

Ατομική Διπλωματική Εργασία

**ΜΕΛΕΤΗ ΤΟΥ ΠΡΟΣΕΓΓΙΣΤΙΚΟΥ ΥΠΟΛΟΓΙΣΜΟΥ ΚΑΤΑ ΤΗΝ ΕΚΤΕΛΕΣΗ
ΕΠΕΡΩΤΗΜΑΤΩΝ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ**

Χάρης Βουτουρή

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2016

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μελέτη Του Προσεγγιστικού Υπολογισμού

Κατά Την Εκτέλεση

Επερωτημάτων Βάσεων Δεδομένων

Χάρης Βουτουρή

Επιβλέπων Καθηγητής

Pedro Trancoso

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

Μάιος 2016

Ευχαριστίες

Θα ήθελα να εκφράσω τις ευχαριστίες μου στον επιβλέποντα καθηγητή μου Δρ. Pedro Trancoso, για την βοήθειά του, την υποστήριξη του και για την πολύ καλή συνεργασία που είχαμε καθ' όλο το χρονικό διάστημα εκπόνησης της διπλωματικής αυτής εργασίας. Οι συμβουλές του, οι οδηγίες του αλλά και οι κατευθύνσεις που μου έδωσε, συνέβαλαν στο μέγιστο, για την επιτυχή κατά την άποψη μου ολοκλήρωση της εργασίας αυτής.

Επίσης θα ήθελα να εκφράσω ένα μεγάλο ευχαριστώ στον διδακτορικό φοιτητή Διαβαστό Ανδρέα για την πολύτιμη υποστήριξη που μου πρόσφερε, καθώς επίσης και για τις πολύτιμες συμβουλές που μου έδινε όλο αυτό το διάστημα.

Ακολούθως, θα ήθελα να ευχαριστήσω τους συνεργάτες – συμφοιτητές μου από το CASPER Group του επιβλέποντα καθηγητή μας, οι οποίοι φρόντιζαν να υπάρχει πάντα ένα ευχάριστο περιβάλλον εργασίας στην ομάδα μας.

Τέλος, θα ήθελα να εκφράσω ιδιαίτερες ευχαριστίες στην οικογένεια και τους φίλους μου που με στήριξαν ηθικά και ψυχολογικά σε ολόκληρη την πορεία μου κατά τη διάρκεια φοίτησης μου στο Πανεπιστήμιο Κύπρου, αλλά και κατά τη δύσκολη αυτή περίοδο ολοκλήρωσης της διπλωματικής μου εργασίας.

Περίληψη

Από την πρώτη εμφάνιση των υπολογιστών, σκοπός τους ήταν να υπολογίζουν γρήγορα και με ακρίβεια αυτό που τους αναθετόταν. Η ανάπτυξη στην υπολογιστική ισχύ, στην χωρητικότητα και την ταχύτητα του αποθηκευτικού χώρου καθώς και η μείωση του μεγέθους των συσκευών που τα εμπεριέχουν, είναι ραγδαία. Κάτι το οποίο δεν βελτιώνετε με τους ίδιους ρυθμούς είναι η κατανάλωση τους σε ενέργεια. Ένα νέο μοντέλο που εμφανίστηκε για να βοηθήσει σε αυτό το σημείο είναι ο προσεγγιστικός υπολογισμός (approximate computing).

Ο προσεγγιστικός υπολογισμός στοχεύει στη βελτίωση της ενεργειακής απόδοσης μέσω της μείωσης/χαλάρωσης της ακρίβειας στο παραγόμενο αποτέλεσμα. Η υπερβολική χαλάρωση όμως μπορεί να επιφέρει ένα αποτέλεσμα του οποίου η προσέγγιση του να μην είναι καλή. Για αυτό χρειάζεται διερεύνηση ο τρόπος και το ποσό της χαλαρότητας. Η μείωση στην ακρίβεια μπορεί να επιτευχθεί με διάφορους τρόπους, είτε σε λογισμικό επίπεδο είτε σε υλικό επίπεδο.

Στόχος της παρούσας διπλωματικής εργασίας είναι η εφαρμογή μεθόδων χαλαρότητας στην κύρια μνήμη και η μελέτη τους. Όπου αυτό συμπεριλαμβάνει την αξιολόγηση των αποτελεσμάτων, τον υπολογισμό της εξοικονόμησης καθώς και την διαπίστωση αιτιών που επηρεάζουν αυτούς τους δύο παράγοντες. Προσεγγίσαμε το θέμα της χαλαρότητας, από δύο πλευρές, την μείωση προσβάσεων και την μείωση του ρυθμού ανανέωσης (refresh-rate).

Η χρήση του υπολογισμού κατά προσέγγιση είναι αρκετά διαδεδομένη και ανεκτή σε εργαλεία που διαχειρίζονται αρχεία πολυμέσων (εικόνες και βίντεο). Αλλά εδώ γίνεται η προσπάθεια εφαρμογής του μοντέλου σε άλλον τομέα, ο οποίος είναι τα επρωτήματα (queries) σε βάσεις δεδομένων.

Ως σημείο εφαρμογής επιλέχθηκε η Υποστήριξη Αποφάσεων (DSS Workloads) και πιο συγκεκριμένα μία ομάδα από επρωτήματα, η TPC-H. Για να είναι εφικτή η αξιολόγηση και σύγκριση των αποτελεσμάτων, χρησιμοποιήθηκε η μετρική Quality of Result (QoR), έτσι ώστε να εκφραστεί η αλλοίωση των παραγόμενων δεδομένων και με την βοήθεια της βιβλιοθήκης PAPI αναλύσαμε το κέρδος σε επίδοση το οποίο εκφράστηκε και ως μείωση σε κατανάλωση.

Ακολουθως μελετήθηκε το κέρδος στην κατανάλωση σε σύγκριση με το σφάλμα στο αποτέλεσμα.

Μέσα από τα αποτελέσματα διαπιστώθηκε ότι σε πολλά από τα επερωτήσαντα είναι εφικτό να εξοικονομηθεί περίπου το 10% σε ενέργεια ενώ η απώλεια στο αποτέλεσμα να είναι σχεδόν μηδενική (μικρότερη του 5%). Σε άλλες περιπτώσεις όπου η αλλοίωση στο αποτέλεσμα είναι περισσότερη και πάλι παραμένει σε χαμηλότερο ποσοστό από την εξοικονόμηση αλλά η αναλογία τους μειώνετε.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή	1
1.1	Υποκίνηση Εργασίας	1
1.2	Στόχος και Συνεισφορά Εργασίας	2
Κεφάλαιο 2	Σχετικές δουλειές.....	3
Κεφάλαιο 3	Προσεγγιστικός Υπολογισμός	5
3.1	Γενικά για τον Προσεγγιστικό Υπολογισμό	5
3.2	Μέθοδοι Χαλαρότητας	6
Κεφάλαιο 4	Πειραματική Διάταξη.....	8
4.1	Τρόπος Αξιολόγησης.....	8
4.2	Μεθοδολογία.....	10
4.3	Προσομοίωση των Μεθόδων Χαλαρότητας	12
4.4	Σύστημα Αξιολόγησης.....	12
Κεφάλαιο 5	Αποτελέσματα	13
5.1	Μείωση Ρυθμού Ανανέωσης	13
5.1.1	Ποιότητα Αποτελέσματος.....	13
5.1.2	Κατανάλωση	14
5.1.3	Ωφελιμότητα.....	16
5.2	Μείωση των Προσβάσεων	18
5.2.1	Ποιότητα Αποτελέσματος.....	18
5.2.2	Κατανάλωση	19
5.2.3	Ωφελιμότητα	20
5.3	Σύγκριση των Μεθόδων Χαλαρότητας	22
5.3.1	Ίδιο ποσοστό εφαρμογής	23
5.3.2	Εφαρμογή στις ίδιες πλειάδες	25
5.4	Περεταίρω για τη Μείωση των Προσβάσεων.....	29
5.4.1	Έλεγχος στις πλειάδες που αφαιρούνται.....	29

5.4.2 Έλεγχος στην χρήση διαφορετικών αλγόριθμων κατά την εκτέλεση των επερωτημάτων	35
5.5 Συνοπτικά Αποτελέσματα	37
Κεφάλαιο 6 Συμπεράσματα – Μελλοντική Εργασία	39
6.1 Συμπεράσματα	39
6.2 Μελλοντική Εργασία.....	40
Βιβλιογραφία	41
Παράρτημα Α	A-1
Παράρτημα Β	B-1
Παράρτημα Γ	Γ-1
Παράρτημα Δ	Δ-1

Κεφάλαιο 1

Εισαγωγή

1.1	Υποκίνηση Εργασίας	1
1.2	Στόχος και Συνεισφορά Εργασίας	2

1.1 Υποκίνηση Εργασίας

Τα τελευταία χρόνια έχει παρατηρηθεί ραγδαία αύξηση στην ζήτηση των φορητών υπολογιστικών συσκευών (φορητοί υπολογιστές, tablets και κινητά τηλέφωνα). Την ίδια στιγμή, όπου η μεταφορά μας προς τη φορητότητα συνεχίζεται και οι συσκευές γίνονται μικρότερες, η ποσότητα ισχύος που χρειάζονται για να λειτουργήσουν δεν μειώνεται στον ίδιο βαθμό. Αυτό οδηγεί σε συσκευές με μικρή χρονική διάρκεια σε μπαταρία και που υπερθερμαίνονται εύκολα. Μια λύση σε αυτά τα προβλήματα είναι η μείωση του ποσού ενέργειας που χρησιμοποιείται από μία υπολογιστική συσκευή εισάγοντας λιγότερη ακρίβεια κατά την επεξεργασία. Με αυτό τον τρόπο επιτυγχάνεται μείωση της κατανάλωσης χωρίς να εμφανίζεται μείωση στην επίδοση, το οποίο είναι πολύ επιθυμητό.

Η ιδέα του προσεγγιστικού υπολογισμού είναι ότι η υπερβολική ακρίβεια που υπάρχει στους υπολογιστές δεν είναι πάντοτε αναγκαία. Ενώ η ακρίβεια είναι ζωτικής σημασίας σε ορισμένες εφαρμογές, υπάρχουν παραδείγματα εφαρμογών όπου το αποτέλεσμα τους εξ ορισμού είναι προσεγγιστικό όπως η επεξεργασία και συμπίεση αρχείων πολυμέσων (εικόνες, βίντεο κλπ), οι τυχαίοι αλγόριθμοι, η μηχανική μάθηση (machine learning) καθώς και η ανάλυση δεδομένων από αισθητήρες. Η μειωμένη ακρίβεια έχει ήδη ενταχθεί σε μερικά σημεία της καθημερινής μας επαφής με υπολογιστικές συσκευές και η επέκταση σε περισσότερα σημεία είναι καλό να μελετηθεί.

Οι τεχνικές /μηχανισμοί με τους οποίους εφαρμόζεται ο προσεγγιστικός υπολογισμός ονομάζονται μέθοδοι χαλαρότητας και μπορούν να διαχωριστούν σε τρεις περιοχές, βάση του πώς υλοποιούνται. Αυτές οι περιοχές είναι η υλοποίηση αλγορίθμων οι οποίοι δεν παράγουν πάντοτε ορθό αποτέλεσμα, η χρήση λογισμικού το οποίο πιθανόν να τερματίζει μόλις πιστεύει ότι έχει βρει μίαν αρκετά καλή λύση ή γενικά να μην στοχεύει στην εύρεση της λύσης με απόλυτη ακρίβεια και σε υλικό επίπεδο όπου επηρεάζονται ή σχεδιάζονται νέα κομμάτια τα οποία έχουν καλύτερη ενεργειακή απόδοση αλλά δεν παράγουν αποτέλεσμα με πλήρη ακρίβεια.

Αξίζει μεγάλη προσοχή η κύρια μνήμη γιατί είναι κομμάτι σχεδόν σε όλες τις υπολογιστικές συσκευές. Η μνήμη συμβάλλει σημαντικά στην συνολική κατανάλωση ενός συστήματος.

Επίσης ξεχωρίζει γιατί σε πολλά εξαρτήματα μπορούν να βρεθούν στιγμές που είναι σχεδόν αχρείαστα και έτσι τα θέτουμε σε κατάσταση sleep mode [6], για να καταναλώνουν πολύ λιγότερη ενέργεια. Ενώ στην περίπτωση της κύριας μνήμης δεν ισχύει γιατί οι απαιτήσεις της σε ενέργεια είναι συνεχές και σχετικά σταθερές γιατί θέλουμε τα δεδομένα που φυλάγονται μέσα να μην αλλοιωθούν.

Η χρήση βάσεων δεδομένων υπάρχει σε πάρα πολλές εφαρμογές και αυτές οι εφαρμογές ολοένα και αυξάνονται καθώς η ύπαρξη Συστημάτων Διαχείρισης Βάσεων (DataBase Management System) επεκτείνεται σε νέες πλατφόρμες και λειτουργικά έχοντας καλύψει σχεδόν κάθε μηχανή. Η ιδέα της εφαρμογής του προσεγγιστικού μοντέλου σε βάσεις δεδομένων βασίζεται στο γεγονός ότι συχνά χρησιμοποιούνται για υπολογισμούς πάνω σε μεγάλο όγκο δεδομένων όπου το αποτέλεσμα σε πολλές περιπτώσεις δεν θεωρείται τόσο κρίσιμο και ευαίσθητο.

1.2 Στόχος και Συνεισφορά Εργασίας

Ο προσεγγιστικός υπολογισμός ανταλλάζει ακρίβεια για καλύτερη επίδοση και ενεργειακή αποδοτικότητα. Η απώλεια στην ακρίβεια υπάρχει ήδη στην λειτουργία των υπολογιστών αλλά σε μικρό ποσοστό το οποίο μπορεί να διαχειριστεί και να διορθωθεί ή να αγνοηθεί. Τέτοιες περιπτώσεις βρίσκονται στην λειτουργία της κύριας μνήμης, όπου μέσω του μηχανισμού Κώδικας Διόρθωσης Σφαλμάτων (ECC) διορθώνονται τυχόν λάθη κατά την ανάρτηση πληροφοριών και μία άλλη περίπτωση είναι στην συμπίεση αρχείων πολυμέσων όπως ψηφιακές εικόνες και βίντεο.

Στόχος αυτής της εργασίας είναι η μελέτη της ύπαρξης/εισαγωγής λαθών κατά την χρήση βάσεων δεδομένων. Η ύπαρξη λαθών θεωρούμε ότι εμφανίστηκε λόγω της εφαρμογής δύο μεθόδων χαλαρότητας, της μείωσης των προσβάσεων και της μείωσης του ρυθμού ανανέωσης (refresh rate) στην κύρια μνήμη. Άρα η μελέτη μας συμπεριλαμβάνει την προσομοίωση αυτών των μεθόδων, για να είναι δυνατή η μέτρηση των σφαλμάτων και της εξοικονόμησης σε ενέργεια που προσφέρουν, έτσι ώστε να εκτιμηθεί σε ποιες περιπτώσεις είναι επιθυμητή αυτή η ανταλλαγή και από ποιους παράγοντες επηρεάζονται.

Για την προσομοίωση και την συλλογή των μετρήσεων και αποτελεσμάτων, έχει δημιουργηθεί ένα εργαλείο το οποίο είναι ικανό να αξιολογήσει τις επιπτώσεις της εφαρμογής των μεθόδων χαλαρότητας, στην ποιότητα του αποτελέσματος καθώς και στην κατανάλωση την παραγωγής του.

Σε πολλές από τις περιπτώσεις που μελετήθηκαν το ποσοστό εξοικονόμησης ενέργειας υπερβαίνει του ποσοστού απώλειας στο αποτέλεσμα. Σε αρκετά σημεία, η παραγωγή ενός αποτελέσματος το οποίο προσεγγίζει πάρα πολύ το αυθεντικό αποτέλεσμα (ποιότητα αποτελέσματος μεγαλύτερη του 95%) αντιστοιχεί σε μία μείωση στην κατανάλωση κατά ένα ποσοστό κοντά στο 10% και σε μεμονωμένες περιπτώσεις φθάνει σχεδόν στο 30%. Ενώ αν αφήσουμε το σφάλμα να γίνει πιο αισθητό η εξοικονόμηση ολοένα και αυξάνεται.

Κεφάλαιο 2

Σχετικές Εργασίες

Η ιδέα πάνω στην οποία βασίζεται ο προσεγγιστικός υπολογισμός (approximate computing) [13], [3] είναι ότι η αυστηρή έννοια της ορθότητας είναι περιττή και επιβάλλει σημαντική αναποτελεσματικότητα, έτσι η αρχή του είναι να γίνετε αποτελεσματικός υπολογισμός για την παραγωγή αποτελεσμάτων που είναι αρκετά ανεκτά ή επαρκούς ποιότητας. Είναι σχετικά ένα νέο θέμα προς έρευνα το οποίο προσφέρει μία νέα διάσταση για τη βελτιστοποίηση υπολογιστών πλατφόρμων. Οι τομείς βελτιστοποίησης που καλύπτει είναι στην ενεργειακή κατανάλωση και στην υπολογιστική απόδοση.

Έχουν δημιουργηθεί διάφορα εργαλεία τα οποία αυτοματοποιούν την χαλαρότητα της ακρίβειας και διευκολύνουν ως ένα βαθμό τον προγραμματιστή. Για παράδειγμα το ACCEPT [9], όπου ένα κύριο συστατικό του είναι ένας μεταγλωττιστής για προσεγγιστικό προγραμματισμό, που συνδυάζει την αυτοματοποίηση και τον χειρισμό από τον προγραμματιστή. Όπως ένας τυπικός μεταγλωττιστής μπορεί να προσφέρει κάποιες βελτιστοποιήσεις έτσι και το ACCEPT, αλλά οι βελτιστοποιήσεις του επιφέρουν απώλεια στο αποτέλεσμα και χρειάζονται καθοδήγηση από την προγραμματιστή, ο οποίος έχει μία δυναμική ανάδραση (feedback) των επιπτώσεων καθώς εφαρμόζει αυτές τις βελτιστοποιήσεις. Ακόμα ένα παράδειγμα είναι το REACT [14], το οποίο είναι ένα πλαίσιο από εργαλεία μοντελοποίησης (modeling framework) που επεκτείνει τις ικανότητες του ACCEPT. Δίνει την ικανότητα στον χρήστη να εισάγει επιπρόσθετα λάθη κατά την εκτέλεση κάποιας εφαρμογής. Έτσι επιτρέπει να εφαρμοστούν διαφορετικές τεχνικές προσέγγισης σε επιλεγμένα σημεία ενός προγράμματος. Επίσης με την χρήση ενός μοντέλου ενέργειας μπορεί να υπολογίσει δυναμικά την ανταλλαγή μεταξύ αποδοτικότητας-ακρίβειας.

Έχει παρατηρηθεί μεγαλύτερη απήχηση στην ενεργειακή εξοικονόμηση που μπορεί να επιτευχθεί μέσω της εφαρμογής του προσεγγιστικού υπολογισμού [1], [12]. Πολλοί πιστεύουν ότι είναι ένας ελπιδοφόρος τομέας και αξίζει την προσοχή από το κοινό. Όμως ένας σημαντικός παράγοντας που καθορίζει το πόσο πολύ μπορεί να χαλαρωθεί η ακρίβεια στο σύστημα, είναι ο τύπος του προγράμματος που θα εκτελεστεί πάνω σε αυτό το σύστημα. Διαφορετικές εφαρμογές, παρέχουν διαφορετικά ποσοστά λάθους κάτω από τις ίδιες συνθήκες. Έτσι έχει παρουσιαστεί το Rumba [2], ένα διαδικτυακό σύστημα που διαχειρίζεται την ποιότητα των αποτελεσμάτων σε πλατφόρμες προσεγγιστικού υπολογισμού. Μέσω διαφόρων τεχνικών μπορεί να εξακριβώσει πότε υπάρχει μεγάλο σφάλμα στα αποτελέσματα και τα διορθώνει με τον επανυπολογισμό μερικών σημείων της εκτέλεσης.

Σχετικά με την μείωση των προσβάσεων στη μνήμη, μέσω της μείωσης των επαναλήψεων (loop perforation) σε βρόγχους έχει παρατηρηθεί [7] ότι δεν έχει παρόμοια αποτελέσματα για όλους τους τύπους εφαρμογών. Πιο συγκεκριμένα μέσα από την μελέτη του Sidiroglou και Misailovic [10], οι οποίοι με τη χρήση ενός διαφοροποιημένου μεταγλωττιστή και μίας μετρικής

για την ακρίβεια, διαπίστωσαν ότι σε συγκεκριμένες περιπτώσεις μπορεί να επωφεληθείς με αυξημένη απόδοση μέχρι και 7 φορές έναντι μικρού ποσοστού ακρίβειας, περίπου 10%.

Στο θέμα της μείωσης του ρυθμού ανανέωσης (refresh rate) της μνήμης, έχουν υπάρξει αρκετές μελέτες οι οποίες η μία κλιμακώνει την επόμενη. Στην περίπτωση του RAIDR [4] μείωσαν το ρυθμό σχεδόν 75% χωρίς να προκαλέσουν επιπλέον λάθη. Εκμεταλλεύτηκαν το γεγονός ότι κάθε κομμάτι της μνήμης μπορεί να κρατήσει αναλλοίωτη την πληροφορία του για διαφορετικό χρονικό διάστημα. Με απλές μεθόδους εντόπιζαν το πόσο μπορούσαν να διατηρήσουν τα δεδομένα τους τα διαφορετικά μέρη (cells) της μνήμης. Έτσι τα χώριζαν σε ομάδες (bins) με διαφορετικό ρυθμό ανανέωσης. Το ενδιαφέρον είναι ότι κέρδισαν σε ενεργειακή απόδοση χωρίς να χρειάζεται έλεγχος λαθών.

Στη συνέχεια, το ArchShield [8] ασχολήθηκε σε πιο μεγάλη μείωση η οποία έφθασε μέχρι και τέσσερις φορές αυτήν του RAIDR. Άρα σχεδόν 94% μείωση σε σύγκριση με τον αρχικό ρυθμό ανανέωσης. Σε αυτό το επίπεδο όμως το φόρτο για την εύρεση και διόρθωση των λαθών δεν ωφελούσε το σύστημα σε επίδοση και κατανάλωση ενέργειας.

Για την επιπλέον βελτίωση αυτών των μεθόδων, χρειάστηκε να επέμβουν με περισσότερη προσοχή. Έτσι το Flicker [5] δίνει την δυνατότητα στον προγραμματιστή να διαχωρίζει τα δεδομένα του σε κρίσιμα και μη κρίσιμα. Κατά την εκτέλεση αποθηκεύονται σε διαφορετικές σελίδες της μνήμης, όπου μόνο στις σελίδες που περιέχουν μη κρίσιμα δεδομένα μειώνεται ο ρυθμός ανανέωσης. Τα αποτελέσματα δείχνουν, ότι σε ένα καλά ορισμένο πρόγραμμα, πέτυχαν μέχρι και 25% μείωση στην ενεργειακή κατανάλωση με πολύ χαμηλά λάθη στο αποτέλεσμα των προγραμμάτων.

Μία διαφορετική μέθοδο χρησιμοποίησε ο Lucas [6] ο οποίος έκανε την παρατήρηση ότι η σημαντικότητα των δεδομένων μπορεί να φθάσει σε επίπεδο bit και ότι στα περισσότερα συστήματα σήμερα υπάρχουν πολλαπλές κάρτες κύριας μνήμης. Παρουσίασε το Sparkk, μία μέθοδο προσεγγιστικής αποθήκευσης (approximate storage). Κάθε byte που πρόκειται να φυλαχτεί στην κύρια μνήμη το χωρίζει σε μέρη (όσα είναι και οι κάρτες μνήμης του συστήματος) και αποθηκεύει τα χαμηλής σημασίας bits (low significant bits) σε μνήμη με πολύ χαμηλό ρυθμό ανανέωσης και μετά κάθε μέρος σε άλλη μνήμη που έχει μεγαλύτερο ρυθμό ανανέωσης. Έτσι τα μεγαλύτερης σημασίας bits (most significant bits) καταλήγουν να είναι αποθηκευμένα σε έναν χώρο που δεν έχει υποστεί μεγάλη χαλαρότητα. Τα αποτελέσματα ήταν καλύτερα και από την περίπτωση του Flicker. Για να έχει αποτελέσματα παρόμοιας ποιότητας με το Flicker χρειάστηκε να φθάσει σχεδόν στο μισό ρυθμό ανανέωσης.

Η μεθοδολογία που θα ακολουθηθεί βασίζεται στα βήματα που ακολουθήθηκαν από τον Trancoso [11] στην προσπάθειά του να αξιολογήσει την γενική επίπτωση του προσεγγιστικού υπολογισμού σε επερωτήματα Υποστήριξης Αποφάσεων (DSSW). Με τη διαφορά ότι στην περίπτωση του, εκμεταλλεύταν την παράλληλη εκτέλεση του ίδιου επερωτήματος. Δηλαδή υπήρχε η προσπάθεια εντοπισμού του πόσο λάθος προκύπτει το αποτέλεσμα μέσα από τις πρώτες απαντήσεις που παράγονταν από τις παράλληλες εκτελέσεις, έτσι ώστε να κριθεί το πόσες φορές συνολικά θα εκτελεστεί αυτό το επερωτήμα έτσι ώστε να παρθεί μία μέση λύση.

Κεφάλαιο 3

Προσεγγιστικός Υπολογισμός

3.1	Γενικά για τον Προσεγγιστικό Υπολογισμό	5
3.2	Μέθοδοι Χαλαρότητας	6

3.1 Γενικά για τον Προσεγγιστικό Υπολογισμό

Από την πρώτη εμφάνιση των υπολογιστών, σκοπός τους ήταν να υπολογίζουν γρήγορα και με ακρίβεια αυτό που τους αναθετόταν. Με το πέρασμα των χρόνων, οι υπολογιστικές δυνάμεις αυτών των συσκευών έχουν αυξηθεί και το μέγεθος τους έχει μειωθεί. Όμως κάτι το οποίο δεν βελτιώνετε με τους ίδιους ρυθμούς είναι η κατανάλωση τους σε ενέργεια. Σε συνδυασμό με αυτό το πρόβλημα και την παρατήρηση ότι δεν χρειάζονται όλες οι εργασίες/εφαρμογές την ίδια ακρίβεια, τον τελευταίο καιρό έχει φανερωθεί εντονότερα ο όρος προσεγγιστικός υπολογισμός (approximate computing).

Η αρχική χρήση του προσεγγιστικού υπολογισμού εμφανίστηκε κυρίως για να φέρει τρόπους επίλυσης σε προβλήματα τα οποία χαρακτηρίζονταν από την πολύ μεγάλη τους πολυπλοκότητα (εκθετική). Όπου η γρήγορη εύρεση μίας λύσης, έστω και αν δεν ήταν εντελώς ορθή, ήταν επιθυμητή. Αφού αυτός ο τρόπος, της μειωμένης ακρίβειας έγινε αποδεκτός, ξεκίνησαν οι έρευνες εφαρμογής του σε άλλα σημεία στα οποία η ύπαρξη προσεγγιστικού αποτελέσματος μπορεί να μην είναι αντιληπτή ή γενικά να είναι αποδεκτή και έτσι η χρήση του προσεγγιστικού υπολογισμού να επιφέρει βελτίωση στην επίδοση και στην κατανάλωση. Η μείωση στην ακρίβεια μπορεί να επιτευχθεί με διάφορους τρόπους, είτε σε λογισμικό επίπεδο είτε σε υλικό επίπεδο και ονομάζονται μέθοδοι χαλαρότητας.

Τα βασικά στάδια του σχεδιασμού και της εφαρμογής ενός μοντέλου για προσεγγιστικό υπολογισμό, κάτω από μία μέθοδο χαλαρότητας, είναι ο εντοπισμός στοιχείων μιας εφαρμογής που μπορεί να ανεχθεί σφάλματα, η μέτρηση/υπολογισμός της συχνότητας και του μεγέθους στο σφάλμα καθώς και η παρατήρηση της βελτίωσης της επίδοσης ή της εξοικονόμησης σε ενέργεια.

Η κρισιμότητα και το μέγεθος του λάθους πολλές φορές εξαρτάται από το είδος της εφαρμογής που θα χρησιμοποιήσει το μοντέλο του προσεγγιστικού υπολογισμού. Μπορούμε πιο εύκολα να ανεχτούμε λάθη σε περιπτώσεις που έχουμε μεγάλο αριθμό από στοιχεία να λάβουμε υπόψη ή να τα επεξεργαστούμε. Ένας τύπος τέτοιων περιπτώσεων είναι στην επεξεργασία και

συμπύεση εικόνων [14] ή βίντεο, όπου υπάρχει εξ ορισμού απώλεια στο αποτέλεσμα. Μία άλλη περίπτωση είναι σε βάσεις δεδομένων [11]. Σίγουρα υπάρχουν περιπτώσεις που χειρίζονται πολύ ευαίσθητα δεδομένα όπου δεν είναι ανεκτό κανένα λάθος. Παρ' όλα αυτά, χρησιμοποιούνται ταυτόχρονα και για υπολογισμούς πάνω σε μεγάλο όγκο δεδομένων όπου το αποτέλεσμα δεν θεωρείται τόσο κρίσιμο και ευαίσθητο.

3.2 Μέθοδοι Χαλαρότητας

Η χρήση κάποιου είδους υπολογισμού κατά προσέγγιση γίνεται ήδη σε αρκετούς τομείς. Μία κατηγορία αλγορίθμων η οποία εμπίπτει σε αυτό το μοντέλο είναι οι τυχαίοποιημένοι αλγόριθμοι. Απευθύνονται στον αποδοτικότερο υπολογισμό της απάντησης σε σύγκριση με την εύρεση της ακριβής απάντησης. Δηλαδή, για να μειωθεί η πολυπλοκότητα αυτών των προβλημάτων και να αυξηθεί η επίδοση στην επίλυση των προβλημάτων, επιτρέπουν την χαλαρότητα/παράλειψη μερικών συνθηκών του προβλήματος αλλά σε μερικές περιπτώσεις η απάντηση τους ενδέχεται να είναι λανθασμένη.

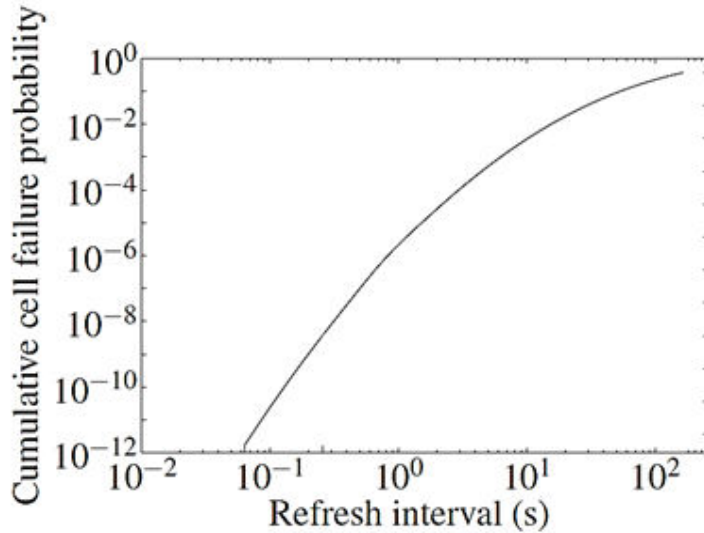
Σε τομείς όπου η ανθρώπινη αίσθηση και αντίληψη πρόκειται να κρίνει την ποιότητα του αποτελέσματος, γίνεται εκτεταμένη χρήση του προσεγγιστικού υπολογισμού. Αυτοί οι τομείς περιλαμβάνουν τις ψηφιακές εικόνες, ήχους, βίντεο καθώς και ανάλυση/απεικόνιση των τριών διαστάσεων. Αυτά τα είδη πληροφορίας έχουν ψηφιοποιηθεί με μεγάλη λεπτομέρεια την οποία ο άνθρωπος πολλές φορές δεν μπορεί να την διακρίνει, έτσι χρησιμοποιούνται αλγόριθμοι συμπίεσης οι οποίοι έχουν απώλεια στο τελικό τους αποτέλεσμα αλλά αποσκοπούν σε μεγάλο λόγο συμπίεσης. Αυτό μας επιτρέπει στο να έχουμε μικρά (σε μέγεθος) αρχεία για αποθήκευση καθώς και για ανάρτηση σε περιπτώσεις μετάδοσης τους.

Σε πιο προχωρημένες πτυχές του προσεγγιστικού υπολογισμού γίνονται αλλαγές για επίτευξη μίας γενικής εξοικονόμησης και όχι κατά την εφαρμογή κάποιων συγκεκριμένων αλγορίθμων. Μία μέθοδος για αυτού του είδους αλλαγή, είναι ο χειρισμός της τάσης (voltage) και συχνότητας (frequency) της κύριας μνήμης έτσι ώστε να μειωθεί η κατανάλωση και να αυξηθεί η ενεργειακή απόδοση. Όμως με την μείωση της συχνότητας μειώνετε και η επίδοση της μνήμης. Άρα επωφελείσαι σε περιπτώσεις όπου η χρησιμοποίηση της μνήμης δεν είναι εκτεταμένη.

Σε αυτή την εργασία το θέμα της χαλαρότητας επικεντρώνεται στη μνήμη και προσεγγίσθηκε από δύο πλευρές, την μείωση προσβάσεων και την μείωση του ρυθμού ανανέωσης (refresh-rate).

Τα κελιά (cells) της κύριας μνήμης πρέπει να ανανεώνονται σε τακτά χρονικά διαστήματα για να διατηρούνται τα δεδομένα τους ανέπαφα. Η επέκταση του χρονικού διαστήματος ανάμεσα στις ανανεώσεις (μείωση του ρυθμού ανανέωσης) οδηγεί σε αλλοίωση δεδομένων γιατί οι πυκνωτές που περιέχονται μέσα στα κελιά χάνουν την φόρτιση τους με την πάροδο του χρόνου.

Ωστόσο, το κάθε κελί έχει το δικό του χρονικό διάστημα ανοχής για διατήρηση της πληροφορίας του. Έτσι οι κατασκευαστές θέτουν αυτό το διάστημα χαμηλά, για να εξασφαλίζουν την ακεραιότητα των δεδομένων σε όλα τα κελιά (συνήθως στα 64ms). Αυτή η αυτόματη λειτουργία χρειάζεται ενέργεια για να επιτευχθεί. Άρα η μείωση του ρυθμού ανανέωσης επιφέρει μείωση της ενέργειας που καταναλώνεται από την κύρια μνήμη αλλά



Σχήμα 1: Χρονικό περιθώριο διατήρησης πληροφορίας σε κελιά (πιθανότητα αλλοίωσης δεδομένων μετά για κάθε χρονικό διάστημα) [4]

Refresh Cycle (s)	Error Rate
0.064	1.0×10^{-12}
0.1	1.0×10^{-10}
0.2	1.0×10^{-9}
0.5	4.2×10^{-8}
0.7	2.6×10^{-7}
1	3.8×10^{-6}
2	2.1×10^{-5}
5	1.3×10^{-4}
10	5.0×10^{-4}
20	1.5×10^{-2}
45	1.0×10^{-1}

Πίνακας 1: Χρονικό διάστημα ανάμεσα στις ανανεώσεις και πιθανότητα σφάλματος (bit flip)

ταυτοχρόνως φέρει και αλλοίωση κάποιων δεδομένων της. Μέσα από διάφορες έρευνες [4] [5] έχει βρεθεί το ποσοστό σφάλματος (error rate) σε διαφορετικούς ρυθμούς ανανέωσης, τα οποία φαίνονται πιο κάτω. Έτσι μπορούμε να αναφερόμαστε σε ποσοστό σφάλματος και να υπονοούμε τον αντίστοιχο ρυθμό ανανέωσης.

Το ποσοστό σφάλματος εκφράζει ότι κατά την ανάρτηση πληροφοριών από τη μνήμη, υπάρχει κάποια πιθανότητα (ίση με το ποσοστό σφάλματος) η πληροφορία να είναι αλλοιωμένη. Η αλλοίωση της πληροφορίας σε αυτή την περίπτωση είναι ότι έχει γίνει αντιστροφή σε κάποιο bit της.

Η επόμενη μέθοδος χαλαρότητας που εφαρμόζεται σε αυτή την εργασία είναι η μείωση των προσβάσεων στην κύρια μνήμη. Μπορεί να υλοποιηθεί σε λογισμικό επίπεδο και επιτυγχάνεται με την μορφοποίηση των βρόχων (loop perforation) έτσι ώστε να εκτελούν ένα υποσύνολο των επαναλήψεων τους. Ο στόχος είναι να μειωθεί η ποσότητα της υπολογιστικής εργασίας (και επομένως η ποσότητα του χρόνου ή / και άλλων πόρων, όπως ισχύς) που ο υπολογισμός απαιτεί για την παραγωγή των επιθυμητών αποτελεσμάτων. Φυσικά το ποια στοιχεία θα παραλειφθούν και το πλήθος τους ενδέχεται να έχουν μεγάλη σημασία στο πόσο ασφαλή θα είναι το αποτέλεσμα.

Κεφάλαιο 4

Πειραματική Διάταξη

4.1	Τρόπος Αξιολόγησης	8
4.2	Μεθοδολογία	10
4.3	Προσομοίωση των Μεθόδων Χαλαρότητας	12
4.4	Σύστημα Αξιολόγησης	12

4.1 Τρόπος Αξιολόγησης

Τα ερωτήματα των βάσεων δεδομένων χωρίζονται σε διάφορα ήδη. Ένα από αυτά ονομάζεται Υποστήριξης Αποφάσεων (DSS) και περιλαμβάνει τα ερωτήματα τα οποία αποσκοπούν στη στήριξη διαχειριστικών αποφάσεων που συχνά στηρίζονται σε ιστορικά δεδομένα. Αυτό το είδος χαρακτηρίζεται από το γεγονός ότι περιέχει “δύσκολες” ερωτήσεις, που είναι πολύπλοκες και χρειάζονται να χρησιμοποιήσουν ένα μεγαλύτερο όγκο δεδομένων. Μία ομάδα από τέτοιου είδους ερωτημάτων είναι η TPC-H [16]. Περιλαμβάνει μία σειρά από 22 ερωτήματα (από το q1.sql μέχρι το q22.sql) επιχειρηματικού προσανατολισμού. Χρησιμοποιούνται ως μία μέθοδος συγκριτικής αξιολόγησης, η οποία μιμείται τα συστήματα υποστήριξης αποφάσεων που εξετάζουν μεγάλους όγκους δεδομένων, συμπεριλαμβάνονται ερωτήματα με υψηλό βαθμό πολυπλοκότητας, που δίνουν απαντήσεις σε κρίσιμα επιχειρηματικά ζητήματα. Μελετήθηκαν μόνο τα 18 από αυτά γιατί τα υπόλοιπα υπήρχαν περιπτώσεις όπου ο χρόνος εκτέλεσης τους ήταν πολύ μεγάλος. Αυτά που εξαιρέθηκαν είναι τα q7, q17, q20 και q21. Για να είναι εφικτή η εκτέλεση τους, χρειάζεται η συσκευή η οποία θα τα τρέξει να έχει εγκατεστημένο ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων (DBMS) το οποίο στην περίπτωση μας ήταν το PostgreSQL. Επίσης τα ερωτήματα της ομάδας TPC-H χωρίστηκαν σε κατηγορίες και ο διαχωρισμός τους φαίνεται πιο κάτω:

Value	Records
q1, q4, q5, q6, q8, q9, q12, q13, q14, q15, q19, q22	q2, q3, q10 q11, q16, q18

Πίνακας 2α: Ομάδες των ερωτημάτων

Η κατηγορία Value είναι σε περιπτώσεις όπου το αποτέλεσμα είναι συνοπτικό, δηλαδή μία ή περισσότερες τιμές που δεν καθορίζουν μία εγγραφή που βρίσκεται στην βάση, αλλά μίαν ομάδα από εγγραφές. Η κατηγορία records είναι για τις υπόλοιπες περιπτώσεις, δηλαδή όπου το αποτέλεσμα είναι μία ακολουθία από εγγραφές.

Αποτέλεσμα του q6 - Value	revenue ----- 123141078.2283 (1 row)
Αποτέλεσμα του q11 - Records	ps_partkey value -----+----- 129760 17538456.86 191287 16474801.97 (1048 rows)

Πίνακας 2b: Παράδειγμα αποτελέσματος από κάθε ομάδα των ερωτημάτων

Για να είναι εφικτή η αξιολόγηση και σύγκριση των αποτελεσμάτων, χρησιμοποιήθηκε η μετρική Quality of Result (QoR) [11]. Συνήθως τα αποτελέσματα των ερωτημάτων είναι ένα πλήθος από πλειάδες, έτσι τα λάθη χωρίζονται σε 3 διαφορετικές περιπτώσεις. Να συμπεριληφθεί στο αποτέλεσμα μία μη επιθυμητή πλειάδα. Να μην συμπεριληφθεί στο αποτέλεσμα μία επιθυμητή πλειάδα και στο να συμπεριληφθεί μία επιθυμητή πλειάδα αλλά για λανθασμένο λόγο. Έτσι η πράξη υπολογισμού της μετρικής QoR είναι:

$$QoR = Accuracy_{TP} \times Accuracy_{FP} \times Accuracy_{Own}$$

Όπου το Accuracy_{TP} αντιπροσωπεύει την ακρίβεια στις Ορθά Θετικές πλειάδες στο αποτέλεσμα. Το Accuracy_{FP} αντιπροσωπεύει την ακρίβεια στις Λανθασμένα Θετικές πλειάδες και το Accuracy_{Own} αντιπροσωπεύει την ακρίβεια στις τιμές στα πεδία των πλειάδων του αποτελέσματος. Άρα πιο αναλυτικά η πράξη μπορεί να εκφραστεί ως:

$$QoR = \frac{\#TP}{\#ANS} \times \frac{\#TP}{\#RES} \times \left(1 - \frac{\sum |ERR|}{\#RES}\right)$$

	Παράδειγμα εφαρμογής μετρικής			
	Αυθεντικό	Αλλοιωμένο	Σύγκριση αλλοιωμένου με αυθεντικό	
Πλειάδες Αποτελέσματος	100	105	Δεν εμφανίστηκαν 5 σωστές	Εμφανίστηκαν 10 νέες
QoR	100%	85.95%	(100-5)/100	(100-5)/105

Πίνακας 3: Παράδειγμα εφαρμογής μετρικής σε ένα αποτέλεσμα της ομάδας Records, όπου δεν υπάρχει αλλοίωση στα πεδία των πλειάδων

Η μέγιστη τιμή που μπορεί να πάρει αυτή η μετρική είναι 100% σε περιπτώσεις όπου το αλλοιωμένο αποτέλεσμα είναι εντελώς το ίδιο με το αυθεντικό, ενώ καθώς διαφοροποιείται η

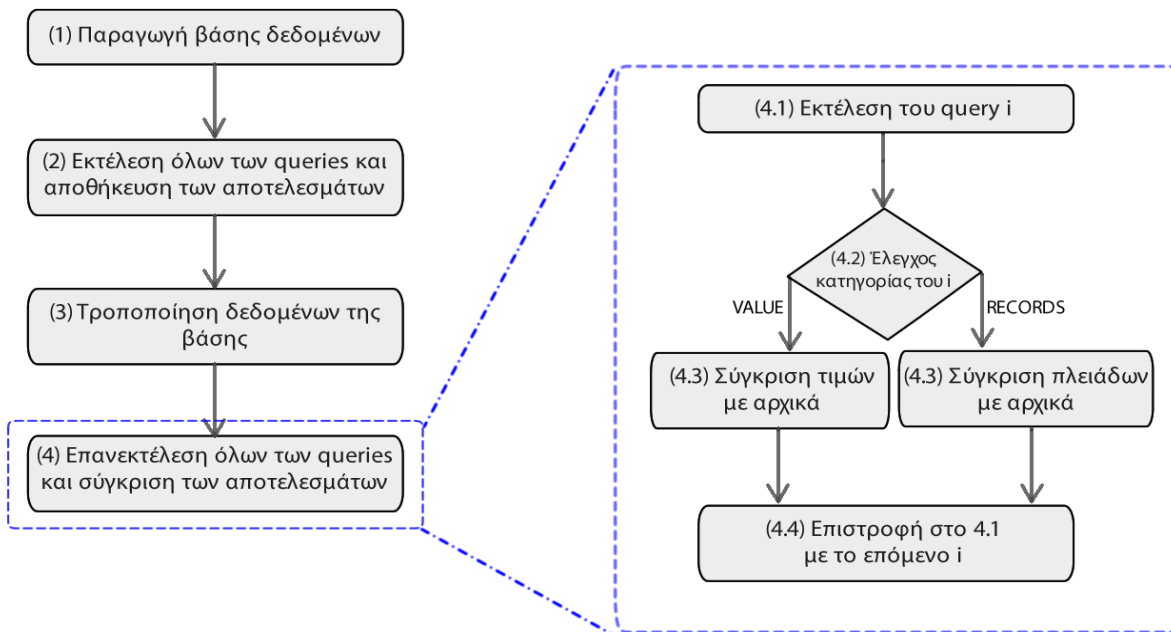
τιμή μειώνετε. Δεν υπάρχει κάποια τιμή η οποία να είναι η ελάχιστη, αλλά εμείς την φράξαμε στο 0%, έτσι σε περιπτώσεις όπου αυτή η μετρική πρόκυπτε αρνητική, η τιμή της απλά μηδενιζόταν.

Το επόμενο χαρακτηριστικό το οποίο μετρήθηκε ήταν η κατανάλωση σε ενέργεια για το οποίο χρησιμοποιήθηκαν μετρητές που παρέχονται από τη Performance Application Programming Interface, και πιο συγκεκριμένα μετρητές του κύκλου μηχανής (clock cycle). Η βιβλιοθήκη PAPI είναι ικανή να παρέχει πρόσβαση σε μετρητές επίδοσης οι οποίοι είναι διαθέσιμοι στους περισσότερους σύγχρονους επεξεργαστές. Η παρακολούθηση αυτών των μετρητών έχει μία ποικιλία χρήσεων στην ανάλυση επιδόσεων, όπως σε περιπτώσεις βελτιστοποίησης από μεταγλωττιστές, εντοπισμός σφαλμάτων, συγκριτικής αξιολόγησης, παρακολούθηση και αξιολόγηση των επιδόσεων ενός συστήματος.

4.2 Μεθοδολογία

Έχοντας αποφασίσει τις μεθόδους χαλαρότητας που θα χρησιμοποιηθούν και το σημείο εφαρμογής τους, απομένει να επεξηγηθεί η διαδικασία που χρειάστηκε να ακολουθηθεί για να παρθούν τα αναγκαία αποτελέσματα.

Μέσο των εργαλείων που προσφέρει η TPC-H, δίνεται η δυνατότητα χρήσης μίας γεννήτριας, η οποία παράγει δεδομένα (1) τα οποία χρησιμοποιούν μία συγκεκριμένη ποσότητα αποθηκευτικού χώρου ανάλογα με τον συντελεστή κλιμάκωσης που θα της δοθεί. Στις εκτελέσεις μας χρησιμοποιήθηκε συντελεστής ίσος με ένα (παραγωγή δεδομένων 1GB). Έτσι μετά την παραγωγή τους και την εισαγωγή τους στο σύστημα DBMS, μπορούν να εκτελεστούν τα ερωτήματα (2) για να έχουμε στην διάθεση μας τα αρχικά και αυθεντικά αποτελέσματα.



Σχήμα 2: Σχεδιάγραμμα των βημάτων για την παραγωγή των αποτελεσμάτων

Ακολούθως χρειάζεται να τροποποιηθούν τα παραγόμενα δεδομένα (3) της βάσης με τέτοιο τρόπο ώστε να προσομοιωθούν οι μέθοδοι χαλαρότητας και να ξαναεισαχθούν τα αλλοιωμένα δεδομένα στο σύστημα. Σε αυτό το σημείο μπορούμε να αποφασίσουμε το ποσοστό αλλοίωσης των δεδομένων, με διαφορετικά λόγια, το ποσό της χαλαρότητας που επιθυμούμε να εφαρμοστεί.

Στη συνέχεια είμαστε έτοιμοι να εκτελέσουμε τα επρωτήματα ακόμα μία φορά (4) αλλά σε αυτή την περίπτωση τα αποτελέσματα που θα πάρουμε θα πάσχουν από σφάλματα. Κρίθηκε αναγκαίο να διαχωριστούν τα queries σε δύο κατηγορίες (4.2) λόγω του αποτελέσματος που παράγουν. Όταν η κατηγορία είναι του τύπου value ο τρόπος σύγκρισης είναι η εύρεση της απόστασης του νέου αποτελέσματος από το αυθεντικό αποτέλεσμα (| Νέα_Τιμή – Αυθεντική_Τιμή |). Το οποίο είναι ένα υπομέρος από την μετρική QoR έτσι ώστε να είναι συμβατό με αυτή την περίπτωση. Στην άλλη κατηγορία αποτελέσματος, η σύγκριση γίνεται με την εφαρμογή της μετρικής Quality of Result (QoR).

Για την παραγωγή αποτελεσμάτων, όσο το δυνατόν πιο αντικειμενικών, τα βήματα 3 και 4 επαναλαμβάνονται πολλαπλές φορές με την ίδια μέθοδο χαλαρότητας και με ίδιο ποσό χαλαρότητας. Έτσι ώστε να παρθεί ο μέσος όρος στις μετρήσεις μας. Μέχρι εδώ έχει εξηγηθεί ο τρόπος καταμέτρησης της αλλαγής στα αποτελέσματα, αλλά όχι της αλλαγής στην κατανάλωση ενέργειας.

Η εξοικονόμηση ενέργειας ήταν εφικτό να καταμετρηθεί μόνο στην μέθοδο που αφορά την μείωση των προσβάσεων στην μνήμη (αφαίρεση στοιχείων από την εκτέλεση), γιατί ο τρόπος προσομοίωσης, μας το επέτρεπε. Δηλαδή η πραγματική μηχανή μας επωφελούταν κατά την εφαρμογή αυτής της μεθόδους. Όσο αφορά την περίπτωση μείωσης του ρυθμού ανανέωσης, η μηχανή μας δεν επωφελούταν γιατί δεν μειώθηκε πραγματικά ο ρυθμός ανανέωσης αλλά προσομοιώθηκαν τα λάθη που θα υπήρχαν σε περίπτωση μείωσης του.

Σε αυτό το σημείο μέτρησης, για να παραχθεί κάπως πιο αντικειμενικό αποτέλεσμα έγινε προσπάθεια καταμέτρησης μόνο του χρόνου εκτέλεσης του επρωτήματος. Δηλαδή, ανοιγόταν η σύνδεση με τον DBMS και μετά ξεκινούσαν οι μετρήσεις. Όταν το αποτέλεσμα ήταν έτοιμο σταματούσαν οι μετρητές. Άρα δεν μετριόταν ούτε η εμφάνιση του αποτελέσματος ούτε ο χρόνος ανοίγματος και κλεισίματος της σύνδεσης.

Αξίζει να αναφερθεί ότι το σημείο 3 είναι εντελώς ανεξάρτητο από το υπόλοιπα μέρη. Δηλαδή στην προκειμένη περίπτωση είναι ένα εκτελέσιμο το οποίο παίρνει ως είσοδο το ποσό χαλαρότητας και έναν πίνακα από την βάση και δίνει ως έξοδο τον αλλοιωμένο πίνακα. Επιπλέον για την σύγκριση των αποτελεσμάτων το μόνο που πρέπει να προκαθοριστεί πριν την εκτέλεση είναι η κατηγορία του κάθε επρωτήματος. Έτσι, πρώτον μπορούμε εύκολα να αλλάζουμε μεθόδους χαλαρότητας και δεύτερον παρατηρούμε ότι η δουλειά αυτή ουσιαστικά έχει δημιουργήσει ένα εργαλείο αξιολόγησης των queries όταν εκτελούνται σε μηχανές που ακολουθούν το μοντέλο του προσεγγιστικού υπολογισμού.

Η εκτέλεση των βημάτων και λειτουργιών αυτοματοποιούνται κυρίως με την χρήση εντολών Bash έτσι ώστε να μπορούν να επαναχρησιμοποιηθούν από οποιαδήποτε πλατφόρμα που είναι POSIX-συμβατή. Με εξαίρεση το σημείο τροποποίησης των δεδομένων και την μέτρησης των clock cycles τα οποία υλοποιήθηκαν σε C/C++ έτσι ώστε να έχουμε διαθέσιμο ένα αντίστοιχο εκτελέσιμο.

4.3 Προσομοίωση των Μεθόδων Χαλαρότητας

Η μείωση των προσβάσεων/επαναλήψεων, στην περίπτωση μας, μπορεί να επιτευχθεί με την μείωση των στοιχείων που υπάρχουν στη βάση. Έτσι πριν από κάθε εκτέλεση των επερωτημάτων αφαιρούνται μερικές εγγραφές από τα δεδομένα της βάσης. Όσο αφορά το ποιες ήταν αυτές οι εγγραφές, δοκιμάστηκαν διάφορα σενάρια τα οποία συμπεριλαμβάνουν την διαγραφή τυχαίων στοιχείων, αρχικών στοιχείων, των τελευταίων και με σταθερό ρυθμό. Ο σκοπός αυτών των διαφορετικών τρόπων επιλογής εγγραφών είναι να διαπιστωθεί κατά πόσο το αποτέλεσμα επηρεάζεται διαφορετικά σε παράλειψη διαφορετικών πλειάδων αλλά ίσων σε πλήθος. Σε αυτή την μέθοδο επιτυγχάνεται μία αντιπροσωπευτική εκτέλεση. Δηλαδή τα παραγόμενα δεδομένα είναι ικανά να μας υποδείξουν την μείωση της εγκυρότητας του αποτελέσματος και την μείωση της κατανάλωσης σε ενέργεια. Για την μέτρηση της εξοικονόμησης σε ενέργεια γίνεται μέτρηση του χρόνου εκτέλεσης σε κύκλους μηχανής με την βοήθεια της βιβλιοθήκης RAPI. Βασισμένοι στο ότι ένα συγκεκριμένο επερώτημα ακολουθά πάντα τα ίδια βήματα κατά την εκτέλεση του, θεωρήθηκε ότι μία μείωση στον χρόνο εκτέλεσης κατά xx% οδηγεί σε μία μείωση στην ενέργεια που καταναλώθηκε επίσης κατά το ίδιο ποσοστό xx%.

Για την μείωση του ρυθμού ανανέωσης αντιστοιχήθηκε κάθε μείωση με κάποια πιθανότητα αλλοίωσης των δεδομένων. Για να προσομοιωθεί αυτή η περίπτωση, πριν από την εκτέλεση των επερωτημάτων μορφοποιούνται τα δεδομένα της βάσης. Πιο συγκεκριμένα γίνεται αντιστροφή όλων των bits ενός byte σε τυχαία bytes, αλλά φροντίζεται να είναι τόσα bytes όσα αντιστοιχούν στο γινόμενο μεταξύ του συνολικού μεγέθους των δεδομένων επί την πιθανότητα. Με αυτόν τον τρόπο επιτυγχάνεται να συλλεχτούν δεδομένα που θα είναι ικανά να μας υποδείξουν την μείωση της εγκυρότητας του αποτελέσματος αλλά όχι την μείωση της κατανάλωσης σε ενέργεια. Γι' αυτό, για το σημείο της εξοικονόμησης, χρησιμοποιήθηκαν αποτελέσματα από άλλες έρευνες [4] [5].

4.4 Σύστημα Αξιολόγησης

Το λειτουργικό στο οποίο έγιναν οι εκτελέσεις βασίζεται στο Linux kernel 3.16 και ως DBMS χρησιμοποιήθηκε το PostgreSQL 9.3.10 [17]. Το σύστημα αποτελείται από 8gb κύρια μνήμη DRAM, επεξεργαστή τεσσάρων πυρήνων και με χωρητικότητα δευτερεύουσας μνήμης 30gb τύπου SSD.

Για το πειραματικό στάδιο επιλέχθηκε ως τύπος της μνήμης το DRAM DDR3 [15], γιατί είναι το πιο διαδεδομένο την παρούσα στιγμή. Λήφθηκε ως δεδομένο ότι οι θερμοκρασίες στην μνήμη δεν φτάνουν σε ακραία επίπεδα (παραμένουν χαμηλότερα των 85°C) [4] και ότι ο προκαθορισμένος ρυθμός ανανέωσης (default refresh rate) είναι 64ms, αφού αυτό είναι που χρησιμοποιείται στις πλείστες μνήμες.

Επίσης θέσαμε ως περιορισμό ότι μόνο τα δεδομένα που φορτώνονται από τη βάση θα μεταφέρονται στην χαλαρωμένη μνήμη, για να μην υποφέρουν από αλλοίωση άλλα δεδομένα.

Κεφάλαιο 5

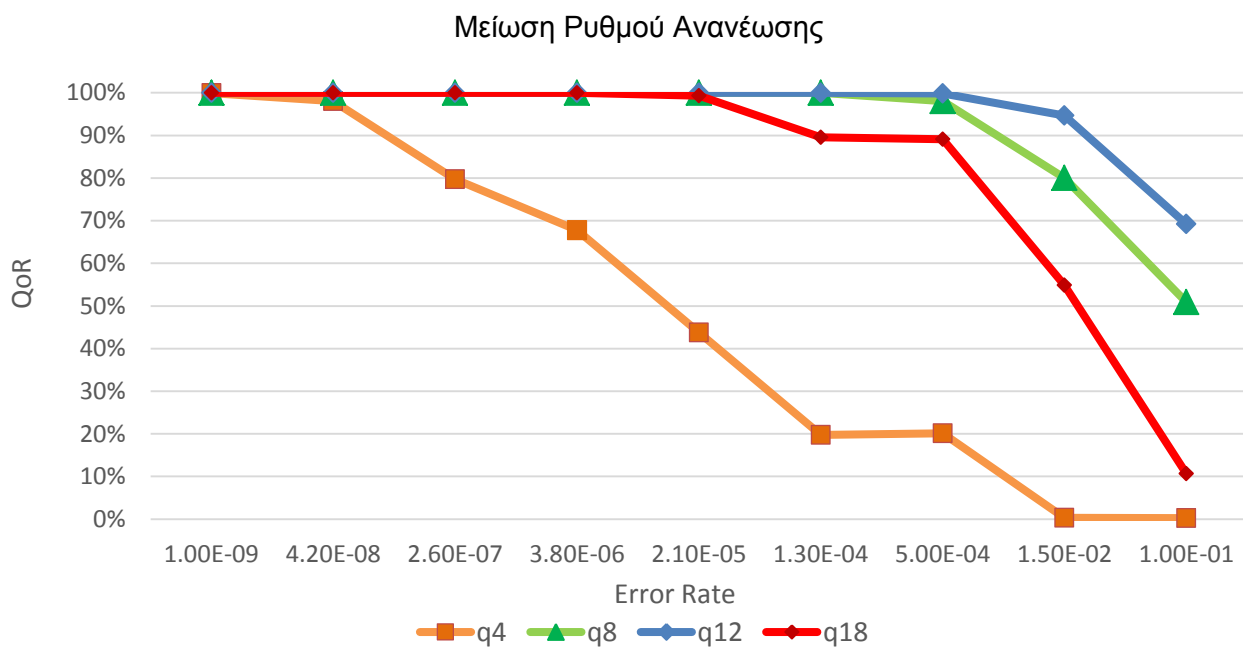
Αποτελέσματα

5.1	Μείωση Ρυθμού Ανανέωσης	13
5.2	Μείωση των Προσβάσεων	18
5.3	Σύγκριση των Μεθόδων Χαλαρότητας	22
5.4	Περεταίρω για τη Μείωση των Προσβάσεων	29

5.1 Μείωση Ρυθμού Ανανέωσης

5.1.1 Ποιότητα Αποτελέσματος

Πιο κάτω φαίνονται τα αποτελέσματα της μετρικής QoR στην περίπτωση της μείωσης του ρυθμού ανανέωσης, για μερικά ερωτήματα τα οποία κρίθηκαν ικανά να μας δείξουν την διαφορετικότητα που υπάρχει μεταξύ όλων των ερωτημάτων .



Σχήμα 3: QoR, κατά την εφαρμογή της μείωσης του ρυθμού ανανέωσης με διαφορετικά ποσοστά σφάλματος

Στο παράρτημα Α υπάρχουν γραφικές οι οποίες περιλαμβάνουν τις μετρήσεις για όλα τα επερωτήματα. Ο σχολιασμός βασίζεται κυρίως στις μετρήσεις για όλα τα επερωτήματα.

Οι μετρήσεις που παρουσιάζονται ξεκινούν με ποσοστό σφάλματος $1e-09$ γιατί όλα τα επερωτήματα εκτός από το q1 σε αυτό το ποσοστό, παράγουν αποτελέσματα με πολύ ψηλό QoR. Επίσης τα ποσοστά δεν ξεπερνούν το $1.0e-1$ γιατί τα περισσότερα επερωτήματα έχουν πολύ αλλοιωμένο αποτέλεσμα.

Παρατηρούμε ότι τα αποτελέσματα διαφέρουν αρκετά μεταξύ τους. Υπάρχουν περιπτώσεις όπου η πτώση είναι απότομη αλλά και περιπτώσεις όπου διατηρείται σχετικά ψηλή η τιμή της μετρικής. Ο λόγος αυτής της διαφοροποίησης είναι ότι το κάθε επερωτήμα χρειάζεται διαφορετικά δεδομένα για την παραγωγή του αποτελέσματος του, διαφορετικά και στο ποια είναι αυτά τα δεδομένα και στο πόσα είναι.

Για να είναι πιο εύκολος ο σχολιασμός και η σύγκριση των αποτελεσμάτων, τα επερωτήματα ομαδοποιήθηκαν. Λαμβάνοντας υπόψη το πόσες φορές η τιμή του QoR ήταν κάτω από το 50%, τα επερωτήματα χωρίστηκαν στις ομάδες Very Good, Good, Bad και Very Bad. Ιδιαίτερη προσοχή αξίζουν τα q12 και q14 όπου σε καμία περίπτωση το αποτέλεσμα τους δεν διέφερε πάνω από το 50% από το αυθεντικό αποτέλεσμα

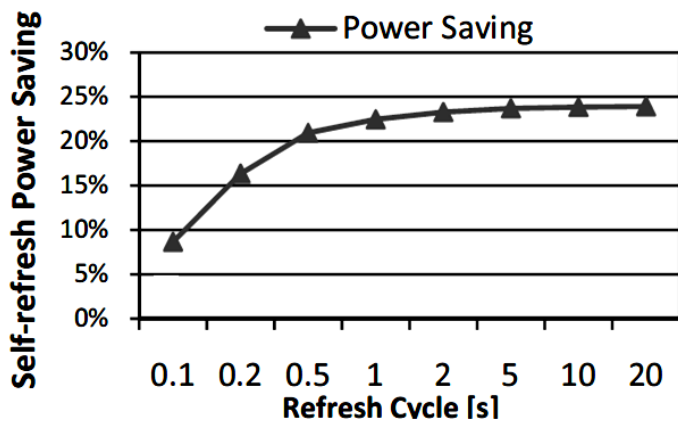
Very Good	q2, q3, q8, q12, q13, q14, q19	(0-1 φορές με QoR <50%)
Good	q5, q6, q16, q18,	(2 φορές με QoR <50%)
Bad	q9, q15, q22	(3 φορές με QoR <50%)
Very Bad	q1, q4, q9, q10, q11, q16, q18	(>3 φορές με QoR <50%)

Ο διαχωρισμός έγινε με αυτό τον τρόπο γιατί η προσέγγιση ενός αποτελέσματος κατά το ήμισυ θεωρείτε ως μία πολύ άσχημη επίπτωση, έτσι η παραγωγή τέτοιων αποτελεσμάτων αρκετές φορές χαρακτηρίζει την ανοχή του επερωτήματος ως πολύ άσχημη. Καθώς μειώνεται το πλήθος αυτών των αποτελεσμάτων καλύτερεύει και η ομάδα στην οποία εμπίπτει το επερωτήμα.

Με εξαίρεση πολύ λίγα επερωτήματα, τα υπόλοιπα στις τρεις πρώτες περιπτώσεις έχουν σχεδόν την μέγιστη τιμή της QoR (πολύ κοντά στο 100%). Έτσι η μείωση του ρυθμού ανανέωσης φαίνεται να είναι καλή επιλογή για περιπτώσεις όπου το ποσό χαλαρότητας θέλουμε να είναι σχετικά χαμηλό.

5.1.2 Κατανάλωση

Το θέμα της εξοικονόμησης ενέργειας η οποία επιτυγχάνεται με την χρήση της μείωσης του ρυθμού ανανέωσης, έχει ερευνηθεί ήδη στο παρελθόν. Για αυτό είναι εφικτό να χρησιμοποιηθούν στατιστικά και μετρήσεις από σχετικές μελέτες, κυρίως του Liu [5]. Η εξοικονόμηση που φαίνεται στο σχήμα 4 και πίνακα 4 αναλογεί στο ποσοστό κατανάλωσης της μνήμης και όχι την συνολική κατανάλωση του υπολογιστή.



Σχήμα 4: Ποσοστό εξοικονόμηση ενέργειας που εντιστοιχεί στην μνήμη κατά την αλλαγή του χρονικού διαστήματος ανάμεσα στις ανανεώσεις [5]

Refresh Cycle (s)	Memory Power Saving
0.1	9 %
0.2	15.5 %
0.5	20.5 %
0.7	21.0 %
1	22.0 %
2	23.0 %
5	23.5 %
10	24.0 %
20	24.5 %
45	25.0 %

Πίνακας 4: Χρονικό διάστημα ανάμεσα στις ανανεώσεις και εξοικονόμηση της μνήμης

Η κύρια μνήμη είναι ένας σημαντικός παράγοντας στην συνολική κατανάλωση ενέργειας για μίαν υπολογιστική μηχανή. Σε περιπτώσεις εκτεταμένης χρήσης της, όπως και στην περίπτωση μας, μπορεί να της αναλογεί μέχρι και 30% της συνολικής κατανάλωσης [5]. Άρα τα ποσοστά εξοικονόμησης του πίνακα 4 αναλογούν σε 30% της συνολικής εξοικονόμησης. Συνδυάστηκαν αυτά τα στοιχεία με τα στοιχεία που φαίνονται στο τμήμα 3.2, έχουμε συσχέτιση του κάθε ρυθμού ανανέωσης με ένα ποσοστό σφάλματος και εξοικονόμησης ενέργειας (πίνακας 5).

Για όλα τα επερωτήματα η εξοικονόμηση είναι η ίδια. Αυτό ευθύνεται στο ότι η μείωση στην κατανάλωση προκύπτει από μηχανικά πλεονεκτήματα λόγω της μείωσης του ρυθμού ανανέωσης και έτσι δεν διαφοροποιείται το ποσοστό από επερωτήμα σε επερωτήμα.

Refresh Cycle (sec)	Refresh-Rate (percent)	Error Rate	Overall Power Saving
0.064	100 %	1.0×10^{-12}	0 %
0.2	32 %	1.0×10^{-9}	4.65 %
0.5	12.8 %	4.2×10^{-8}	6.15 %
0.7	9.14 %	2.6×10^{-7}	6.30 %
1	6.4 %	3.8×10^{-6}	6.60 %
2	3.2 %	2.1×10^{-5}	6.90 %
5	1.28 %	1.3×10^{-4}	7.05 %
10	0.64 %	5.0×10^{-4}	7.20 %
20	0.32 %	1.5×10^{-2}	7.35 %
45	0.142 %	1.0×10^{-1}	7.50 %

Πίνακας 5: Ρυθμός ανανέωσης – Ποσοστό λάθους – Συνολική εξοικονόμηση Ενέργειας

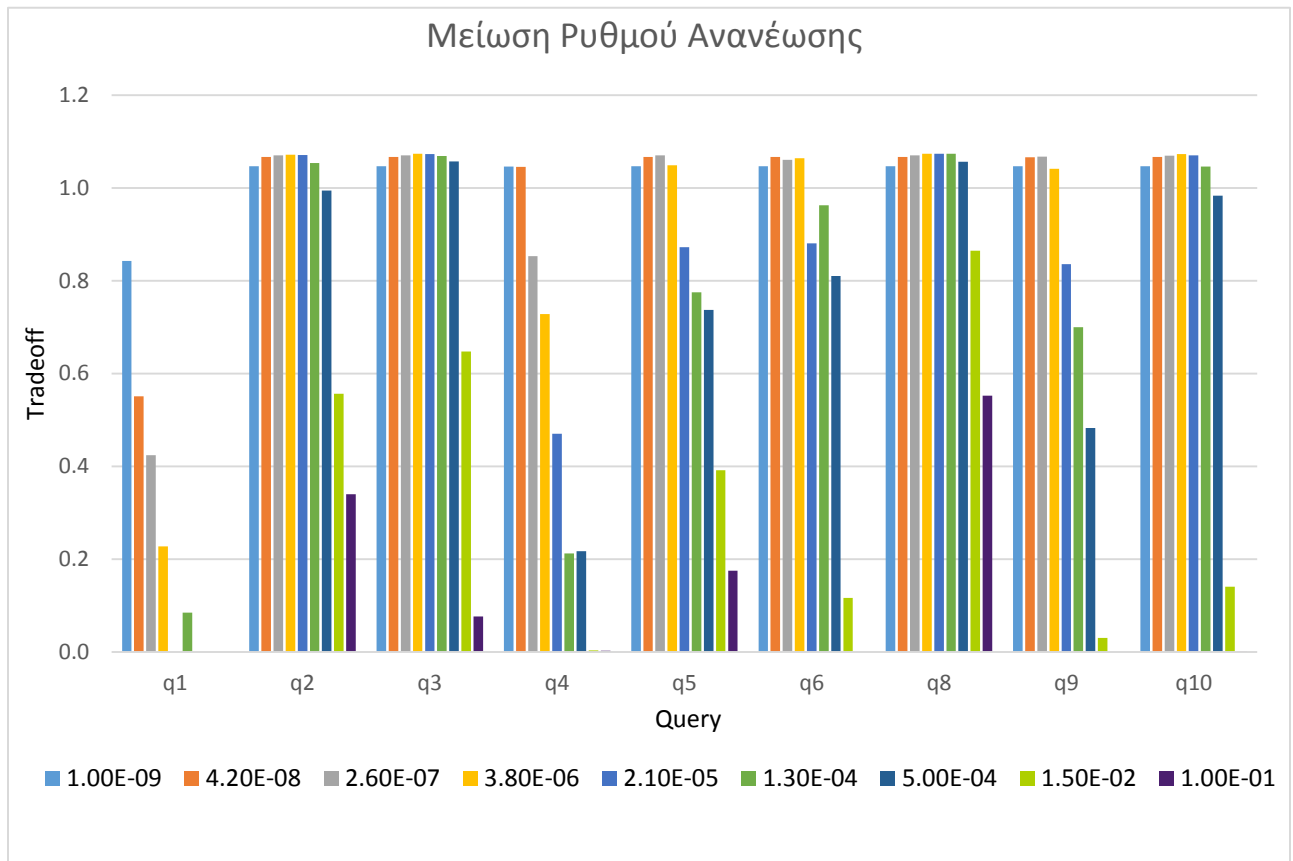
5.1.3 Ωφελιμότητα

Ακολούθως, για να γίνει ευκολότερη η σύγκριση της αλλοίωσης του αποτελέσματος με την εξοικονόμηση στην ενέργεια υπολογίσθηκε μία νέα τιμή ανταλλαγής/ωφελιμότητας (Tradeoff), η οποία προκύπτει από την διαίρεση της QoR με το Power Consumption. Αυτή η πράξη δικαιολογείται γιατί η επιθυμητή επίπτωση της χαλαρότητας πάνω στην εκτέλεση ενός ερωτήματος είναι ανάλογη με τη QoR και αντίστροφος ανάλογη με την κατανάλωση της ενέργειας που χρειάζεται.

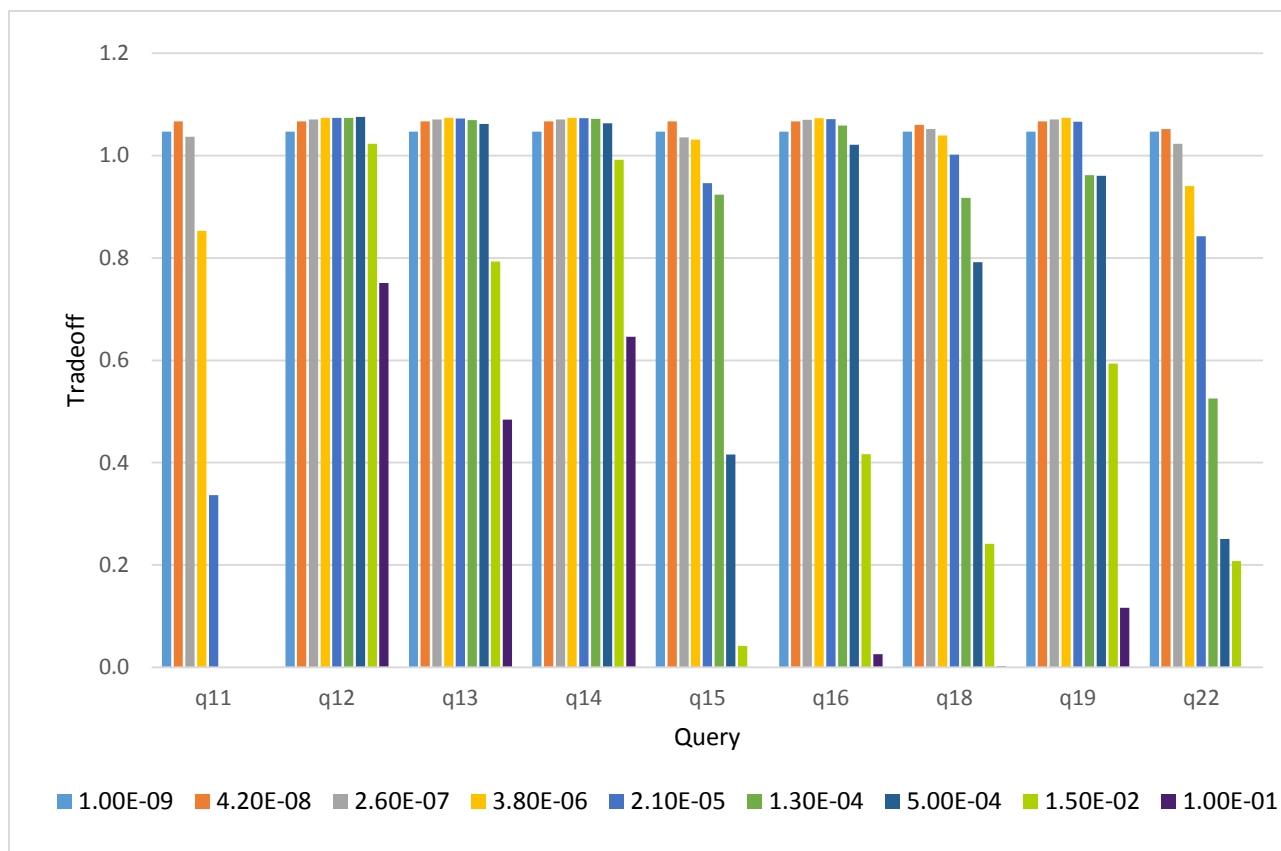
$$\text{Tradeoff} = \frac{QoR}{\text{Power Consumption}}$$

Δηλαδή όταν η τιμή του Tradeoff είναι ίση με τη μονάδα, τότε η απώλεια στο αποτέλεσμα είναι η ίδια με το ποσοστό εξοικονόμησης σε ενέργεια. Ενώ σε περιπτώσεις όπου ξεπερνά την μονάδα τότε η ποιότητα του αποτελέσματος (όπως εκφράζεται από τη μετρική) έχει μεγαλύτερη τιμή από το ποσοστό της κατανάλωσης, και το αντίθετο.

Αφού σε όλα τα ερωτήματα η εξοικονόμηση ήταν ίδια, τότε είναι προφανές ότι αυτή τη μέθοδο είναι επιθυμητό να εφαρμόζεται στα ερωτήματα της κατηγορίας very good.



Σχήμα 5α: Tradeoff για κάθε ερωτήριο, κατά την εφαρμογή της μείωσης του ρυθμού ανανέωσης με διαφορετικά ποσοστά



Σχήμα 5b: Tradeoff για κάθε ερωτήματα, κατά την εφαρμογή της μείωσης του ρυθμού ανανέωσης με διαφορετικά ποσοστά

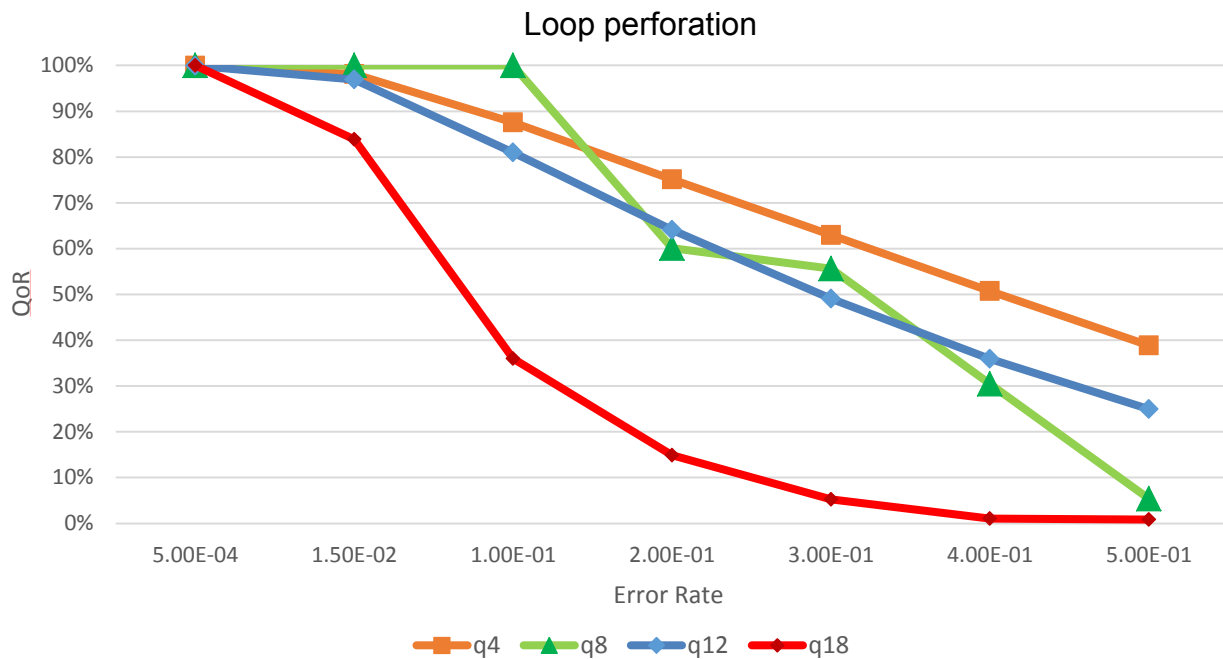
Στις πρώτες περιπτώσεις η τιμή του Tradeoff είναι μεγαλύτερη της μονάδα, το οποίο δείχνει ότι το ποσοστό αλλοίωσης του αποτελέσματος είναι μικρότερο από το ποσοστό εξοικονόμησης σε ενέργεια. Στη συνέχεια υπάρχει απότομη μείωση του Tradeoff, το οποίο είναι αναμενόμενο γιατί η μείωση του ρυθμού ανανέωσης πέρα της 4^{ης} περίπτωσης, δηλαδή όταν το χρονικό διάστημα ανάμεσα στις ανανεώσεις (refresh cycle) φθάσει ή ξεπεράσει τα 2 δευτερόλεπτα, η εξοικονόμηση σε ενέργεια παραμένει σχεδόν σταθερή.

Μέχρι και με ποσοστό σφάλματος ίσο με 3.8×10^{-6} σχεδόν όλα τα ερωτήματα, έχουν τιμή κοντά στο 100% στην QoR και εξοικονόμηση 6.60%. Αυτό υποδεικνύει ότι αυτή η μέθοδος είναι πολύ προσιτή για εφαρμογή σε μικρό ποσό σφάλματος το οποίο επιφέρει σχεδόν αμελητέα αλλοίωση στο αποτέλεσμα και μία μικρή αλλά αισθητή μείωση στην κατανάλωση.

5.2 Μείωση των Προσβάσεων

5.2.1 Ποιότητα Αποτελέσματος

Πιο κάτω φαίνονται τα αποτελέσματα της μετρικής QoR, για τα επιλεγμένα ερωτήματα, στην περίπτωση της μείωσης των προσβάσεων/επαναλήψεων (loop perforation):



Σχήμα 6: QoR κατά την εφαρμογή της μείωσης των επαναλήψεων με διαφορετικά ποσοστά σφάλματος

Στο παράρτημα Β βρίσκονται τα αποτελέσματα για όλα τα ερωτήματα.

Τα ποσοστά μείωσης επαναλήψεων που παρουσιάζονται ξεκινούν από το 5e-4 γιατί όλα τα ερωτήματα εκτός από το q1 σε αυτό το ποσοστό παράγουν αποτελέσματα με πολύ ψηλό QoR. Επίσης τα ποσοστά δεν ξεπερνούν το 5e-1 γιατί τα περισσότερα ερωτήματα έχουν πολύ αλλοιωμένο αποτέλεσμα.

Και πάλι, τα ερωτήματα χωρίστηκαν στις ομάδες Very Good, Good, Bad και Very Bad. Συγκρίνοντας τις νέες ομάδες με τις παλιές, φαίνεται ότι υπάρχουν αρκετές αλλαγές. Άρα το πόσο θα αλλοιωθεί το αποτέλεσμα δεν έχει να κάνει μόνο με το σημείο εφαρμογής του προσεγγιστικού υπολογισμού αλλά και με την μέθοδο χαλαρότητας που θα επιλεγεί για αυτό το συγκεκριμένο σημείο εφαρμογής. Ιδιαίτερη προσοχή αξίζουν τα q6, q13 και q14 όπου σε καμία περίπτωση το αποτέλεσμα τους δεν διέφερε πάνω από το 50% από το αυθεντικό αποτέλεσμα.

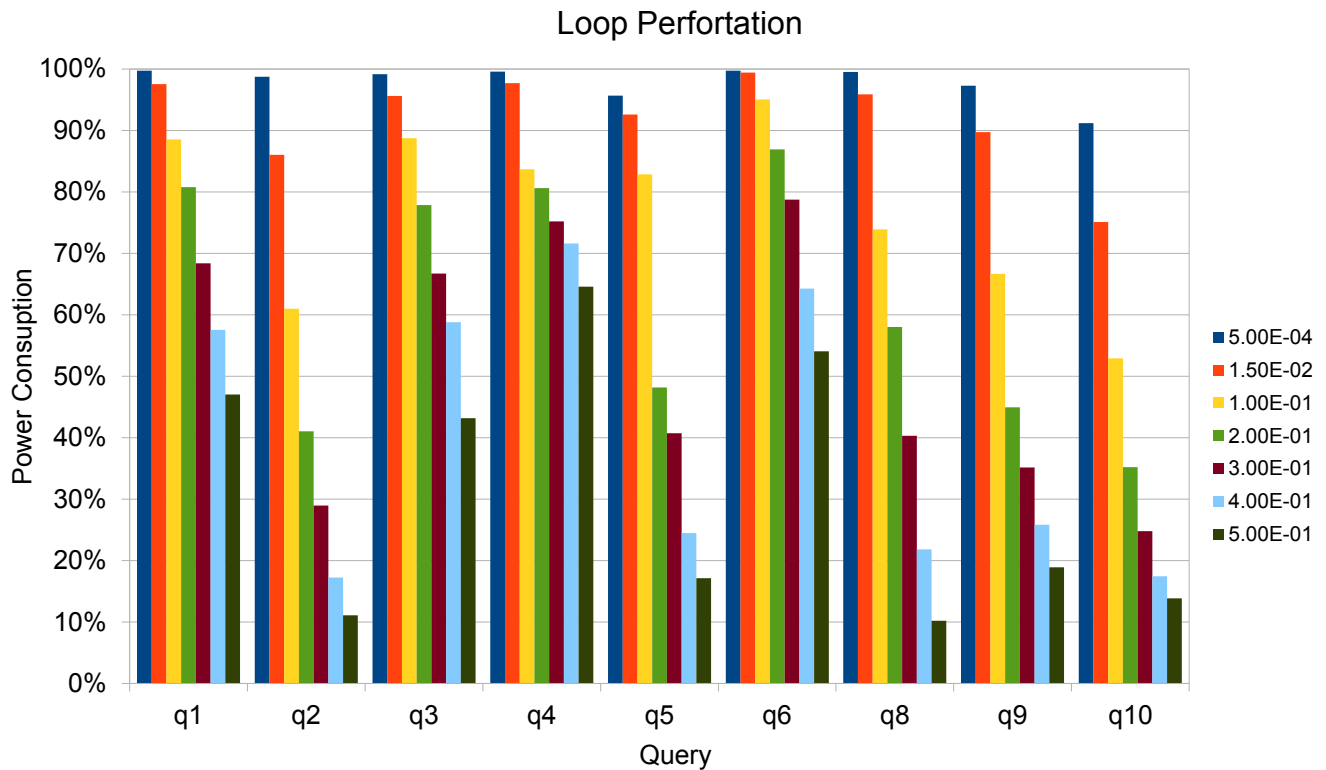
Very Good	q4, q6, q8, q13, q14, q15, q22	(0-2 φορές με QoR <50%)
Good	q12, q19	(3 φορές με QoR <50%)
Bad	q2, q3, q11	(4 φορές με QoR <50%)

Very Bad q1, q5, q9, q10, q16, q18 (>4 φορές με QoR <50%)

Όλα τα ερωτήματα που βρίσκονται στην ομάδα Very Good είναι της κατηγορίας value. Έτσι διαπιστώνεται ότι αυτή η κατηγορία τείνει να είναι πιο ανεκτική στο ποσό χαλαρότητας που εφαρμόζεται με την μέθοδο μείωσης των προσβάσεων.

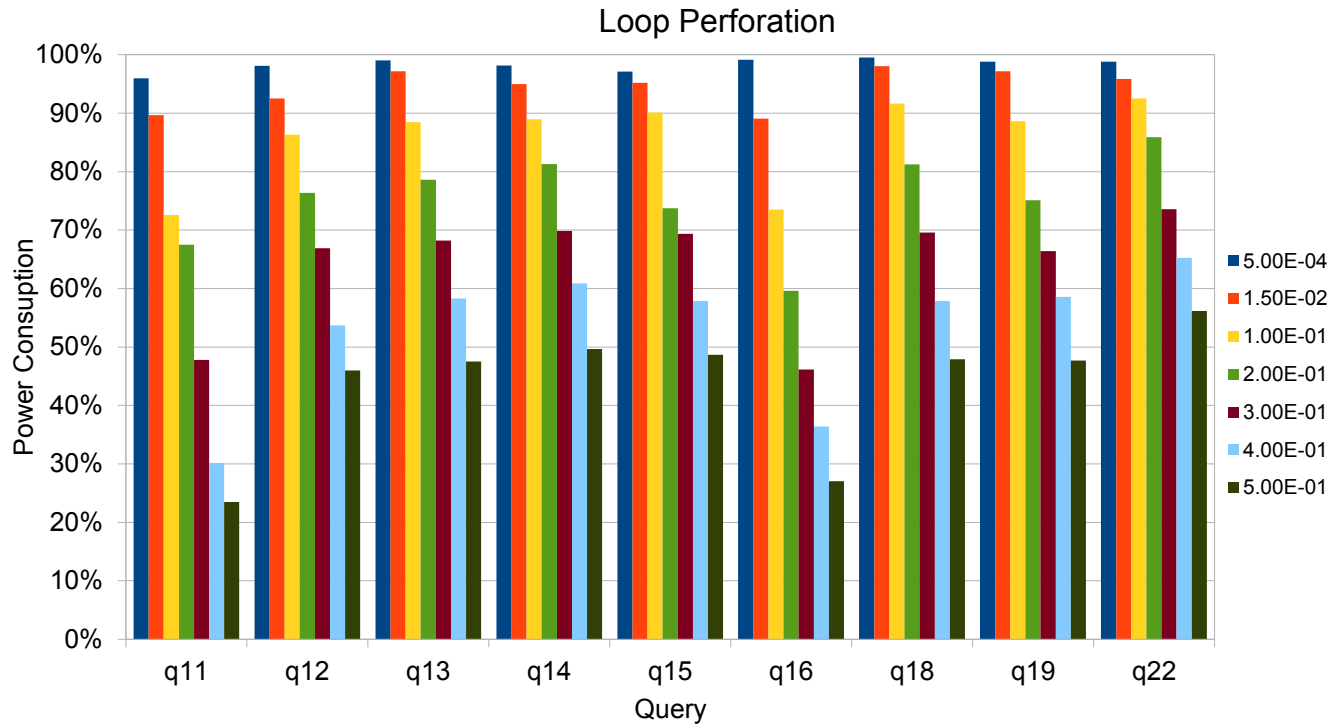
5.2.2 Κατανάλωση

Πιο κάτω φαίνονται οι αλλαγές σε κατανάλωση της ενέργειας η οποία επιτυγχάνεται με την χρήση της μείωσης των προσβάσεων:



Σχήμα 7α: Κατανάλωση για κάθε ερωτήματα, κατά την εφαρμογή της μείωσης των επαναλήψεων με διαφορετικά ποσοστά σφάλματος

Παρατηρούμε ότι ο ρυθμός μείωσης της ενέργειας που χρειάζεται κάποιο ερωτήματα για να εκτελεστεί, δεν συνάδει με το πόσο αλλοιωμένο αποτέλεσμα παράγεται. Δηλαδή υπάρχουν περιπτώσεις όπου το αποτέλεσμα δεν αλλοιώνεται πολύ (ψηλό QoR) και η κατανάλωση μειώνεται αρκετά, π.χ. q8.



Σχήμα 7b: Κατανάλωση για κάθε ερωτήματα, κατά την εφαρμογή της μείωσης των επαναλήψεων με διαφορετικά ποσοστά σφάλματος

Με τον ίδιο τρόπο όπως και προηγουμένως, διαχωρίστηκαν και πάλι τα ερωτήματα σε ομάδες, όμως αυτή τη φορά το επιθυμητό είναι να υπάρχουν πολλές φορές χαμηλές τιμές και όχι λίγες φορές.

Very Good	q2, q5, q9, q10	(4 φορές με κατανάλωση <50%)
Good	q8, q11, q16	(3 φορές με κατανάλωση <50%)
Bad	q1, q3, q12, q13, q14, q15, q18, q19	(1 φορά με κατανάλωση <50%)
Very Bad	q4, q6, q22	(0 φορές με κατανάλωση <50%)

5.2.3 Ωφελιμότητα

Ακολουθούν, τα αποτελέσματα για την τιμή του Tradeoff, η οποία είναι ίση με το πηλίκο της QoR με το Power Consumption. Σε αυτή τη μέθοδο, δόθηκε περισσότερη έμφαση σε αυτό το πηλίκο γιατί το κάθε ερωτήματα έχει διαφορετική αλλοίωση στο αποτέλεσμα και διαφορετική εξοικονόμηση από τα υπόλοιπα. Έτσι το Tradeoff μας δείχνει τις αναλογίες ανάμεσα σε αυτές τις τιμές.

Παρουσιάζονται στο σχήμα 8 συνοπτικά αποτελέσματα που δείχνουν μόνο την τιμή του Tradeoff των ερωτημάτων, ενώ στο Παράρτημα Γ βρίσκονται πιο αναλυτικές γραφικές όπου φαίνεται για κάθε ερωτήματα ξεχωριστά τα 3 χαρακτηριστικά (QoR, PowerConsumption και Tradeoff) και το πώς αλλάζουν σε κάθε ποσοστό χαλαρότητας.



Σχήμα 8: Tradeoff για κάθε επερώτηση, κατά την εφαρμογή της μείωσης των επαναλήψεων με διαφορετικά ποσοστά σφάλματος

Υπάρχουν και πάλι περιπτώσεις στις οποίες η τιμή του Tradeoff είναι μεγαλύτερη του 100% αλλά σε αρκετές από αυτές τις περιπτώσεις η τιμή της QoR είναι χαμηλή και απλά είναι ακόμα χαμηλότερο το ποσοστό χρήσης σε ενέργεια. Σε αυτές τις εκτελέσεις δεν είναι επιθυμητή η χρήση αυτής της μεθόδου χαλαρότητας γιατί παρόλο της εξοικονόμησης, το αποτέλεσμα δεν θεωρείται χρήσιμο. Έτσι σε αυτό το σημείο, τα επερωτήματα πρέπει να κριθούν κατά πόσο αξίζει ή δεν αξίζει να εκτελεστούν με την εφαρμογή της μείωσης των επαναλήψεων. Μόνο σε περιπτώσεις όπου η QoR έχει τιμή πολύ κοντά στο 80% ή μεγαλύτερη θεωρείται το αποτέλεσμα χρήσιμο (acceptable) και για αυτό ο σχολιασμός εστιάζει σε αυτές τις περιπτώσεις:

q14	(5 φορές με acceptable αποτέλεσμα)
q13	(4 φορές με acceptable αποτέλεσμα)
q4, q8, q12, q15	(3 φορές με acceptable αποτέλεσμα)

Το επερωτήμα q14 έχει πολύ καλά αποτελέσματα γιατί παράγει χρήσιμο αποτέλεσμα σε πολλές εφαρμογές της μείωσης των επαναλήψεων και ταυτόχρονα έχει περισσότερη εξοικονόμηση σε ενέργεια, σε σύγκριση με την απώλεια του. Ποιο συγκεκριμένα η τιμή του Tradeoff είναι συνεχώς >90% και στις περισσότερες περιπτώσεις ξεπερνά και το 100%. Στην εφαρμογή της μεθόδου χαλαρότητας με ποσοστό παράληψης στοιχείων 30% έχει περίπου 115% Tradeoff.

Όσο αφορά το q13, επίσης τα αποτελέσματα του δείχνουν ότι παράγει χρήσιμο αποτέλεσμα αρκετές φορές αλλά το Tradeoff είναι κοντά στο 100% και όταν το ξεπερνά τότε δεν είναι κατά πολύ.

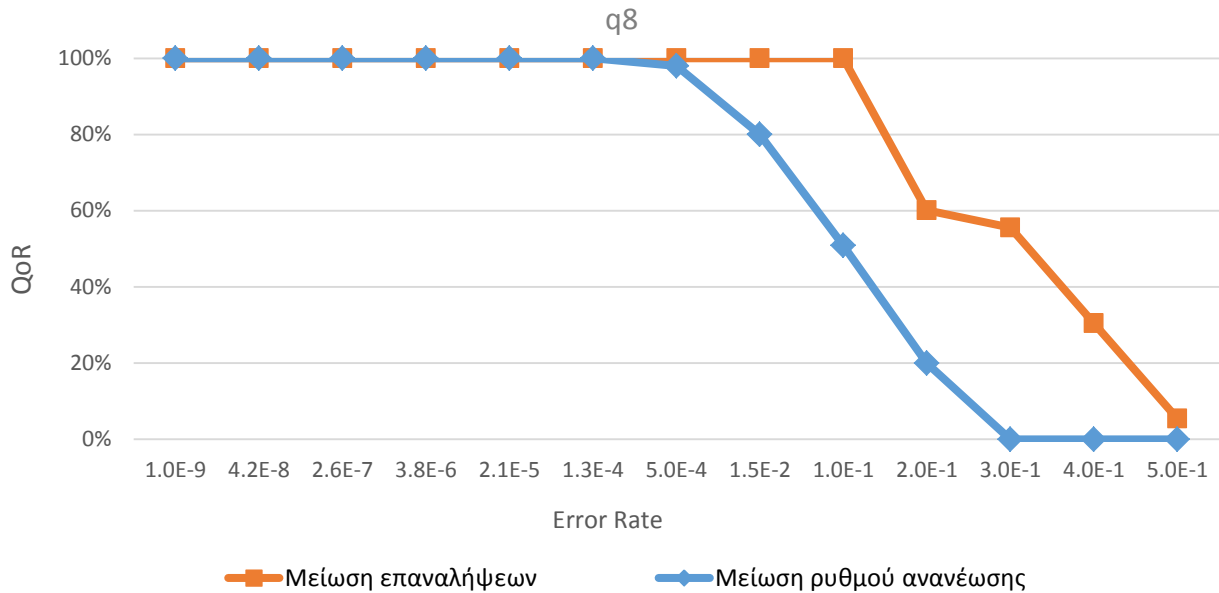
Τα επερωτήματα q4, q8, q12 και q15 έχουν λιγότερες φορές χρήσιμο αποτέλεσμα αλλά σχεδόν πάντα η τιμή του Tradeoff είναι μεγαλύτερη του 100%. Επίσης το q8 είναι το μόνο επερωτήμα στο οποίο το Tradeoff ξεπερνά το 120% και τυγχάνει να πλησιάζει κατά πολύ το 140%. Υπάρχει σημείο στο οποίο η QoR βρίσκεται πολύ κοντά στην μέγιστη τιμή της ενώ το Power Consumption είναι σχεδόν στα τρία τέταρτα.

5.3 Σύγκριση των Μεθόδων Χαλαρότητας

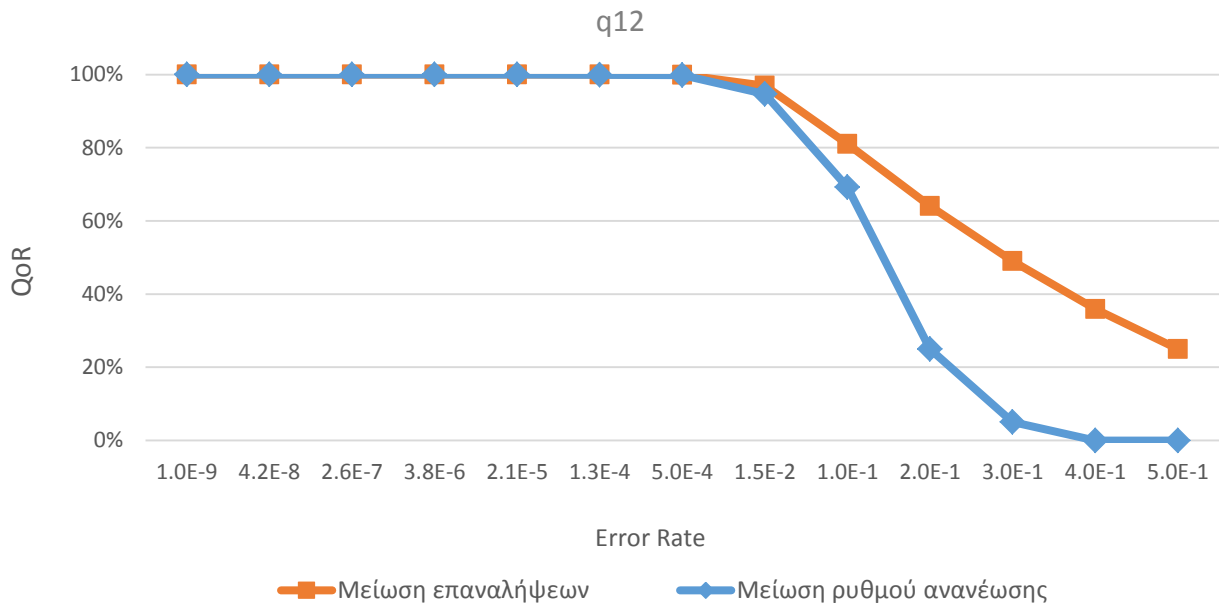
Η σύγκριση των μεθόδων είναι εφικτή αν έχουν κάποια κοινά στοιχεία κατά την εφαρμογή τους. Ένα κοινό στοιχείο που μελετήθηκε, είναι σε περιπτώσεις όπου εφαρμόζονται με το ίδιο ποσοστό και ένα άλλο κοινό σημείο, είναι η εφαρμογή να γίνεται στις ίδιες εγγραφές.

5.3.1 Ίδιο ποσοστό εφαρμογής

Στην περίπτωση της μείωσης των επαναλήψεων το ποσοστό σφάλματος αντιπροσωπεύει το ποσοστό εξαιρεμένων πλειάδων από το συνολικό πλήθος των πλειάδων στα δεδομένα. Ενώ για την περίπτωση μείωσης του ρυθμού ανανέωσης αντιπροσωπεύει το ποσοστό αλλοιωμένων bytes από το συνολικό όγκο δεδομένων.



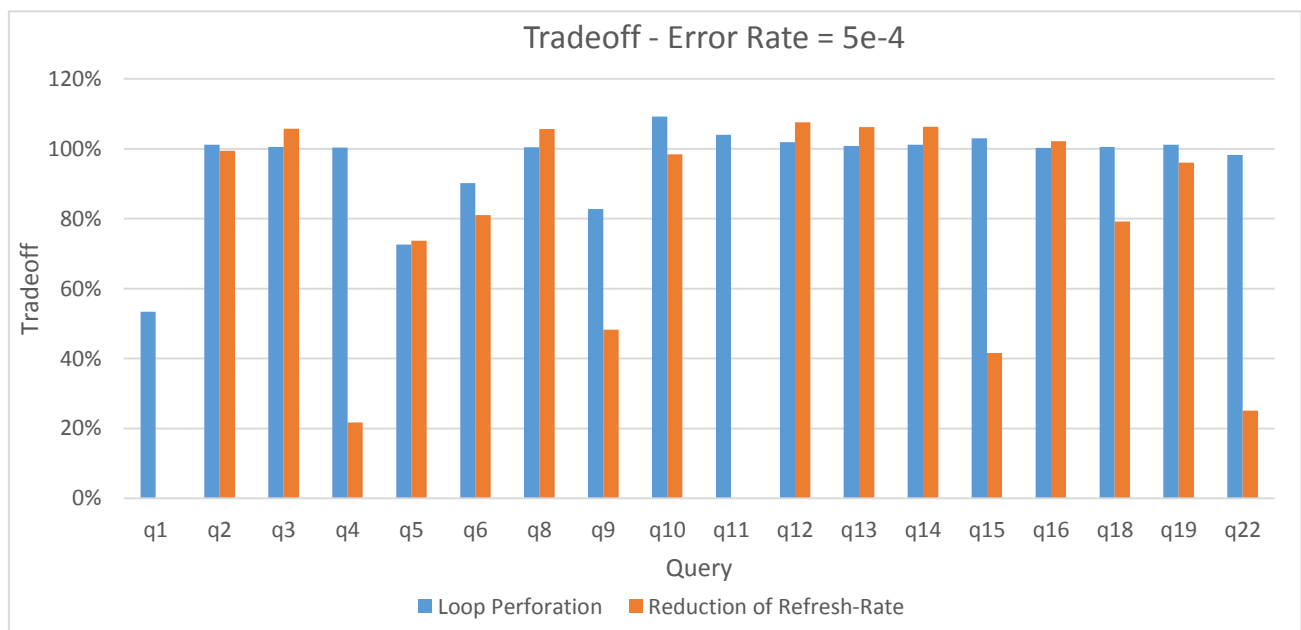
Σχήμα 9: QoR για το q8, κατά την εφαρμογή της μείωσης των επαναλήψεων και μείωσης ρυθμού ανανέωσης



Σχήμα 10: QoR για το q12, κατά την εφαρμογή της μείωσης των επαναλήψεων και μείωσης ρυθμού ανανέωσης

Στα κοινά ποσοστά, η αλλοίωση του αποτελέσματος είναι μεγαλύτερη στην μέθοδο όπου μειώνεται ο ρυθμός ανανέωσης. Αυτό ευθύνεται στο γεγονός ότι τα ερωτήματα χρησιμοποιούν με διαφορετικό τρόπο τα δεδομένα (μερικά απλά μέτρηση εμφάνισης ενώ άλλα νοιάζονται για την τιμή αυτής της εμφάνισης). Επίσης ένας άλλος παράγοντας αυτής της διαφοράς είναι ότι σε ένα κοινό ποσοστό εφαρμογής έχουν αλλοιωθεί περισσότερα bytes απ' όσο έχουν διαγραφεί εγγραφές. Γιατί, αν πάρουμε ένα τυχαίο παράδειγμα, έστω στην περίπτωση όπου το ποσοστό είναι 1.5%, έχοντας 1000 εγγραφές και κάθε εγγραφή χρειάζεται κατά μέσο όρο 50 bytes. Στην αλλοίωση δεδομένων θα έχουν διαμορφωθεί 750 bytes τα οποία το πιο πιθανόν να ανήκουν σε 750 διαφορετικές εγγραφές, ενώ στην παράλειψη στοιχείων θα έχουν διαγραφεί μόλις 15 εγγραφές. Αυτή η διαφορά δείχνει ότι τα αλλοιωμένα bytes βρίσκονται διασκορπισμένα σε πολλές εγγραφές και έτσι είναι ευκολότερο να επηρεάσουν το τελικό αποτέλεσμα.

Αλλά η εξοικονόμηση σε ενέργεια είναι σχεδόν πάντα μικρότερη στην περίπτωση όπου μειώνεται ο ρυθμός ανανέωσης, έτσι με τη χρήση του Tradeoff μπορούμε να διαπιστώσουμε ποιες εκτελέσεις μας δίνουν καλύτερη ανταλλαγή μεταξύ ποιότητας αποτελέσματος και εξοικονόμησης ενέργειας.



Σχήμα 11: Tradeoff για τις δύο μεθόδους χαλαρότητας κάτω από το ίδιο ποσοστό σφάλματος (5e-4)

Συγκρίνοντας τις τιμές του Tradeoff, στις περιπτώσεις με ποσοστό εφαρμογής ίσο με 5E-4, δεν υπάρχει μία μέθοδος που να έχει μεγαλύτερη τιμή για όλα τα ερωτήματα. Πιο συγκεκριμένα, στην μέθοδο με την μείωση του ρυθμού ανανέωσης είναι προτιμότερο να χρησιμοποιείται στα ερωτήματα q3, q8, q12, q13, q14 και q16 ενώ στα υπόλοιπα είναι προτιμότερο να χρησιμοποιηθεί η μέθοδος μείωσης των επαναλήψεων.

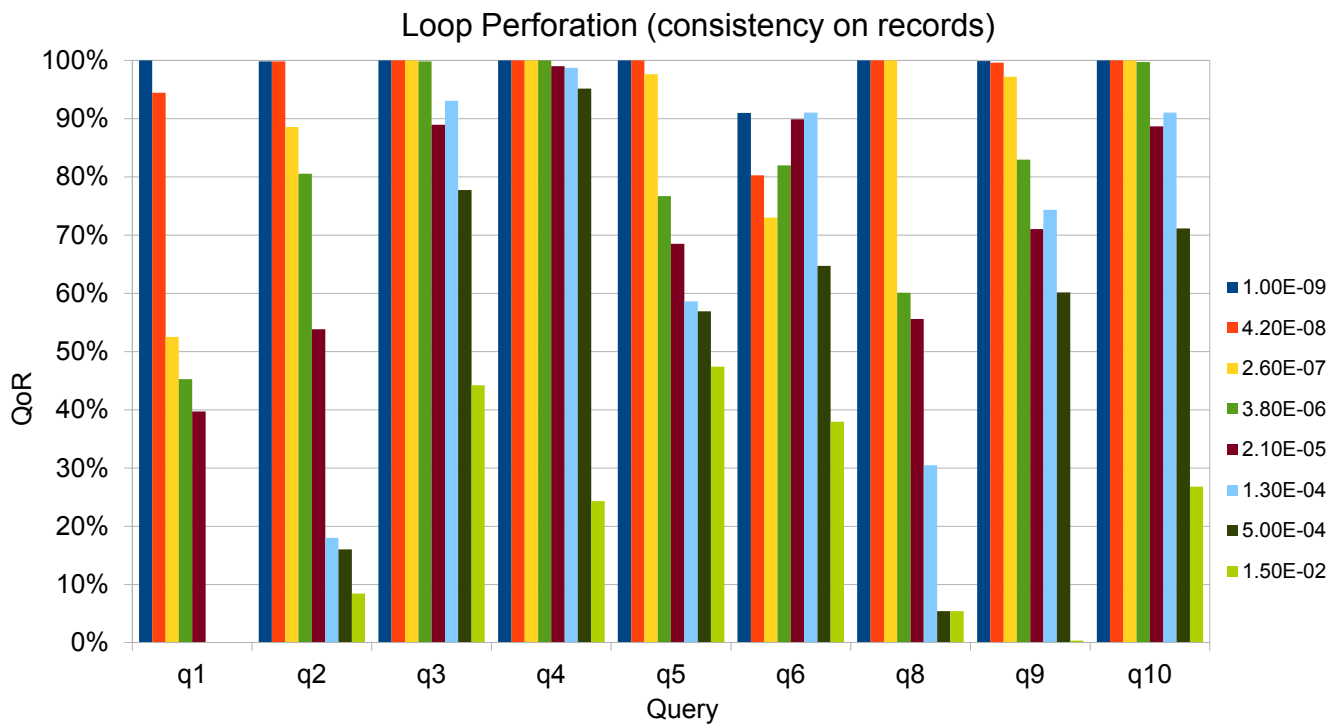
Στις υπόλοιπες περιπτώσεις που έχουν κοινά ποσοστά εφαρμογής, πάντα η μείωση των επαναλήψεων έχει μεγαλύτερη τιμή στο Tradeoff σε σύγκριση με την τιμή της μείωσης των

ανανεώσεων.

5.3.2 Εφαρμογή στις ίδιες πλειάδες

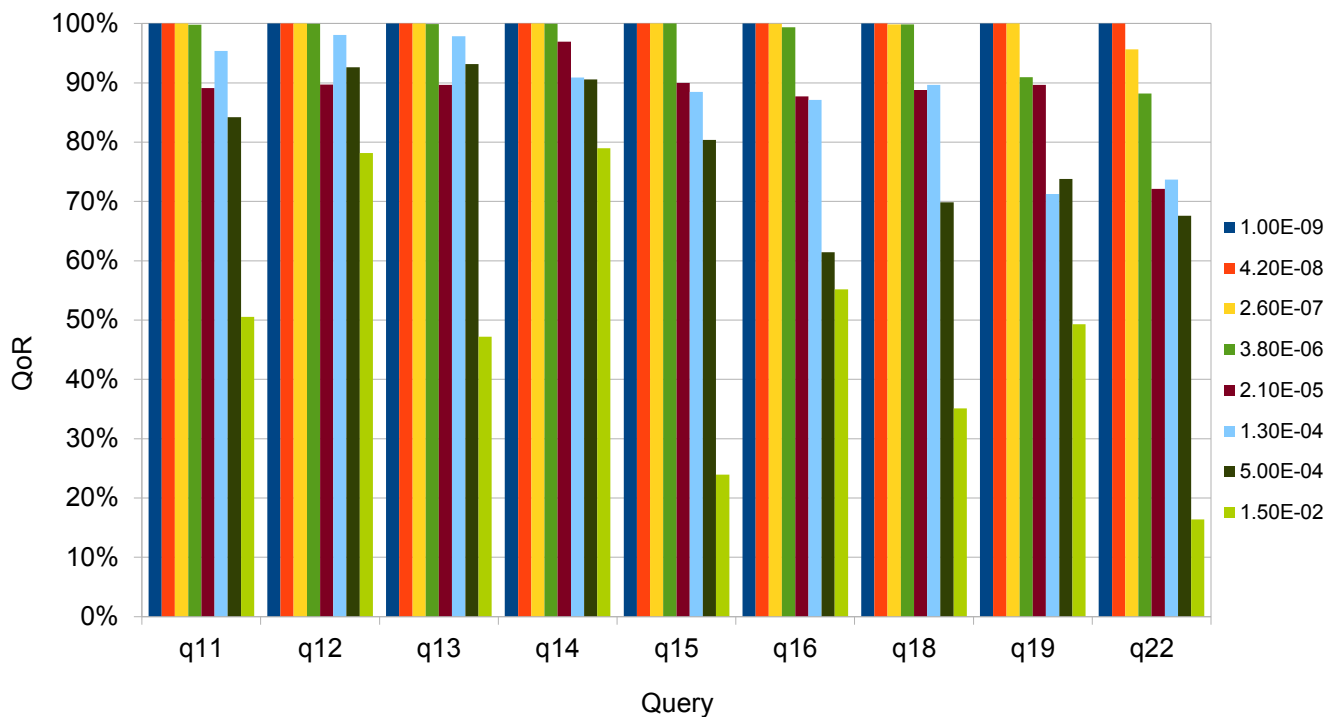
Παρατηρώντας ότι με το ίδιο ποσοστό εφαρμογής δεν υπήρξε ίδιος αριθμός εφαρμογής, δηλαδή το πλήθος των αλλοιωμένων bytes δεν αντιστοιχούσε στο πλήθος των παραλειπόμενων εγγραφών, σε αυτή τη σύγκριση διατηρείται αυτό το κοινό σημείο. Πιο συγκεκριμένα, για κάποιο ποσοστό εφαρμογής που παρουσιάζεται, σημαίνει ότι το ποσοστό αναφέρεται στην περίπτωση αλλοίωσης δεδομένων (μείωση του refresh rate) ενώ για την μέθοδο της μείωσης των επαναλήψεων θα ισχύει ότι παραλείπονται όλες οι πλειάδες για τις οποίες υπήρξε κάποια αλλοίωση στα στοιχεία τους.

Χρειάζεται να παρουσιαστούν νέα αποτελέσματα (θα αναγράφονται ως περίπτωση με consistency on records) μόνο για την μέθοδο της μείωσης των επαναλήψεων, γιατί διαφοροποιούνται από τα προηγούμενα. Ενώ για την μέθοδο με την μείωση του ρυθμού ανανέωσης δεν χρειάζονται νέα αποτελέσματα.

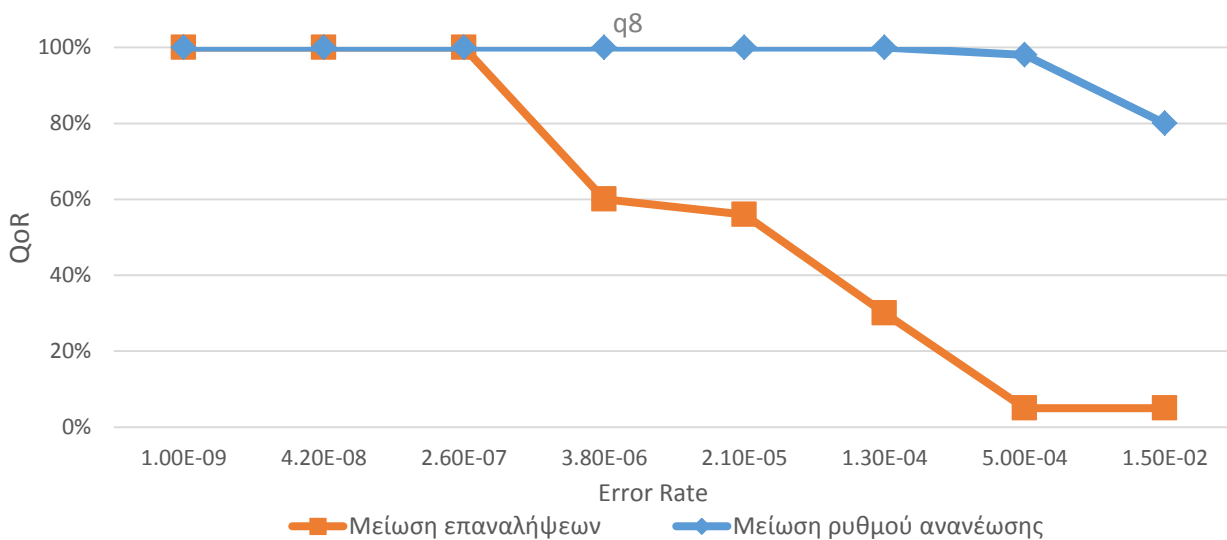


Σχήμα 12α: QoR για κάθε ερωτήματα, κατά την εφαρμογή της μείωσης των επαναλήψεων με συνέπεια στις εγγραφές

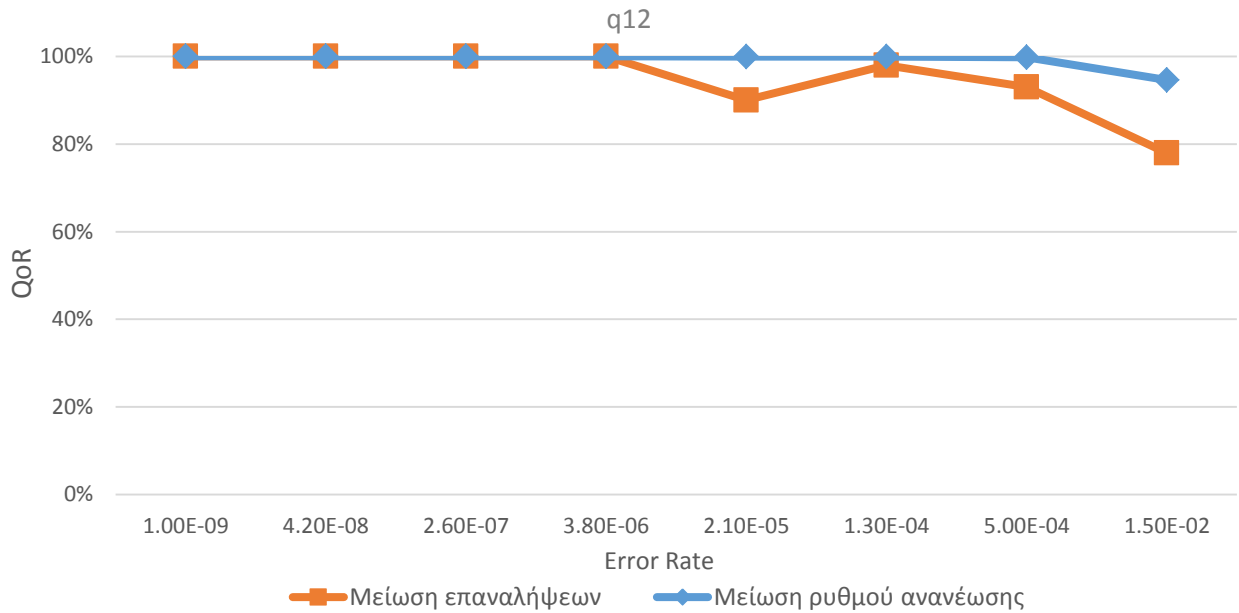
Loop Perforation (consistency on records)



Σχήμα 12b: QoR για κάθε επερώτημα, κατά την εφαρμογή της μείωσης των επαναλήψεων με συνέπεια στις εγγραφές



Σχήμα 12c: QoR για το επερώτημα q8, κατά την εφαρμογή της μείωσης των επαναλήψεων με συνέπεια στις εγγραφές σε σύγκριση με την εφαρμογή της μεθόδου της μείωσης του ρυθμού ανανέωσης

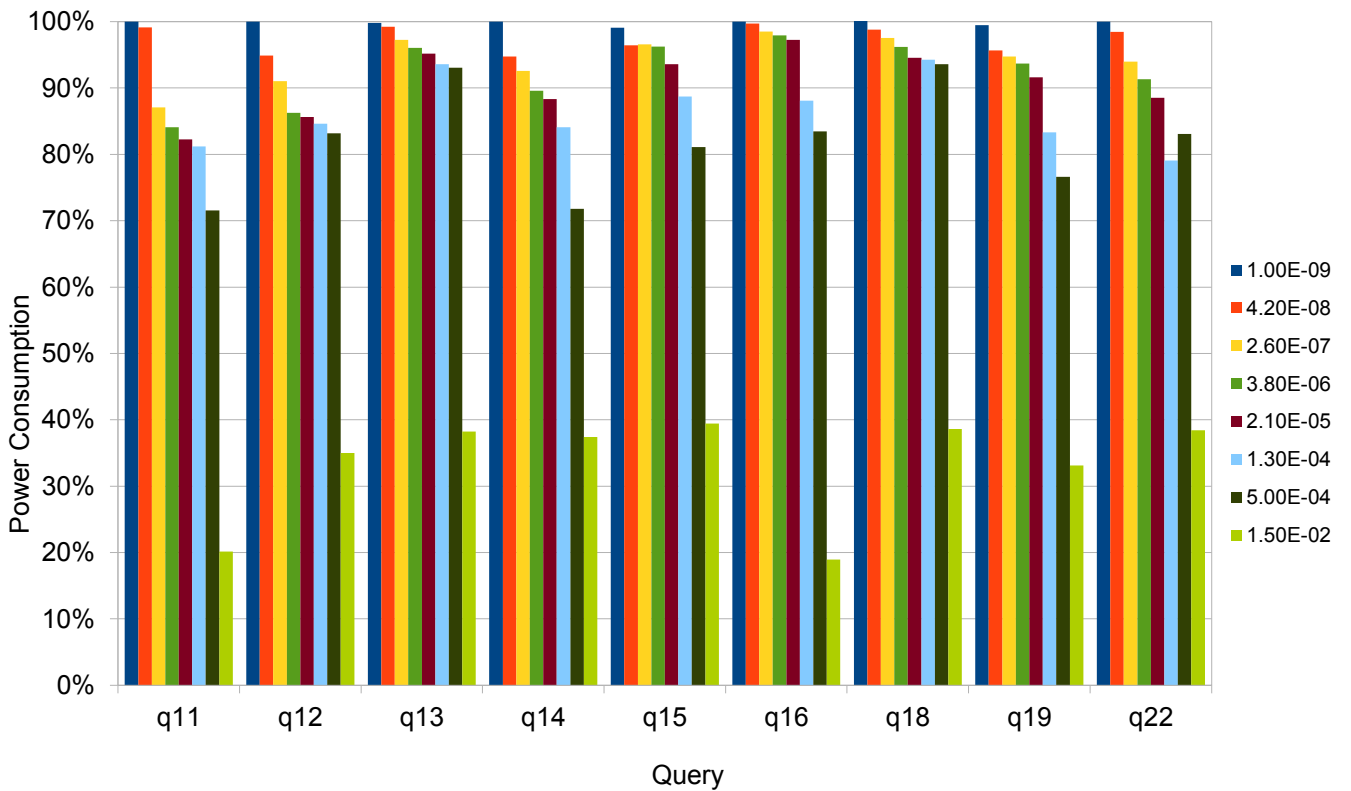
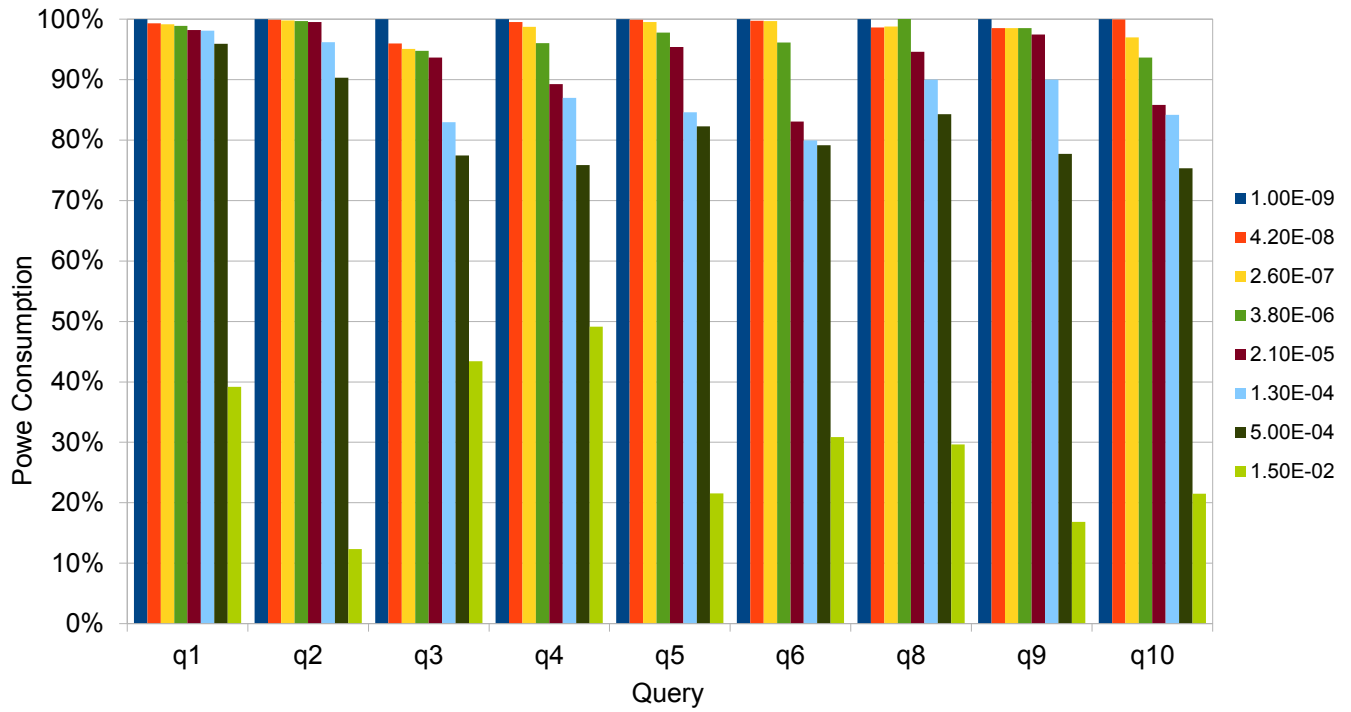


Σχήμα 12d: QoR για το επερωτήμα q12, κατά την εφαρμογή της μείωσης των επαναλήψεων με συνέπεια στις εγγραφές σε σύγκριση με την εφαρμογή της μεθόδου της μείωσης του ρυθμού ανανέωσης

Σχεδόν σε όλες τις εφαρμογές η τιμή της QoR είναι ψηλότερη για τη μέθοδο της μείωσης του ρυθμού ανανέωσης. Άρα, από πλευράς ποιότητας αποτελέσματος, είναι προτιμότερο να χρησιμοποιηθεί μία εγγραφή με αλλοιωμένο περιεχόμενο παρά να παραλειφθεί εντελώς από τον υπολογισμό.

Για ποσοστό εφαρμογής ίσο με 0.1% δεν παρουσιάζονται τα αποτελέσματα γιατί ήταν πάρα πολύ χαμηλά, για όλα τα επερωτήματα, και έτσι δεν είναι προσιτό να εφαρμοστεί ανεξάρτητα της εξοικονόμησης που τυχόν να μας προσφέρει.

Loop Perfertation (consistency on records)



Σχήμα 13: Κατανάλωση για κάθε επερώτηση, κατά την εφαρμογή της μείωσης των επαναλήψεων με συνέπεια στις εγγραφές

Όσο αφορά την κατανάλωση, στην περίπτωση με αλλοιωμένα δεδομένα δεν μειωνόταν εύκολα κάτω από 93%, πράγμα το οποίο δεν ισχύει στην μείωση των προσβάσεων (consistency on records). Για τα πρώτα ποσοστά εφαρμογής ($1e-09$ μέχρι $2.6e-07$) στα περισσότερα ερωτήματα συμφέρει η μέθοδος μείωσης του ρυθμού ανανέωσης ενώ για τα μεγαλύτερα ποσοστά συμφέρει η μείωση των επαναλήψεων.

Άρα για ποσοστά μικρότερα ή ίσα με $2.6e-07$ έχουμε καλύτερη ποιότητα αποτελέσματος και περισσότερη εξοικονόμηση ενέργειας (επομένως μεγαλύτερη τιμή και στο Tgrfeoff) με την αλλοίωση πληροφοριών στις εγγραφές και όχι με την παράληψη τους.

5.4 Περιτταίρω για τη Μείωση των Προσβάσεων

Μέσω της αξιολόγησης και της σύγκρισης των δύο μεθόδων, παρατηρήθηκε μεγαλύτερο ενδιαφέρον στη μείωση των προσβάσεων, γιατί τα αποτελέσματα του έδειξαν ότι όσο αυξάνεται το ποσό χαλαρότητας κλιμακώνονται και οι αλλοιώσεις του αποτελέσματος αλλά και η εξοικονόμηση. Για αυτό έγιναν επιπλέον αξιολογήσεις οι οποίες έχουν ως σκοπό να εντοπίζουν την αιτία αλλοίωσης των αποτελεσμάτων και από ποιόν παράγοντα επηρεάζεται η εξοικονόμηση της ενέργειας.

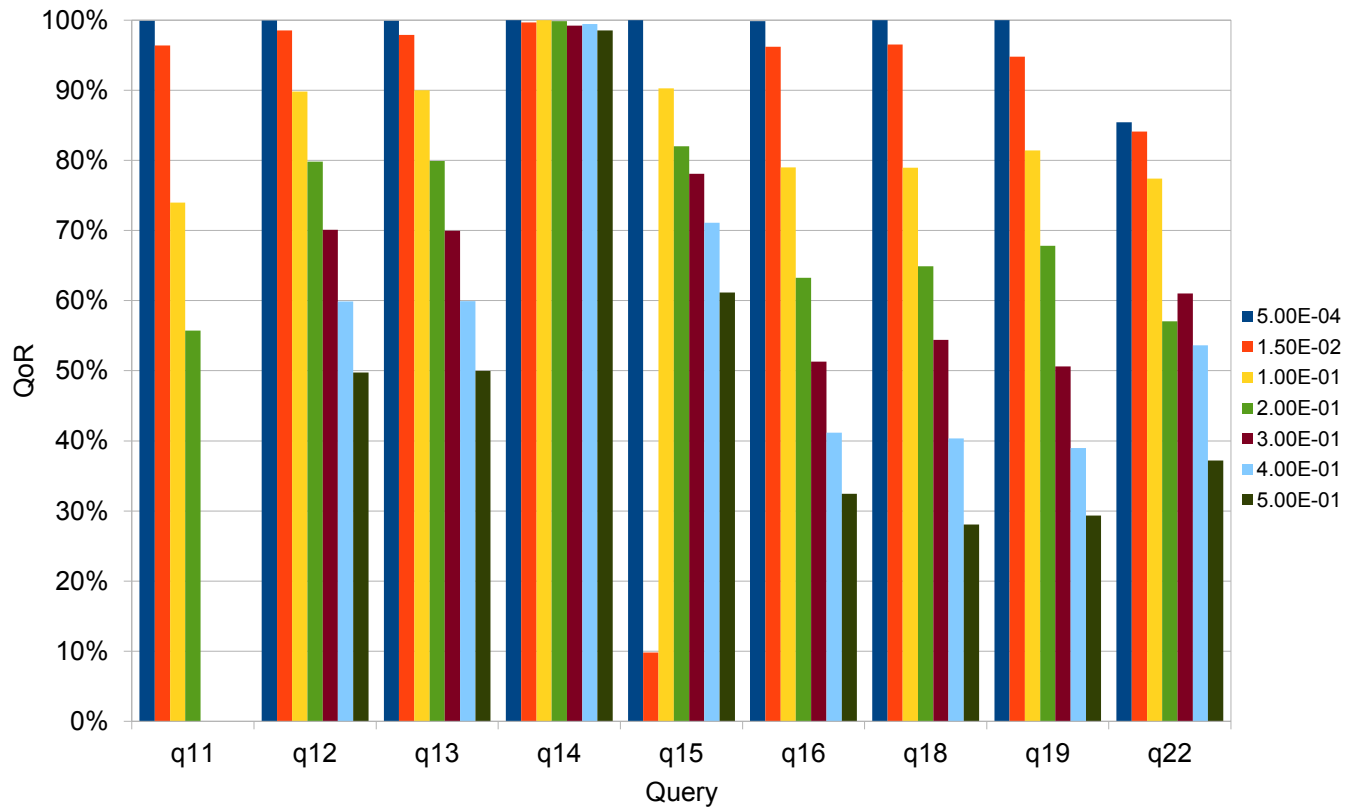
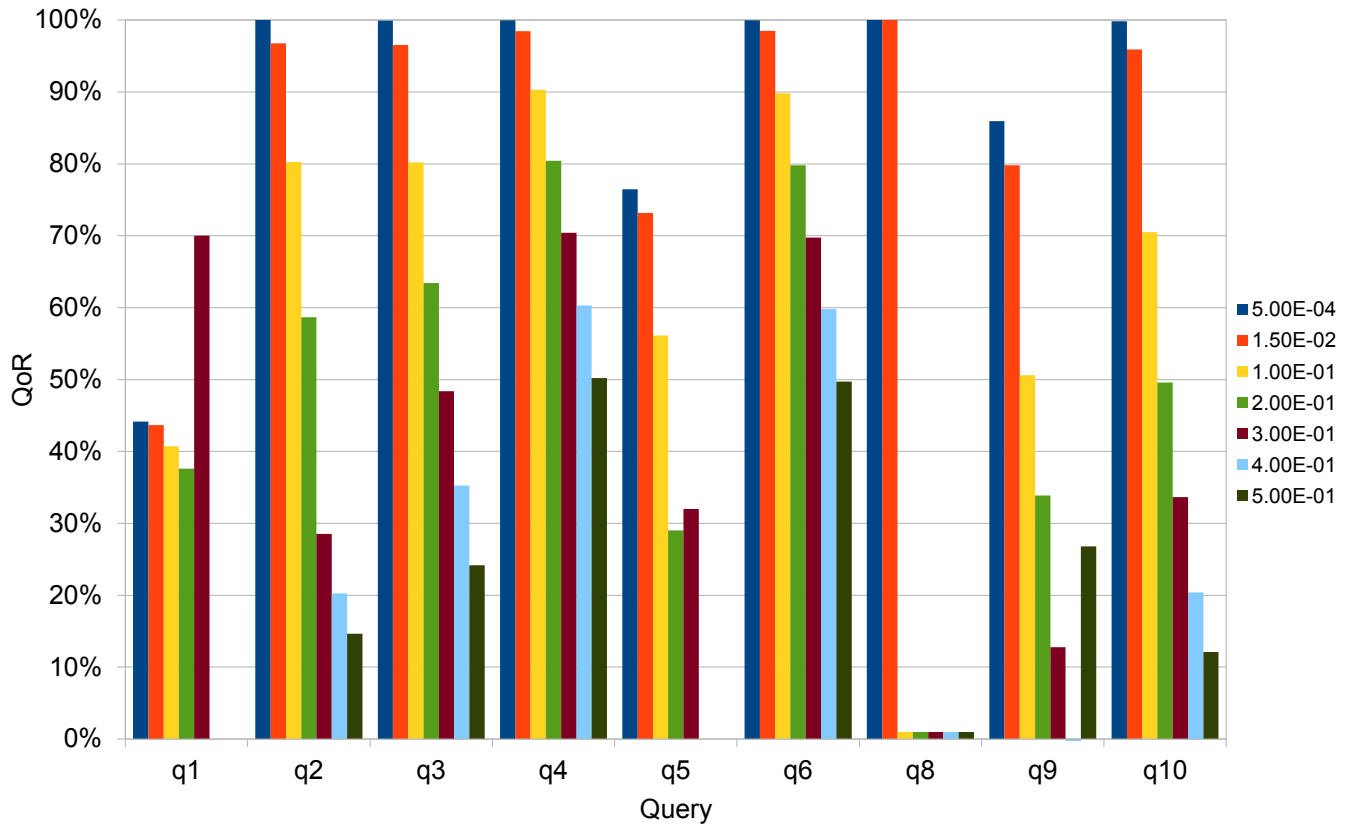
5.4.1 Έλεγχος στις πλειάδες που αφαιρούνται

Για τη συλλογή αυτών των αποτελεσμάτων χρειάστηκε να γίνουν επιπρόσθετες εκτελέσεις στις οποίες να μην διαγράφονται τυχαία οι πληροφορίες αλλά να γίνεται έλεγχος στο ποιες πλειάδες αφαιρούνται.

Στην κατανάλωση δεν αναμενόταν κάποια αλλαγή γιατί ο βασικός παράγοντας που επηρεάζει την εξοικονόμηση είναι το πλήθος των πλειάδων που παραλείφθηκαν και όχι το ποιες ήταν αυτές.

Ο πρώτος έλεγχος στις πλειάδες που έγινε ήταν να αφαιρούνται όλες οι εγγραφές από την αρχή. Αφού η αφαίρεση γινόταν από αρχεία, αυτό σημαίνει ότι η διαγραφή ξεκινούσε από την αρχή (begin) του αρχείου και αφαιρούσε συνεχόμενα στοιχεία, μέχρι να διαγραφούν όσα ταίριαζαν στο αντίστοιχο ποσοστό σφάλματος κάθε φορά.

Loop Perforation (begin)

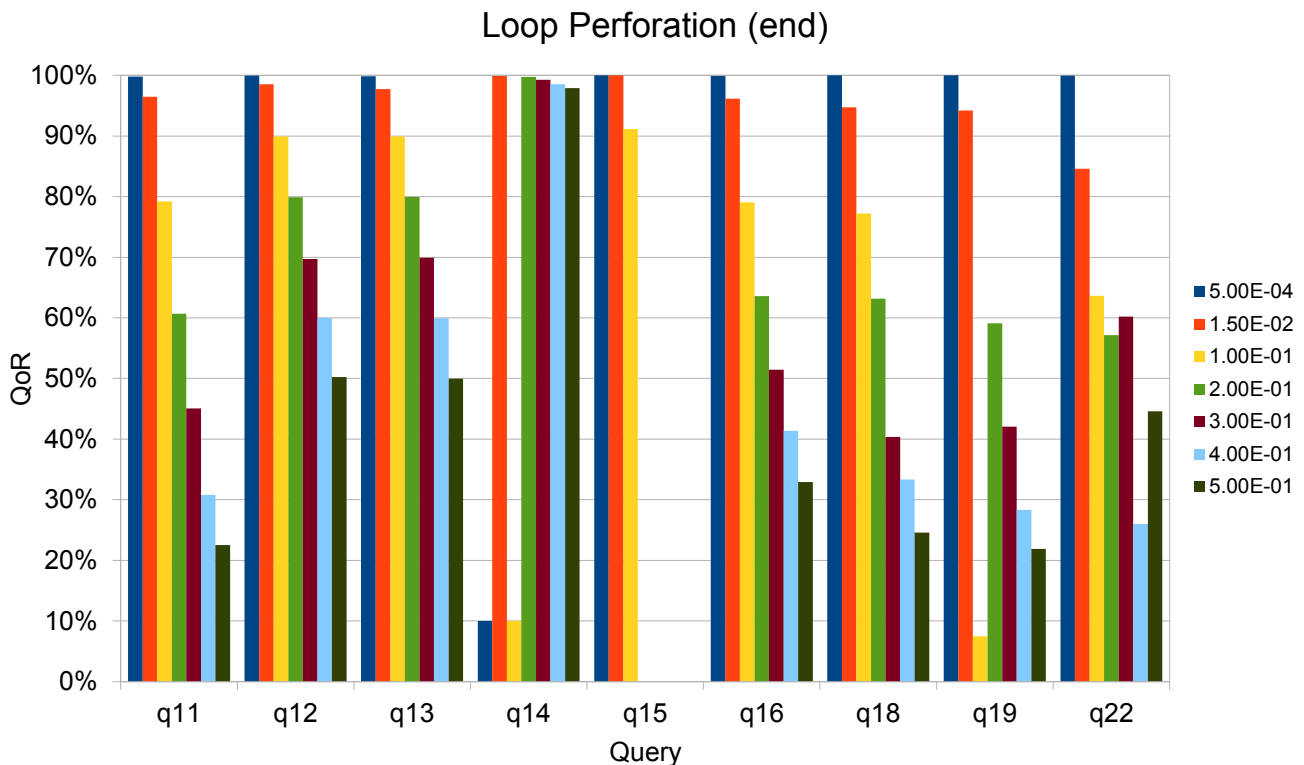


Σχήμα 14: QoR για κάθε επερώτημα, κατά την εφαρμογή της μείωσης των επαναλήψεων με διαγραφή αρχικών στοιχείων

Είναι εμφανές ότι υπάρχει διαφορά στην ποιότητα του αποτελέσματος με σύγκριση την τυχαία διαγραφή στοιχείων. Στα περισσότερα ερωτήματα το αποτέλεσμα βελτιώθηκε, χωρίς όμως αυτό να σημαίνει ότι δεν υπάρχουν περιπτώσεις στις οποίες η ποιότητα του αποτελέσματος να μειώθηκε αρκετά (π.χ. q8). Αυτό μας υποδεικνύει την διαφορετική σημαντικότητα της κάθε εγγραφής για το αποτέλεσμα. Δηλαδή ότι η παράλειψη μερικών εγγραφών έχουν μεγαλύτερο αντίκτυπο από την παράλειψη κάποιων άλλων. Οι ομάδες των ερωτημάτων σε αυτή την περίπτωση έχουν διαμορφωθεί ως εξής:

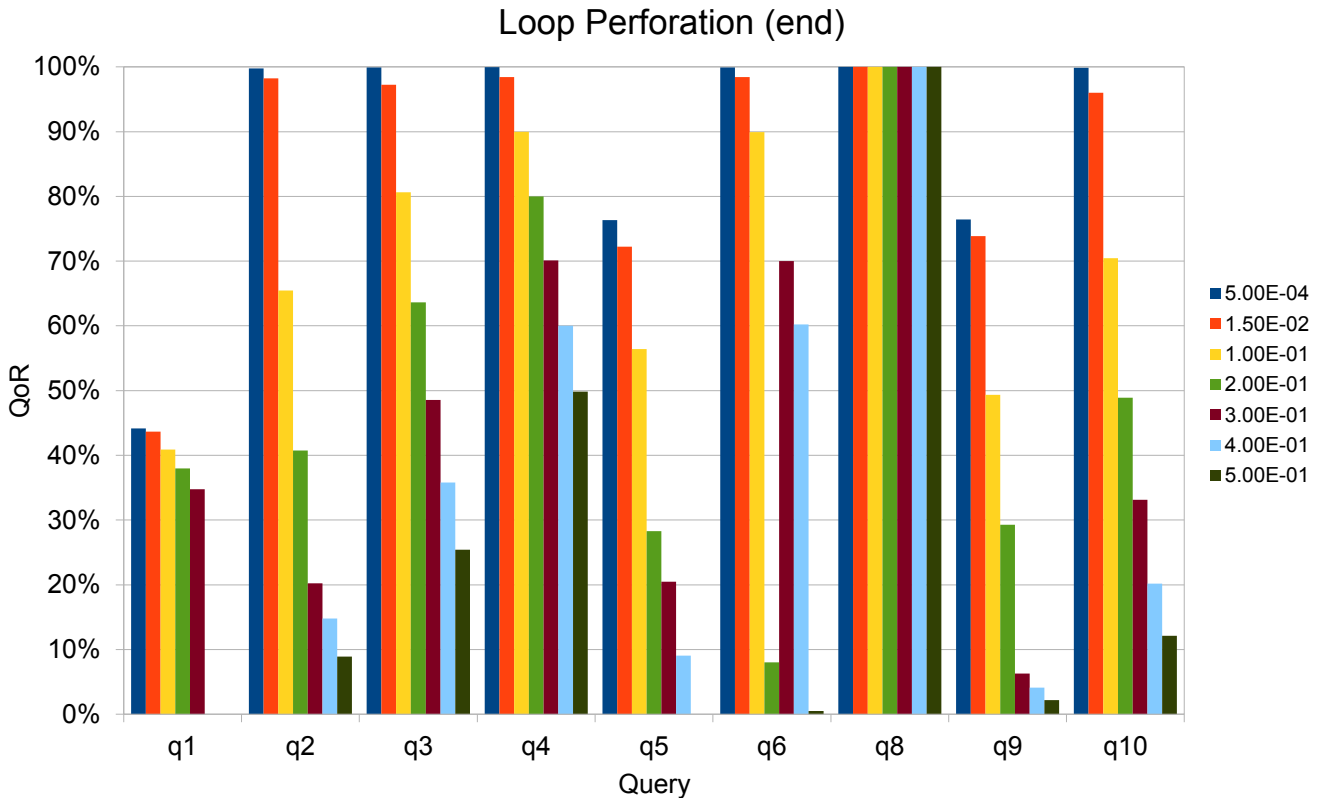
Very Good	q4, q6, q12, q13, q14, q15, q16, q18, q19, q22	(0-2 φορές με QoR <50%)
Good	q2, q3, q11	(3 φορές με QoR <50%)
Bad	q5, q9, q10	(4 φορές με QoR <50%)
Very Bad	q1, q8	(>4 φορές με QoR <50%)

Επίσης υπάρχουν φαινόμενα αστάθειας, δηλαδή να αυξάνεται το ποσό της χαλαρότητας της μεθόδου αλλά η ποιότητα του αποτελέσματος να αυξάνεται αντί να μειώνεται. Αυτό συμβαίνει μόνο σε ερωτήματα που ανήκουν στην κατηγορία value. Οφείλεται στο γεγονός ότι κάθε αύξηση του ποσοστού εφαρμογής καλύπτει την διαγραφή όλων των στοιχείων που αφαιρέθηκαν από την προηγούμενη εφαρμογή, συν κάποιων επιπλέον εγγραφών. Τα κοινά στοιχεία μπορεί να προκάλεσαν μείωση ή αύξηση της τιμής σε σύγκριση με την αυθεντική, και η διαγραφή των επιπλέον στοιχείων μπορεί να επιφέρει την αντίθετη αλλαγή στο αποτέλεσμα. Έτσι το γεγονός ότι διαγράφηκαν έχει θετική επίπτωση στην τιμή της QoR.



Σχήμα 15α: QoR για κάθε ερωτήματα, κατά την εφαρμογή της μείωσης των επαναλήψεων με διαγραφή τελευταίων στοιχείων

Ο επόμενος έλεγχος αφαιρούσε πλειάδες από το τέλος των στοιχείων μας. Δηλαδή ξεκινούσε από το τέλος (end) των αρχείων και διάγραφε το κατάλληλο πλήθος από στοιχεία.



Σχήμα 15b: : QoR για κάθε επερώτημα, κατά την εφαρμογή της μείωσης των επαναλήψεων με διαγραφή τελευταίων στοιχείων

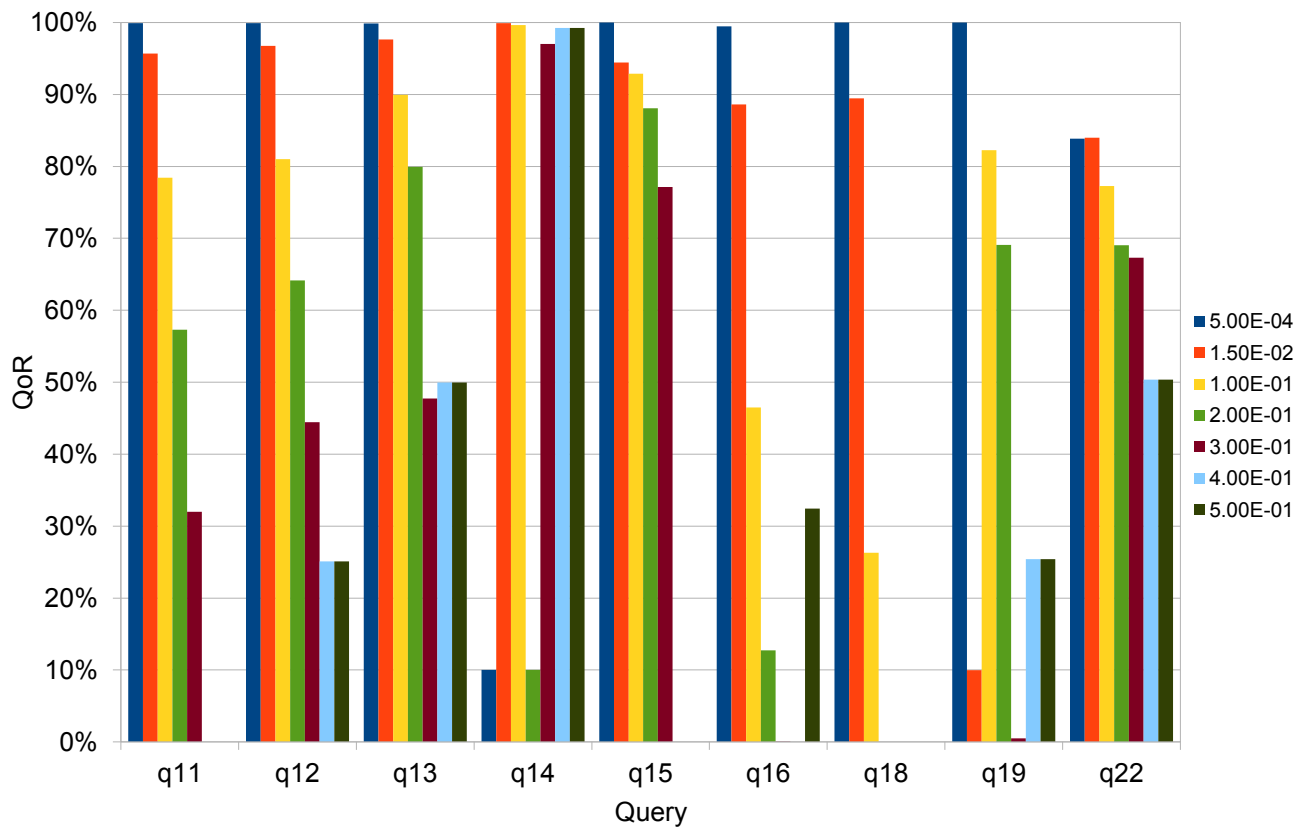
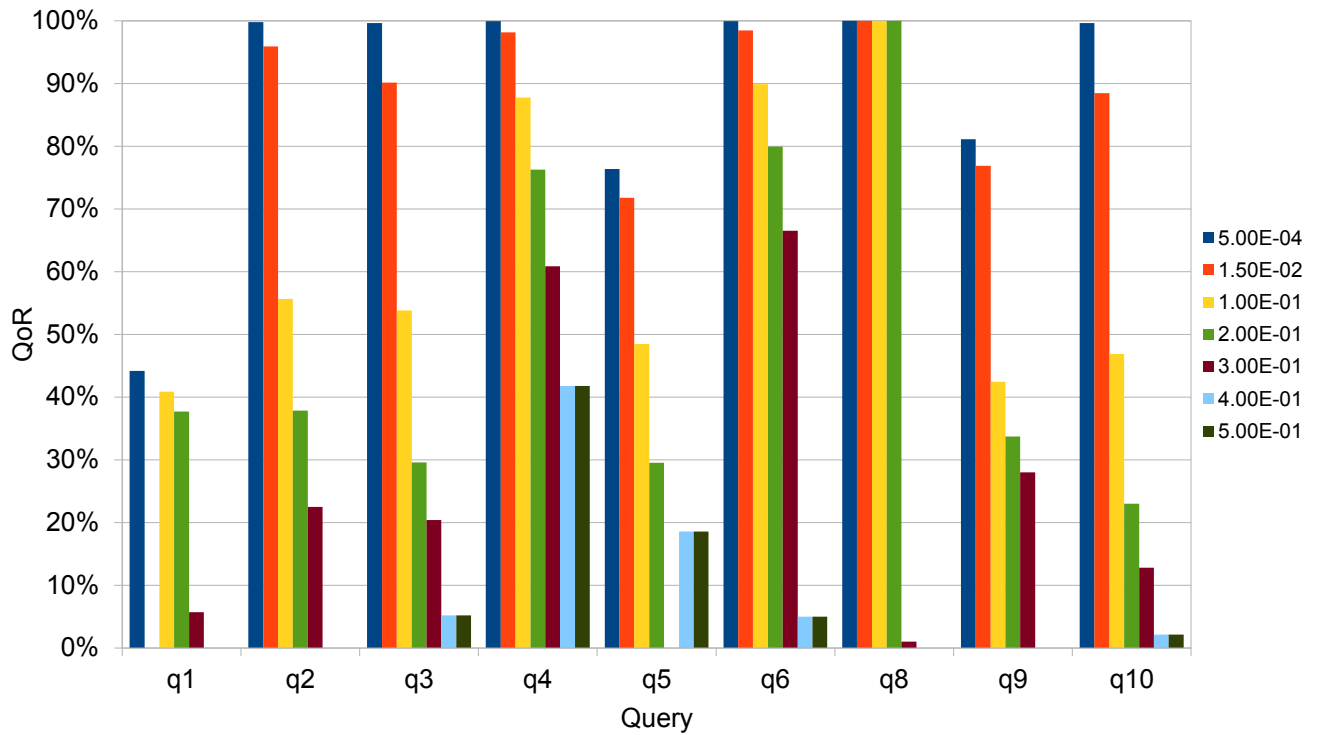
Η επίπτωση στο αποτέλεσμα είναι παρόμοια με τον προηγούμενο έλεγχο (begin), δηλαδή υπάρχει και θετική αλλά και αρνητική αλλαγή. Επίσης υπάρχουν και πάλι σημεία αστάθειας στα αποτελέσματα. Πιο κάτω παρουσιάζονται οι ομάδες που αντιστοιχούν σε αυτό τον έλεγχο:

Very Good	q4, q6, q8, q12, q13, q14, q16, q22	(0-2 φορές με QoR <50%)
Good	q3, q11, q18	(3 φορές με QoR <50%)
Bad	q2, q5, q10, q15, q19	(4 φορές με QoR <50%)
Very Bad	q1, q9	(>4 φορές με QoR <50%)

Με τον επόμενο έλεγχο, επιδιώχθηκε η αποφυγή του κοινού στοιχείου των δύο προηγούμενων ελέγχων, δηλαδή η αύξηση στην χαλαρότητα τώρα επιθυμούμε να επιφέρει αύξηση στο πλήθος των διαγραμμένων εγγραφών αλλά αν συγκρίνουμε τις εγγραφές που αφαιρέθηκαν δεν επιδιώκετε να συμπεριλαμβάνουν όλες όσες ήταν με την εφαρμογή της λιγότερο χαλαρωμένης περίπτωσης.

Αυτό επιτυγχάνεται με το να διαγράφονται στοιχεία με έναν στατικό (stride) ρυθμό. Πιο συγκεκριμένα για κάποιο ποσοστό εφαρμογής, έστω 0.1%, τότε θα διαγράφεται πάντοτε το πρώτο στοιχείο ανά 1000 στοιχεία.

Loop Perforation (stride)



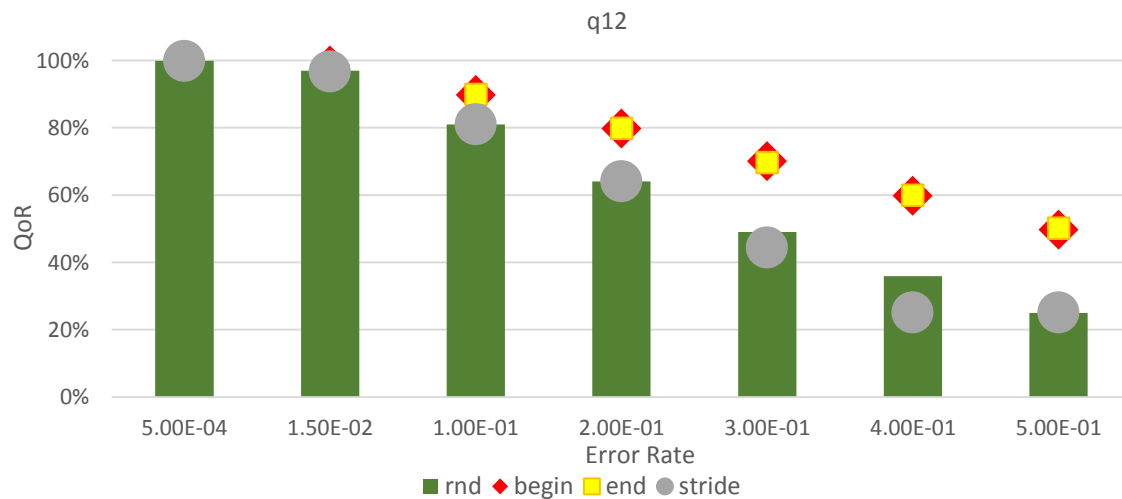
Σχήμα 16α: QoR για κάθε ερώτημα, κατά την εφαρμογή της μείωσης των επαναλήψεων με χρήση στατικού ρυθμού διαγραφής

Με την εφαρμογή της μείωσης των επαναλήψεων με στατικό ρυθμό πάνω στις πλειάδες, παρατηρούμε ότι τα αποτελέσματα είναι αρκετά όμοια με την περίπτωση της εφαρμογής πάνω σε τυχαίες πλειάδες (τμήμα **Error! Reference source not found.**). Έτσι και οι ομάδες των επερωτημάτων είναι σχεδόν οι ίδιες.

Very Good	q4, q6, q13, q14, q15, q22	(0-2 φορές με QoR <50%)
Good	q8, q11, q12	(3 φορές με QoR <50%)
Bad	q2, q3, q19	(4 φορές με QoR <50%)
Very Bad	q1, q5, q9, q10, q16, q18	(>4 φορές με QoR <50%)



Σχήμα 16b: QoR για το q4, κατά την εφαρμογή της μείωσης των επαναλήψεων με χρήση όλων των τρόπων αφαίρεσης πλειάδων



Σχήμα 16c: QoR για το q12, κατά την εφαρμογή της μείωσης των επαναλήψεων με χρήση όλων των τρόπων αφαίρεσης πλειάδων

Η εφαρμογή των δύο πρώτων ελέγχων (begin, end), στις περισσότερες περιπτώσεις βελτίωσαν την ποιότητα του αποτελέσματος γιατί τα επερωτήματα έχουν την τάση να χρησιμοποιούν δεδομένα τα οποία βρίσκονται διασκορπισμένα στην μνήμη. Άρα η μαζική παράλειψη στοιχείων από την αρχή ή από το τέλος δεν συμπεριλαμβάνει πολλά από αυτά τα διασκορπισμένα δεδομένα.

Η διαγραφή με στατικό ρυθμό θεωρείται καλύτερη από άποψη υλοποίησης, γιατί είναι πιο απλή και μπορεί να εφαρμοστεί χωρίς να προϋποθέτει την γνώση του πλήθους των πληροφοριών/εγγραφών που έχει στην διάθεση του η βάση δεδομένων.

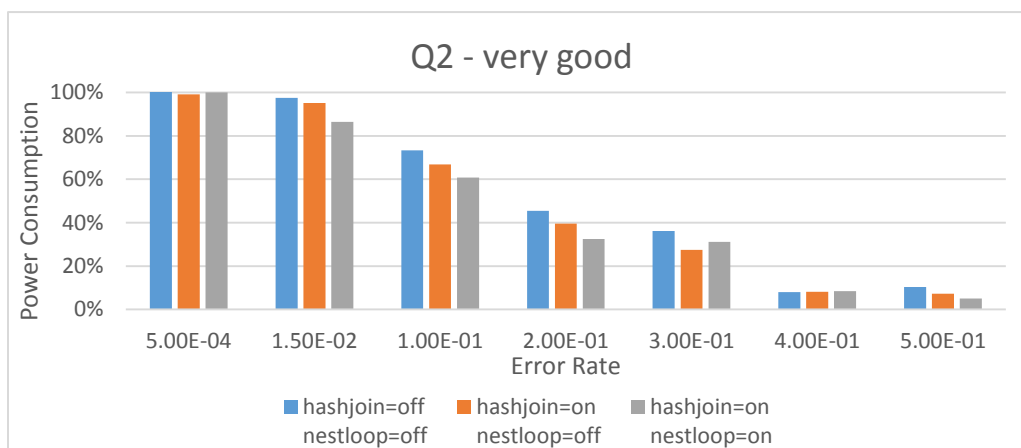
Επειδή υπήρξε ιδιαίτερο ενδιαφέρον στην σύγκριση αυτού του ελέγχου (stride) μαζί με την τυχαία εφαρμογή, παρουσιάζονται στο Παράρτημα Δ επτά γραφικές παραστάσεις, μία για κάθε ποσοστό εφαρμογής, οι οποίες επιδεικνύουν τις τιμές της QoR και για τους δύο τρόπου επιλογής στην διαγραφή πλειάδων καθώς και την μέση τιμή της.

5.4.2 Έλεγχος στην χρήση διαφορετικών αλγόριθμων κατά την εκτέλεση των επερωτημάτων

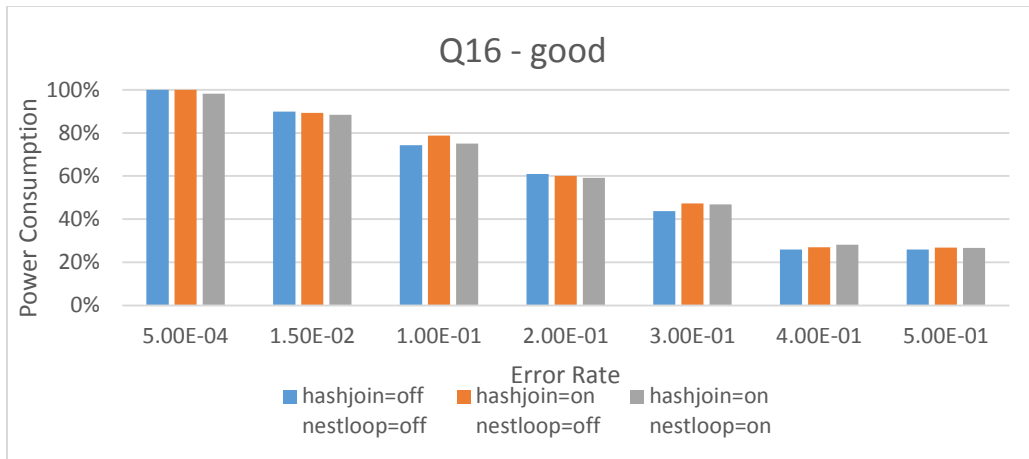
Στην προσπάθεια να βρεθούν παράγοντες οι οποίοι επηρεάζουν το ποσοστό εξοικονόμησης σε ενέργεια κατά την εφαρμογή της μείωσης των προσβάσεων στην εκτέλεση ενός επερωτήματος, δοκιμάστηκε η απενεργοποίηση και ενεργοποίηση κάποιων αλγορίθμων οι οποίοι χρησιμοποιούνταν από το DBMS. Αυτοί οι αλγόριθμοι είναι το hash join και το nest loop.

Με αυτή την διαφοροποίηση δεν αναμενόταν να υπάρξουν αλλαγές στην ποιότητα του αποτελέσματος, λόγο του τρόπου με τον οποίο προσομοιώθηκαν οι μέθοδοι χαλαρότητας. Αλλά αναμενόταν να παρατηρηθούν αλλαγές μόνο στους κύκλους εκτέλεσης των επερωτημάτων, ο οποίος ήταν και ο βασικός σκοπός αυτού του ελέγχου.

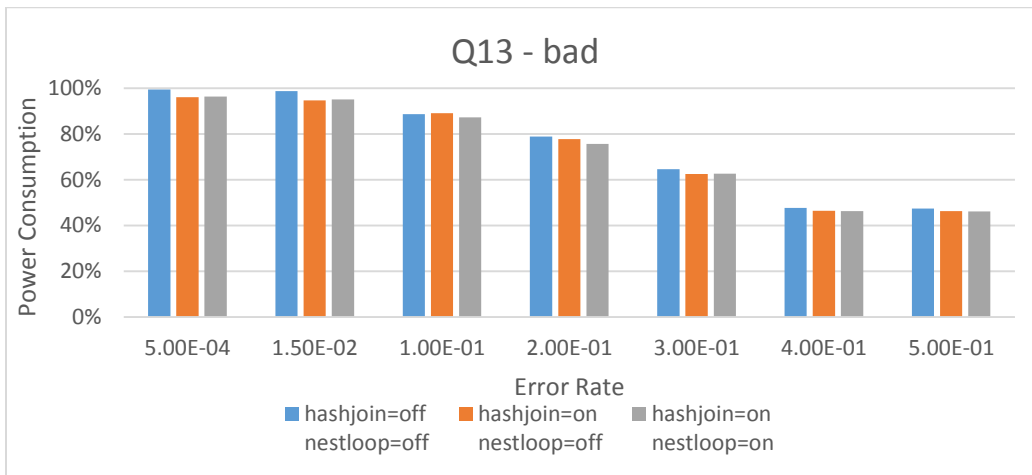
Για να είναι πιο ευδιάκριτα τα αποτελέσματα, επιλέχθηκε ένα επερωτήμα από κάθε ομάδα (very good, good, bad, very bad) όπως είχαν διαχωριστεί βάση των αποτελεσμάτων τους στο Power Consumption κατά την χρήση της μεθόδους με μείωση των επαναλήψεων.



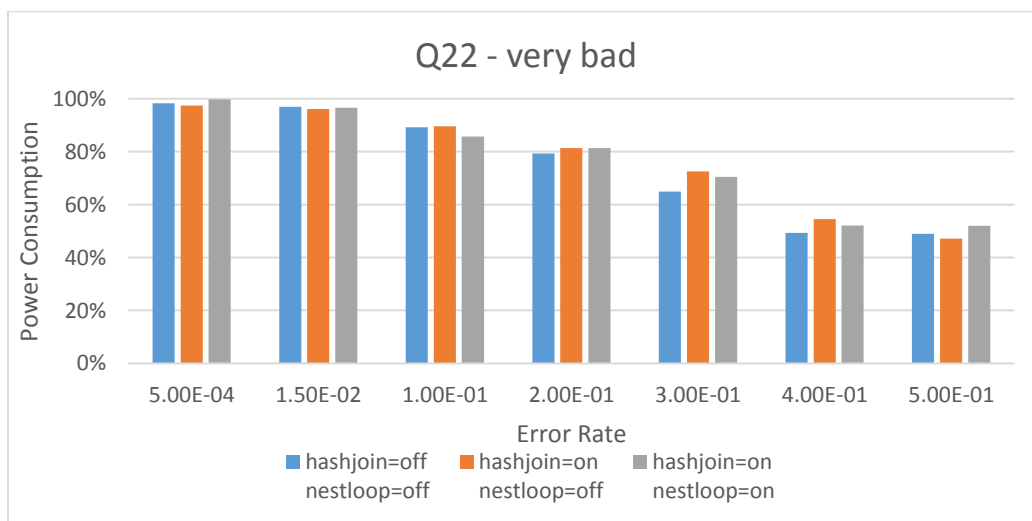
Σχήμα 17: Κατανάλωση για το q2, κατά τον έλεγχο χρήσης διαφορετικών αλγορίθμων



Σχήμα 18: Κατανάλωση για το q16, κατά τον έλεγχο χρήσης διαφορετικών αλγορίθμων



Σχήμα 19: Κατανάλωση για το q13, κατά τον έλεγχο χρήσης διαφορετικών αλγορίθμων



Σχήμα 20: Κατανάλωση για το q22, κατά τον έλεγχο χρήσης διαφορετικών αλγορίθμων

Ο συνολικός χρόνος εκτέλεσης του κάθε επερωτήματος κατά την χρήση διαφορετικών αλγορίθμων άλλαξε, αλλά παρατηρούμε ότι οι τιμές του ποσοστού κατανάλωσης σε ενέργεια δεν διαφέρουν κατά πολύ μεταξύ τους, όταν αναφερόμαστε στην ίδια συχνότητα λάθους. Δηλαδή το ποσοστό της ενέργειας που θα εξοικονομηθεί καθορίζεται από τη γενική δομή του επερωτήματος και όχι από το ποιο αλγόριθμοι χρησιμοποιούνται.

Αυτή η παρατήρηση είναι θετική, γιατί αν γνωρίζουμε το ποσοστό εξοικονόμηση που επιτυγχάνουμε κατά την εφαρμογή αυτή της μεθόδου (loop perforation) σε κάποιο επερωτήμα, σημαίνει ότι αυτό δεν διαφοροποιείται κατά μεγάλο βαθμό από το σε ποιο DBMS θα εκτελεσθεί, ή ποια υλοποίηση και ποιοι αλγόριθμοι θα χρησιμοποιηθούν.

5.5 Συνοπτικά Αποτελέσματα

Τα αποτελέσματα μας μέχρι τώρα αναφερόντουσαν σε κάθε επερωτήμα ξεχωριστά, το οποίο μας έδινε μία πολύ καλή εικόνα για τις επιπτώσεις του προσεγγιστικού υπολογισμού σε μία συγκεκριμένη εφαρμογή.

Αλλά τα αποτελέσματα μπορούν να μας δώσουν μία πιο γενική εικόνα, η οποία θα αναφέρεται για την κάθε μέθοδο χαλαρότητας γενικότερα. Σε αυτό το σημείο έγινε περαιτέρω σύμπτυξη των αποτελεσμάτων για τις βασικές περιπτώσεις, δηλαδή από το τμήμα 5.1 και 5.2.

	Μέθοδος Χαλαρότητας:	Μείωση Ρυθμού Ανανέωσης			Μείωση Προσβάσεων		
	Μέτρηση:	QoR	Power Consumption	Tradeoff	QoR	Power Consumption	Tradeoff
Ποσοστό Σφάλματος	1.00E-09	98.91%	95.35%	1.04	-	-	-
	4.20E-08	97.08%	93.85%	1.03	-	-	-
	2.60E-07	94.74%	93.70%	1.01	-	-	-
	3.80E-06	91.22%	93.40%	0.98	-	-	-
	2.10E-05	81.89%	93.10%	0.88	-	-	-
	1.30E-04	75.41%	92.95%	0.81	-	-	-
	5.00E-04	66.92%	92.80%	0.72	93.74%	98.07%	0.96
	1.50E-02	36.28%	92.65%	0.39	86.01%	93.29%	0.92
	1.00E-01	16.25%	92.50%	0.18	65.17%	81.44%	0.80
	2.00E-01	-	-	-	46.74%	68.49%	0.68
	3.00E-01	-	-	-	38.94%	57.58%	0.68
	4.00E-01	-	-	-	27.00%	46.56%	0.58
5.00E-01	-	-	-	19.31%	37.45%	0.52	

Πίνακας 6: Συνοπτικά αποτελέσματα για κάθε μέθοδο χαλαρότητας

Τα αποτελέσματα στον πίνακα 6 δείχνουν μία μέση τιμή των μετρήσεων σε κάθε ποσοστό σφάλματος για τις δύο μεθόδους χαλαρότητας. Αυτή η μέση τιμή είναι ο μέσος όρος από όλα τα επερωτήματα για το αντίστοιχο σενάριο. Άρα η κάθε τιμή προκύπτει από 18 δείγματα, έτσι θεωρούνται γενικές και ικανές να μας προϊδεάσουν για την επίπτωση που μπορεί να έχει το αντίστοιχο σενάριο (μέθοδος χαλαρότητας και ποσοστό εφαρμογής) σε άλλες εφαρμογές, πάνω στις οποίες πιθανόν να επιθυμήσουμε να το χρησιμοποιήσουμε.

Κεφάλαιο 6

Συμπεράσματα – Μελλοντική Εργασία

6.1	Συμπεράσματα	39
6.2	Μελλοντική Εργασία	40

6.1 Συμπεράσματα

Σε αυτή την εργασία, μελετήθηκε το μοντέλο του προσεγγιστικού υπολογισμού κατά την εκτέλεση μερικών επερωτημάτων, μέσω της χρήσης και αξιολόγησης δύο μεθόδων χαλαρότητας. Αυτές οι μέθοδοι είναι η μείωση των προσβάσεων στην μνήμη και η μείωση του ρυθμού ανανέωσης της κύριας μνήμης.

Από τα αποτελέσματα που συλλέχτηκαν (QoR, Power Consumption, Tradeoff), μπορέσαμε να αξιολογήσουμε το πότε είναι προτιμότερο η χρήση της κάθε μεθόδου και να διαπιστώσουμε μερικούς από τους παράγοντες οι οποίοι επηρεάζουν την ποιότητα του αποτελέσματος και την εξοικονόμηση.

Υπάρχουν πολλές περιπτώσεις όπου η μείωση στην κατανάλωση υπερβαίνει το ποσοστό αλλοίωσης του αποτελέσματος. Σε αρκετά επερωτήματα επιτυγχάνεται εξοικονόμηση πολύ κοντά στο 10% με πολύ χαμηλό αντίκτυπο στο αποτέλεσμα και σε πιο μεμονωμένες περιπτώσεις, διατηρώντας αυτή την ποιότητα στο αποτέλεσμα, η εξοικονόμηση έφθασε το 30%.

Όσο αφορά την καλύτερη παραγωγή προσεγγιστικού αποτελέσματος, είναι προτιμότερη η χρήση των αλλοιωμένων δεδομένων έναντι της παράλειψης των ίδιων δεδομένων. Επίσης φάνηκε ότι τα επερωτήματα q8, q13 και q14 ήταν πολύ ανεκτικά ανεξάρτητα της μεθόδου χαλαρότητας, γιατί βρίσκονται στην ομάδα very good και για τις δύο βασικές περιπτώσεις των μεθόδων χαλαρότητας που μελετήθηκαν.

Ενώ για εξοικονόμηση, σε περιπτώσεις μικρού ποσοστού (περίπου 6%) συνιστάτε η χρήση της μείωσης του ρυθμού ανανέωσης και για μεγαλύτερα ποσοστά η μείωση των επαναλήψεων.

Στη συνέχεια επικεντρωθήκαμε στην μέθοδο της μείωσης των προσβάσεων. Διαπιστώθηκε ότι η ποιότητα του αποτελέσματος έχει άμεση σχέση με το ποιες εγγραφές θα παραλειφθούν και ότι το ποσοστό εξοικονόμηση δεν επηρεάζεται από το ποιους αλγόριθμοι θα χρησιμοποιηθούν κατά την εκτέλεση του επερωτήματος.

6.2 Μελλοντική Εργασία

Μέσα από αυτή την εργασία διερευνήθηκαν μερικές πτυχές και εφαρμογές του προσεγγιστικού υπολογισμού, αλλά σίγουρα υπάρχουν περεταίρω σημεία για μελέτη. Ένα επιπρόσθετο σημείο το οποίο θα μπορούσε να ερευνηθεί, είναι ο συνδυασμός των δύο μεθόδων που χρησιμοποιήθηκαν. Δηλαδή η ταυτόχρονη εφαρμογή τους και το τι είναι ικανό να επιτευχθεί. Σε θεωρητικό επίπεδο μπορούν να χρησιμοποιηθούν τα αποτελέσματα αυτής της έρευνας αλλά θα είχε ακόμα μεγαλύτερο ενδιαφέρον η πειραματική μελέτη τους.

Ένα στοιχείο αυτής της έρευνας το οποίο μπορεί να βελτιωθεί, είναι η εύρεση μιας καλύτερης σχέσης ανάμεσα στην εξοικονόμηση και στην επίδοση ή ο διαχωρισμός της μέτρησης τους γιατί στην παρούσα έρευνα αυτά τα δύο εξισώθηκαν και θεωρήθηκαν ως ποσοστό κατανάλωσης.

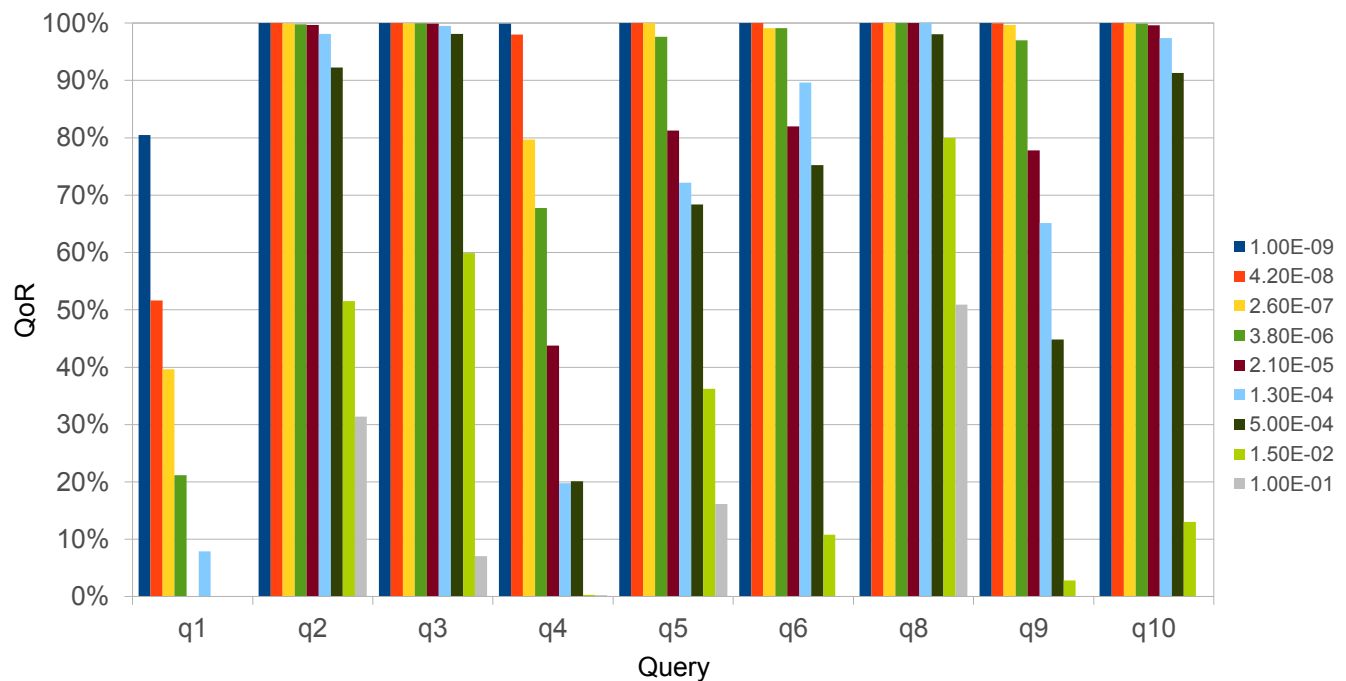
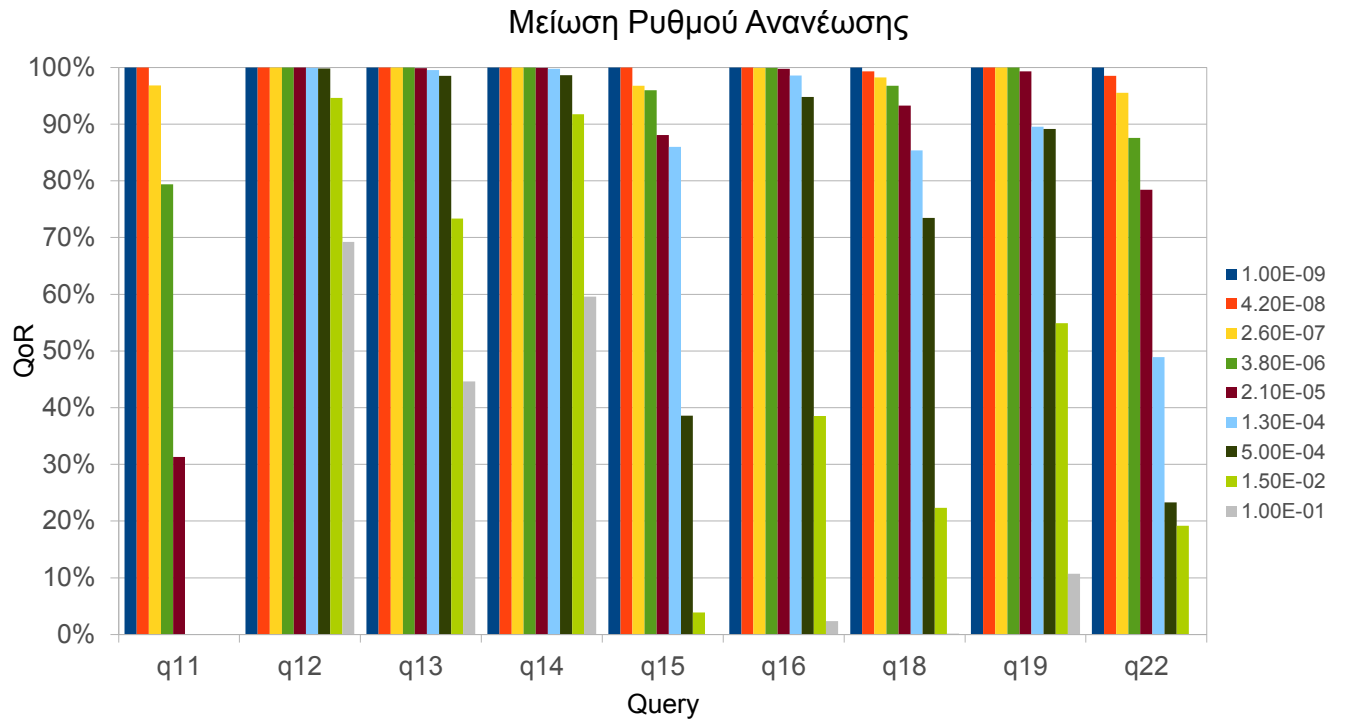
Επίσης, οι μελλοντικές έρευνες, θα μπορούσαν να επικεντρωθούν στην εύρεση περισσότερων παραγόντων οι οποίοι επηρεάζουν το αποτέλεσμα και την εξοικονόμηση. Έτσι ώστε, στη συνέχεια θα μπορούσε να εξεταστεί κατά πόσο με τη σωστή/κατάλληλη επιλογή και διαχείρισης αυτών των παραγόντων, θα υπάρξει θετική επίπτωση στην ωφελιμότητα χρήσης τους. Δηλαδή να διαπιστωθεί κατά πόσο υπάρχει σημαντική βελτίωση στις δύο μετρήσεις, ποιότητα αποτελέσματος και εξοικονόμηση στην ενέργεια.

Βιβλιογραφία

- [1] H. Esmailzadeh, A. Sampson, M. Ringenburt, L. Ceze, D. Grossman, and D. Burger, "Addressing dark silicon challenges with disciplined approximate computing," in *Proc. 4th Workshop on Energy-Efficient Design*, 2012, pp. 1-2.
- [2] D. S. Khudia, B. Zamirai, M. Samadi, and S. Mahlke, "Rumba: an online quality management system for approximate computing," in *Proceedings of the 42nd Annual International Symposium on Computer Architecture*, 2015, pp. 554-566.
- [3] L. Kugler, "Is good enough computing good enough?," *Communications of the ACM*, vol. 58, pp. 12-14, 2015.
- [4] J. Liu, B. Jaiyen, R. Veras, and O. Mutlu, "RAIDR: Retention-aware intelligent DRAM refresh," in *Computer Architecture (ISCA), 2012 39th Annual International Symposium on*, 2012, pp. 1-12.
- [5] S. Liu, K. Pattabiraman, T. Moscibroda, and B. G. Zorn, "Flicker: saving DRAM refresh-power through critical data partitioning," *ACM SIGPLAN Notices*, vol. 47, pp. 213-224, 2012.
- [6] J. Lucas, M. Alvarez-Mesa, M. Andersch, and B. Juurlink, "Sparkk: Quality-scalable approximate storage in DRAM," in *The Memory Forum*, 2014, pp. 1-9.
- [7] S. Misailovic, S. Sidiroglou, H. Hoffmann, and M. Rinard, "Quality of service profiling," in *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering—Volume 1*, 2010), pp. 25-34.
- [8] P. J. Nair, D.-H. Kim, and M. K. Qureshi, "ArchShield: Architectural framework for assisting DRAM scaling by tolerating high error rates," in *ACM SIGARCH Computer Architecture News*, 2013, pp. 72-83.
- [9] A. Sampson, A. Baixo, B. Ransford, T. Moreau, J. Yip, L. Ceze, *et al.*, "Accept: A programmer-guided compiler framework for practical approximate computing," *University of Washington Technical Report UW-CSE-15-01*, vol. 1, 2015.
- [10] S. Sidiroglou-Douskos, S. Misailovic, H. Hoffmann, and M. Rinard, "Managing performance vs. accuracy trade-offs with loop perforation," in *Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering*, 2011, pp. 124-134.
- [11] P. Trancoso, "Getting ready for approximate computing: trading parallelism for accuracy for dss workloads," in *Proceedings of the 11th ACM Conference on Computing Frontiers*, (CF '14), pp. 1-10, Cagliari, Italy, May 2014.
- [12] V. Vassiliadis, C. Chaliotis, K. Parasyris, C. D. Antonopoulos, S. Lalis, N. Bellas, *et al.*, "A significance-driven programming framework for energy-constrained approximate computing," in *Proceedings of the 12th ACM International Conference on Computing Frontiers*, (CF '15), pp. 1-9.
- [13] S. Venkataramani, S. T. Chakradhar, K. Roy, and A. Raghunathan, "Approximate computing and the quest for computing efficiency," in *Proceedings of the 52nd Annual Design Automation Conference*, 2015, pp. 1-6.
- [14] M. Wyse, A. Baixo, T. Moreau, B. Zorn, J. Bornholt, A. Sampson, *et al.*, "REACT: A Framework for Rapid Exploration of Approximate Computing Techniques.," in *WAX 2015*
- [15] JEDEC, "DDR3 SDRAM Specification," www.jedec.org/sites/default/files/docs/JESD79-3E.pdf, 7 2010
- [16] T. P. P. C. (TPC). TPC BenchmarkTM H (Decision Support) Standard Specification Revision 2.17.1. http://www.tpc.org/TPC_Documents_Current_Versions/pdf/tpch2.17.1.pdf, 10 2014.
- [17] T. P. G. D. Group. PostgreSQL 9.3.10 Documentation. <http://www.postgresql.org/docs/9.3/static/index.html>, 6 2013.

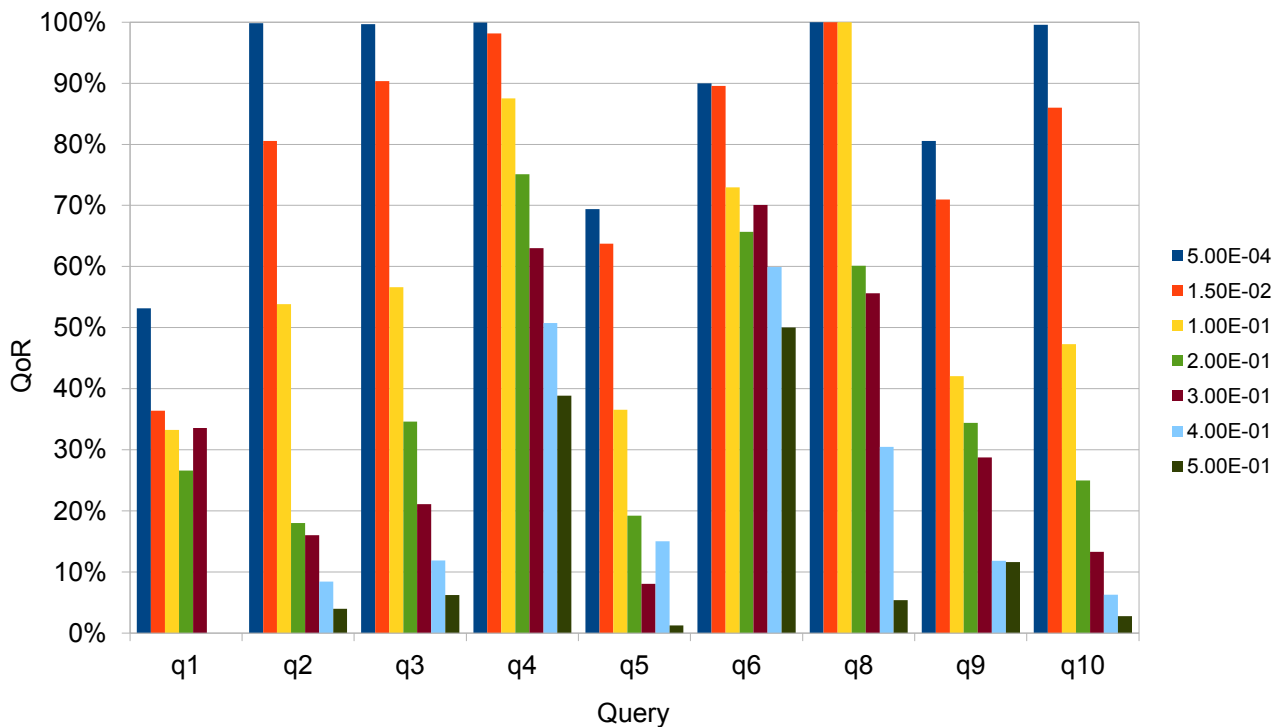
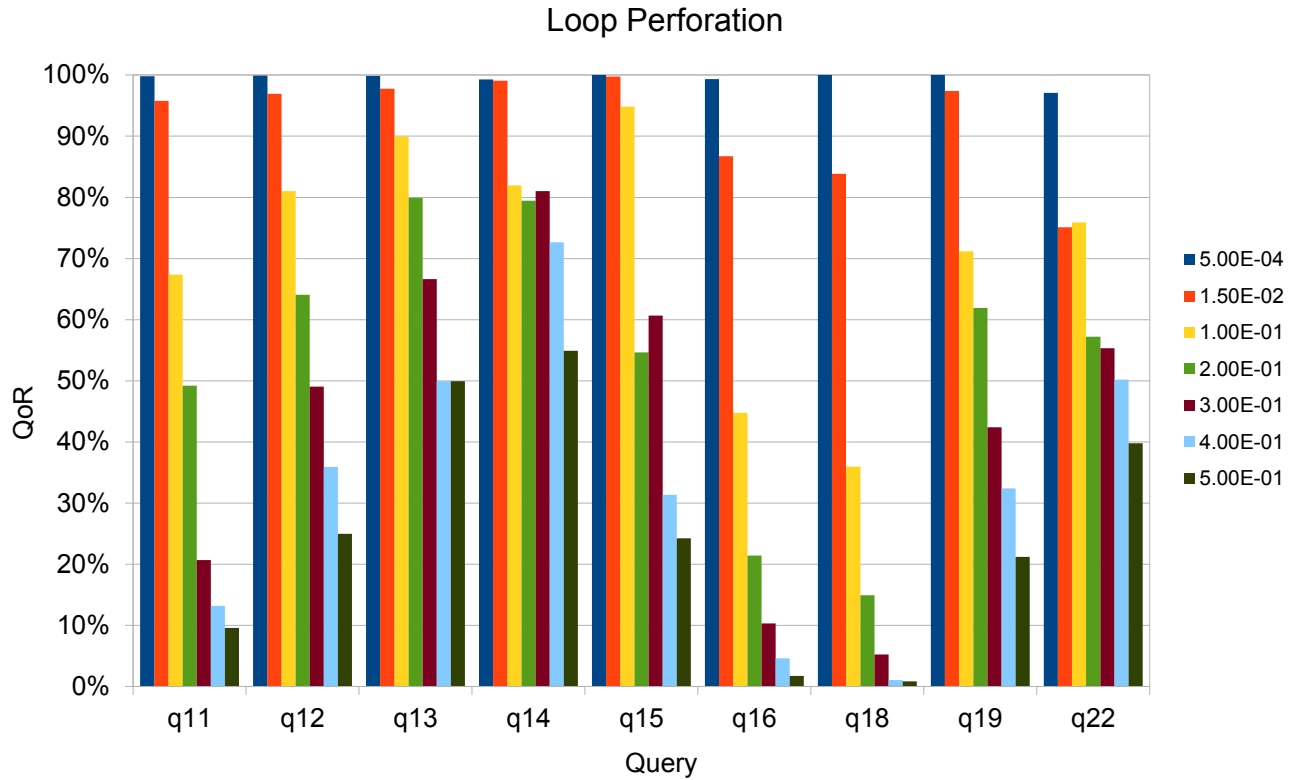
Παράρτημα Α

Παρουσιάζονται οι τιμές του QoR για κάθε επερώτηση, κατά την εφαρμογή της μείωσης του ρυθμού ανανέωσης. Οι μετρήσεις περιλαμβάνουν πολλαπλές τιμές (με διαφορετικό χρώμα) για την ποιότητα του αποτελέσματος και η κάθε τιμή απευθύνεται σε ένα ποσοστό σφάλματος.



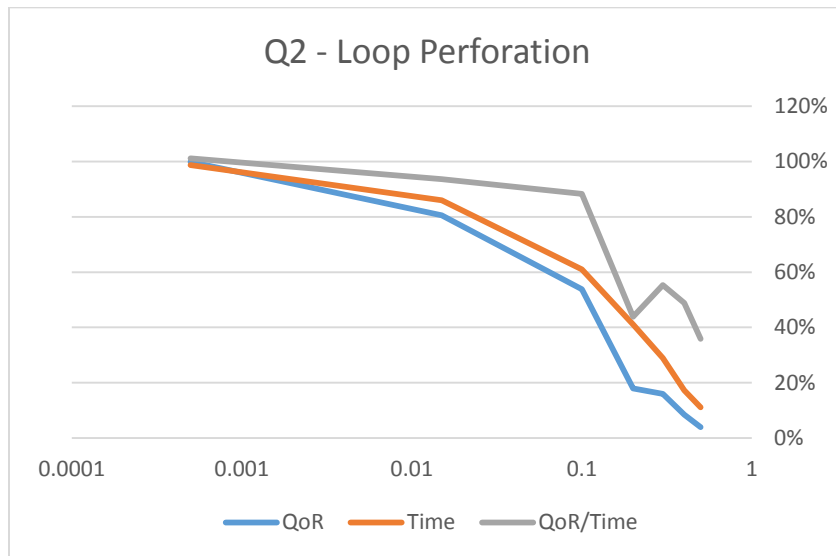
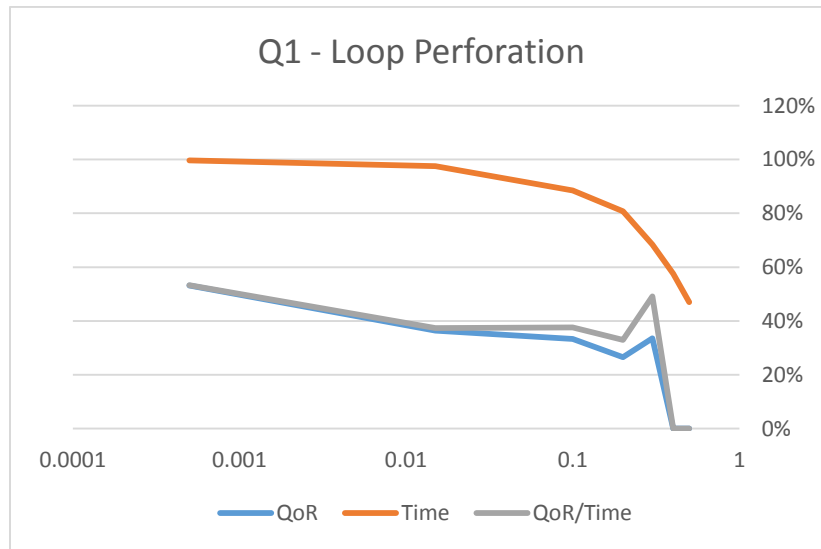
Παράρτημα Β

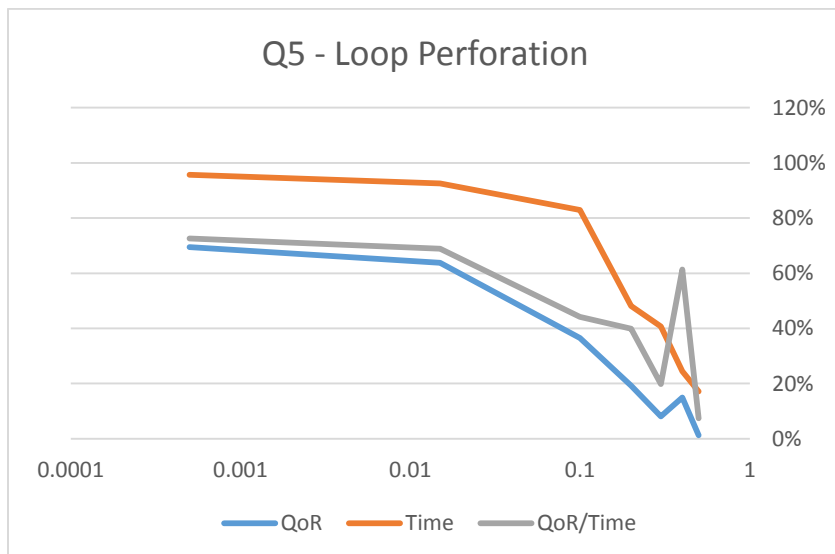
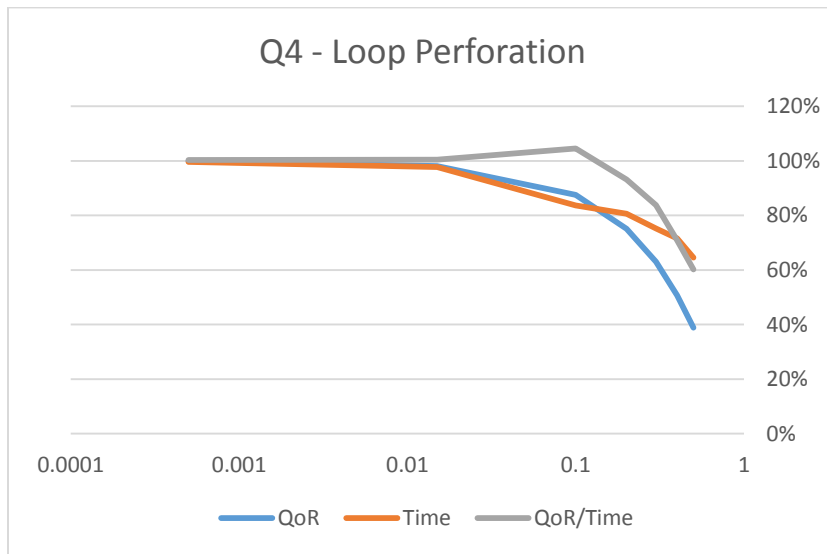
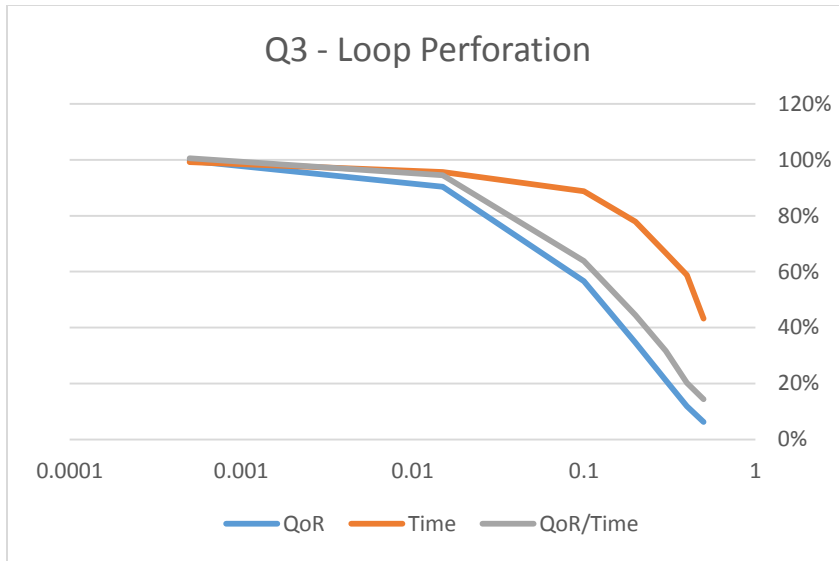
Πιο κάτω βρίσκονται οι τιμές του QoR για κάθε επερώτημα, κατά την εφαρμογή της μείωσης του της μείωσης των επαναλήψεων.

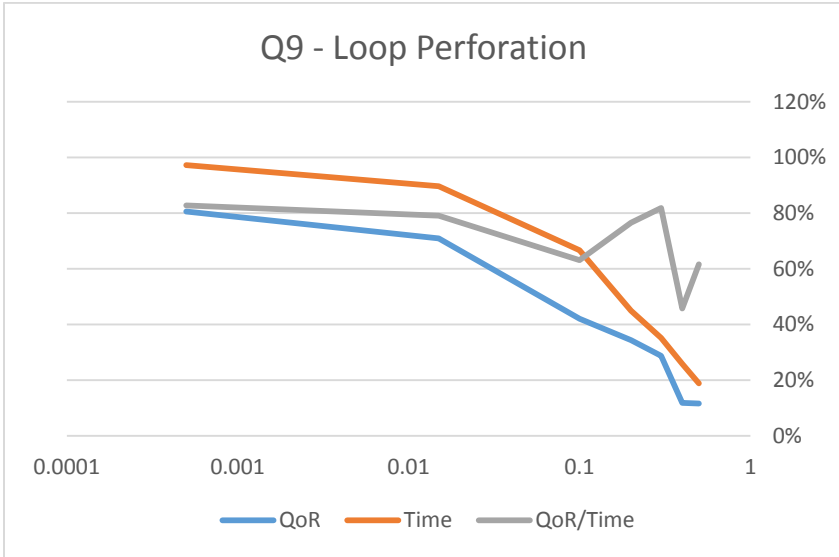
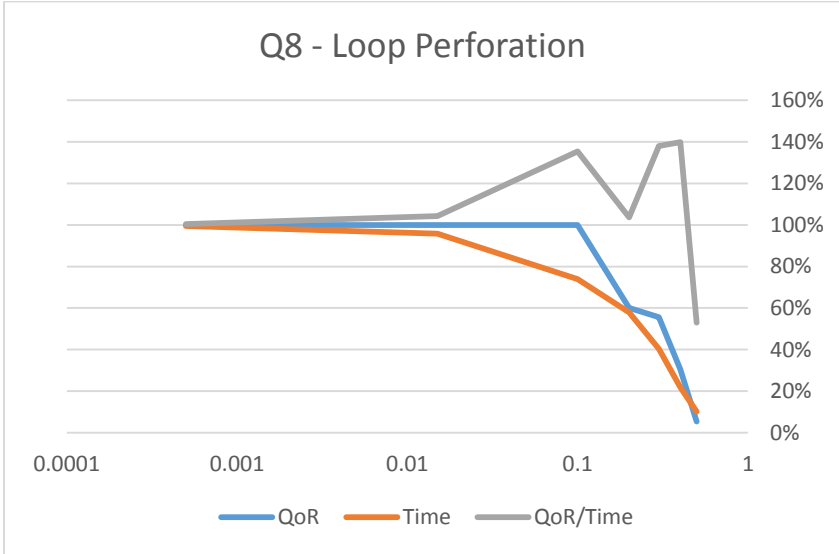
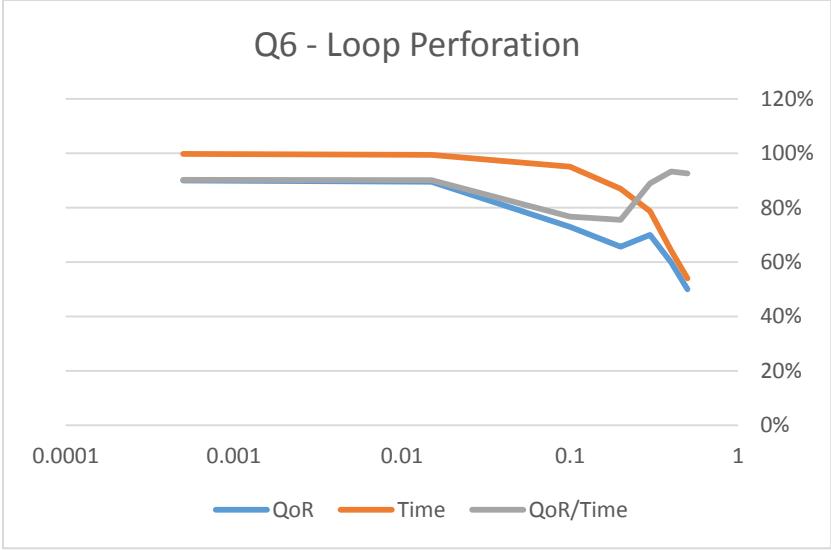


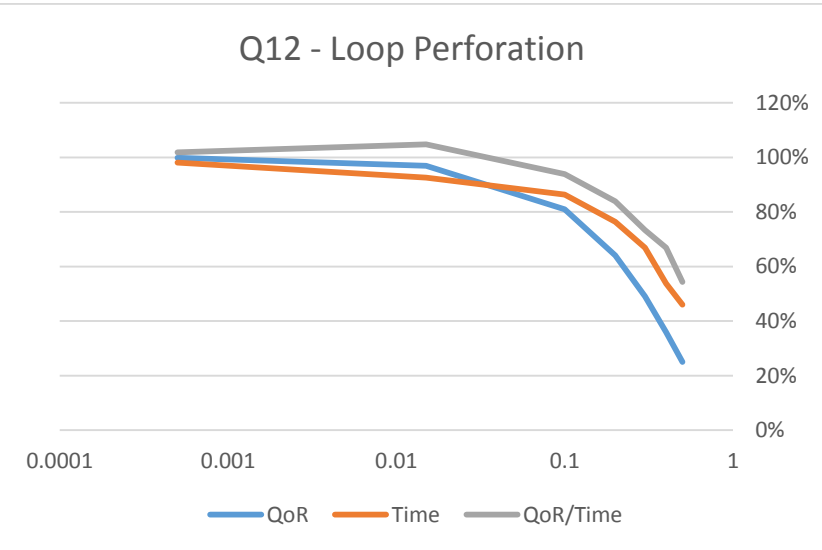
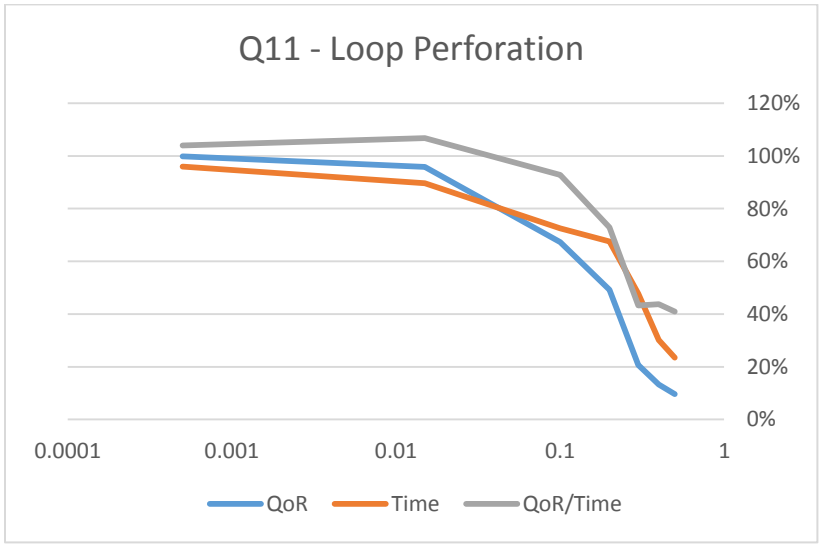
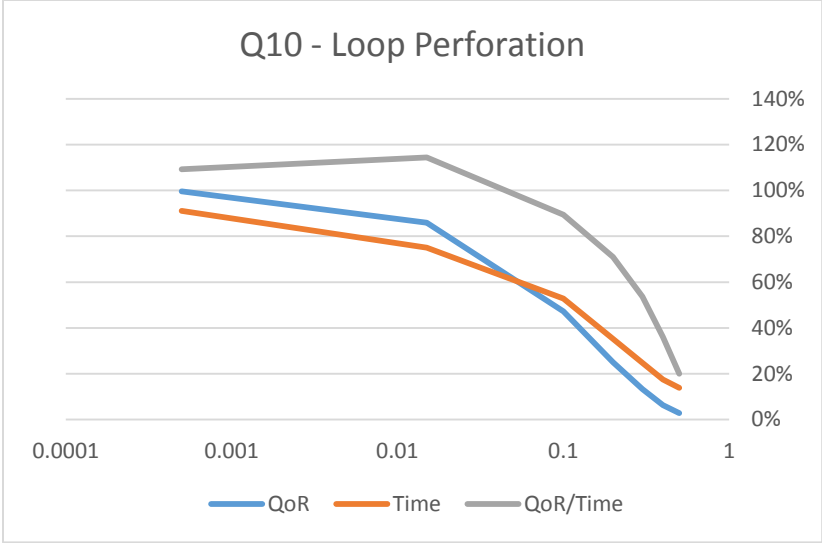
Παράρτημα Γ

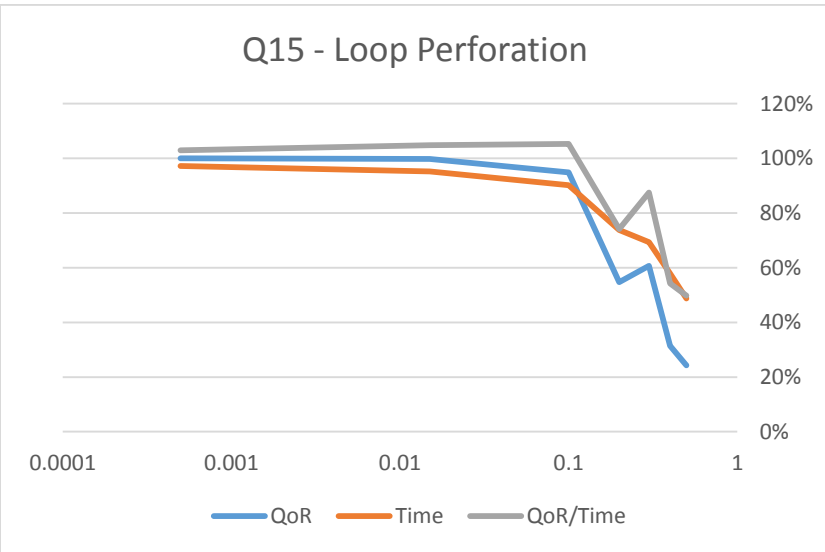
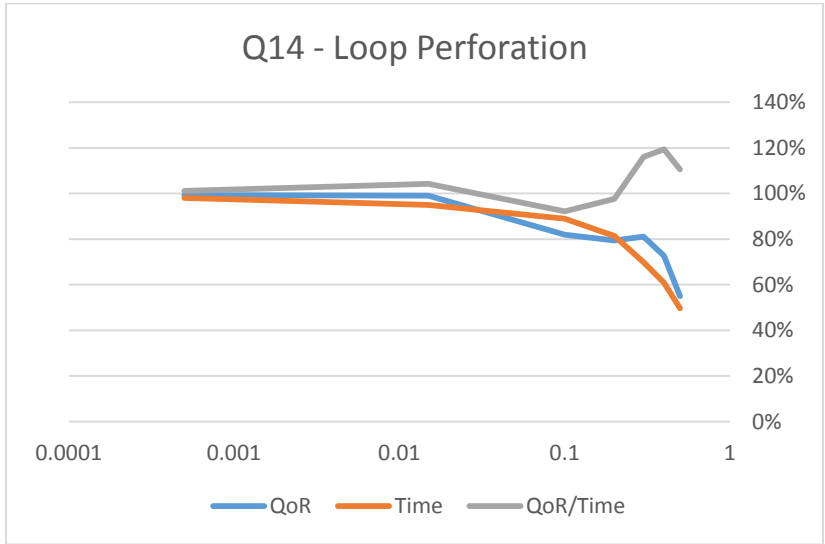
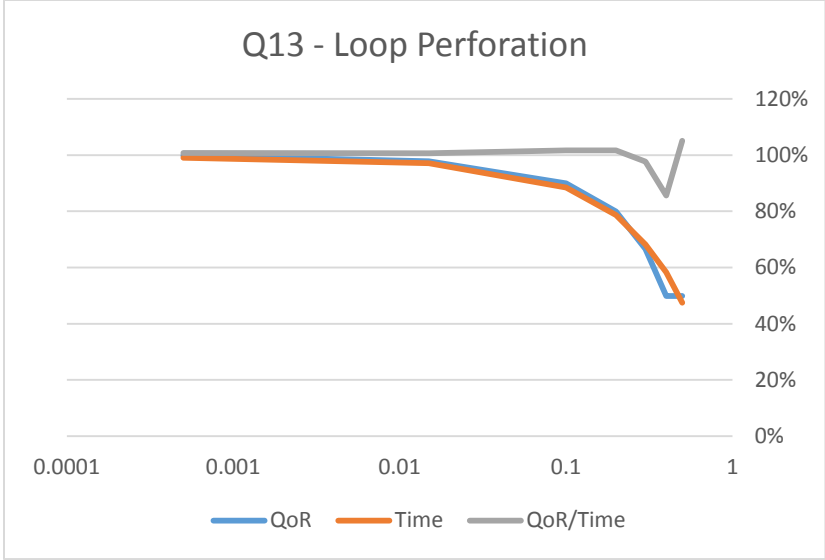
Ακολουθούν αποτελέσματα για την μέθοδο χαλαρότητας με μείωση των επαναλήψεων, τα οποία παρουσιάζουν για κάθε επερώτημα ξεχωριστά, τα 3 χαρακτηριστικά που μετρήθηκαν (QoR, PowerConsumption και Tradeoff) και το πώς αλλάζουν με διαφορετικό ποσοστό χαλαρότητας. Οι τιμές του οριζόντιου άξονα αντιστοιχούν στο ποσοστό σφάλματος (error rate).

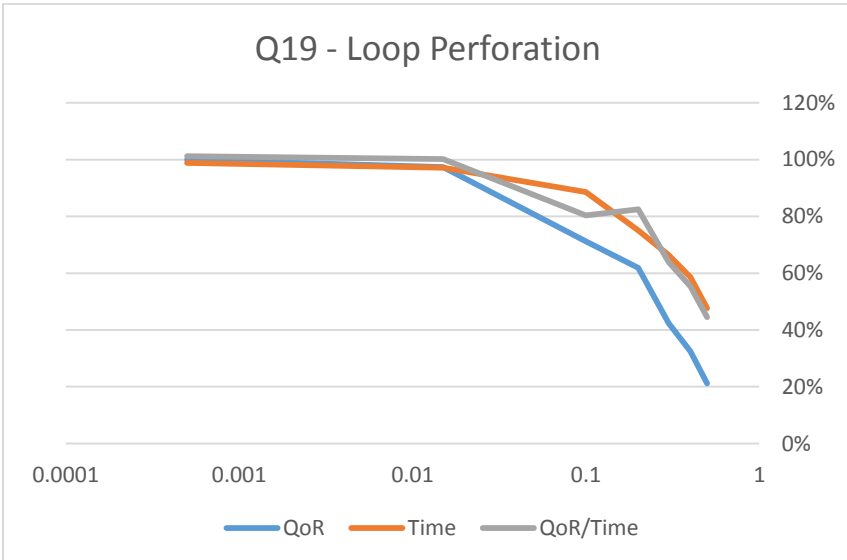
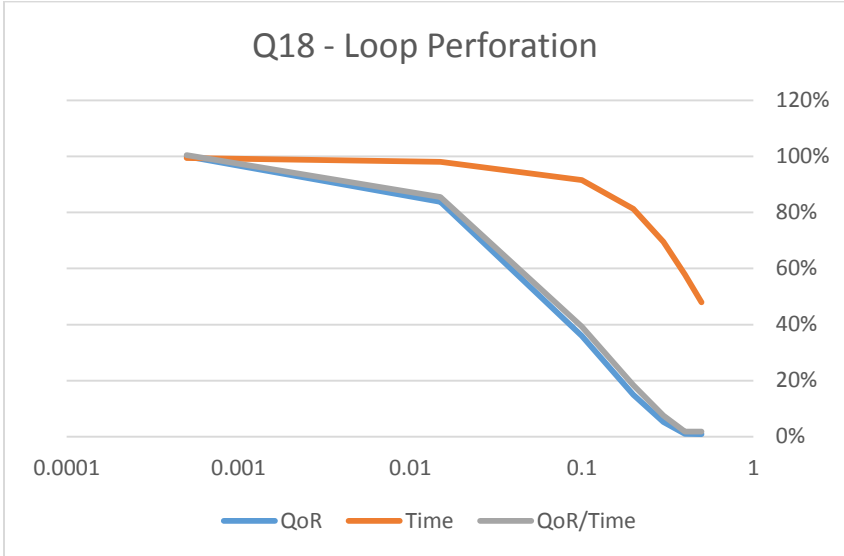
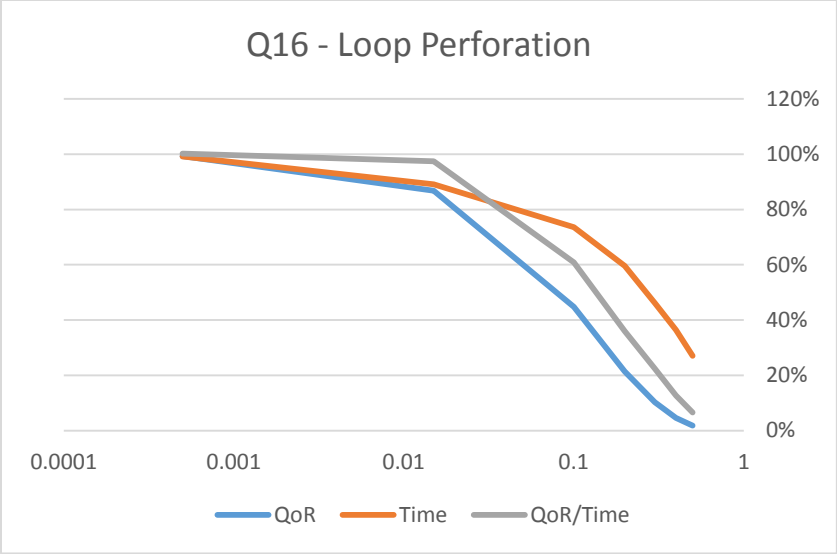


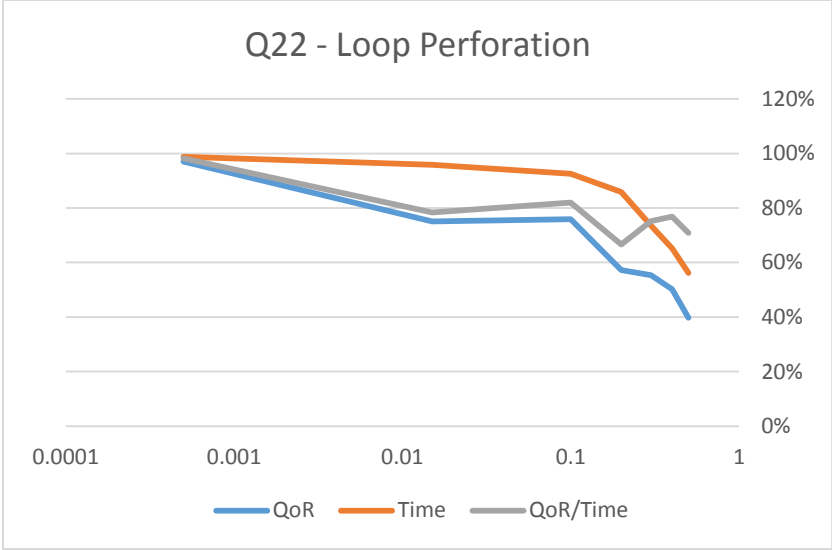












Παράρτημα Δ

Ακολουθως παρουσιάζεται η ποιότητα των αποτελεσμάτων κατά την εφαρμογή της μείωσης των επαναλήψεων με δύο τρόπου επιλογής παράλειψης στοιχείων, τυχαία (Rnd) και με σταθερό ρυθμό (Stride).

