμως οι επερωτήσεις q1 και q11 διαφέρουν κατά πολύ από τις άλλες επερωτήσεις, συγκεκριμένα στο πείραμα τέσσερα με τους οχτώ πυρήνες, θα το κατατάξουμε σε μια άλλη υποκατηγορία της γενικής κατηγορίας αυτής όπου ο κανονικοποιημένος χρόνος εκτέλεσης των επερωτήσεων δεν επηρεάζεται με την εκτέλεση με τις άλλες επερωτήσεις, αλλά επηρεάζεται από το χώρο στην κρυφή μνήμη(cache). Η επερώτηση q13 χρησιμοποιεί τη cpu ελάχιστα για να διαβάσει, χρησιμοποιεί hash join και επεξεργάζεται περισσότερο και για την επερώτηση q16 έχουμε να επεξεργάζεται και διαβάζει σχεδόν το ίδιο, χρησιμοποιεί nested loop join μαζί με sort. Για την επερώτηση q11 έχουμε περισσότερο να διαβάζουν από τη μνήμη και λιγότερο να επεξεργάζονται. Ενώ για την q1 έχουμε sequential scan, όπου ελάχιστα διαβάζει από τη μνήμη και με την εκτέλεση σε οχτώ πυρήνες παρατηρούμε ότι αυξάνεται ο χρόνος εκτέλεσης. Έτσι στη γενική κατηγορία θα έχουμε τις επερωτήσεις q13 και q16 όπως φαίνονται στη γραφική παράσταση 4.4.1. Και στην υποκατηγορία θα έχουμε μόνο την επερώτηση q1, q11 όπως φαίνεται στη γραφική παράσταση 4.4.2.



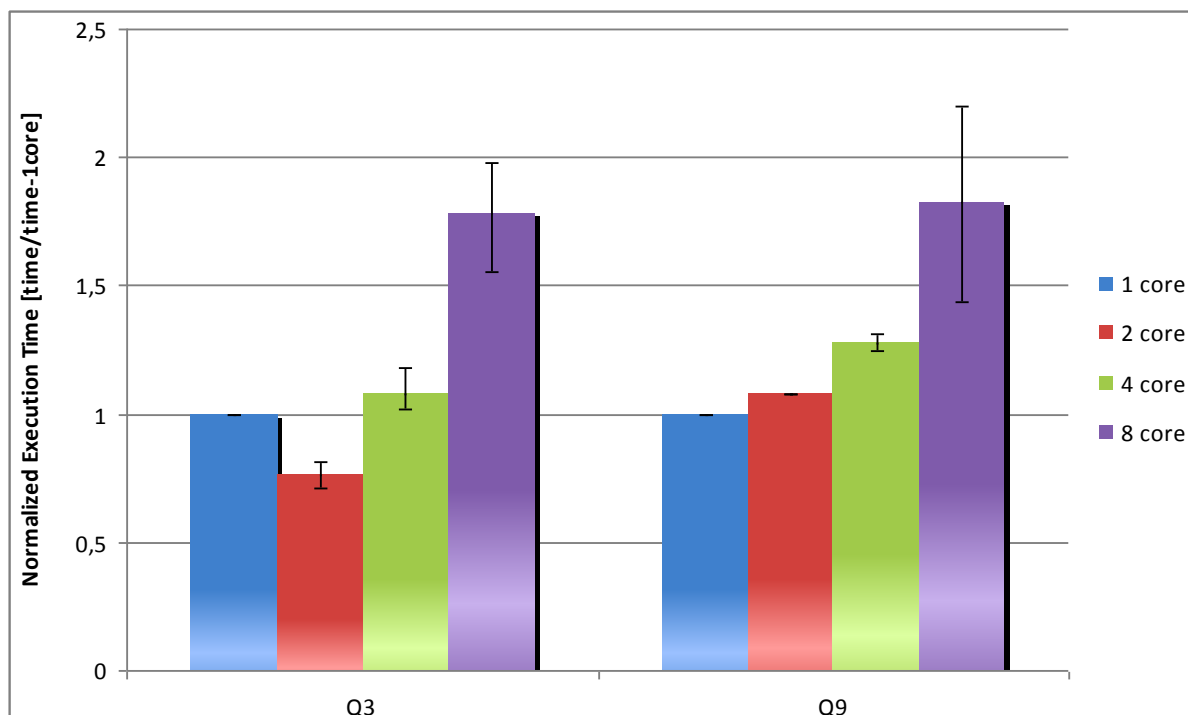
Γραφική παράσταση 4.4.1 Κανονικοποιημένος Χρόνος Εκτέλεσης-Πρώτη Κατηγορία



Γραφική παράσταση 4.4.2 Κανονικοποιημένος Χρόνος Εκτέλεσης-Πρώτη Κατηγορία

Κατηγορία 2

Στην δεύτερη κατηγορία θα έχουμε τις επρωτήσεις αυτές, όπου με την αλλαγή του αριθμού των πυρήνων έχουμε κάποια σημαντική αλλαγή στα αποτελέσματα, δηλαδή ο κανονικοποιημένος χρόνος εκτέλεσης των επρωτήσεων επηρεάζεται με την εκτέλεση της κάθε επρωτήσης με τις άλλες επρωτήσεις. Οι επρωτήσεις που έχουμε στην κατηγορία αυτή είναι οι επρωτήσεις q3, και q9 όπως φαίνονται στη γραφική παράσταση 4.5.



Γραφική παράσταση 4.5 Κανονικοποιημένος Χρόνος Εκτέλεσης-Δεύτερη Κατηγορία

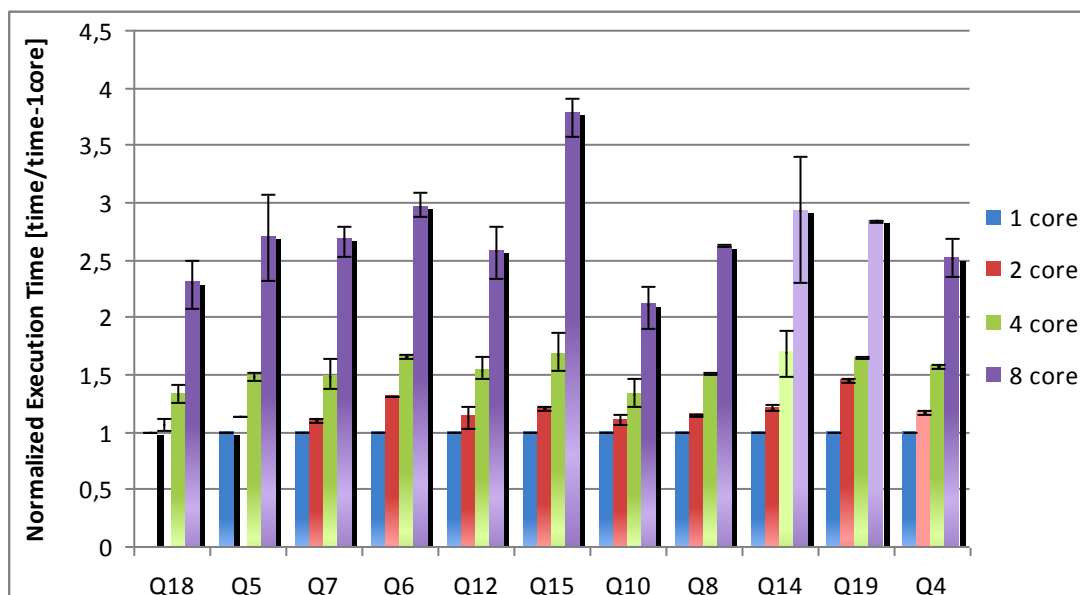
Παρατηρήσεις

Στη γραφική παράσταση 4.5 φαίνονται οι επερωτήσεις για την δεύτερη κατηγορία. Υπάρχει αλλαγή στα αποτελέσματα μεταξύ των τεσσάρων πειραμάτων, διότι ο κανονικοποιημένος χρόνος εκτέλεσης των επερωτήσεων είναι διαφορετικός σε κάθε πείραμα για την κάθε επερώτηση και αυξάνεται με την αλλαγή της αύξησης του αριθμού πυρήνων στο κάθε πείραμα. Επίσης ο κανονικοποιημένος χρόνος επηρεάζεται με την εκτέλεση της κάθε επερώτησης με τις άλλες επερωτήσεις. Ο ρυθμός αύξησης του κανονικοποιημένου χρόνου στα πρώτα τρία πειράματα δηλαδή για ένα πυρήνα, δύο και τέσσερις πυρήνες είναι σταδιακά προς τα πάνω σχεδόν για την επερώτηση q9. Ενώ για την επερώτηση q3 έχουμε μεγάλη αύξηση του κανονικοποιημένου χρόνου στο τέταρτο πείραμα με τους οχτώ πυρήνες και για το δεύτερο πείραμα με τους δύο πυρήνες έχουμε μείωση του χρόνου. Μετά από τη μελέτη της συμπεριφοράς της εκτέλεσης των επερωτήσεων που ανήκουν στη δεύτερη κατηγορία αυτή, οι διακυμάνσεις ήταν πολύ μικρές έτσι και πάλι έχουμε την αιτία του χώρου στην κρυφή μνήμη (cache) γιατί έχουμε πολύ περισσότερο όγκο δεδομένων από όσο είχαμε στα άλλα τρία πειράματα με ένα, δύο και τέσσερις πυρήνες. Για την επερώτηση q3 στο δεύτερο πείραμα με τους δύο πυρήνες σε συνδυασμό hash join και nested loop join έχουμε καλύτερο κανονικοποιημένο χρόνο εκτέλεσης απ' ότι στο βασικό μας σενάριο, διαβάζει πολύ

περισσότερο από τη μνήμη. Βλέπουμε όμως ότι για τέσσερις και οχτώ πυρήνες πως αυξάνεται ο κανονικοποιημένος χρόνος εκτέλεσης σε σύγκριση με τα πρώτα μας πειράματα αυτό συμβαίνει διότι αυξάνεται το κόστος για cache misses. Για την επερώτηση q9 όμως που χρησιμοποιεί nested loop join μόνο έχουμε αύξηση του κανονικοποιημένου χρόνου για όλα τα πειράματα σε σύγκριση με το βασικό σενάριο διαβάζει πολύ λίγο από τη μνήμη υπάρχει το κόστος για cache misses, όμως σε σύγκριση με την επερώτηση q3 στο δεύτερο πείραμα βλέπουμε ότι διαφέρουν κατά πολύ και αυτό συμβαίνει επειδή χρησιμοποιεί hash join όπου περιορίζει τον αριθμό πλειάδων που θα χρησιμοποιούσε χωρίς hash join.

Κατηγορία 3

Στην τρίτη κατηγορία θα έχουμε τις επερωτήσεις αυτές, όπου με την αλλαγή του αριθμού των πυρήνων έχουμε μεγάλη αλλαγή στα αποτελέσματα, δηλαδή ο κανονικοποιημένος χρόνος εκτέλεσης των επερωτήσεων επηρεάζεται με την εκτέλεση της κάθε επερωτήσης με τις άλλες επερωτήσεις πάρα πολύ. Οι επερωτήσεις που έχουμε στην κατηγορία αυτή είναι οι επερωτήσεις q4, q5, q6, q7, q8, q10, q12, q14, q15, q18 και q19 όπως φαίνονται στη γραφική παράσταση 4.6.



Γραφική παράσταση 4.6 Κανονικοποιημένος Χρόνος Εκτέλεσης -Τρίτη Κατηγορία

Παρατηρήσεις

Στη γραφική παράσταση 4.6 φαίνονται οι επερωτήσεις για την τρίτη κατηγορία. Σε αυτή τη κατηγορία ανήκουν και οι περισσότερες επερωτήσεις που έχουμε στη λίστα με τις δεκαεπτά επερωτήσεις. Υπάρχει αλλαγή στα αποτελέσματα μεταξύ των τεσσάρων πειραμάτων, διότι ο

κανονικοποιημένος χρόνος εκτέλεσης των επερωτήσεων είναι διαφορετικός σε κάθε πείραμα για την κάθε επερώτηση και αυξάνεται με την αλλαγή της αύξησης του αριθμού πυρήνων στο κάθε πείραμα. Επίσης ο κανονικοποιημένος χρόνος επηρεάζεται με την εκτέλεση της κάθε επερώτησης με τις άλλες επερωτήσεις. Ο ρυθμός αύξησης του κανονικοποιημένου χρόνου στα πρώτα τρία πειράματα δηλαδή για ένα πυρήνα, δύο και τέσσερις πυρήνες είναι σταδιακά προς τα πάνω σχεδόν για όλες τις επερωτήσεις. Ενώ για όλες τις επερωτήσεις έχουμε μεγάλη αύξηση του κανονικοποιημένου χρόνου στο τέταρτο πείραμα με τους οχτώ πυρήνες. Οι επερωτήσεις όπως φαίνονται στη γραφική παράσταση 4.6, ανήκουν σε αυτή τη κατηγορία λόγω του ρυθμού αυξομείωσης των διακυμάνσεων που είναι μεγάλος για το κάθε πείραμα που είχαν εκτελεσθεί. Για τις επερωτήσεις που έχουμε στην κατηγορία τρία στα πρώτα δύο πειράματα έχουν παρόμοιες τιμές αλλά στο τρίτο πείραμα με τους τέσσερις πυρήνες αυξάνεται ο χρόνος εκτέλεσης κατά πολύ. Επίσης στο πείραμα τέσσερα έχουμε μεγάλο ρυθμό αυξομείωσης των διακυμάνσεων μεταξύ των εκτελέσεων των επερωτήσεων στους οχτώ πυρήνες που αποδεικνύει με τη συμπεριφορά αυτή, ότι ο κανονικοποιημένος χρόνος των χρόνων εκτέλεσης που είχαμε από τις επερωτήσεις σε όλα τα πειράματα πως επηρεάζεται πάρα πολύ με την αλλαγή του αριθμού των πυρήνων και την εκτέλεση των επερωτήσεων αυτών με τις υπόλοιπες επερωτήσεις. Δεν φταίει τόσο πολύ ο χώρος στην κρυφή μνήμη λόγω των διακυμάνσεων που δεν έχουν σταθερό χρόνο. Αυτό μπορούμε να το πούμε διότι έχουμε μελετήσει για τις προηγούμενες κατηγορίες τον ρυθμό αυξομείωσης των διακυμάνσεων μεταξύ των εκτελέσεων των επερωτήσεων σε όλους τους πυρήνες για όλα τα πειράματα και γνωρίζουμε πως οι διακυμάνσεις είχαν σχεδόν σταθερούς χρόνους. Γι' αυτές τις μεγάλες αλλαγές στις διακυμάνσεις των χρόνων εκτέλεσης των επερωτήσεων στα πειράματα αυτά που πραγματοποιήθηκαν φταίει και η σειρά που εκτελούνται οι επερωτήσεις. Για αυτές τις επερωτήσεις έχουμε περίπου το ίδιο CPI(Cycle per Instruction, Παράρτημα Α), και οι επερωτήσεις μας επεξεργάζονται περισσότερο από ότι να διαβάσουν στη μνήμη[6].

Η επερώτηση q14 παρόλο που διαβάζει λίγο από τη μνήμη, χρησιμοποιεί μόνο index scan που έχει αυξημένο κόστος για cache misses ενόσω αυξάνεται και ο αριθμός των πυρήνων. Ενώ η επερώτηση q4 που χρησιμοποιεί index scan σε συνδυασμό με nested loop join, διαβάζει πάρα πολύ από τη μνήμη και έχει καλύτερους χρόνους για όλα τα πειράματα, όμως και εδώ υπάρχει αυξημένο το κόστος για cache misses.

Η επερώτηση q10, διαβάζει λίγο από τη μνήμη και χρησιμοποιεί nested loop join έχει τους χαμηλότερους χρόνους σε αυτή την κατηγορία όμως λόγω των cache misses που αυξάνονται στο τέταρτο πείραμα με τους οχτώ πυρήνες την κατατάξαμε εδώ.

Οι επερωτήσεις q7 και q8 έχουν παρόμοια συμπεριφορά χρόνων διότι χρησιμοποιούν hash join και nested loop join με ελάχιστη διαφορά στο διάβασμα από τη μνήμη. Ίσως να ήταν καλύτερα να είχαμε χρησιμοποιήσει nested loop join, όμως αν προσέξουμε στην επερώτηση

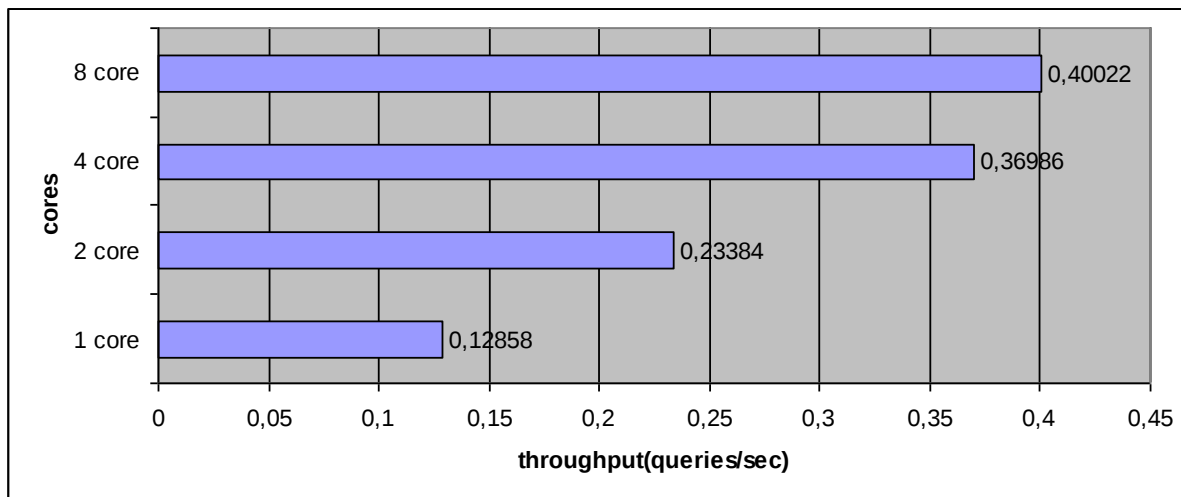
q19 που χρησιμοποιεί και αυτή nested loop join μόνο με ελάχιστα περισσότερο διάβασμα από τη μνήμη από ότι είχαμε για την q10 πλησιάζει τους χρόνους των επερωτήσεων q7 και q8. Επίσης η επερώτηση q15 χρησιμοποιεί nested loop join με περισσότερο διάβασμα με Cpu πάνω από το 50% , έχουμε χειρότερο χρόνο σε εκτέλεση στους οχτώ πυρήνες επειδή είναι πολύ αυξημένο το κόστος των cache misses. Οι επερωτήσεις q5, q6, q12 και q18 χρησιμοποιούν όλες sort με διαφορετικό συνδυασμό, καλύτερους χρόνους παρουσίασε η q5 και q18 διότι με μεγαλύτερο διάβασμα από τη μνήμη και σε συνδυασμό με nested loop join είχαν καλύτερα αποτελέσματα απ' ότι hash με συνδυασμό sort είτε sort μόνο για τις επερωτήσεις q12 και q6 αντίστοιχα όπου είχαν και λιγότερο διάβασμα από τη μνήμη.

Ρυθμοαπόδοση

Η ρυθμοαπόδοση (throughput) για τη βάση δεδομένων του ενός Gigabyte (1Gb) φαίνεται πιο κάτω στο πίνακα 4.2 και στη γραφική παράσταση 4.7 των επερωτήσεων συνολικά που έχουμε εκτελέσει. Με τη ρυθμοαπόδοση δείχνουμε πόσες επερωτήσεις έχουν εκτελεστεί ανά μονάδα χρόνου σε δευτερόλεπτα (queries/sec). Έχοντας όλους τους χρόνους που εκτελέστηκαν από όλες τις επερωτήσεις σε όλα τα πειράματα που πραγματοποιήθηκαν για τη βάση αυτή, παίρνουμε ένα μέσο όρο από τα αθροίσματα από όλες τις επαναλήψεις των εκτελέσεων (Sum of queries average(sec)). Ακολουθώς θα πάρουμε για το πρώτο πείραμα στον ένα πυρήνα, τον αριθμό των επερωτήσεων, στην συγκεκριμένη περίπτωση είναι ίσος με δεκαεπτά, και θα διαιρέσουμε με το άθροισμα των χρόνων αυτών. Με αυτό τον τρόπο μπορούμε να δείξουμε πόσες επερωτήσεις έχουν εκτελεστεί ανά δευτερόλεπτο (throughput (queries/sec)), στη συγκεκριμένη περίπτωση έχουμε 0,12858 επερωτήσεις που εκτελούνται ανά δευτερόλεπτο.

Cores	1core	2core	4core	8core
Number of Queries	17	34	68	136
Sum of queries average(sec)	(2,2035075)x60	(2,4232245)x60	(3,064202)x60	(5,6634107)x60
Throughput(queries/sec)	0,12858	0,23384	0,36986	0,40022

Πίνακας 4.2 Ρυθμοαπόδοση για την βάση δεδομένων του ενός Gigabyte (1Gb)



Γραφική παράσταση 4.7 Ρυθμοαπόδοση (Throughput)

Παρατηρήσεις

Όπως έχουμε αναφέρει και στο κεφάλαιο 3 για να έχουμε την έννοια του παραλληλισμού θα πρέπει να υπάρχουν περισσότεροι από ένα πυρήνες. Για το πείραμα με δύο πυρήνες έχουμε ρυθμοαπόδοση ίση με το πηλίκο του συνολικού αριθμού επερωτήσεων (δηλαδή 34) και του αθροίσματος των χρόνων από όλες τις εκτελέσεις και των 34 επερωτήσεων, πολλαπλασιάζουμε με 60 για να το μετατρέψουμε σε δευτερόλεπτα. Το αποτέλεσμα που έχουμε είναι 0,23384 δηλαδή παρόλο που εκτελούνται οι διπλάσιες επερωτήσεις απ' ότι στο πρώτο πείραμα με ένα πυρήνα, βλέπουμε ότι ο αριθμός των επερωτήσεων που εκτελούνται υπερέρχει από τον αριθμό 0,12858, σχεδόν διπλάσιο. Συνεπώς βλέπουμε ότι με τον παραλληλισμό που έχουμε χρησιμοποιήσει στο δεύτερο πείραμα, έχουμε καλύτερα αποτελέσματα. Έχουμε βελτιωμένη επίδοση (performance) στο δεύτερο πείραμα, 1,81 φορές καλύτερη ρυθμοαπόδοση, δηλαδή σχεδόν δύο φορές τον αριθμό επερωτήσεων που μπορούν να εκτελεστούν, που το έχουμε δείξει μέσω της ρυθμοαπόδοσης (throughput) της μηχανής.

Για τα άλλα δύο πειράματα με τέσσερις πυρήνες και οχτώ πυρήνες πράττουμε παρομοίως και έχουμε τιμή της ρυθμοαπόδοσης ίση με 0,36986 και 0,40022 αντίστοιχα. Για το πείραμα με τέσσερις πυρήνες που και εδώ έχουμε χρησιμοποιήσει παραλληλισμό, η ρυθμοαπόδοση είναι σχεδόν ο τριπλάσιος αριθμός επερωτήσεων (2,87 φορές) που μπορούν να εκτελεστούν απ' ότι τη ρυθμοαπόδοση σε ένα πυρήνα, δεν είναι η αναμενόμενη δηλαδή 3,62 φορές, όμως και πάλι έχουμε ακόμη περισσότερη βελτίωση στην επίδοση της μηχανής. Για το πείραμα με οχτώ πυρήνες θα αναμέναμε να ήταν 7-8 φορές καλύτερη τιμή για την ρυθμοαπόδοση σε

σύγκριση με τη τιμή της ρυθμοαπόδοσης για ένα πυρήνα και κατά συνέπεια καλύτερη επίδοση από το αρχικό μας πείραμα. Βλέπουμε όμως ότι η τιμή της ρυθμοαπόδοσης ξεπερνάει λίγο το τριπλάσιο (3,11 φορές) από τη τιμή της ρυθμοαπόδοσης στον ένα πυρήνα. Έχουμε μεν πολύ καλύτερη ρυθμοαπόδοση από όλα τα άλλα πειράματα όμως δεν έχουμε την αναμενόμενη, χάνουμε ένα πολύ μεγάλο ποσοστό, για αυτό έχουμε ως αιτία που δημιουργείται η καθυστέρηση και χάνεται αυτό το ποσοστό χρόνου, τον αριθμό επερωτήσεων και τον όγκο πληροφοριών, τη σειρά που εκτελούνται και ο ρόλος της κρυφής μνήμης. Με τη σειρά που εκτελούνται εννοούμε ότι σε μία συγκεκριμένη μονάδα χρόνου μπορεί να έχουμε πολλές επερωτήσεις που να ζητούν μεγάλο όγκο πληροφοριών από τη βάση και να καθυστερούν τη εκτέλεση των επόμενων επερωτήσεων που είναι σε αναμονή. Επίσης μπορεί να έχουμε το γέμισμα της ιδιωτικής μνήμης του πυρήνα και να δημιουργείται περισσότερη καθυστέρηση, αφού ο κάθε πυρήνας στη μελέτη μας εκτελεί 17 επερωτήσεις παρόλο που έχουμε συνεργασία και μοιράζονται οι πληροφορίες μεταξύ των διεργασιών στο συγκεκριμένο πυρήνα σε κάποια στιγμή ο χώρος στην ιδιωτική μνήμη του πυρήνα θα φτάσει μέχρι το όριο που έχει και θα έχουμε καθυστέρηση συνολικά διότι θα χρειαστεί περισσότερος χρόνος για να ολοκληρωθούν οι διεργασίες που εκτελούνταν και περισσότερο χρόνο αναμονής των επόμενων επερωτήσεων προς εκτέλεση. Επίσης μπορεί να γεμίσει και η κοινή μνήμη μεταξύ των πυρήνων που και πάλι έχουμε ως θετικό στοιχείο τη συνεργασία και μοίρασμα πληροφοριών μεταξύ των πυρήνων, αλλά και πάλι σε κάποια στιγμή μπορεί να φτάσει στο όριο του χώρου της και να δημιουργηθεί καθυστέρηση. Συνεπώς βλέπουμε στη συγκεκριμένη περίπτωση του πειράματος με τους οχτώ πυρήνες να εξυπηρετούνται περισσότερες επερωτήσεις από όλα τα άλλα πειράματα όμως επίσης βλέπουμε ότι μειώνεται πιο πολύ η ρυθμοαπόδοση από την αναμενόμενη ρυθμοαπόδοση. Οι περισσότερες επερωτήσεις χρησιμοποίησαν nested loop join έτσι μπορούμε να δικαιολογήσουμε την περισσότερη μείωση της ρυθμοαπόδοσης στους οχτώ πυρήνες λόγο του αυξημένου κόστους για cache misses.

Αυτά τα προβλήματα μπορούμε να τα λύσουμε σίγουρα εάν λιγοστέψουμε τις επερωτήσεις για τους οχτώ πυρήνες. Για μια εταιρία είναι πολύ σημαντικός ο χρόνος που εξυπηρετούνται οι πελάτες τους μέσω ενός server, συνεπώς στην προκειμένη περίπτωση με οχτώ πυρήνες για παρόμοιο μέγεθος βάσης.

Για να λύσουμε το υπάρχον πρόβλημα χωρίς να κάνουμε τις αλλαγές αυτές, πρέπει να θέσουμε διαφορετικό συνδυασμό επερωτήσεων προς εκτέλεση από τους πυρήνες, έτσι που να μην έχω εκτέλεση πολλών επερωτήσεων που ζητούν μεγάλο όγκο πληροφοριών από τη βάση την ίδια χρονική στιγμή. Θα μπορούσα να έχω λιγότερες τέτοιες επερωτήσεις και να έχω συνδυασμό με επερωτήσεις που ζητούν λιγότερες πληροφορίες από τη βάση δηλαδή οι επερωτήσεις που θα συνδυάσω να είναι πολύ πιο μικρός ο χρόνος εκτέλεσής τους, και έτσι

θα έχω λιγότερη καθυστέρηση συνολικά στο τέλος, και κατά συνέπεια καλύτερη ρυθμοαπόδοση και συνολική επίδοση.

4.4 Συμπεράσματα

Με την εκτέλεση των επερωτήσεων στα τέσσερα πειράματα σε μία βάση δεδομένων μπορούμε να έχουμε καμία αλλαγή είτε σημαντική αλλαγή είτε μεγάλη αλλαγή στα αποτελέσματα μας.

Καμία αλλαγή έχουμε όταν με την αλλαγή του αριθμού των πυρήνων, ο κανονικοποιημένος χρόνος εκτέλεσης των επερωτήσεων δεν επηρεάζεται με την εκτέλεση της κάθε επερώτησης με τις άλλες επερωτήσεις. Είτε επίσης να μην έχουμε αλλαγή στα αποτελέσματα μας για τα τρία πρώτα πειράματα στους ένα , δύο και τέσσερις πυρήνες και για το τέταρτο πείραμα να έχουμε μεγάλη αλλαγή αλλά ο χρόνος αυτός να είναι σταθερά μεγάλος και δικαιολογείται λόγω του χώρου στην κρυφή μνήμη (cache).

Σημαντική αλλαγή έχουμε όταν με την αλλαγή του αριθμού των πυρήνων, ο κανονικοποιημένος χρόνος εκτέλεσης των επερωτήσεων επηρεάζεται αρκετά με την εκτέλεση της κάθε επερώτησης με τις άλλες επερωτήσεις. Ο ρυθμός αύξησης του κανονικοποιημένου χρόνου στα πρώτα τρία πειράματα δηλαδή για ένα πυρήνα, δύο και τέσσερις πυρήνες είναι σταδιακά με αύξηση προς τα πάνω σχεδόν για όλες τις επερωτήσεις, και γι αυτό φταίει η εκτέλεση της κάθε επερώτησης με άλλες επερωτήσεις. Ενώ για όλες τις επερωτήσεις στο τέταρτο πείραμα με οχτώ πυρήνες, έχουμε μεγάλη αύξηση του κανονικοποιημένου χρόνου και γι αυτό παίζει μεγάλο ρόλο και ο χώρος στην κρυφή μνήμη (cache).

Μεγάλη αλλαγή έχουμε όταν με την αλλαγή του αριθμού των πυρήνων, ο κανονικοποιημένος χρόνος εκτέλεσης των επερωτήσεων επηρεάζεται πάρα πολύ με την εκτέλεση της κάθε επερώτησης με τις άλλες επερωτήσεις. Μεγάλη αλλαγή έχουμε σε μεγάλες διακυμάνσεις στις αυξομειώσεις στους χρόνους εκτέλεσης των επερωτήσεων.

Για μια βάση δεδομένων του ενός Gigabyte (1GB), θα είναι πιο αποδοτικό να δουλεύει με τουλάχιστον οχτώ πυρήνες. Και για να βελτιώσουμε την ρυθμοαπόδοση και συνάμα την όλη επίδοση στα υπάρχον προβλήματα θα πρέπει να αλλάξουμε το συνδυασμό επερωτήσεων για κάθε πυρήνα.

Όλες οι επερωτήσεις που διαβάζουν περισσότερο από τη μνήμη με cpi πέραν του 50% και χρησιμοποιούν είτε hash, είτε sort, είτε index scan καλύτερα να χρησιμοποιούνται σε συνδυασμό με nested loop join. Ενώ για τις επερωτήσεις με λιγότερο από το 50% διάβασμα από τη μνήμη είναι καλύτερα τα αποτελέσματα με χρήση του hash και μόνο.

Κεφάλαιο 5

Βάση Δεδομένων Μεγέθους 10GB

5.1 Περιγραφή Πειράματος	28
5.2 Υλοποίηση Πειράματος	29
5.3 Πειραματική Αξιολόγηση Πειράματος	30
5.4 Συμπεράσματα	38

5.1 Περιγραφή Πειράματος

Αρχικά θα αλλάξουμε τη βάση του ενός Gigabyte (1Gb) σε βάση των δέκα Gigabyte (10Gb), με περισσότερο όγκο από δεδομένα δηλαδή απ' ότι είχαμε προηγουμένως στην άλλη βάση, όπου θα πραγματοποιήσουμε και πάλι τέσσερα πειράματα συνολικά. Τα τέσσερα πειράματα αυτά είναι τα ίδια πειράματα που εκτελέσαμε και για τη βάση του ενός Gigabyte (1Gb). Δηλαδή το πρώτο πείραμα με ένα πυρήνα που θα είναι και το βασικό μας σενάριο για τη καινούργια βάση, το δεύτερο πείραμα με δύο πυρήνες, το τρίτο πείραμα με τέσσερις πυρήνες και το τέταρτο και τελευταίο πείραμα για αυτή την βάση δεδομένων μεγέθους των δέκα Gigabyte (10Gb) θα ασχοληθούμε με τους οχτώ πυρήνες της μηχανής. Με την βοήθεια των επερωτήσεων (queries) των TPC-H Benchmarks που θα εκτελεστούν στους πυρήνες της ίδιας μηχανής που είχαν εκτελεσθεί και προηγουμένως για τη βάση μεγέθους ενός Gigabyte. Και πάλι θα έχουμε σαν αποτελέσματα στην έξοδο δύο αρχεία που έχουμε ορίσει για να έχουμε για την κάθε επερώτηση, το ένα αρχείο θα περιέχει την απάντηση της επερώτησης (query), δηλαδή πληροφορίες που ικανοποιούν την κάθε επερώτηση που θα ανακτήσει από τη βάση μέσω των πινάκων της βάσης, και το δεύτερο αρχείο θα είναι ο χρόνος εκτέλεσης της κάθε επερώτησης που έχει ολοκληρώσει την ανάκτηση των πληροφοριών που ζητούσε από τη βάση δεδομένων. Με τα αποτελέσματα αυτά που θα μας δοθούν από τις εκτελέσεις των πειραμάτων θα δημιουργήσουμε γραφικές παραστάσεις που να δείχνουν τον χρόνο εκτέλεσης των επερωτήσεων στους πυρήνες (cores). Ακολούθως θα αναλύσουμε και θα μελετήσουμε την συμπεριφορά των χρόνων αυτών. Ανάλογα με την συμπεριφορά των χρόνων αυτών θα χωριστούν σε κατηγορίες. Οι κατηγορίες αυτές θα αναφέρονται για τον βαθμό που οι παράγοντες επηρεάζουν την απόδοση της εφαρμογής συγκεκριμένα οι

παράγοντες αυτοί είναι ο όγκος των αποθηκευμένων δεδομένων στην βάση, ο αριθμός των πυρήνων που χρησιμοποιούνται για την εκτέλεση των επερωτήσεων και η σειρά της εκτέλεσης των επερωτήσεων στον κάθε πυρήνα. Τέλος θα δώσουμε γενικά συμπεράσματα για τις εκτελέσεις των επερωτήσεων και για τα τέσσερα πειράματα που έχουν διεξαχθεί.

5.2 Υλοποίηση Πειράματος

Για να δημιουργήσουμε την καινούργια βάση των δέκα Gigabyte, με τα ίδια ονόματα στους οχτώ πίνακες που είχαμε και πριν, διαγράφουμε όλα τα δεδομένα που υπήρχαν στους πίνακες και γεμίζουμε τους πίνακες με τα καινούργια δεδομένα με τους πιο κάτω τρόπους και εντολές:

Για να διαγράψουμε τα δεδομένα που υπήρχαν στους πίνακες, εκτελούμε αρχικά την εντολή `psql`, για να μπούμε στην βάση, μετά αφού είμαστε στην βάση έχουμε την εντολή:

```
<όνομα της βάσης> => SELECT * from <όνομα του πίνακα> ;
```

Με αυτή την εντολή βλέπουμε εάν υπάρχουν εγγραφές στον πίνακα που έχουμε επιλέξει να δούμε τα περιεχόμενά του, εάν υπάρχει τότε εκτελούμε μία άλλη εντολή την:

```
<όνομα της βάσης> =>DELETE FROM <όνομα του πίνακα> ;
```

όπου θα διαγραφούν όλες οι εγγραφές που υπήρχαν μέσα στον πίνακα που έχουμε επιλέξει για να διαγράψουμε τα περιεχόμενα του. Ακολούθως βγαίνουμε εκτός της βάσης με την εντολή:

```
<όνομα της βάσης> => \q
```

Μετά εκτελούμε την εντολή :

```
<όνομα ιδιοκτήτη>@<όνομα μηχανής>:~$ cat <όνομα πίνακα>.tbl | psql -c "COPY <όνομα πίνακα> FROM STDIN DELIMITER AS '|' "
```

Με αυτή την εντολή έχουμε αντιγράψει όλα τα περιεχόμενα από τον πίνακα που έχουμε στον φάκελο εκτός της βάσης μέσα στον πίνακα της βάσης, έχοντας τα περιεχόμενα ως τις εγγραφές στον πίνακα της βάσης που επιλέξαμε. Η πρώτη εγγραφή του πίνακα της βάσης αντιστοιχεί με τα περιεχόμενα των πρώτων γραμμών μέχρι τον πρώτο χαρακτήρα '|' του πίνακα που βρισκόταν εκτός της βάσης. Επαναλαμβάνουμε τις εντολές αυτές για όλους τους πίνακες που θέλουμε να μεταφέρουμε τα δεδομένα τους, ως τις εγγραφές των πινάκων της βάσης μας. Όταν έχουμε τελειώσει με το γέμισμα των πινάκων της βάσης, τότε συνεχίζουμε με την εκτέλεση των αρχείων τύπου shell script (SH), δηλαδή script αρχεία που το κάθε ένα θα περιέχει δεκαεπτά επερωτήσεις προς εκτέλεση. Παράδειγμα ενός τέτοιου αρχείου είχαμε

δείξει στο προηγούμενο κεφάλαιο 3. Και ακολούθως τρέχουμε για το πρώτο πείραμα ένα αρχείο nameOfscript.sh με την εντολή :

```
screen bash ./nameOfscript.sh
```

ο λόγος που έχουμε τη λέξη screen είναι επειδή τρέχουμε εξ' αποστάσεως οτιδήποτε, θέλουμε να συνεχίσει να εκτελείται το αρχείο τύπου shell script με όνομα nameOfscript.sh, ακόμη και όταν εμείς θα έχουμε απενεργοποιήσει τον προσωπικό μας υπολογιστή.

Για το πρώτο πείραμα της βάσης αυτής για ένα πυρήνα, εμείς θα έχουμε το αρχείο τύπου shell script που είχαμε χρησιμοποιήσει στο πρώτο πείραμα για τη βάση του ενός Gigabyte σε επαναλήψεις των είκοσι για καλύτερα αποτελέσματα και θα εκτελούνται οι δεκαεπτά επερωτήσεις σε κάθε επανάληψη, σύμφωνα με την σειρά που έχει καθοριστεί από τα TPC-H Benchmarks (βλέπε Appendix A). Μετά για το δεύτερο πείραμα θα έχω τα δύο αρχεία τύπου shell script που χρησιμοποίησα στο δεύτερο πείραμα της βάσης του ενός Gigabyte. Εκτελώ δύο φορές την εντολή για να εκτελεστούν τα αρχεία τύπου shell script για δύο πυρήνες ταυτόχρονα. Παρομοίως για το τρίτο πείραμα με τέσσερις πυρήνες και για το τέταρτο πείραμα με του οχτώ πυρήνες.

5.3 Πειραματική Αξιολόγηση Πειράματος

Μεθοδολογία Πειραματικής Αξιολόγησης

Στο πρώτο πείραμα που πραγματοποιείται μόνο για έναν πυρήνα θα εξαχθούν συνολικά δεκαεπτά αρχεία κειμένου με τα αποτελέσματα των επερωτήσεων, δηλαδή για κάθε τέτοιο αρχείο έχουμε τις εγγραφές που απαντούν την επερώτηση μέσω των πινάκων της βάσης δεδομένων που έχουμε αρχικά δημιουργήσει. Επίσης θα εξαχθούν και δεκαεπτά αρχεία κειμένου που θα υπάρχουν οι χρόνοι που έκαναν να ολοκληρωθούν οι δεκαεπτά επερωτήσεις και σε κάθε τέτοιο αρχείο έχουμε είκοσι χρόνους της εκτέλεσης της επερώτησης. Ο λόγος που η επερώτηση έχει εκτελεσθεί είκοσι φορές είναι επειδή στην μηχανή μπορούν να δουλεύουν και άλλοι χρήστες και επίσης για να έχουμε πιο σωστά αποτελέσματα. Για το δεύτερο πείραμα θα έχουμε διπλάσιο αριθμό αρχείων γιατί εκτελούμε σε δύο πυρήνες τις δεκαεπτά επερωτήσεις. Παρομοίως και για τα άλλα δύο πειράματα που πραγματοποιήθηκαν στη βάση για τέσσερις και οχτώ πυρήνες.

Αποτελέσματα Πειραματικής Αξιολόγησης

Με τις μετρήσεις των χρόνων που έχουμε στα αρχεία κειμένου με τους χρόνους της κάθε επερώτησης αναλύουμε για να φτιάξουμε τις γραφικές παραστάσεις με τα τελικά μας αποτελέσματα. Πιο κάτω στον πίνακα 5.1 βλέπουμε τις τελικές μετρήσεις για όλα τα πειράματα που έχουν πραγματοποιηθεί στην βάση δεδομένων των δέκα Gigabyte (10Gb). Ο χρόνος εκτέλεσης είναι σε λεπτά. Ο μέσος χρόνος (Average) είναι ο μέσος όρος από τους ολικούς μέγιστους χρόνους που έχουμε από τις είκοσι εκτελέσεις όλων των επερωτήσεων. Με τον ολικό μέγιστο χρόνο σε μία εκτέλεση του πρώτου πειράματος σε ένα πυρήνα(1core) εννοούμε τον μέγιστο συνολικά χρόνο που είναι ο ίδιος με το άθροισμα των δεκαεπτά τελικών χρόνων εκτέλεσης των δεκαεπτά επερωτήσεων. Για το δεύτερο πείραμα με δύο πυρήνες έχουμε τα αποτελέσματα των είκοσι εκτελέσεων των δεκαεπτά επερωτήσεων για τον πρώτο πυρήνα και τα αποτελέσματα των είκοσι εκτελέσεων των δεκαεπτά επερωτήσεων για τον δεύτερο πυρήνα. Αθροίζουμε για κάθε εκτέλεση των δεκαεπτά επερωτήσεων τους χρόνους εκτέλεσής τους για τη συγκεκριμένη εκτέλεση. Παρομοίως πράττουμε και για τη κάθε εκτέλεση των δεκαεπτά επερωτήσεων στο δεύτερο πυρήνα. Ακολουθώς παίρνουμε το μέγιστο χρόνο από τα αθροίσματα των δύο πυρήνων για την συγκεκριμένη εκτέλεση και τέλος έχουμε τον χρόνο με τον μέσο όρο από τους μέγιστους χρόνους και των είκοσι εκτελέσεων. Για το τρίτο πείραμα με τέσσερις πυρήνες και για το τέταρτο πείραμα με οχτώ πυρήνες πράττουμε παρομοίως με το δεύτερο πείραμα. Με τη γραμμή plus στον πίνακα 5.1 εννοούμε τον επιπρόσθετο χρόνο που θα μπορούσε να έχει ο συνολικός χρόνος εκτέλεσης του όλου πειράματος. Αυτό ισχύει και για τα τέσσερα πειράματα που έγιναν στη νέα μας βάση. Ο χρόνος αυτός είναι ο μέγιστος χρόνος από όλους τους μέγιστους χρόνους που έχουμε από όλες τις εκτελέσεις αφαιρώντας τον μέσο όρων των μέγιστων χρόνων αυτών. Πιο αναλυτικά, αθροίζουμε για κάθε εκτέλεση για οποιοδήποτε πείραμα, τους χρόνους εκτέλεσης των επερωτήσεων για τον κάθε πυρήνα ξεχωριστά και ακολουθώς παίρνουμε τον μέγιστο χρόνο από αυτά τα αθροίσματα. Ακολουθώς παίρνουμε τον μέγιστο χρόνο από αυτούς τους είκοσι μέγιστους χρόνους των είκοσι εκτελέσεων και αφαιρούμε το μέσο όρο από είκοσι μέγιστους χρόνους αυτούς και έτσι έχουμε τελικά τον επιπρόσθετο μέγιστο χρόνο που θα μπορούσε να έχει συνολικά η εκτέλεση ολόκληρου του πειράματος. Με τη γραμμή minus στον πίνακα 5.1 εννοούμε τον ελάχιστο χρόνο που θα μπορούσε να έχει η όλη εκτέλεση του πειράματος. Αυτό ισχύει και για τα τέσσερα πειράματα που πραγματοποιήθηκαν για τη βάση δεδομένων αυτή. Ο χρόνος αυτός μπορεί να βρεθεί με το να αφαιρέσουμε από το μέσο όρο των μέγιστων χρόνων που είχαμε στη γραμμή average του πίνακα 5.1, τον ελάχιστο χρόνο που έχουμε βρει από όλους τους μέγιστους χρόνους των είκοσι εκτελέσεων.

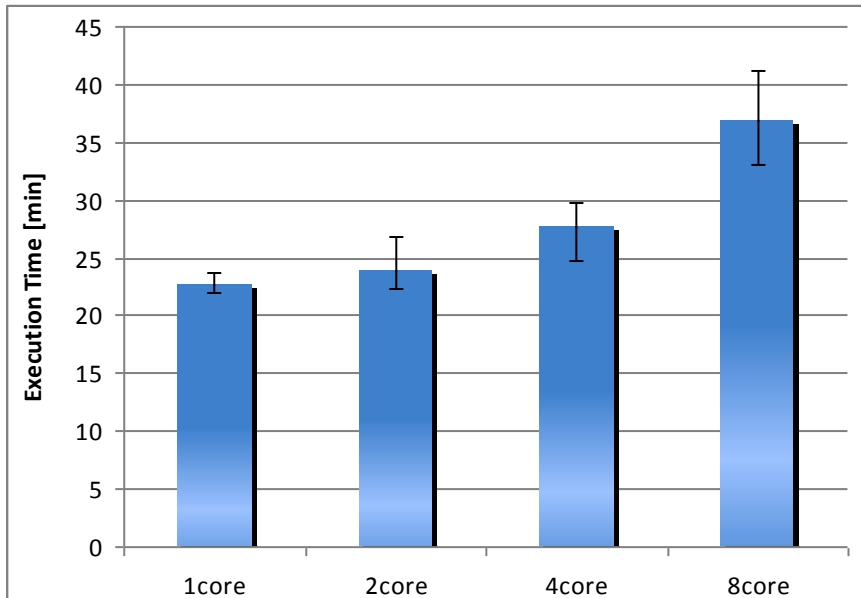
	1core	2core	4core	8core
Average	22,601337	23,856853	27,7279485	36,7832365
Plus	1,063013	3,025447	1,9877415	4,3412735
Minus	0,553207	1,507933	3,0448785	3,7828265

Πίνακας 5.1 Χρόνοι για την βάση δεδομένων των δέκα Gigabyte (10Gb)

Μπορούμε να δούμε πιο καθαρά τους χρόνους του πίνακα 5.1 στην πιο κάτω γραφική παράσταση 5.2 πώς φαίνονται.

Παρατηρήσεις

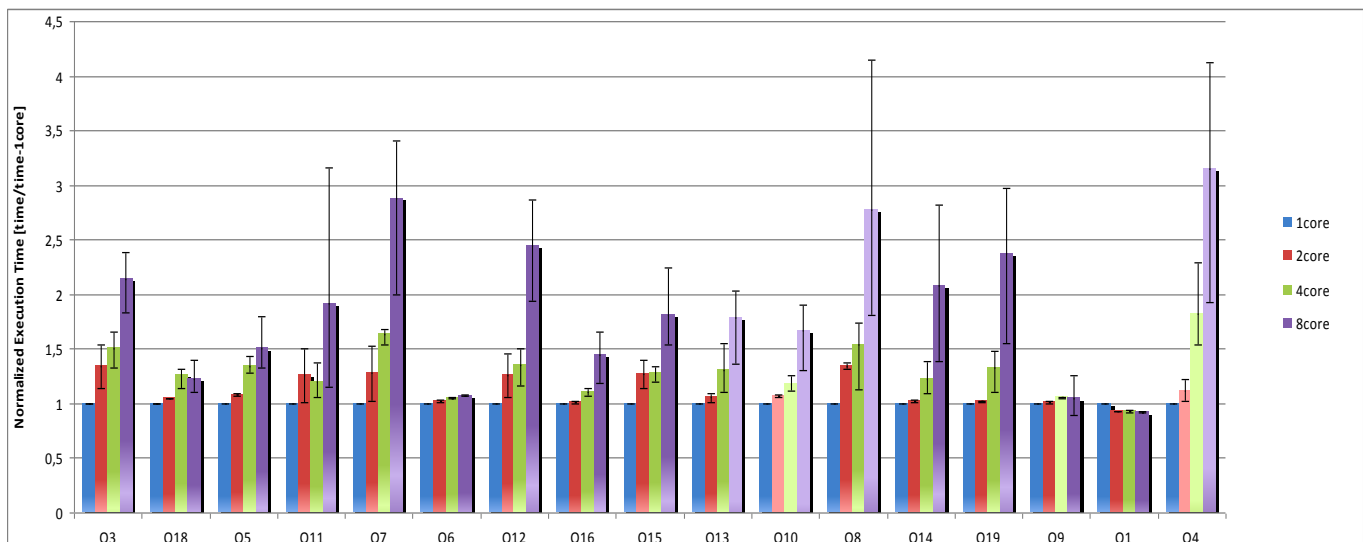
Στη γραφική παράσταση 5.2 οι χρόνοι της εκτέλεσης για το πρώτο πείραμα σε ένα πυρήνα έχει χρόνο είκοσι δύο λεπτά και εξήντα δευτερόλεπτα και για τις διπλάσιες επερωτήσεις στο δεύτερο πείραμα για δύο πυρήνες χρόνος εκτέλεσης είναι είκοσι τρία λεπτά και ογδόντα πέντε δευτερόλεπτα. Ενώ θα αναμέναμε να ήταν ο διπλάσιος χρόνος, με βάση το πρώτο πείραμα έχουμε σχεδόν τον ίδιο χρόνο εκτέλεσης. Αυτό συμβαίνει επειδή οι επερωτήσεις ξεκινούν ταυτόχρονα και τρέχουν παράλληλα στους δύο πυρήνες, δεκαεπτά επερωτήσεις στον πρώτο πυρήνα και δεκαεπτά επερωτήσεις στο δεύτερο πυρήνα. Μοιράζονται το φόρτο εργασίας οι δύο πυρήνες ενώ αν ήταν σε ένα πυρήνα και οι 34 επερωτήσεις θα είχαμε διπλάσιο χρόνο. Η μνήμη του cache δεν επιβαρύνεται με το να τρέχουμε σε δύο πυρήνες ταυτόχρονα και παράλληλα. Επίσης θα αναμέναμε να τελείωναν τον ίδιο χρόνο με το πρώτο πείραμα αφού τρέχουν οι ίδιες επερωτήσεις και στον πρώτο πυρήνα και στο δεύτερο. Έχουμε μία μικρή καθυστέρηση του ενός λεπτού και είκοσι πέντε δευτερόλεπτα, αυτή τη καθυστέρηση θα την μελετήσουμε αργότερα για κάθε επερωτηση ξεχωριστά για να δούμε τον ρόλο του cache και το ρόλο της σειράς εκτέλεσης των επερωτήσεων στα πειράματα αυτά. Παρομοίως για το τρίτο πείραμα έχουμε καθυστέρηση πέντε λεπτών και δώδεκα δευτερολέπτων περίπου και για το τέταρτο πείραμα έχουμε τη μεγαλύτερη καθυστέρηση με δεκατέσσερα λεπτά και δεκαοκτώ δευτερόλεπτα.



Γραφική παράσταση 5.2 Χρόνος εκτέλεσης σε λεπτά των τεσσάρων πειραμάτων

Στη γραφική παράσταση 5.2 φαίνονται με μικρές γραμμές κατακόρυφες, οι ελάχιστοι (minus) χρόνοι και οι μέγιστοι (plus) χρόνοι που θα μπορούσαν οι εκτελέσεις να ολοκληρωθούν στα πειράματα για ένα πυρήνα για δύο, τέσσερις και οχτώ πυρήνες.

Πιο κάτω στη γραφική παράσταση 5.3 έχουμε για όλες τις επερωτήσεις που εκτελέστηκαν τη συμπεριφορά του κανονικοποιημένου χρόνου εκτέλεσής τους ξεχωριστά.



Γραφική παράσταση 5.3 Κανονικοποιημένος Χρόνος εκτέλεσης

Παρατηρήσεις

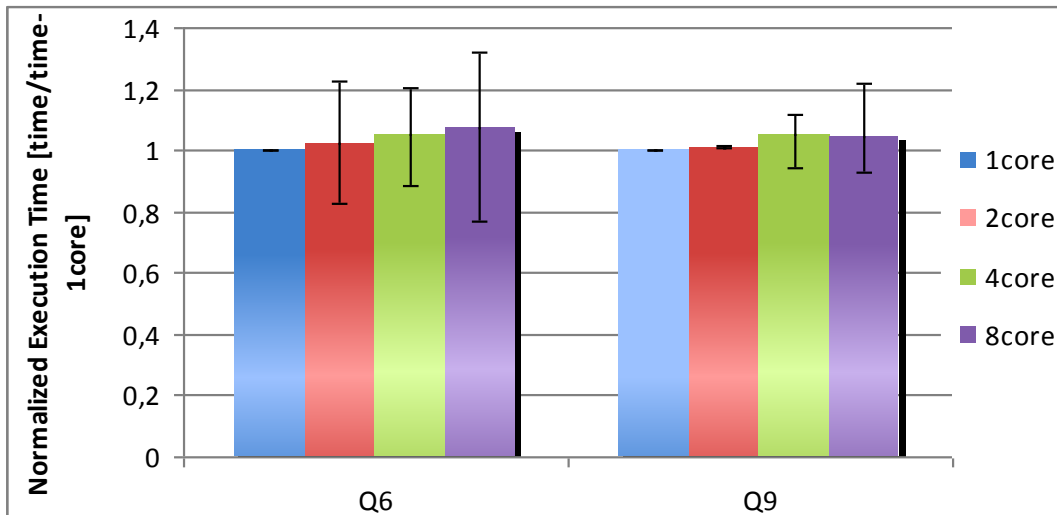
Στη γραφική παράσταση 5.3 βλέπουμε όλες τις επερωτήσεις που χρησιμοποιήθηκαν από τα TPC-H Benchmarks και τα λάθη (error bars) για την κάθε επερώτηση ξεχωριστά και για τα τέσσερα πειράματα. Στη γραφική παράσταση ξεχωρίζει με μεγάλη διαφορά ο κανονικοποιημένος χρόνος εκτέλεσης σχεδόν για όλες τις επερωτήσεις για το πείραμα σε οχτώ πυρήνες. Επίσης διαφορά αλλά όχι και τόσο μεγάλη από τον κανονικοποιημένο χρόνο για ένα πυρήνα έχει και ο κανονικοποιημένος χρόνος στο τρίτο πείραμα σχεδόν σε όλες τις επερωτήσεις με τους τέσσερις πυρήνες. Την πιο μικρή διαφορά κάνει ο κανονικοποιημένος χρόνος των επερωτήσεων για το πείραμα με τους δύο πυρήνες. Για να δείξουμε τους λόγους που συμβαίνει αυτό θα πρέπει να χωρίσουμε τις επερωτήσεις μας σε κατηγορίες για να μπορούμε να μελετήσουμε καλύτερα τη συμπεριφορά αυτή. Πιο κάτω θα αναφέρουμε τις κατηγορίες αυτές.

Κατηγορίες των επερωτήσεων

Για να δείξουμε τους λόγους που συμβαίνουν αυτές οι συμπεριφορές των επερωτήσεων, θα πρέπει να χωρίσουμε τις επερωτήσεις μας σε κατηγορίες. Με τις κατηγορίες αυτές θα μπορούμε πιο εύκολα να αναλύσουμε την συμπεριφορά των επερωτήσεων που εκτελέστηκαν στα πειράματα της βάσης δεδομένων των δέκα Gigabyte (10Gb). Οι κατηγορίες αυτές θα είναι σε σχέση με την αλλαγή του αριθμού πυρήνων και την εκτέλεση με άλλες επερωτήσεις μαζί έχοντας ως βασικό σενάριο το πείραμα σε έναν πυρήνα.

Κατηγορία 1

Στην πρώτη κατηγορία θα έχουμε τις επερωτήσεις αυτές, όπου με την αλλαγή του αριθμού των πυρήνων δεν έχουμε σημαντική αλλαγή στα αποτελέσματα, δηλαδή ο κανονικοποιημένος χρόνος εκτέλεσης των επερωτήσεων δεν επηρεάζεται με την εκτέλεση της κάθε επερώτησης με τις άλλες επερωτήσεις. Οι επερωτήσεις που ανταποκρίνονται στην κατηγορία αυτή είναι οι επερωτήσεις q6 και q9 όπως φαίνονται στη γραφική παράσταση 5.4.



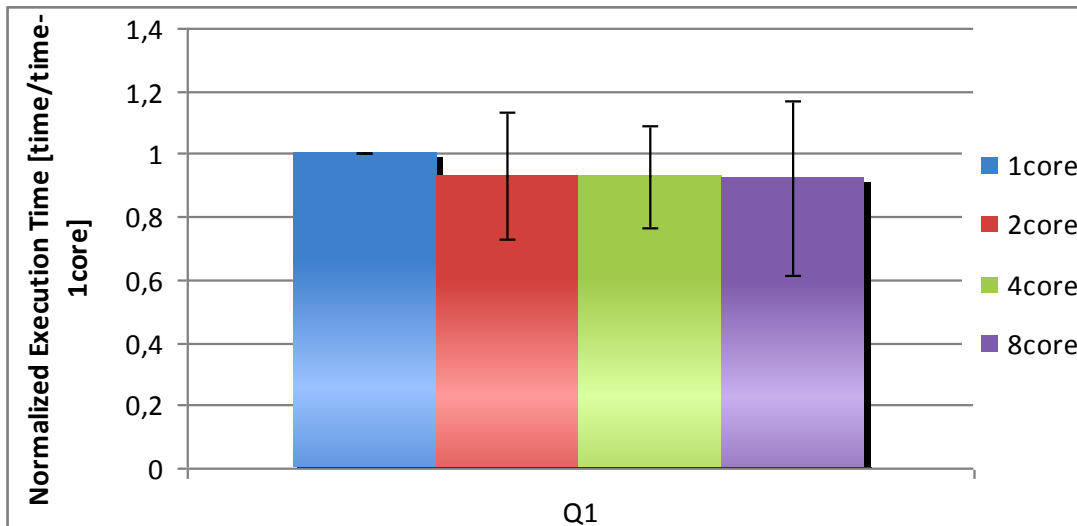
Γραφική παράσταση 5.4 Κανονικοποιημένος Χρόνος Εκτέλεσης

Παρατηρήσεις

Στη γραφική παράσταση 5.4 φαίνονται οι επερωτήσεις για την πρώτη κατηγορία. Οι επερωτήσεις q6 και q9 έχουν σχεδόν την ίδια συμπεριφορά και στα τέσσερα πειράματα, συνεπώς με την αλλαγή του αριθμού των πυρήνων δεν έχουμε σημαντική αλλαγή στα αποτελέσματα, και ο κανονικοποιημένος χρόνος εκτέλεσης των επερωτήσεων δεν επηρεάζεται με την εκτέλεση με τις άλλες επερωτήσεις. Για τις επερωτήσεις αυτές έχουμε εξίσου επεξεργασία και διάβασμα από τη μνήμη. Η επερώτηση q6 χρησιμοποιεί sort και διαβάζει ελάχιστα από τη μνήμη έχει σχεδόν ίδιους χρόνους για όλα τα πειράματα συνεπώς δεν επηρεάζεται από τον αριθμό των επερωτήσεων και την εκτέλεση με άλλες επερωτήσεις όταν η χρησιμοποίηση του cpu για i/o είναι λιγότερο από 50%. Για την επερώτηση q9 που χρησιμοποιεί cpu για i/o λιγότερο από 50% και nested loop join όπου παρατηρούμε πως ελάχιστα αλλάζει με την εκτέλεση με περισσότερες επερωτήσεις στους τέσσερις και οχτώ πυρήνες.

Κατηγορία 2

Στην δεύτερη κατηγορία θα έχουμε την επερώτηση q1 όπως φαίνεται στη γραφική παράσταση 5.5, όπου με την αλλαγή του αριθμού των πυρήνων έχουμε μικρή αλλαγή προς τα κάτω στα αποτελέσματα σε σχέση με το πρώτο πείραμα σενάριο, δηλαδή ο κανονικοποιημένος χρόνος εκτέλεσης της επερώτησης επηρεάζεται με την εκτέλεση της επερώτησης με τις άλλες επερωτήσεις αλλά με καλύτερα αποτελέσματα.



Γραφική παράσταση 5.5 Κανονικοποιημένος Χρόνος Εκτέλεσης

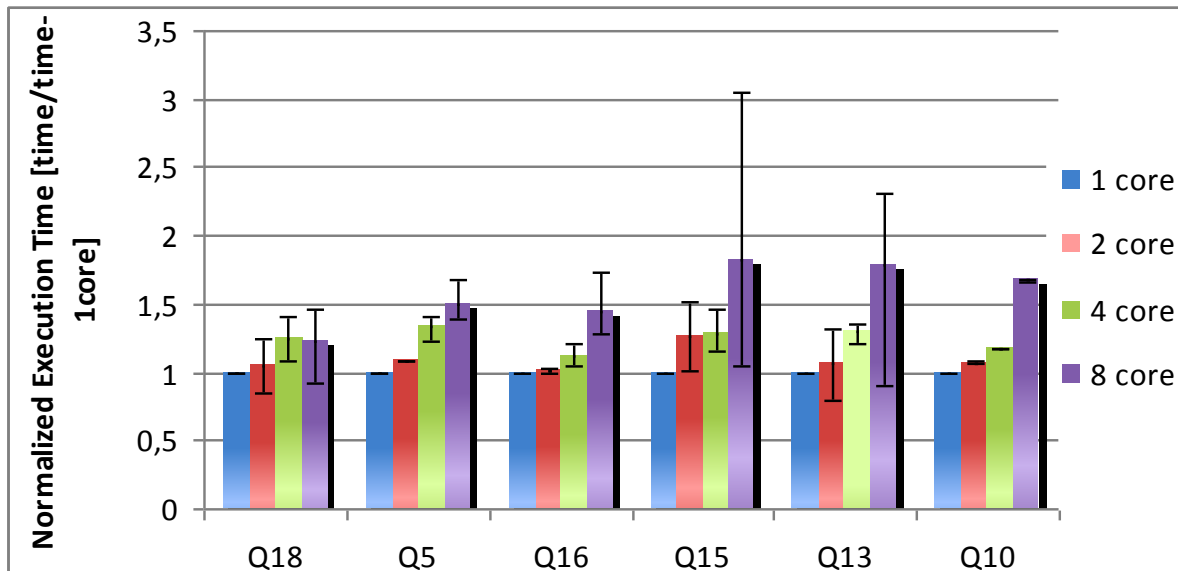
Παρατηρήσεις

Στη γραφική παράσταση 5.5 φαίνεται η επερώτηση για την δεύτερη κατηγορία. Η επερώτηση q1 έχει την ίδια συμπεριφορά για τα πειράματα με δύο, τέσσερις και οχτώ πυρήνες και επίσης έχει καλύτερο κανονικοποιημένο χρόνο από ότι στο βασικό μας σενάριο δηλαδή στο πείραμα με ένα πυρήνα. Θα αναμέναμε από την επερώτηση αυτή να είχε παρόμοια συμπεριφορά όπως είχε και στην βάση του ενός Gigabyte, όμως βλέπουμε ότι διαφέρει κατά πολύ. Επίσης αυτή η επερώτηση έχει διαφορετική συμπεριφορά από όλες τις άλλες επερωτήσεις σε αυτή τη βάση. Οι λόγοι για αυτή τη συμπεριφορά είναι αρχικά ο όγκος δεδομένων στη βάση αυτή των δέκα Gigabyte που δημιουργεί τη διαφορά αυτή και δεν επηρεάζεται όπως βλέπουμε από το χώρο στην κρυφή μνήμη. Αυτό σημαίνει πως χρειάζεται λιγότερο χώρο στη κρυφή μνήμη από όσο χρειαζότανε για το βασικό μας σενάριο με ένα πυρήνα. Επίσης ο συνδυασμός των επερωτήσεων που εκτελούνται πριν από την επερώτηση q1 στα πειράματα με δύο, τέσσερις και οχτώ πυρήνες, επηρεάζει τον χρόνο εκτέλεσης της q1 και κατά συνέπεια και τον κανονικοποιημένο χρόνο εκτέλεσης. Αυτό θα το αποδείξουμε στη συνέχεια μετά την κατηγοριοποίηση των αποτελεσμάτων. Για την επερώτηση αυτή, έχουμε τη λειτουργία sequential scan μόνο και πολύ μικρό ποσοστό χρησιμοποίησης της cpu για i/o, παρατηρούμε ότι ωφελείται ο χρόνος εκτέλεσης κατά την αύξηση των πυρήνων και δεν επηρεάζεται τόσο πολύ με την εκτέλεση με άλλες επερωτήσεις.

Κατηγορία 3

Στην τρίτη κατηγορία θα έχουμε τις επερωτήσεις αυτές, όπου με την αλλαγή του αριθμού των πυρήνων έχουμε κάποια σημαντική αλλαγή στα αποτελέσματα, δηλαδή ο κανονικοποιημένος χρόνος εκτέλεσης των επερωτήσεων επηρεάζεται με την εκτέλεση της

κάθε επερώτησης με τις άλλες επερωτήσεις. Οι επερωτήσεις που έχουμε στην κατηγορία αυτή είναι οι επερωτήσεις q5, q10, q13, q15, q16 και q18 όπως φαίνονται στη γραφική παράσταση 5.6.



Γραφική παράσταση 5.6 Κανονικοποιημένος Χρόνος Εκτέλεσης

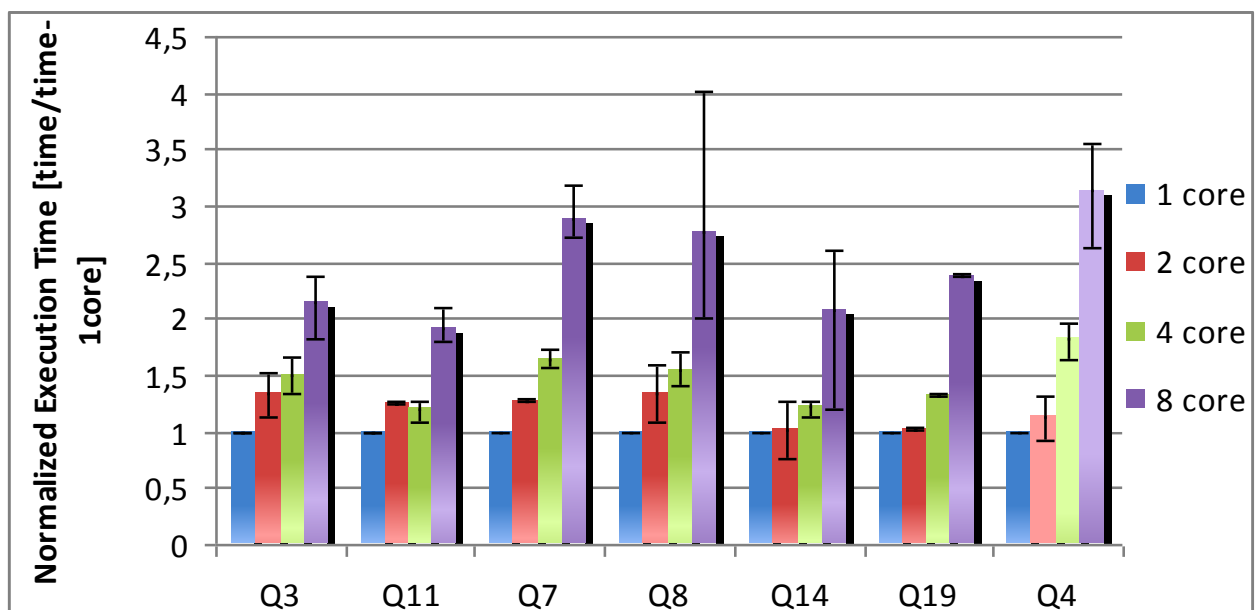
Παρατηρήσεις

Στη γραφική παράσταση 5.6 φαίνονται οι επερωτήσεις για την τρίτη κατηγορία. Υπάρχει αλλαγή στα αποτελέσματα μεταξύ των τεσσάρων πειραμάτων, διότι ο κανονικοποιημένος χρόνος εκτέλεσης των επερωτήσεων είναι διαφορετικός σε κάθε πείραμα για την κάθε επερώτηση και αυξάνεται με την αλλαγή της αύξησης του αριθμού πυρήνων στο κάθε πείραμα. Επίσης ο κανονικοποιημένος χρόνος επηρεάζεται με την εκτέλεση της κάθε επερώτησης με τις άλλες επερωτήσεις. Ο ρυθμός αύξησης του κανονικοποιημένου χρόνου στα πρώτα τρία πειράματα δηλαδή για ένα πυρήνα, δύο και τέσσερις πυρήνες είναι σταδιακά προς τα πάνω σχεδόν για όλες τις επερωτήσεις. Για το πείραμα με τους οχτώ πυρήνες έχουμε αυξημένο κανονικοποιημένο χρόνο εκτός από την επερώτηση q18 που έχουμε μικρότερο κανονικοποιημένο χρόνο από το τρίτο πείραμα στους τέσσερις πυρήνες. Ενώ ξεχωρίζουμε για τις επερωτήσεις q13, q15 να έχουμε μεγάλη αύξηση του κανονικοποιημένου χρόνου στο τέταρτο πείραμα με τους οχτώ πυρήνες από ότι οι άλλες επερωτήσεις. Μετά από τη μελέτη της συμπεριφοράς της εκτέλεσης των επερωτήσεων που ανήκουν στη τρίτη κατηγορία αυτή, οι διακυμάνσεις ήταν πολύ μικρές και ως αιτία έχουμε τον χώρο στην κρυφή μνήμη (cache) γιατί έχουμε δεκαπλάσιο όγκο δεδομένων από όσο είχαμε στα άλλα τρία πειράματα

με ένα, δύο και τέσσερις πυρήνες. Επίσης πολύ σημαντικό ρόλο για αυτή τη συμπεριφορά του κανονικοποιημένου χρόνου των επερωτήσεων έχει και ο συνδυασμός των επερωτήσεων. Δηλαδή με οποιοδήποτε άλλο συνδυασμό επερωτήσεων θα έχουμε διαφορετικούς χρόνους για όλες τις επερωτήσεις, αυτό θα το αποδείξουμε αργότερα. Για όλες τις επερωτήσεις σε αυτή την κατηγορία έχουμε την λειτουργία του nested loop join, επεξεργάζονται περισσότερο παρά να διαβάζουν από τη μνήμη. Για τις επερωτήσεις q18,q5 και q16 χρησιμοποιούν nested loop join και sort έχουμε καλύτερους χρόνους όταν πλησιάζει το 50% cpu για i/o δηλαδή τους χρόνους για την επερώτηση q16. Οι επερωτήσεις q15 και q10 που χρησιμοποιούν μόνο nested loop join είναι καλύτερα να εκτελείται όταν χρησιμοποιεί λιγότερο από 50% του cpu για i/o. Και για την επερώτηση q13 που χρησιμοποιεί hash join έχει χειρότερα αποτελέσματα από τις επερωτήσεις που χρησιμοποίησαν nested loop join σε συνδυασμό με sort αλλά έχει καλύτερους χρόνους από τις επερωτήσεις με nested loop join με cpu πέραν των 50% για i/o.

Κατηγορία 4

Στην τέταρτη κατηγορία θα έχουμε τις επερωτήσεις αυτές, όπου με την αλλαγή του αριθμού των πυρήνων έχουμε μεγάλη αλλαγή στα αποτελέσματα, δηλαδή ο κανονικοποιημένος χρόνος εκτέλεσης των επερωτήσεων επηρεάζεται με την εκτέλεση της κάθε επερώτησης με τις άλλες επερωτήσεις πάρα πολύ. Οι επερωτήσεις που έχουμε στην κατηγορία αυτή είναι οι επερωτήσεις q3, q4, q7, q8, q11, q14 και q19 όπως φαίνονται στη γραφική παράσταση 5.7.



Γραφική παράσταση 5.7 Κανονικοποιημένος Χρόνος Εκτέλεσης

Παρατηρήσεις

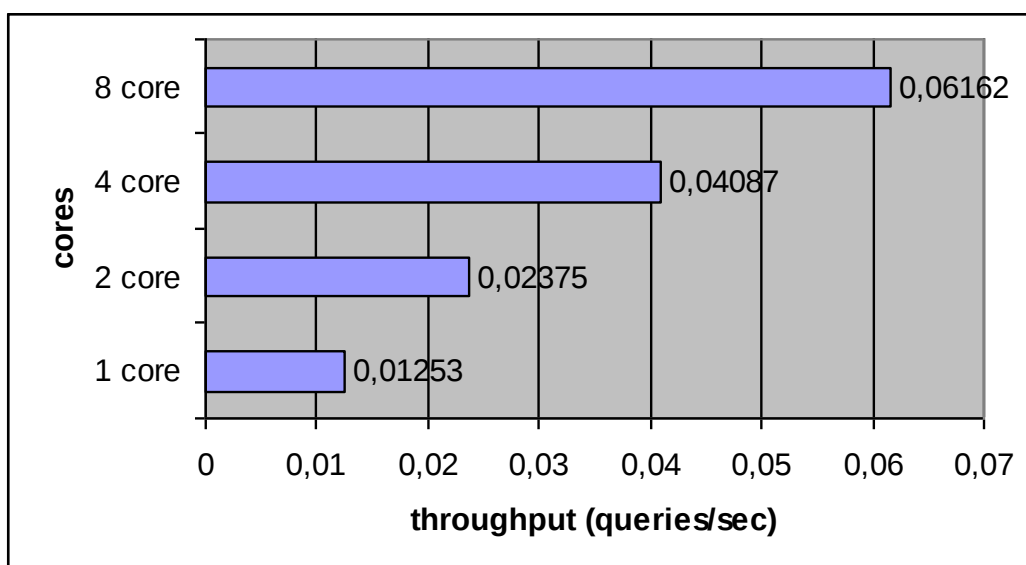
Στη γραφική παράσταση 5.7 φαίνονται καθαρά οι μεγάλες αυξομειώσεις χρόνου στις επερωτήσεις q8 και q14, με τα errors bars, για το πείραμα με τους οχτώ πυρήνες. Οι εκτελέσεις των επερωτήσεων αυτών επηρεάζεται πολύ με την εκτέλεση μαζί με τις άλλες επερωτήσεις γιατί έχουμε μεγάλες διακυμάνσεις μεταξύ των χρόνων τους κατά την εκτέλεσή τους. Στο πείραμα στους δύο πυρήνες έχουμε σχετικά μικρή αύξηση του κανονικοποιημένου χρόνου σε σύγκριση με το πρώτο πείραμα με ένα πυρήνα. Στο τρίτο πείραμα με τέσσερις πυρήνες έχουμε αρκετά μεγάλη διαφορά σε σύγκριση με το πρώτο πείραμα στον ένα πυρήνα και στο τέταρτο πείραμα έχουμε τη μεγαλύτερη διαφορά που ξεπερνά τα δύο λεπτά σχεδόν για όλες τις επερωτήσεις εκτός από την επερώτηση q11. Επίσης για το πείραμα τέσσερα με τους οχτώ πυρήνες έχουμε μεγάλο ρυθμό αυξομείωσης των διακυμάνσεων μεταξύ των εκτελέσεων των επερωτήσεων, με τη συμπεριφορά αυτή ο κανονικοποιημένος χρόνος των χρόνων εκτέλεσης που είχαμε από τις επερωτήσεις της κατηγορίας αυτής, σε όλα τα πειράματα επηρεάζεται πάρα πολύ με την αλλαγή του αριθμού των πυρήνων και την εκτέλεση των επερωτήσεων αυτών με τις υπόλοιπες επερωτήσεις. Εάν είχαμε σταθερά μεγάλους χρόνους θα είχαμε ως αιτία το χώρο στην κρυφή μνήμη, όμως λόγω των μεγάλων διακυμάνσεων που έχουμε στους χρόνους των επερωτήσεων στην κατηγορία αυτή, δεν ισχύει. Για αυτές τις αλλαγές φταίει η σειρά εκτέλεσης των επερωτήσεων δηλαδή ο συνδυασμός επερωτήσεων προς εκτέλεση και ο όγκος δεδομένων που θα ανακτήσει η κάθε επερώτηση από τη βάση. Οι επερωτήσεις q3, q7 και q8 hash join σε συνδυασμό με nested loop join με διαφορετικές τιμές cpu για i/o, παρατηρούμε από τη γραφική ότι έχουμε καλύτερους χρόνους όταν έχουμε περισσότερο από 50% cpu για i/o. Οι επερωτήσεις q14 και q4 χρησιμοποιούν index scan και index scan σε συνδυασμό με nested loop join αντίστοιχα, παρατηρούμε ότι έχουμε καλύτερους χρόνους όταν χρησιμοποιείται index scan για λιγότερο από 50% cpu για i/o μέχρι τους τέσσερις πυρήνες. Οι επερωτήσεις q11 και q19 που χρησιμοποιούν nested loop join έχουμε πολύ καλύτερους χρόνους όταν χρησιμοποιείται πέραν του 50% cpu για i/o.

Πυθμοαπόδοση

Η ρυθμοαπόδοση (throughput) για τη βάση δεδομένων των δέκα Gigabyte (10Gb) φαίνεται πιο κάτω στο πίνακα 5.2 και στη γραφική παράσταση 5.8 των επερωτήσεων συνολικά που έχουμε εκτελέσει. Με τη ρυθμοαπόδοση δείχνουμε πόσες επερωτήσεις έχουν εκτελεστεί ανά μονάδα χρόνου σε δευτερόλεπτα (queries/sec). Έχοντας όλους τους χρόνους που εκτελέστηκαν από όλες τις επερωτήσεις σε όλα τα πειράματα που πραγματοποιήθηκαν, παίρνουμε ένα μέσο όρο από τα αθροίσματα από όλες τις επαναλήψεις των εκτελέσεων (Sum of queries average(sec)). Ακολούθως θα πάρουμε για το πρώτο πείραμα στον ένα πυρήνα, τον αριθμό των επερωτήσεων, στην συγκεκριμένη περίπτωση είναι ίσος με δεκαεπτά, και θα διαιρέσουμε με το άθροισμα των χρόνων αυτών. Με αυτό τον τρόπο μπορούμε να δείξουμε πόσες επερωτήσεις έχουν εκτελεστεί ανά δευτερόλεπτο (throughput (queries/sec)), στη συγκεκριμένη περίπτωση έχουμε 0,01253 επερωτήσεις που εκτελούνται ανά δευτερόλεπτο.

Cores	1core	2core	4core	8core
Number of Queries	17	34	68	136
Sum of queries average(sec)	(22,601337)x60	(23,856853)x60	(27,7279485)x60	(36,7832365)x60
Throughput(queries/sec)	0,01253	0,02375	0,04087	0,06162

Πίνακας 5.2 Ρυθμοαπόδοση για την βάση δεδομένων των δέκα Gigabyte (10Gb)



Γραφική παράσταση 5.8 Ρυθμοαπόδοση (Throughput)

Παρατηρήσεις

Για το πείραμα με δύο πυρήνες έχουμε ρυθμοαπόδοση ίση με το πηλίκο του συνολικού αριθμού επερωτήσεων (δηλαδή 34) και του αθροίσματος των χρόνων από όλες τις εκτελέσεις και των 34 επερωτήσεων, πολλαπλασιάζουμε με 60 για να το μετατρέψουμε σε δευτερόλεπτα. Το αποτέλεσμα που έχουμε είναι 0,02375 δηλαδή παρόλο που εκτελούνται οι διπλάσιες επερωτήσεις απ' ότι στο πρώτο πείραμα με ένα πυρήνα, βλέπουμε ότι ο αριθμός των επερωτήσεων που εκτελούνται υπερέχει από τον αριθμό 0,01253, σχεδόν διπλάσιο. Συνεπώς βλέπουμε ότι με τον παραλληλισμό που έχουμε χρησιμοποιήσει στο δεύτερο πείραμα, έχουμε καλύτερα αποτελέσματα. Έχουμε βελτιωμένη επίδοση (performance) στο δεύτερο πείραμα, 1,89 φορές καλύτερη ρυθμοαπόδοση, δηλαδή σχεδόν δύο φορές τον αριθμό επερωτήσεων που μπορούν να εκτελεστούν, που το έχουμε δείξει μέσω της ρυθμοαπόδοσης (throughput) της μηχανής.

Για τα άλλα δύο πειράματα με τέσσερις πυρήνες και οχτώ πυρήνες πράττουμε παρομοίως και έχουμε τιμή της ρυθμοαπόδοσης ίση με 0,04087 και 0,06162 αντίστοιχα. Για το πείραμα με τέσσερις πυρήνες, η ρυθμοαπόδοση ξεπερνάει τον τριπλάσιο αριθμό επερωτήσεων (3,26 φορές) που μπορούν να εκτελεστούν απ' ότι τη ρυθμοαπόδοση σε ένα πυρήνα, δεν είναι η αναμενόμενη δηλαδή 3,78 φορές, όμως και πάλι έχουμε ακόμη περισσότερη βελτίωση στην επίδοση της μηχανής. Για το πείραμα με οχτώ πυρήνες θα αναμέναμε να ήταν 6-7 φορές καλύτερη τιμή για την ρυθμοαπόδοση σε σύγκριση με τη τιμή της ρυθμοαπόδοσης για ένα πυρήνα και κατά συνέπεια καλύτερη επίδοση από το αρχικό μας πείραμα. Βλέπουμε όμως ότι η τιμή της ρυθμοαπόδοσης είναι σχεδόν πενταπλάσια (4,91 φορές) από τη τιμή της ρυθμοαπόδοσης στον ένα πυρήνα. Έχουμε καλύτερη ρυθμοαπόδοση από όλα τα άλλα πειράματα όμως δεν έχουμε την αναμενόμενη, χάνουμε ένα αρκετά καλό ποσοστό, για αυτό έχουμε ως αιτία που δημιουργείται η καθυστέρηση και χάνεται αυτό το ποσοστό χρόνου, τον αριθμό επερωτήσεων και τον όγκο πληροφοριών, τη σειρά που εκτελούνται και ο ρόλος της κρυφής μνήμης. Με τη σειρά που εκτελούνται εννοούμε ότι σε μία συγκεκριμένη μονάδα χρόνου μπορεί να έχουμε πολλές επερωτήσεις που να ζητούν μεγάλο όγκο πληροφοριών από τη βάση και να καθυστερούν τη εκτέλεση των επόμενων επερωτήσεων που είναι σε αναμονή. Επίσης μπορεί να έχουμε το γέμισμα της ιδιωτικής μνήμης του πυρήνα και να δημιουργείται περισσότερη καθυστέρηση, αφού ο κάθε πυρήνας στη μελέτη μας εκτελεί 17 επερωτήσεις παρόλο που έχουμε συνεργασία και μοιράζονται οι πληροφορίες μεταξύ των διεργασιών στο συγκεκριμένο πυρήνα σε κάποια στιγμή ο χώρος στην ιδιωτική μνήμη του πυρήνα θα φτάσει

μέχρι το όριο που έχει και θα έχουμε καθυστέρηση συνολικά διότι θα χρειαστεί περισσότερος χρόνος για να ολοκληρωθούν οι διεργασίες που εκτελούνταν και περισσότερο χρόνο αναμονής των επόμενων επερωτήσεων προς εκτέλεση. Επίσης μπορεί να γεμίσει και η κοινή μνήμη μεταξύ των πυρήνων που και πάλι έχουμε ως θετικό στοιχείο τη συνεργασία και μοίρασμα πληροφοριών μεταξύ των πυρήνων, αλλά και πάλι σε κάποια στιγμή μπορεί να φτάσει στο όριο του χώρου της και να δημιουργηθεί καθυστέρηση. Συνεπώς βλέπουμε στη συγκεκριμένη περίπτωση του πειράματος με τους οχτώ πυρήνες να εξυπηρετούνται περισσότερες επερωτήσεις από όλα τα άλλα πειράματα όμως επίσης βλέπουμε ότι μειώνεται πιο πολύ η ρυθμοαπόδοση από την αναμενόμενη ρυθμοαπόδοση.

Για μια εταιρία που χρησιμοποιεί παρόμοιο μέγεθος βάσης είναι καλύτερα να δουλεύει με 8 πυρήνες έχουμε τη μεγαλύτερη ρυθμοαπόδοση όμως πέφτει η επίδοση σε σύγκριση με τον αναμενόμενο, παρόλα αυτά έχουμε πολύ περισσότερες επερωτήσεις να εκτελούνται.

Αυτά τα προβλήματα μπορούμε να τα λύσουμε σίγουρα εάν λιγοστέψουμε τις επερωτήσεις για τους οχτώ πυρήνες είτε να αυξήσω τον αριθμό πυρήνων και να έχω τον ίδιο αριθμό επερωτήσεων, αλλά για να λύσουμε το υπάρχον πρόβλημα χωρίς να κάνουμε τις αλλαγές αυτές, πρέπει να θέσουμε διαφορετικό συνδυασμό επερωτήσεων προς εκτέλεση από τους πυρήνες, έτσι που να μην έχω εκτέλεση πολλών επερωτήσεων που ζητούν μεγάλο όγκο πληροφοριών από τη βάση την ίδια χρονική στιγμή. Θα μπορούσα να έχω λιγότερες τέτοιες επερωτήσεις και να έχω συνδυασμό με επερωτήσεις που ζητούν λιγότερες πληροφορίες από τη βάση δηλαδή οι επερωτήσεις που θα συνδυάσω να είναι πολύ πιο μικρός ο χρόνος εκτέλεσής τους, και έτσι θα έχω λιγότερη καθυστέρηση συνολικά στο τέλος, και κατά συνέπεια καλύτερη ρυθμοαπόδοση και συνολική επίδοση.

5.4 Συμπεράσματα

Με την εκτέλεση των επερωτήσεων στα τέσσερα πειράματα που έχουμε πραγματοποιήσει σε αυτή τη βάση δεδομένων των δέκα Gigabyte, μπορούμε να έχουμε σε σύγκριση με το πρώτο πείραμα σε ένα πυρήνα, καμία αλλαγή στα αποτελέσματα, είτε σημαντική αλλαγή με μείωση του χρόνου, είτε σημαντική αλλαγή με αύξηση του χρόνου, είτε μεγάλη αλλαγή στα αποτελέσματα μας.

Καμία αλλαγή έχουμε όταν με την αλλαγή του αριθμού των πυρήνων, ο κανονικοποιημένος χρόνος εκτέλεσης των επερωτήσεων δεν επηρεάζεται με την εκτέλεση της κάθε επερωτήσης με τις άλλες επερωτήσεις.

Σημαντική αλλαγή με μείωση του χρόνου, έχουμε όταν με την αλλαγή του αριθμού των πυρήνων, ο κανονικοποιημένος χρόνος εκτέλεσης των επερωτήσεων επηρεάζεται με την εκτέλεση της κάθε επερώτησης με τις άλλες επερωτήσεις όμως με θετικό τρόπο. Θετικό γιατί έχουμε καλύτερο χρόνο από ότι είχαμε στο πρώτο πείραμα με τον ένα πυρήνα. Συνεπώς σημαντικό ρόλο έχουμε για τη σειρά εκτέλεσης των επερωτήσεων, που σε μια δεδομένη στιγμή που εκτελείται μία συγκεκριμένη επερώτηση με ένα συγκεκριμένο συνδυασμό επερωτήσεων σε οποιοδήποτε πείραμα κι αν εκτελείται, δεν έχουμε μεγάλο όγκο πληροφοριών να ανακτώνται από τη βάση δεδομένων και έτσι δεν έχουμε καθυστέρηση στον ολικό χρόνο εκτέλεσής της.

Σημαντική αλλαγή με αύξηση του χρόνου, έχουμε όταν με την αλλαγή του αριθμού των πυρήνων, ο κανονικοποιημένος χρόνος εκτέλεσης των επερωτήσεων επηρεάζεται αρκετά με την εκτέλεση της κάθε επερώτησης με τις άλλες επερωτήσεις. Έχουμε μικρές αυξομειώσεις των χρόνων εκτέλεσης των επερωτήσεων σε όλα τα πειράματα. Λόγοι για τη συμπεριφορά αυτή είναι ο χώρος στην κρυφή μνήμη (cache) και ελάχιστα ο συνδυασμός των επερωτήσεων που εκτελούνται.

Μεγάλη αλλαγή έχουμε όταν με την αλλαγή του αριθμού των πυρήνων, ο κανονικοποιημένος χρόνος εκτέλεσης των επερωτήσεων επηρεάζεται πάρα πολύ με την εκτέλεση της κάθε επερώτησης με τις άλλες επερωτήσεις. Λόγοι για αυτή τη μεγάλη αλλαγή έχουμε τον συνδυασμό επερωτήσεων προς εκτέλεση που έχουμε στα αποτελέσματα μεγάλες διακυμάνσεις στις αυξομειώσεις των χρόνων εκτέλεσης των επερωτήσεων και ελάχιστα φταίει ο χώρος στην κρυφή μνήμη (cache).

Με τη χρήση του sort έχουμε καλύτερους χρόνους όταν είναι λιγότερο από 50% cpu για i/o. Με τη χρήση nested loop join έχουμε καλύτερους χρόνους όταν είναι λιγότερο από 50% cpu για i/o παρά με τη χρήση nested loop join σε συνδυασμό με hash join. Όταν χρησιμοποιείται sequential scan έχουμε καλύτερους χρόνους με λιγότερο από 50% cpu για i/o. Επίσης όταν χρησιμοποιείται nested loop join σε συνδυασμό με sort έχουμε καλύτερους χρόνους από ότι θα είχαμε μόνο nested loop join είτε μόνο hash join. Επίσης έχουμε καλύτερους χρόνους όταν χρησιμοποιείται index scan παρά nested loop join με λιγότερο από 50% cpu για i/o.

Για μια βάση δεδομένων μεγέθους των δέκα Gigabyte (10GB), έχουμε καλύτερη ρυθμοαπόδοση και συνάμα καλύτερη γενική επίδοση της μηχανής όταν χρησιμοποιεί 8 πυρήνες γιατί εκτελείται πολύ περισσότερος αριθμός επερωτήσεων ανά δευτερόλεπτο. Όμως βλέπουμε ότι για τους οχτώ πυρήνες έχουμε χειρότερους χρόνους εκτέλεσης λόγο των περισσότερων cache misses που δημιουργούνται.

Κεφάλαιο 6

Σύγκριση πειραμάτων

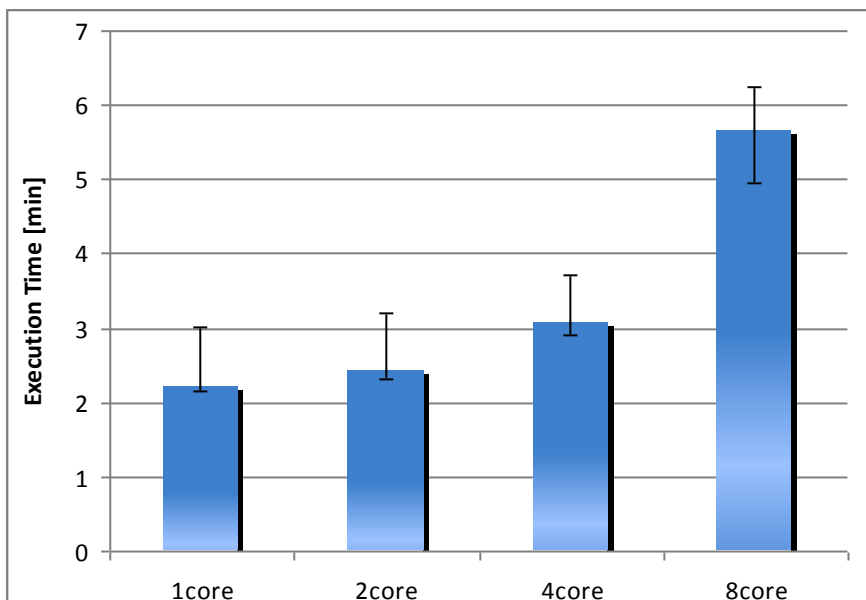
6.1 Σύγκριση Πειραμάτων	70
6.2 Συμπεράσματα	73

6.1 Σύγκριση Πειραμάτων

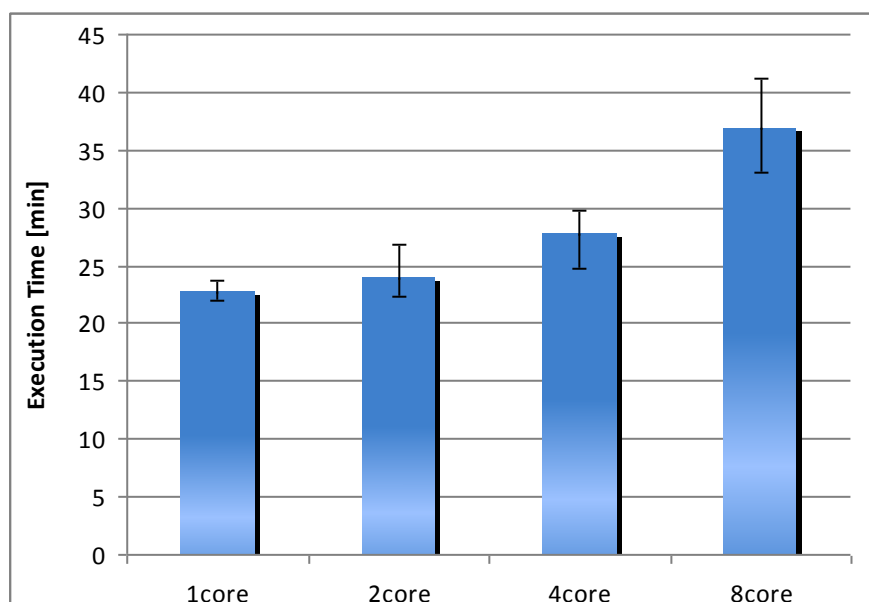
Αφού μελετήθηκαν και αξιολογήθηκαν ξεχωριστά οι δύο βάσεις δεδομένων μεγέθους ενός και δέκα Gigabyte, μένει να γίνει η μεταξύ τους σύγκριση για να αποφασιστούν οι παράγοντες που επηρεάζουν τη ρυθμοαπόδοση και τη γενική επίδοση της μηχανής server που φιλοξενεί τη βάση δεδομένων. Η σύγκριση θα γίνει ξεχωριστά για τη κάθε παράμετρο που επηρεάζει τη ρυθμοαπόδοση στις δύο βάσεις δεδομένων και στο τέλος θα οδηγήσει στα γενικά συμπεράσματα.

Όγκος δεδομένων

Όπως μπορούμε να δούμε από τις πιο κάτω γραφικές παραστάσεις 6.1.1 και 6.1.2, έχουμε μεγάλη διαφορά σε σύγκριση με τους χρόνους εκτέλεσης των πειραμάτων που πραγματοποιήθηκαν στη βάση του ενός Gigabyte και της βάσης των δέκα Gigabyte, φαίνεται ξεκάθαρα ότι ο όγκος δεδομένων παίζει πολύ σημαντικό ρόλο στην εκτέλεση αποτελεσμάτων.



Γραφική παράσταση 6.1.1 Χρόνος εκτέλεσης σε λεπτά των τεσσάρων πειραμάτων για τη βάση του ενός Gigabyte



Γραφική παράσταση 6.1.2 Χρόνος εκτέλεσης σε λεπτά των τεσσάρων πειραμάτων για τη βάση των δέκα Gigabyte

Αυτό μπορούμε να το ελέγξουμε βλέποντας τους χρόνους πιο κάτω στους πίνακες με τους ακριβείς χρόνους που είχαν όταν είχαν ολοκληρωθεί τα πειράματα και για τις δύο βάσεις δεδομένων.

	1core	2core	4core	8core
Average	2,2035075	2,4232245	3,064202	5,6634107
Plus	0,8130725	0,7913955	0,661598	0,5917893
Minus	0,0596775	0,0957545	0,153042	0,7170907

Πίνακας 6.1.1 Χρόνοι για την βάση δεδομένων του ενός Gigabyte (1Gb)

	1core	2core	4core	8core
Average	22,601337	23,856853	27,7279485	36,7832365
Plus	1,063013	3,025447	1,9877415	4,3412735
Minus	0,553207	1,507933	3,0448785	3,7828265

Πίνακας 6.1.2 Χρόνοι για την βάση δεδομένων των δέκα Gigabyte (10Gb)

Για τα πειράματα που εκτελέστηκαν στη βάση δεδομένων του ενός Gigabyte και δέκα Gigabyte, χρόνος εκτέλεσης όλων των επερωτήσεων για το κάθε πείραμα ξεχωριστά δίνεται στους πίνακες στους χρόνους average. Για ένα πυρήνα έχουμε 2,2035075 και 22,601337 λεπτά εκτέλεσης για τη βάση ενός και δέκα Gigabyte αντίστοιχα. Είναι δεκαπλάσιος ο χρόνος εκτέλεσης στο πείραμα με ένα πυρήνα στη βάση των δέκα Gigabyte απ' ότι στη βάση του ενός Gigabyte.

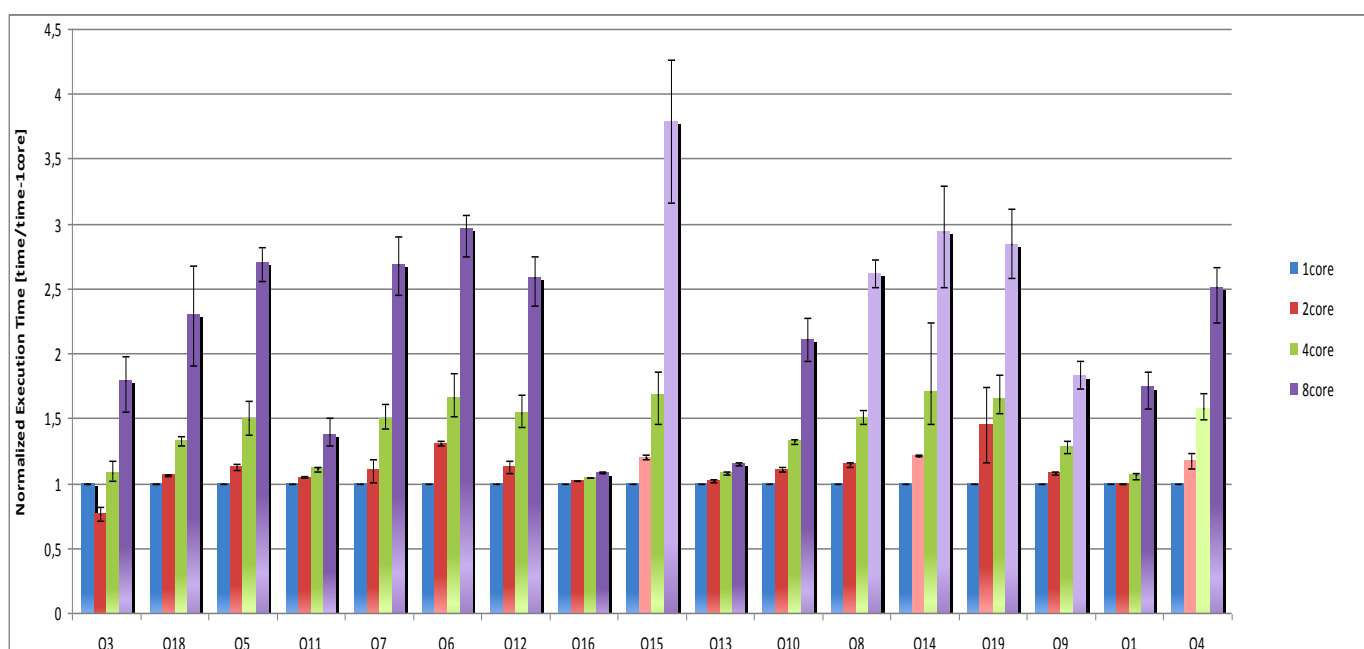
Για δύο πυρήνες έχουμε 2,4232245 και 23,856853 λεπτά εκτέλεσης για τη βάση ενός και δέκα Gigabyte αντίστοιχα. Και πάλι είναι σχεδόν δεκαπλάσιος ο χρόνος εκτέλεσης στο πείραμα με δύο πυρήνες στη βάση των δέκα Gigabyte απ' ότι στη βάση του ενός Gigabyte.

Για τέσσερις πυρήνες έχουμε 3,064202 και 27,7279485 λεπτά εκτέλεσης για τη βάση ενός και δέκα Gigabyte αντίστοιχα. Είναι 3 λεπτά σχεδόν λιγότερο από το δεκαπλάσιο χρόνο εκτέλεσης στο πείραμα με τέσσερις πυρήνες στη βάση των δέκα Gigabyte απ' ότι στη βάση του ενός Gigabyte.

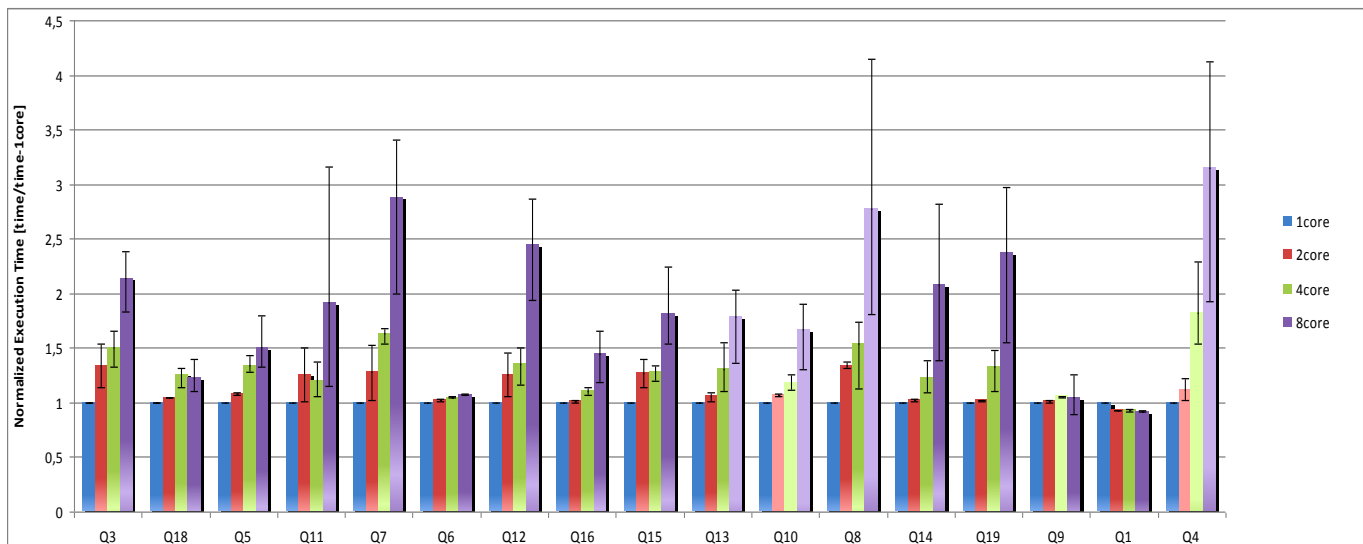
Για οχτώ πυρήνες έχουμε 5,6634107 και 36,7832365 λεπτά εκτέλεσης για τη βάση ενός και δέκα Gigabyte αντίστοιχα. Παρατηρούμε ότι δεν είναι το δεκαπλάσιο όπως αναμέναμε, αλλά λίγο πιο πάνω από το πενταπλάσιο του χρόνου εκτέλεσης στο πείραμα με οχτώ πυρήνες στη βάση των δέκα Gigabyte απ' ότι στη βάση του ενός Gigabyte.

Επομένως παίζει σημαντικό ρόλο ο όγκος των δεδομένων για κάθε βάση είναι ένας από τους παράγοντες που επηρεάζουν τον χρόνο εκτέλεσης, αλλά όπως έχουμε παρατηρήσει από το πείραμα με τους οχτώ πυρήνες θα πρέπει να υπάρχουν κι άλλοι παράγοντες που επηρεάζουν τη ρυθμοαπόδοση.

Πιο κάτω στις γραφικές παραστάσεις 6.1.3 και 6.1.4 μπορούμε να δούμε για κάθε επερώτηση που έχει εκτελεστεί τον κανονικοποιημένο χρόνο εκτέλεσης τους. Οι γραφικές αυτές διαφέρουν πολύ μεταξύ τους, βλέπουμε ότι με την αύξηση των πυρήνων σχεδόν για όλες τις επερωτήσεις αυξάνεται ο κανονικοποιημένος χρόνος εκτέλεσης τους. Υπάρχουν όμως περιπτώσεις που μειώνεται ο κανονικοποιημένος χρόνος, παράδειγμα για αυτό μπορούμε να δούμε την επερώτηση q3 στη γραφική παράσταση 6.1.3 για τους δύο πυρήνες έχουμε λιγότερο κανονικοποιημένο χρόνο απ' ό τι στον ένα πυρήνα είτε στην επερώτηση q1 στη γραφική παράσταση 6.1.4 όπου για τα πειράματα σε 2,4,8 πυρήνες έχουμε λιγότερο κανονικοποιημένο χρόνο απ' ό τι στον ένα πυρήνα. Είτε έχουμε σταθερότητα όπως για παράδειγμα η επερώτηση q16 στη γραφική παράσταση 6.1.3, και επίσης έχουμε άλλες περιπτώσεις που έχουμε αυξημένο κανονικοποιημένο χρόνο για μία επερώτηση όμως παραμένει σταθερά αυξημένος καθ' όλη τη διάρκεια εκτέλεσης του με πολλές μικρές διακυμάνσεις μεταξύ των πυρήνων του. Αυτές οι μεταβολές μας κινούν την περιέργεια για να εμβαθύνουμε ακόμη περισσότερο για τις αιτίες που συμβαίνει αυτό.



Γραφική παράσταση 6.1.3 Κανονικοποιημένος Χρόνος εκτέλεσης β.δ του 1GB



Γραφική παράσταση 6.1.4 Κανονικοποιημένος Χρόνος εκτέλεσης β.δ των 10GB

Μέγεθος Κρυφής μνήμης (cache)

Ο ρόλος του μεγέθους της κρυφής μνήμης φαίνεται συγκεκριμένα για την επερώτηση αυτή που χρειάζεται σταθερά μεγαλύτερο χρόνο σε κάποιο ή όλα τα πειράματα σε σύγκριση με το αρχικό μας σενάριο για ένα πυρήνα. Αυτό μπορούμε να το δούμε για παράδειγμα στην επερώτηση q14 της γραφικής παράστασης 6.1.4 και πιο συγκεκριμένα στα πειράματα των τεσσάρων και οχτώ πυρήνων όπου ο κανονικοποιημένος χρόνος διαφέρει πολύ από το κανονικοποιημένο χρόνο στο βασικό μας σενάριο και το error bar δηλώνει τις διακυμάνσεις μεταξύ των εκτελέσεων της ίδιας επερώτησης στους διάφορους πυρήνες του ίδιου πειράματος. Παρόλο που οι διακυμάνσεις ήταν πολλές, δηλαδή έχουμε μεγάλο error bar, παρόλα αυτά ο χρόνος εκτέλεσης στον κάθε πυρήνα διαφέρει ελάχιστα από το χρόνο εκτέλεσης στους άλλους πυρήνες για το ίδιο πείραμα, το συγκεκριμένο που έχουμε αναφέρει δεν φαίνεται στις γραφικές θα πρέπει να το ελέγξουμε τιμή προς τιμή από τα αποτελέσματα που έχουμε πάρει από όλες τις εκτελέσεις των επερωτήσεων στη βάση δεδομένων που έχουν εκτελεστεί.

Συνδυασμός του σετ επερωτήσεων

Ο συνδυασμός επερωτήσεων για το σετ που θα εκτελείται από τους πυρήνες, παίζει πολύ σημαντικό ρόλο όπως βλέπουμε γιατί για το ίδιο μέγεθος βάσης δεδομένων του ενός Gigabyte έχουμε διαφορετικούς χρόνους εκτέλεσης για τις ίδιες επερωτήσεις, όπως και διαφορετική τιμή της ρυθμοαπόδοσης και της όλης επίδοσης της μηχανής. Συνεπώς αυτό μας

παραπέμπει σε μια άλλη παράμετρο που είναι αιτία για τις διαφορετικές τιμές των χρόνων μεταξύ των ίδιων επερωτήσεων αλλά με διαφορετικό σετ.

Κοινή μνήμη και buffer

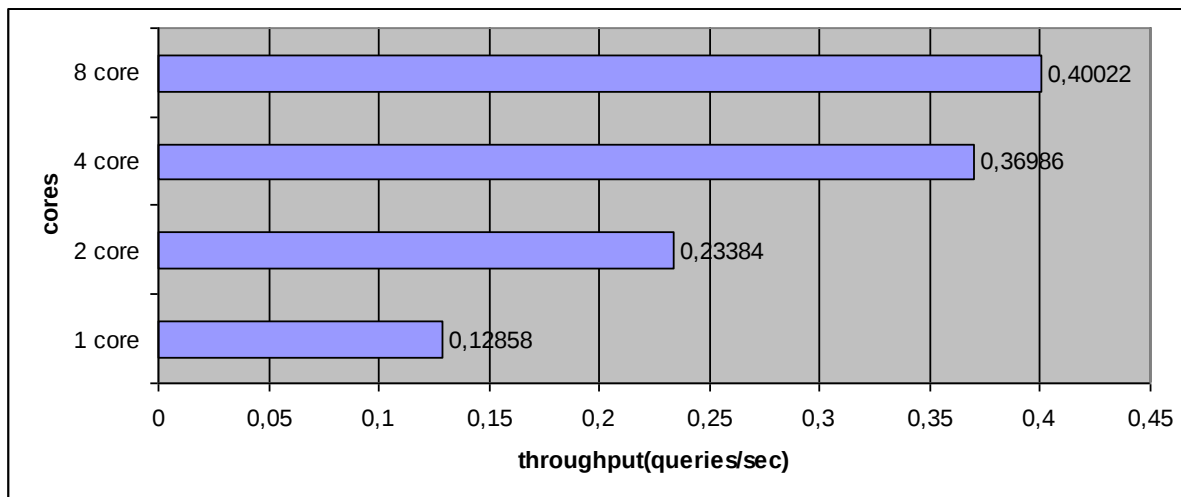
Η επικοινωνία μεταξύ των πυρήνων γίνεται μέσω του buffer, συνεπώς όταν γεμίσει ο χώρος του buffer έχουμε τη δημιουργία καθυστέρησης και για τις επερωτήσεις που εκτελούνται και για τις επόμενες επερωτήσεις που ακολουθούν, διότι οι πληροφορίες που δίνονται από τα νήματα και μοιράζονται μεταξύ τους γεμίζουν το buffer και δημιουργείται η εν λόγω καθυστέρηση. Τώρα για την επίτευξη καλύτερου χρόνου από το χρόνο στο βασικό σενάριο έχουμε μέσω των κοινών μεταβλητών όπου μοιράζονται πληροφορίες και συγχρονίζονται με τέτοιο τρόπο έτσι ώστε στους πυρήνες που εκτελούνται οι επερωτήσεις ως νήματα να μπορούν να μοιράζονται πληροφορίες και εσωτερικά στο κάθε πυρήνα και μεταξύ τους οι πυρήνες όλοι μαζί. Αυτό συμβαίνει μόνο με τη χρήση του παραλληλισμού, που μπορούμε να επιτύχουμε καλύτερη ρυθμοαπόδοση και γενικότερα επίδοση της μηχανής.

Ρυθμοαπόδοση

Για τη ρυθμοαπόδοση που φαίνεται στις πιο κάτω γραφικές παραστάσεις 6.1.6 και 6.1.7, και στους πιο κάτω πίνακες 6.1.3 και 6.1.4, μπορούμε να δούμε πόσο διαφέρουν οι τιμές μεταξύ τους για τα μεγέθη των δύο βάσεων για όλα τα πειράματα που έχουν εκτελεστεί. Επίσης μπορούμε να αποφασίσουμε μέσω της ρυθμοαπόδοσης πόσους πυρήνες μπορούμε να χρησιμοποιήσουμε έτσι ώστε να έχουμε καλύτερη επίδοση της μηχανής server που είναι και ένα από τα σημαντικά ζητήματα των σύγχρονων εταιριών που ασχολούνται με αυτό το τομέα.

Cores	1core	2core	4core	8core
Number of Queries	17	34	68	136
Sum of queries average(sec)	(2,2035075)x60	(2,4232245)x60	(3,064202)x60	(5,6634107)x60
Throughput(queries/sec)	0,12858	0,23384	0,36986	0,40022

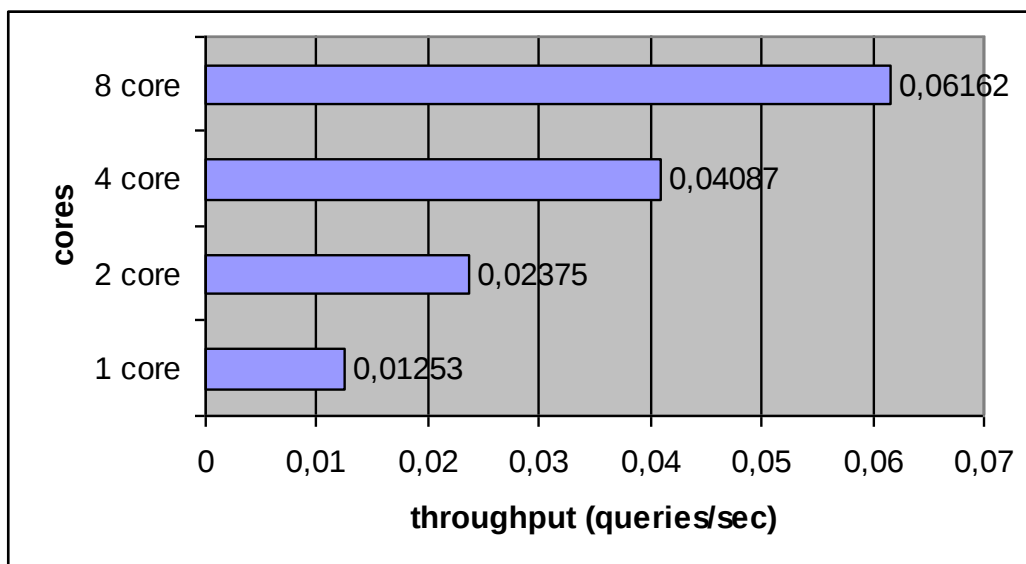
Πίνακας 6.1.3 Ρυθμοαπόδοση για την βάση δεδομένων του ενός Gigabyte (1Gb)



Γραφική παράσταση 6.1.6 Ρυθμοαπόδοση (Throughput) για την βάση δεδομένων του ενός Gigabyte (1Gb)

Cores	1core	2core	4core	8core
Number of Queries	17	34	68	136
Sum of queries average(sec)	(22,601337)x60	(23,856853)x60	(27,7279485)x60	(36,7832365)x60
Throughput(queries/sec)	0,01253	0,02375	0,04087	0,06162

Πίνακας 6.1.4 Ρυθμοαπόδοση για την βάση δεδομένων των δέκα Gigabyte (10Gb)

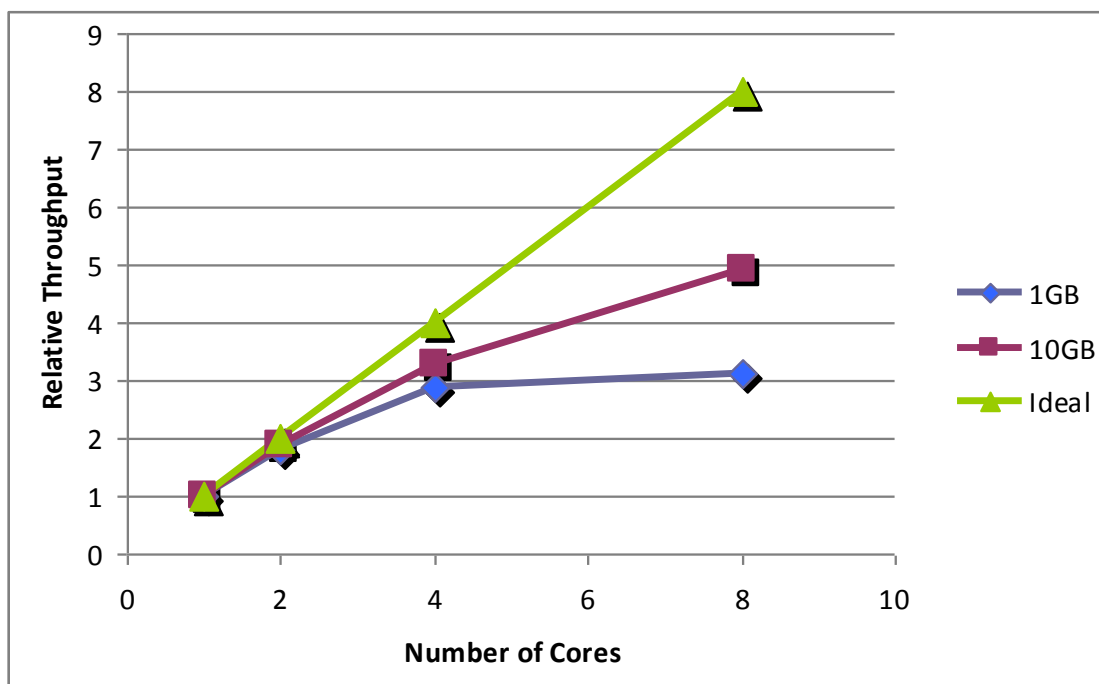


Γραφική παράσταση 6.1.7 Ρυθμοαπόδοση (Throughput) για την βάση δεδομένων των δέκα Gigabyte (10Gb)

Όπως μπορούμε να δούμε για μια βάση δεδομένων του ενός Gigabyte (1GB), θα είναι πιο αποδοτικό να δουλεύει με οχτώ πυρήνες. Και για μία βάση δεδομένων μεγέθους των δέκα Gigabyte (10GB), έχουμε καλύτερη ρυθμοαπόδοση και συνάμα καλύτερη γενική επίδοση της μηχανής όταν χρησιμοποιεί 8 πυρήνες επίσης παρότι πέφτει η ρυθμοαπόδοση όσο αυξάνουμε τον αριθμό πυρήνων. Για να βελτιώσουμε την ρυθμοαπόδοση και συνάμα την όλη επίδοση στα υπάρχον προβλήματα θα πρέπει να αλλάξουμε το σετ συνδυασμού επερωτήσεων για κάθε πυρήνα.

	1core	2cores	4cores	8cores
1GB	1	1,81863431	2,87649712	3,11261471
10GB	1	1,89545092	3,26177175	4,91779729
Ideal	1	2	4	8

Πίνακας 6.1.4 Throughput factor



Γραφική παράσταση 6.1.8 Σύγκριση Ρυθμοαπόδοσης

Για τη γραφική παράσταση 6.1.8 όπου συγκρίνουμε τη ρυθμοαπόδοση του ενός και δέκα Gigabyte και με την αναμενόμενη ρυθμοαπόδοση συνολικά. Σε σύγκριση με τις δύο βάσεις βλέπουμε ότι για δέκα GB έχουμε μεγαλύτερη ρυθμοαπόδοση σε όλα τα πειράματα που εκτελέσαμε και την μέγιστη για αριθμό πυρήνων ίσο με οχτώ. Η αναμενόμενη ρυθμοαπόδοση ξεχωρίζει και μας δείχνει ότι από τους τέσσερις πυρήνες και μετά η

ρυθμοαπόδοση των δύο βάσεων πέφτει πολύ. Για τα πειράματα που εκτελεστήκαν σε δύο πυρήνες έχουμε σχεδόν ίση ρυθμοαπόδοση για όλα. Την μέγιστη ρυθμοαπόδοση έχουμε στους οχτώ πυρήνες για τις δύο βάσεις.

6.2 Συμπεράσματα

Από τη σύγκριση των πειραμάτων στα δύο μεγέθη των βάσεων δεδομένων μπορέσαμε να συμπεράνουμε τους παράγοντες που επηρεάζουν τον χρόνο επίδοσης μιας μηχανής server. Οι παράγοντες που επηρεάζουν την επίδοση είναι ο όγκος των δεδομένων, ο αριθμός των πυρήνων, το μέγεθος της κρυφής μνήμης, ο συνδυασμός του σετ επερωτήσεων στον κάθε πυρήνα, το buffer της κοινής μνήμης και η τιμή της ρυθμοαπόδοσης.

Για τη βάση του ενός Gigabyte έχουμε όλες τις επερωτήσεις που διαβάζουν περισσότερο από τη μνήμη με cpu πέραν του 50% και χρησιμοποιούν είτε hash, είτε sort, είτε index scan καλύτερα να χρησιμοποιούνται σε συνδυασμό με nested loop join. Ενώ για τις επερωτήσεις με λιγότερο από το 50% διάβασμα από τη μνήμη είναι καλύτερα τα αποτελέσματα με χρήση του hash και μόνο. Οι περισσότερες επερωτήσεις χρησιμοποίησαν nested loop join έτσι μπορούμε να δικαιολογήσουμε την περισσότερη μείωση της ρυθμοαπόδοσης στους οχτώ πυρήνες λόγω του αυξημένου κόστους για cache misses.

Για τη βάση των δέκα Gigabyte έχουμε cache misses όμως από τις παρατηρήσεις για αυτή τη βάση έχουμε καλύτερους χρόνους για nested loop join όταν είναι λιγότερο από 50% cpu για i/o. Επειδή οι περισσότερες επερωτήσεις χρησιμοποιούν nested loop join είχαμε ρυθμοαπόδοση για τους οχτώ πυρήνες να πλησιάζει την αναμενόμενη ρυθμοαπόδοση.

Με τη χρήση του sort έχουμε καλύτερους χρόνους όταν είναι λιγότερο από 50% cpu για i/o. Με τη χρήση nested loop join έχουμε καλύτερους χρόνους όταν είναι λιγότερο από 50% cpu για i/o παρά με τη χρήση nested loop join σε συνδυασμό με hash join. Όταν χρησιμοποιείται sequential scan έχουμε καλύτερους χρόνους με λιγότερο από 50% cpu για i/o. Επίσης όταν χρησιμοποιείται nested loop join σε συνδυασμό με sort έχουμε καλύτερους χρόνους από ότι θα είχαμε μόνο nested loop join είτε μόνο hash join. Επίσης έχουμε καλύτερους χρόνους όταν χρησιμοποιείται index scan παρά nested loop join με λιγότερο από 50% cpu για i/o.

Κεφάλαιο 7

Επίλογος

7.1 Γενικά Συμπεράσματα	70
7.2 Μελλοντική Εργασία	73
7.3 Οφέλη από τη Διπλωματική Εργασία	70

7.1 Γενικά Συμπεράσματα

Όντως η PostgreSQL που είναι ένα αντικειμενοστραφές σύστημα διαχείρισης βάσεων δεδομένων και χαρακτηρίζεται από την αξιοπιστία, την ακεραιότητα των δεδομένων και την ορθότητα, μπορεί να τρέξει σε λειτουργικό σύστημα των Linux χωρίς προβλήματα και να διαχειριστεί μεγάλο όγκο πληροφοριών.

Με τον παραλληλισμό γνωρίζουμε ότι πρέπει να έχουμε τουλάχιστον δύο πυρήνες και πως για κάθε πυρήνα γνωρίζουμε ότι έχει την δική του κρυφή μνήμη και μία κοινή διαμοιραζόμενη μνήμη με τους υπόλοιπους πυρήνες που θα εκτελείται. Οι διεργασίες που εκτελούνται παράλληλα συνεργάζονται έτσι ώστε να έχουμε καλύτερο χρόνο εκτέλεσης, με συγχρονισμό, διότι μοιράζονται τις πληροφορίες μεταξύ τους. Για να πετύχει αυτό θα πρέπει να έχουμε κοινές μεταβλητές το buffer semaphore για τον συγχρονισμό τους, αυτά που έχουμε αναφέρει αποτελούν θετικές επιδράσεις για να μοιράζονται οι πληροφορίες μεταξύ των νημάτων και μεταξύ των πυρήνων. Σε σύγκριση με το βασικό μας σενάριο για ένα πυρήνα ο παραλληλισμός υπερέχει στην επίδοση.

Μέσω των πειραμάτων που έχουμε πραγματοποιήσει στην μελέτη αυτή, μπορέσαμε να δείξουμε ότι ο παραλληλισμός μπορεί να βελτιώσει την επίδοση μιας μηχανής server. Η επίδοση μιας παρόμοιας μηχανής με παρόμοια χαρακτηριστικά μπορεί να βελτιωθεί με ποικίλους τρόπους, ανάλογα με τις απαιτήσεις που θέλει να θέσει μια εταιρία. Στην εργασία αυτή πραγματοποιήθηκαν τέσσερα πειράματα για μία βάση του ενός Gigabyte και τα ίδια πειράματα για μία βάση των δέκα Gigabyte στην ίδια μηχανή. Τα πειράματα ήταν το πρώτο βασικό σενάριο για ένα πυρήνα και τα υπόλοιπα πειράματα ήταν για δύο, τέσσερις και οχτώ

πυρήνες. Όμως τα αποτελέσματα των πειραμάτων αυτών ήταν εντελώς διαφορετικά, όμως οι θετικοί και αρνητικοί λόγοι είναι οι ίδιοι.

Οι λόγοι αυτοί είναι ο όγκος των πληροφοριών δηλαδή το μέγεθος των δεδομένων που είναι καταχωρημένα στη βάση, ο αριθμός των πυρήνων όπου με την αύξηση τους βλέπουμε ότι επηρεάζει θετικά τη ρυθμοαπόδοση αρχικά όμως σε κάποιο όριο αρχίζει να πέφτει και θα προτιμηθεί να μειωθεί ο αριθμός των πυρήνων μέχρι το σημείο που μπορούσε να δουλέψει αποδοτικά η μηχανή. Επίσης το μέγεθος της κρυφής μνήμης που επηρεάζει την επίδοση όταν έχει γεμίσει και δημιουργείται καθυστέρηση της εκτέλεσης των επερωτήσεων που ήδη εκτελούνται και καθυστέρηση στις επερωτήσεις που περιμένουν να εκτελεστούν. Επίσης μια άλλη αιτία είναι το σετ με τις συνδυασμένες επερωτήσεις που εκτελούνται σε κάθε πυρήνα όπου δίνουν διαφορετικές τιμές στους χρόνους εκτέλεσης και γενικότερα για την επίδοση της μηχανής.

Μέσω της ρυθμοαπόδοσης μπορούμε να δούμε πόσες επερωτήσεις μπορούν να εκτελούνται ανά δευτερόλεπτο έτσι ώστε να μπορούμε να αποφασίζουμε και πόσους πυρήνες μπορούμε να χρησιμοποιήσουμε για μία βάση δεδομένων.

Με μεγαλύτερο όγκο πληροφοριών για τον ίδιο αριθμό επερωτήσεων έχουμε πολύ καλύτερη ρυθμοαπόδοση. Έχουμε μεγάλες αλλαγές και για τις δύο βάσεις στους οχτώ πυρήνες. Χειρότερους χρόνους εκτέλεσης έχουμε σχεδόν για όλες τις επερωτήσεις για τις εκτελέσεις στους οχτώ πυρήνες. Για διαφορετικά μεγέθη βάσεων δεδομένων με ίδια σειρά εκτέλεσης των επερωτήσεων, έχουμε διαφορετική συμπεριφορά των χρόνων εκτέλεσης για όλες τις επερωτήσεις διότι οι λειτουργίες που εμπεριέχουν επιδρούν διαφορετικά στον διαφορετικό όγκο πληροφοριών.

7.2 Μελλοντική Εργασία

Η αξιολόγηση της επίδοσης μέσω της ρυθμοαπόδοσης της μηχανής server που είχε δοθεί για την πραγματοποίηση αυτής της εργασίας, έγινε αξιολόγηση μέχρι τους οχτώ πυρήνες μόνο και για δύο μεγέθη βάσεων δεδομένων του ενός και των δέκα Gigabyte. Όπως έχουμε δει από τα αποτελέσματα για αυτά τα δύο μεγέθη βάσεων, δεν θα χρειαζόμασταν περισσότερους πυρήνες από τους οχτώ πυρήνες, διότι η ρυθμοαπόδοση πέφτει πάρα πολύ για περισσότερους από τέσσερις πυρήνες. Μία εταιρία που θέλει να χρησιμοποιεί περίπου το ίδιο μέγεθος με μία εκ των δύο βάσεων δεδομένων και τα ίδια χαρακτηριστικά μηχανής που χρησιμοποιήθηκαν

στην έρευνα αυτή, μπορεί να μελετήσει τα αποτελέσματα της επίδοσης που έχουμε βρει με τους συνδυασμούς που χρησιμοποιήσαμε και να αποφασίσει πόσους πυρήνες να χρησιμοποιεί και το όφελος που θα μπορούσε να έχει σε σχέση με τη ρυθμοαπόδοση δηλαδή πόσους πελάτες θα μπορούσε να φιλοξενήσει η μηχανή server ανά δευτερόλεπτο.

Στόχος μίας μελλοντικής εργασίας θα ήταν να ανακαλυφθεί ο άριστος συνδυασμός επερωτήσεων που χρησιμοποιήθηκε για τα δύο μεγέθη βάσεων σε αυτή την έρευνα, έτσι ώστε να βρεθεί η μέγιστη επίδοση. Επίσης μπορεί να διεξαχθεί έρευνα για περισσότερους πυρήνες και για μεγαλύτερα μεγέθη βάσεων δεδομένων.

7.3 Οφέλη από τη Διπλωματική Εργασία

Μέσα από τη διεκπεραίωση των στόχων της διπλωματικής αυτής εργασίας απέκτησα αρκετά οφέλη. Αρχικά μελέτησα και κατανόησα τους λόγους για να μπορώ να χρησιμοποιήσω το κατάλληλο σύστημα διαχείρισης βάσεων δεδομένων που στην έρευνα αυτή ήταν η PostgreSQL και στη χρήση κατάλληλων επερωτήσεων σε SQL που θα μπορούσαν να προσφέρουν αξιοπιστία όπως οι επερωτήσεις από το TPC-H Benchmark suite. Στην συνέχεια μελέτησα τους τρόπους πλοήγησης στο περιβάλλον των Linux, για τη δημιουργία βάσεων δεδομένων και εκτέλεση πειραμάτων μέσω των shell script αρχείων και ακολούθως μέσω της ανάλυσης των αποτελεσμάτων από τα πειράματα αυτά να μπορούσα να καταλήξω στους παράγοντες που επηρεάζουν την επίδοση της συγκεκριμένης μηχανής που χρησιμοποιήθηκε και να προτείνω αριθμό πυρήνων που μπορούν να χρησιμοποιηθούν για τα μεγέθη βάσεων δεδομένων που χρησιμοποίησα στην έρευνα αυτή. Τέλος, το σημαντικότερο όφελος που απέκτησα καθ' όλη τη διάρκεια της εργασίας ήταν η εκμάθηση και η κατανόηση της μεθοδολογίας που ακολουθείται για την εκπόνηση μίας τέτοιας εργασίας ξεκινώντας από το στάδιο της έρευνας σχετικά με το θέμα της εργασίας και καταλήγοντας στο στάδιο της αξιολόγησης της εργασίας και των συμπερασμάτων.

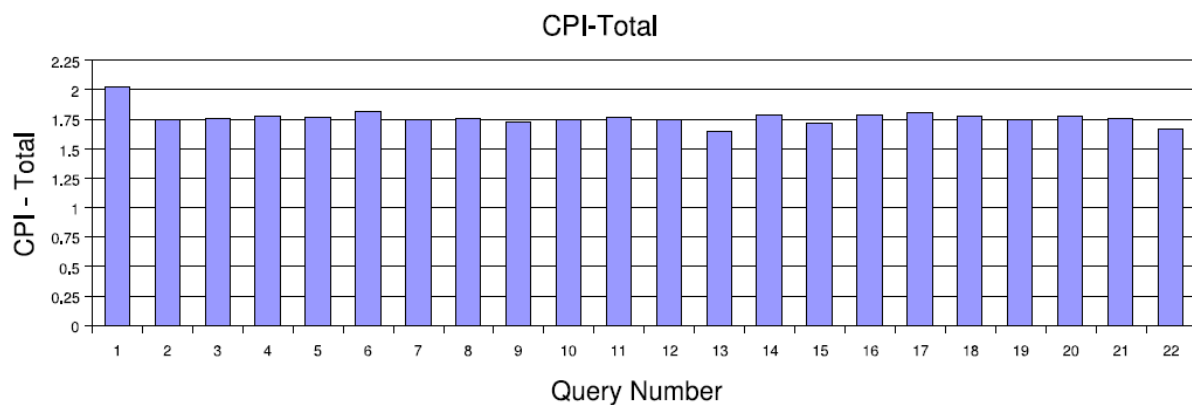
Βιβλιογραφία

- [1] PostgreSQL and version 8.4 of PostgreSQL information. [Online]. Available: <http://www.postgresql.org>
- [2] TPC-H Benchmarks information. [Online]. Available: <http://www.tpc.org/tpch/spec/tpch2.14.4.pdf>
- [3] TPC-H Benchmarks suite, queries information. [Online]. Available: <http://www.tpc.org>
- [4] Throughput scalability information. [Online]. Available: http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.prfungd/doc/prftungd/multiproc_throughput_scal.htm
- [5] Από σημειώσεις του μαθήματος Παράλληλη Επεξεργασία. [Online]. Available: <http://moodle.cs.ucy.ac.cy/>
- [6] <http://oss.sgi.com/projects/kaio>

Παράρτημα Α

Table 1: Query types and corresponding database functions

Query	Sequential Scan	Index Scan	Hash Join	Nested Loop Join	Sort
Q1	✓				
Q2	✓				
Q3			✓	✓	
Q4		✓		✓	
Q5				✓	✓
Q6					✓
Q7			✓	✓	
Q8			✓	✓	
Q9				✓	
Q10				✓	
Q11				✓	
Q12			✓		✓
Q13			✓		
Q14		✓			
Q15				✓	
Q16				✓	✓
Q17				✓	
Q18				✓	✓
Q19				✓	
Q20				✓	
Q21				✓	
Q22			✓		



query	user CPU (%)	system CPU (%)	system CPU breakup (%)				page faults per jiffy	blocks read per jiffy	blocks written per jiffy	packets sent per jiffy	packets received per jiffy	CPU utilization during IO (%)
Q1	27.58	21.44	pread	pwrite	select	ipc	1.50	51.01	23.1151	0.0012	0.0015	26.87
Q2	56.22	15.67	socketcall	pread	select	pwrite	0.39	21.97	1.7718	1.9077	1.9248	53.73
			35.4	32.9	20.1	7.3						
Q3	40.76	17.76	pread	socketcall	select	pwrite	0.99	55.47	4.9591	0.9778	0.9938	55.40
			54.1	15.1	13.1	12.8						
Q4	51.48	15.19	pread	select	socketcall	pwrite	0.00	22.97	1.0478	0.3517	0.3652	68.41
			60.0	17.6	11.2	6.9						
Q5	58.96	15.68	socketcall	pread	select	pwrite	0.05	16.73	1.1369	2.0779	2.0849	42.81
			43.3	29.2	21.7	3.7						
Q6	40.65	22.20	pread	ipc	socketcall		1.73	90.72	0.0020	0.0012	0.0012	33.49
			90.1	4.9	4.3							
Q7	52.44	16.82	pread	pwrite	ipc	select	0.00	16.89	1.9467	2.3880	2.3521	31.04
			72.1	14.8	6.1	6.0						
Q8	20.65	17.71	pread	pwrite	select	ipc	0.01	27.63	4.8078	0.0261	0.0228	12.91
			59.5	23.7	9.4	5.8						
Q9	51.41	13.52	pwrite	pread	select	ipc	0.00	6.69	1.9276	0.0133	0.0136	23.21
			40.8	38.7	13.9	2.5						
Q10	41.79	17.87	pread	socketcall	select	pwrite	0.17	41.99	1.8880	0.8774	0.8859	18.71
			57.7	17.9	13.4	7.0						
Q11	81.00	13.28	socketcall	pread	select	ipc	0.49	18.87	0.0020	2.3794	2.4011	66.46
			43.6	27.0	24.8	3.9						
Q12	14.91	19.73	pread	select	ipc		0.25	40.79	0.0197	0.0102	0.0101	4.8
			78.3	15.2	5.2							
Q13	53.23	21.86	pread	socketcall	select	ipc	1.62	45.86	0.0034	2.0966	2.0935	32.71
			45.7	30.3	18.8	4.6						
Q14	33.57	22.19	pread	select	ipc		1.01	84.78	0.0025	0.1156	0.1175	29.75
			85.6	8.3	5.1							
Q15	55.37	18.69	pread	select	ipc	nanosleep	0.60	75.26	0.0033	0.6260	0.7703	51.68
			71.7	14.0	10.9	1.9						
Q16	51.84	15.30	socketcall	pread	select	ipc	2.32	15.36	0.8454	2.4836	2.4993	46.24
			43.5	27.0	22.2	5.7						
Q17	23.71	18.26	pread	pwrite	select	ipc	0.00	32.76	5.2532	0.0036	0.0037	13.94
			59.4	26.1	8.7	4.9						
Q18	52.64	14.77	pread	socketcall	select	pwrite	0.04	17.16	0.6261	0.3890	0.3803	35.11
			61.1	17.3	13.3	5.2						
Q19	35.48	21.16	pread	select	ipc	socketcall	0.29	78.76	0.0032	0.3343	0.3329	23.38
			80.1	7.3	5.9	5.9						
Q20	56.98	14.72	pread	select	socketcall	ipc	0.00	15.74	0.1601	0.3456	0.2973	47.36
			53.4	25.6	15.3	3.6						

Παράδειγμα shell script που χρησιμοποιήθηκε στην έρευνα

```
#!/bin/bash
for i in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
do
(time psql name < /home/students/cs/.../queries_old/q3.sql) 1>> output81_q3.txt 2>>
time81_q3.txt
(time psql name < /home/students/cs/.../queries_old/q18.sql) 1>> output81_q18.txt 2>>
time81_q18.txt
```

```
(time psql name < /home/students/cs/.../queries_old/q5xwris.sql) 1>> output81_q5.txt 2>>
time81_q5.txt
(time psql name < /home/students/cs/.../queries_old/q11.sql) 1>> output81_q11.txt 2>>
time81_q11.txt
(time psql name < /home/students/cs/.../queries_old/q7.sql) 1>> output81_q7.txt 2>>
time81_q7.txt
(time psql name < /home/students/cs/.../queries_old/q6.sql) 1>> output81_q6.txt 2>>
time81_q6.txt
(time psql name < /home/students/cs/.../queries_old/q12.sql) 1>> output81_q12.txt 2>>
time81_q12.txt
(time psql name < /home/students/cs/.../queries_old/q16.sql) 1>> output81_q16.txt 2>>
time81_q16.txt
(time psql name < /home/students/cs/.../queries_old/q15.sql) 1>> output81_q15.txt 2>>
time81_q15.txt
(time psql name < /home/students/cs/.../queries_old/q13.sql) 1>> output81_q13.txt 2>>
time81_q13.txt
(time psql name < /home/students/cs/.../queries_old/q10xwris.sql) 1>> output81_q10.txt 2>>
time81_q10.txt
(time psql name < /home/students/cs/.../queries_old/q8.sql) 1>> output81_q8.txt 2>>
time81_q8.txt
(time psql name < /home/students/cs/.../queries_old/q14.sql) 1>> output81_q14.txt 2>>
time81_q14.txt
(time psql name < /home/students/cs/.../queries_old/q19.sql) 1>> output81_q19.txt 2>>
time81_q19.txt
(time psql name < /home/students/cs/.../queries_old/q9.sql) 1>> output81_q9.txt 2>>
time81_q9.txt
(time psql name < /home/students/cs/.../queries_old/q1.sql) 1>> output81_q1.txt 2>>
time81_q1.txt
(time psql name < /home/students/cs/.../queries_old/q4.sql) 1>> output81_q4.txt 2>>
time81_q4.txt
done
```

Παράρτημα Β

Χαρακτηριστικά της μηχανής

Mem info:

MemTotal: 33015724 kB
MemFree: 14437116 kB
Buffers: 315544 kB
Cached: 16974952 kB
SwapCached: 0 kB
Active: 1812488 kB
Inactive: 15512240 kB
Active(anon): 30472 kB
Inactive(anon): 4052 kB
Active(file): 1782016 kB
Inactive(file): 15508188 kB
Unevictable: 0 kB
Mlocked: 0 kB
SwapTotal: 15237112 kB
SwapFree: 15237112 kB
Dirty: 16 kB
Writeback: 0 kB
AnonPages: 34260 kB
Mapped: 15332 kB
Slab: 630536 kB
SReclaimable: 611216 kB
SUnreclaim: 19320 kB
PageTables: 2648 kB
NFS_Unstable: 0 kB
Bounce: 0 kB
WritebackTmp: 0 kB
CommitLimit: 31744972 kB
Committed_AS: 252324 kB
VmallocTotal: 34359738367 kB
VmallocUsed: 381820 kB
VmallocChunk: 34359350811 kB

HugePages_Total: 0
HugePages_Free: 0
HugePages_Rsvd: 0
HugePages_Surp: 0
Hugepagesize: 2048 kB
DirectMap4k: 9792 kB
DirectMap2M: 2086912 kB
DirectMap1G: 31457280 kB

Cpu info:

processor : 0
vendor_id : AuthenticAMD
cpu family : 16
model : 8
model name : Six-Core AMD Opteron(tm) Processor 2427
stepping : 0
cpu MHz : 800.000
cache size : 512 KB
physical id : 0
siblings : 6
core id : 0
cpu cores : 6
apicid : 0
initial apicid : 0
fpu : yes
fpu_exception : yes
cpuid level : 5
wp : yes
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36
clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm 3dnowext 3dnow
constant_tsc rep_good nonstop_tsc extd_apicid pni monitor cx16 popcnt lahf_lm cmp_legacy
svm extapic cr8_legacy abm sse4a misalignsse 3dnowprefetch osvw ibs skinit wdt
bogomips : 4389.47
TLB size : 1024 4K pages
clflush size : 64
cache_alignment : 64

address sizes : 48 bits physical, 48 bits virtual
power management: ts ttp tm stc 100mhzsteps hwpstate

processor : 1
vendor_id : AuthenticAMD
cpu family : 16
model : 8
model name : Six-Core AMD Opteron(tm) Processor 2427
stepping : 0
cpu MHz : 800.000
cache size : 512 KB
physical id : 0
siblings : 6
core id : 1
cpu cores : 6
apicid : 1
initial apicid : 1
fpu : yes
fpu_exception : yes
cpuid level : 5
wp : yes
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36
clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm 3dnowext 3dnow
constant_tsc rep_good nonstop_tsc extd_apicid pni monitor cx16 popcnt lahf_lm cmp_legacy
svm extapic cr8_legacy abm sse4a misalignsse 3dnowprefetch osvw ibs skinit wdt
bogomips : 4389.06
TLB size : 1024 4K pages
clflush size : 64
cache_alignment : 64
address sizes : 48 bits physical, 48 bits virtual
power management: ts ttp tm stc 100mhzsteps hwpstate

processor : 2
vendor_id : AuthenticAMD
cpu family : 16
model : 8

model name : Six-Core AMD Opteron(tm) Processor 2427
 stepping : 0
 cpu MHz : 800.000
 cache size : 512 KB
 physical id : 0
 siblings : 6
 core id : 2
 cpu cores : 6
 apicid : 2
 initial apicid : 2
 fpu : yes
 fpu_exception : yes
 cpuid level : 5
 wp : yes
 flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36
 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm 3dnowext 3dnow
 constant_tsc rep_good nonstop_tsc extd_apicid pni monitor cx16 popcnt lahf_lm cmp_legacy
 svm extapic cr8_legacy abm sse4a misalignsse 3dnowprefetch osvw ibs skinit wdt
 bogomips : 4389.07
 TLB size : 1024 4K pages
 clflush size : 64
 cache_alignment : 64
 address sizes : 48 bits physical, 48 bits virtual
 power management: ts ttp tm stc 100mhzsteps hwpstate

processor : 3
 vendor_id : AuthenticAMD
 cpu family : 16
 model : 8
 model name : Six-Core AMD Opteron(tm) Processor 2427
 stepping : 0
 cpu MHz : 800.000
 cache size : 512 KB
 physical id : 0
 siblings : 6
 core id : 3

cpu cores : 6
 apicid : 3
 initial apicid : 3
 fpu : yes
 fpu_exception : yes
 cpuid level : 5
 wp : yes
 flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36
 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm 3dnowext 3dnow
 constant_tsc rep_good nonstop_tsc extd_apicid pni monitor cx16 popcnt lahf_lm cmp_legacy
 svm extapic cr8_legacy abm sse4a misalignsse 3dnowprefetch osvw ibs skinit wdt
 bogomips : 4389.07
 TLB size : 1024 4K pages
 clflush size : 64
 cache_alignment : 64
 address sizes : 48 bits physical, 48 bits virtual
 power management: ts ttp tm stc 100mhzsteps hwpstate

processor : 4
 vendor_id : AuthenticAMD
 cpu family : 16
 model : 8
 model name : Six-Core AMD Opteron(tm) Processor 2427
 stepping : 0
 cpu MHz : 800.000
 cache size : 512 KB
 physical id : 0
 siblings : 6
 core id : 4
 cpu cores : 6
 apicid : 4
 initial apicid : 4
 fpu : yes
 fpu_exception : yes
 cpuid level : 5
 wp : yes

flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36
 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm 3dnowext 3dnow
 constant_tsc rep_good nonstop_tsc extd_apicid pni monitor cx16 popcnt lahf_lm cmp_legacy
 svm extapic cr8_legacy abm sse4a misalignsse 3dnowprefetch osvw ibs skinit wdt
 bogomips : 4389.06
 TLB size : 1024 4K pages
 clflush size : 64
 cache_alignment : 64
 address sizes : 48 bits physical, 48 bits virtual
 power management: ts ttp tm stc 100mhzsteps hwpstate

processor : 5
 vendor_id : AuthenticAMD
 cpu family : 16
 model : 8
 model name : Six-Core AMD Opteron(tm) Processor 2427
 stepping : 0
 cpu MHz : 800.000
 cache size : 512 KB
 physical id : 0
 siblings : 6
 core id : 5
 cpu cores : 6
 apicid : 5
 initial apicid : 5
 fpu : yes
 fpu_exception : yes
 cpuid level : 5
 wp : yes

flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36
 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm 3dnowext 3dnow
 constant_tsc rep_good nonstop_tsc extd_apicid pni monitor cx16 popcnt lahf_lm cmp_legacy
 svm extapic cr8_legacy abm sse4a misalignsse 3dnowprefetch osvw ibs skinit wdt
 bogomips : 4389.08
 TLB size : 1024 4K pages
 clflush size : 64

cache_alignment : 64
address sizes : 48 bits physical, 48 bits virtual
power management: ts ttp tm stc 100mhzsteps hwpstate

processor : 6
vendor_id : AuthenticAMD
cpu family : 16
model : 8
model name : Six-Core AMD Opteron(tm) Processor 2427
stepping : 0
cpu MHz : 800.000
cache size : 512 KB
physical id : 1
siblings : 6
core id : 0
cpu cores : 6
apicid : 8
initial apicid : 8
fpu : yes
fpu_exception : yes
cpuid level : 5
wp : yes
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36
clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm 3dnowext 3dnow
constant_tsc rep_good nonstop_tsc extd_apicid pni monitor cx16 popcnt lahf_lm cmp_legacy
svm extapic cr8_legacy abm sse4a misalignsse 3dnowprefetch osvw ibs skinit wdt
bogomips : 4389.08
TLB size : 1024 4K pages
clflush size : 64
cache_alignment : 64
address sizes : 48 bits physical, 48 bits virtual
power management: ts ttp tm stc 100mhzsteps hwpstate

processor : 7
vendor_id : AuthenticAMD
cpu family : 16

model : 8
 model name : Six-Core AMD Opteron(tm) Processor 2427
 stepping : 0
 cpu MHz : 800.000
 cache size : 512 KB
 physical id : 1
 siblings : 6
 core id : 1
 cpu cores : 6
 apicid : 9
 initial apicid : 9
 fpu : yes
 fpu_exception : yes
 cpuid level : 5
 wp : yes
 flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36
 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm 3dnowext 3dnow
 constant_tsc rep_good nonstop_tsc extd_apicid pni monitor cx16 popcnt lahf_lm cmp_legacy
 svm extapic cr8_legacy abm sse4a misalignsse 3dnowprefetch osvw ibs skinit wdt
 bogomips : 4388.93
 TLB size : 1024 4K pages
 clflush size : 64
 cache_alignment : 64
 address sizes : 48 bits physical, 48 bits virtual
 power management: ts ttp tm stc 100mhzsteps hwpstate

processor : 8
 vendor_id : AuthenticAMD
 cpu family : 16
 model : 8
 model name : Six-Core AMD Opteron(tm) Processor 2427
 stepping : 0
 cpu MHz : 800.000
 cache size : 512 KB
 physical id : 1
 siblings : 6

core id : 2
 cpu cores : 6
 apicid : 10
 initial apicid : 10
 fpu : yes
 fpu_exception : yes
 cpuid level : 5
 wp : yes
 flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36
 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm 3dnowext 3dnow
 constant_tsc rep_good nonstop_tsc extd_apicid pni monitor cx16 popcnt lahf_lm cmp_legacy
 svm extapic cr8_legacy abm sse4a misalignsse 3dnowprefetch osvw ibs skinit wdt
 bogomips : 4389.08
 TLB size : 1024 4K pages
 clflush size : 64
 cache_alignment : 64
 address sizes : 48 bits physical, 48 bits virtual
 power management: ts ttp tm stc 100mhzsteps hwpstate

processor : 9
 vendor_id : AuthenticAMD
 cpu family : 16
 model : 8
 model name : Six-Core AMD Opteron(tm) Processor 2427
 stepping : 0
 cpu MHz : 800.000
 cache size : 512 KB
 physical id : 1
 siblings : 6
 core id : 3
 cpu cores : 6
 apicid : 11
 initial apicid : 11
 fpu : yes
 fpu_exception : yes
 cpuid level : 5

wp : yes
 flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36
 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm 3dnowext 3dnow
 constant_tsc rep_good nonstop_tsc extd_apicid pni monitor cx16 popcnt lahf_lm cmp_legacy
 svm extapic cr8_legacy abm sse4a misalignsse 3dnowprefetch osvw ibs skinit wdt
 bogomips : 4389.07
 TLB size : 1024 4K pages
 clflush size : 64
 cache_alignment : 64
 address sizes : 48 bits physical, 48 bits virtual
 power management: ts ttp tm stc 100mhzsteps hwpstate

processor : 10
 vendor_id : AuthenticAMD
 cpu family : 16
 model : 8
 model name : Six-Core AMD Opteron(tm) Processor 2427
 stepping : 0
 cpu MHz : 800.000
 cache size : 512 KB
 physical id : 1
 siblings : 6
 core id : 4
 cpu cores : 6
 apicid : 12
 initial apicid : 12
 fpu : yes
 fpu_exception : yes
 cpuid level : 5
 wp : yes
 flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36
 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm 3dnowext 3dnow
 constant_tsc rep_good nonstop_tsc extd_apicid pni monitor cx16 popcnt lahf_lm cmp_legacy
 svm extapic cr8_legacy abm sse4a misalignsse 3dnowprefetch osvw ibs skinit wdt
 bogomips : 4389.09
 TLB size : 1024 4K pages

clflush size : 64
cache_alignment : 64
address sizes : 48 bits physical, 48 bits virtual
power management: ts ttp tm stc 100mhzsteps hwpstate

processor : 11
vendor_id : AuthenticAMD
cpu family : 16
model : 8
model name : Six-Core AMD Opteron(tm) Processor 2427
stepping : 0
cpu MHz : 800.000
cache size : 512 KB
physical id : 1
siblings : 6
core id : 5
cpu cores : 6
apicid : 13
initial apicid : 13
fpu : yes
fpu_exception : yes
cpuid level : 5
wp : yes
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36
clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm 3dnowext 3dnow
constant_tsc rep_good nonstop_tsc extd_apicid pni monitor cx16 popcnt lahf_lm cmp_legacy
svm extapic cr8_legacy abm sse4a misalignsse 3dnowprefetch osvw ibs skinit wdt
bogomips : 4389.08
TLB size : 1024 4K pages
clflush size : 64
cache_alignment : 64
address sizes : 48 bits physical, 48 bits virtual
power management: ts ttp tm stc 100mhzsteps hwpstate