ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Χρήση της Απομακρυσμένης Μνήμης για Αξιοπιστία και Βελτίωση Απόδοσης σε Συστήματα Δοσοληψιών

Σωτήρης Ιωαννίδης

Μεταπτυχιακή Εργασία

Ηράκλειο, Άυγουστος 1996

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Χρήση της Απομακρυσμένης Μνήμης για Αξιοπιστία και Βελτίωση Απόδοσης σε Συστήματα Δοσοληψιών

Εργασία που υποβλήθηκε από τον
Σωτήρη Ιωαννίδη
ως μερική εκπλήρωση των απαιτήσεων
για την απόκτηση
ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΕΙΔΙΚΕΥΣΗΣ

Συγγραφέας:

Σωτήρης Ιωαννίδης
Τμήμα Επιστήμης Υπολογιστών

Εισηγητική Επιτροπή:

Ευάγγελος Μαρκάτος, Επίκουρος Καθηγητής, Επόπτης

Χρήστος Νικολάου, Αναπληρωτής Καθηγητής, Μέλος

Πάνος Τραχανιάς, Επίκουρος Καθηγητής, Μέλος

Δεκτή:

Πάνος Κωνσταντόπουλος, Αναπληρωτής Καθηγητής
Πρόεδρος Επιτροπής Μεταπτυχιακών Σπουδών

Ηράκλειο, Άυγουστος 1996

Χρήση της Απομακρυσμένης Μνήμης για Αξιοπιστία και Βελτίωση Απόδοσης σε Συστήματα Δοσοληψιών

Σωτήρης Ιωαννίδης

Μεταπτυχιακή Εργασία

Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Περίληψη

Οι δοσοληψίες χρησιμοποιούνται σε μία πλειάδα συστημάτων, από εφαρμογές CAD μέχρι συστήματα αρχείων και μεγάλης κλίμακας βάσεις δεδομένων. Αυτό που κάνει τη χρήση τους ελκυστική είναι οι ιδιότητες της ατομικότητας και ανάκτησης που έχουν. Για να μπορούν όμως να υποστηρίξουν τις παραπάνω ιδιότητες βασίζονται στη χρήση μαγνητικών δίσκων για σταθερή αποθήκευση. Δυστυχώς οι μαγνητικοί δίσκοι προκαλούν "μποτιλιάρισμα" στην απόδοση λόγω του υψηλού χρόνου προσπέλασης τους. Αυτό κάνει τις δοσοληψίες ακριβές και περιορίζει τη χρήση τους.

Σε αυτή την εργασία παρουσίαζουμε τρόπους με τους οποίους μπορούμε να χρησιμοποιήσουμε την συνολική κύρια μνήμη των σταθμών εργασίας σε ένα δίκτυο υπολογιστών για να αυξήσουμε την απόδοση σε συστήματα δοσοληψιών. Παρουσιάζουμε τις αλλαγές που κάναμε σε δύο υπάρχοντα συστήματα δοσοληψιών, έτσι ώστε να χρησιμοποιήσουν την απομακρυσμένη μνήμη και να αποφύγουν τη σύγχρονη προσπέλαση στο δίσκο. Επιπλέον παρουσιάζουμε την σχεδίαση και υλοποίηση ενός νέου συστήματος που προσφέρει σταθερή μνήμη χρησιμοποιώντας απομακρυσμένη μνήμη. Για τα παραπάνω συστήματα παραθέτουμε μια σειρά πειραμάτων που έγιναν για να αξιολογηθούν οι ιδέες μας.

Συμπεραίνουμε ότι η χρήση απομακρυσμένης μνήμης βοηθάει την απόδοση σε συστήματα δοσοληψιών και αυτά τα οφέλη θα αυξηθούν στο μέλλον.

Επόπτης: Ευάγγελος Μαρκάτος
Επίκουρος Καθηγητής
Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης.

Using the Remote Main Memory in a Workstation Cluster for Reliability and Performance in Transactional Based Systems

Sotiris Ioannidis

Master of Science Thesis

Department of Computer Science
University of Crete

Abstract

Transactions are used in a wide variety of systems, ranging from CAD applications to file systems and large−scale data bases. What makes them atractive is the properties of atomicity and recoverability. In order to support the above properties transactions rely on the use of magnetic disks as a non−volatile storage medium. Unfortunately magnetic disks become a performance bottleneck because of their high access time. This causes transactions to become expensive and limits their uses.

In this paper we describe how to use the collective main memory in a Network of Workstations (NOW) to improve the performance of transaction−based systems. We present the changes we introduced on two existing transaction based systems, in order for them to use the network memory and therefor avoid synchronous disk I/O. Furthermore we present the design and implementation of a new system that provides non−volatile memory by using network memory. For the above systems we present a variety of experiments that were performed in order to evaluate our ideas.

We conclude that the use of remote main memory benefits the performance of transactions and these benefits will probably increase in the near future.

Advisor: Evangelos Markatos
Assistant Professor
Computer Science Department
University of Crete.

Ευχαριστίες

Ευχαριστώ ιδιαίτερα τον επόπτη καθηγητή μου κ. Ευάγγελο Μαρκάτο για την καθοδήγηση και βοήθεια σε όλη τη διάρκεια της εργασίας μου, αλλά ιδιαίτερα γιατί με έμαθε να "μετράω".

Ευχαριστώ τα μέλη της επιτροπής μου Χρήστο Νικολάου και Πάνο Τραχανιά για τα χρήσιμα σχόλια και τις διορθώσεις τους.

Δεν θα ξεχάσω τον φίλο και συμφοιτητή μου Γιώργο Δραμιτινό για τα χρόνια που δουλέψαμε μαζί και τον Κοσμά Παπαχρήστο για τις τόσες απορίες που μου έλυνε, αφού πρώτα γκρίνιαζε.

Για τον ισόβιο κριτικό μου Μανώλη Μαραζάκη για τα πάντα εύστοχα σχόλια και διορθώσεις στα κείμενά μου, εύχομαι καλό κουράγιο.

Θέλω να ευχαριστήσω ξεχωριστά τους καθηγητές και φίλους Θανάση Φειδά για τα χρόνια που με βοήθησε στις προπτυχιακές μου σπουδές και Σαράντο Καπιδάκη για τις λύσεις που μου πρότεινε σε κάθε είδους απορία μου.

Ευχαριστώ τον αδελφό μου Γιάννη για τις επιστημονικές και μη συζητήσεις μας, που χωρίς αυτόν ίσως δεν μπορούσα να συνεχίσω τις σπουδές μου.

Στη παλιοπαρέα Βασίλη Βιρβίλη, Βίκυ Δανηλάτου, Κώστα Καβουσανό–Καβουσανάκη, Φωτεινή Μαραγκουδάκη και Φλώρα Χρυσού για τα όμορφα χρόνια που περάσαμε, καλή τύχη.

Πρέπει ακόμα να ευχαριστήσω όλα τα μέλη της ομάδας ΠΛΕΙΑΔΕΣ για τη βοήθειά τους καθόλη τη διάρκεια των μεταπτυχιακών σπουδών μου.

Τέλος πρέπει να ευχαριστήσω το Τμήμα Επιστήμης Υπολογιστών του Πανεπιστημίου Κρήτης και το Ινστιτούτο Πληροφορικής του Ιδρύματος Τεχνολογίας και Έρευνας για την υλικοτεχνική και οικονομική υποστήριξη που μου παρείχαν κατά τη διάρκεια των σπουδών μου.

Περιεχόμενα

Κατάλογος Πινάκων

Κατάλογος Σχημάτων

Στην οικογένειά μου

Κεφάλαιο 1

Εισαγωγή

1.1 Το Πρόβλημα

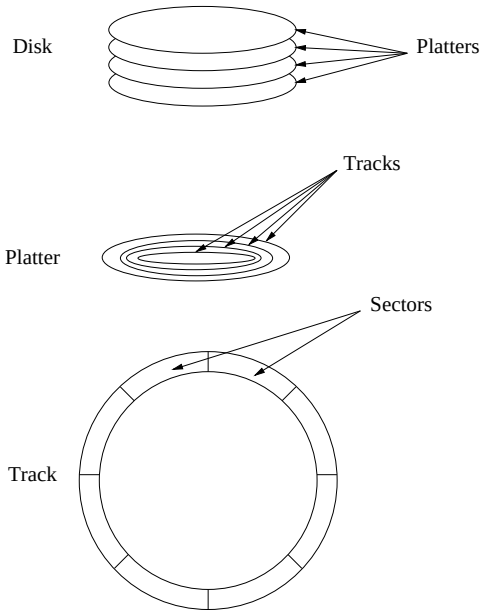
Υπολογιστικά συστήματα όπως συστήματα αρχείων και βάσεις δεδομένων, αλλά και οποιοδήποτε πρόγραμμα χρησιμοποιεί δοσοληψίες για να εξασφαλίσει σταθερότητα στα δεδομένα του, καταφεύγει στη χρήση μαγνητικών δίσκων. Οι μαγνητικοί δίσκοι προσφέρουν ευστάθεια στα δεδομένα τα οποία θέλουμε να προστατεύσουμε από καταστροφές. Η ευστάθεια που μας δίνουν έχει όμως και το αντίτιμό της, και αυτό γιατί σε αντίθεση με τους μικροεπεξεργαστές των οποίων η υπολογιστική ισχύ αυξάνεται συνεχώς με μέσο ρυθμό μεγαλύτερο από 35% το χρόνο κατά τα τελευταία 15 χρόνια, οι δίσκοι δεν ακολουθούν ανάλογη βελτίωση σε απόδοση [HP90].

Ο λόγος για τον οποίο η προσπέλαση στο δίσκο είναι τόσο ακριβή οφείλεται στη δομή του που φαίνεται στο σχήμα 1.1. Για την προσπέλαση δεδομένων πρέπει αρχικά οι κεφαλές του δίσκου να μετακινηθούν μέχρι να φτάσουν στο σωστό ίχνος. Ο χρόνος αυτός ονομάζεται χρόνος αναζήτησης. Επίσης θα πρέπει να περιστραφεί αρκετά ο δίσκος ώστε ο τομέας που περιέχει τα ζητούμενα δεδομένα να βρεθεί κάτω από την κεφαλή (χρόνος περιστροφής). Στη συνέχεια τα δεδομένα μεταφέρονται (γράφονται/διαβάζονται) από την κεφαλή. Επειδή ο δίσκος αποτελείται από μηχανικά μέρη, τόσο ο χρόνος αναζήτησης όσο και ο χρόνος περιστροφής είναι αρκετά σημαντικοί (της τάξης των αρκετών ms).

Εξαιτίας αυτού η απόδοση εφαρμογών που βασίζονται στη χρήση μαγνητικών δίσκων για την λειτουργία τους δεν βελτιώνεται με υψηλούς ρυθμούς όπως θα περίμενε κανείς.

1.2 Διάφορες Λύσεις

Για να επιταχύνουν όσο γίνεται τις διαδικασίες εισόδου−εξόδου τα συστήματα αρχείων και οι βάσεις δεδομένων κρατούν όσο το δυνατόν περισσότερα δεδομένα στη μνήμη. Χρησιμοποιούν μία ιεραρχία μνήμης, η οποία φαίνεται στο σχήμα 1.2. Αυτή η ιεραρχία μνήμης αποτελείται από την κρυφή μνήμη, την κύρια μνήμη και το δίσκο, τα χαρακτηριστικά των οποίων φαίνονται στον πίνακα 1.1. Η κρυφή μνήμη είναι μια πολύ γρήγορη στατική μνήμη τυχαίας προσπέλασης με χρόνο προσπέλασης ίσο τυπικά με λίγους κύκλους ρολογιού, αλλά το μέγεθός της είναι πολύ μικρό λόγω του μεγάλου της κόστους. Η κύρια μνήμη χρησιμοποιείται για την αποθήκευση δεδομένων που δε χωρούν στην κρυφή μνήμη. Είναι μια δυναμική μνήμη τυχαίας προσπέλασης



Σχήμα 1.1: **Η δομή του δίσκου.**

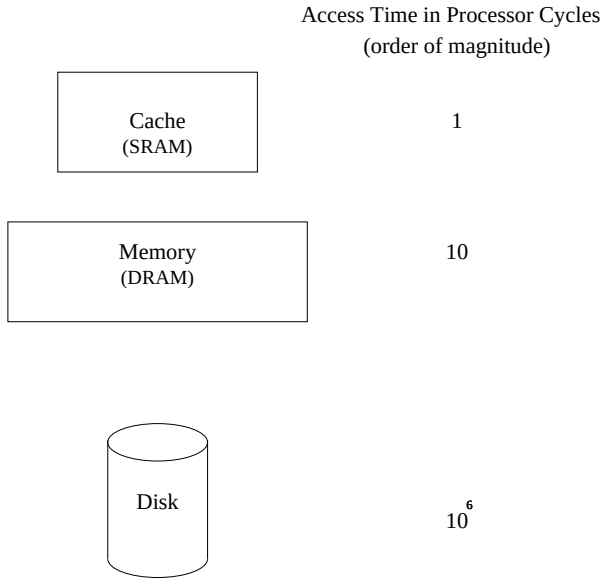
Τύπος μνήμης	Χρόνος πρόσβασης (σε κύκλους ρολογιού)	Χώρος αποθήκευσης (σε MB)
Κρυφή μνήμη	1	0.1 – 1
Κύρια μνήμη	10	10 – 100
Μνήμη Flash	$10/10^3$ (ανάγνωση/εγγραφή)	
Δίσκος	10^6	100 – 1000

Πίνακας 1.1: **Χαρακτηριστικά των επιπέδων της ιεραρχίας μνήμης.**

με χρόνο προσπέλασης της τάξης των λίγων δεκάδων κύκλων ρολογιού και μέγεθος λίγων δεκάδων MB. Τέλος ο δίσκος χρησιμοποιείται για την αποθήκευση δεδομένων που δεν χωρούν ούτε στην κύρια μνήμη. Το μέγεθός του είναι τυπικά ίσο με λίγες εκατοντάδες MB και ο χρόνος προσπέλασής του της τάξης των αρκετών ms, δηλαδή της τάξης των εκατομμυρίων κύκλων ρολογιού.

Μία τέτοια ιεραρχία βελτιώνει κατά πολύ τις αναγνώσεις δεδομένων αλλά όχι και τις εγγραφές [BHK+91]. Αυτό συμβαίνει όχι επειδή δεν έχουμε αρκετή μνήμη, αλλά κατά κύριο λόγο γιατί θέλουμε να γράφουμε τα δεδομένα μας στο δίσκο για να τα προστατεύσουμε από τυχούσες καταστροφές στο σύστημά μας. Για παράδειγμα, στο Σύστημα Αρχείων Sprite [NWO88], τα "βρώμικα" δεδομένα που βρίσκονται στην κύρια μνήμη γράφονται περιοδικά (κάθε 30 δευτερόλεπτα), στο δίσκο. Σε βάσεις δεδομένων, και γενικά σε όποιο σύστημα χρησιμοποιεί δοσοληψίες, αυτό συμβαίνει πολύ πιο συχνά (κάθε φορά που θέλουμε να δεσμεύσουμε μία δοσοληψία).

Παρατηρούμε ότι, ενώ η κύρια μνήμη παρέχει δύο τάξεις μεγέθους μεγαλύτερο αποθηκευτικό χώρο από την κρυφή μνήμη αυξάνοντας τον χρόνο προσπέλασης στα δεδομένα κατά μια τάξη μεγέθους, ο δίσκος παρέχει μόλις μια τάξη μεγέθους μεγα–



Σχήμα 1.2: **Η δομή της ιεραρχίας μνήμης.**

Παράμετρος Υλικού	Ετήσια Βελτίωση
Καθυστέρηση Δίσκου	10%
Ρυθμός Μεταγωγής Δεδομένων Δίσκου	20%
Κόστος Δίσκου	100%
Καθυστέρηση Δικτύου	20%
Ρυθμός Μεταγωγής Δεδομένων Δικτύου	45%
Απόδοση Επεξεργαστή	55%
Ρυθμός Μεταγωγής Δεδομένων Μνήμης	40%
Κόστος Μνήμης	45%

Πίνακας 1.2: **Τάσεις στην αρχιτεκτονική υπολογιστών.**

λύτερο αποθηκευτικό χώρο από την κύρια μνήμη αυξάνοντας το χρόνο προσπέλασης στα δεδομένα κατά πέντε τάξεις μεγέθους. Είναι λοιπόν προφανές ότι εφαρμογές που αναγκάζονται να προσπελάσουν κάποια από τα δεδομένα τους από το δίσκο, υποφέρουν από δραματική αύξηση του χρόνου εκτέλεσής τους.

Μερικές από τις τάσεις που επικρατούν σήμερα στην αρχιτεκτονική υπολογιστών εμφανίζονται στο πίνακα 1.2 [Dah95]. Πιο συγκεκριμένα:

- Ο κύκλος ρολογιού μειώνεται ραγδαία [Dah95, HP90].
- Ο χρόνος προσπέλασης στο δίσκο μειώνεται με πολύ αργό ρυθμό [Dah95, HP90].

Αυτό σημαίνει ότι όποιες εφαρμογές χρειάζεται να χρησιμοποιήσουν δίσκο, δεδομένου ότι ο χρόνος προσπέλασης στο δίσκο, μετρημένος σε κύκλους ρολογιού, θα αυξάνεται συνεχώς, θα αντιμετωπίζουν ολοένα και σοβαρότερο πρόβλημα.

Για την αντιμετώπιση της μεγάλης καθυστέρησης της προσπέλασης στο δίσκο χρησιμοποιούνται διάφορες τεχνικές:

- *Χρησιμοποιούνται σελίδες μεγάλου μεγέθους ώστε ο χρόνος αναζήτησης και περιστροφής για τη μεταφορά της σελίδας από/προς το δίσκο να μοιράζεται σε πολλές προσπελάσεις στη σελίδα.* Η τεχνική αυτή ωφελεί προγράμματα που κάνουν πολλές προσπελάσεις ανά σελίδα (π.χ. εφαρμογές που προσπελαύνουν σειριακά μεγάλους πίνακες). Αυτό όμως δεν βοηθάει στην περίπτωση των δοσοληψιών γιατί τις πιο πολλές φορές τα δεδομένα που μεταβάλλουν έχουν πολύ μικρό μέγεθος και πρέπει να γραφτούν αμέσως στο δίσκο.
- *Χρησιμοποιούνται RAIDs [CLG+94, PGK88].* Το κέρδος από τη χρήση τους είναι σημαντικό αφού μπορεί να γίνεται εγγραφή πολλών σελίδων παράλληλα. Όμως κατά την ανάγνωση της σελίδας, η καθυστέρηση που οφείλεται στο χρόνο αναζήτησης και περιστροφής παραμένει. Επίσης το κόστος των RAIDs είναι υψηλό.

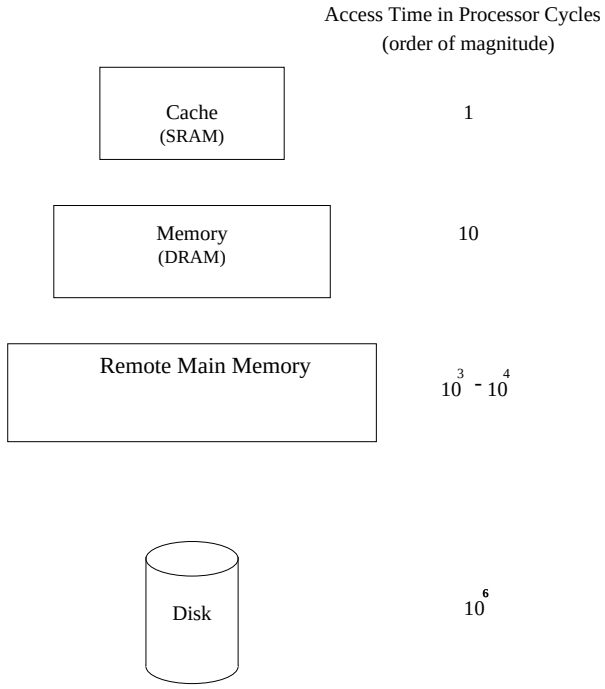
Οι τεχνικές αυτές δεν αντιμετωπίζουν ριζικά το πρόβλημα, απλά προσπαθούν να μειώσουν τις επιπτώσεις του.

Μια άλλη τεχνική η οποία χρησιμοποιείται για να αποσυμφορήσουμε τις εφαρμογές που χρειάζονται σταθερότητα στα δεδομένα τους, είναι μνήμη η οποία μπορεί να διατηρήσει τα δεδομένα της σε περίπτωση που έχουμε πτώση του συστήματος [BAD+92, WZ95, DCK+94]. Με αυτό τον τρόπο προσπαθούμε να επιταχύνουμε τις σύγχρονες εγγραφές. Οι εφαρμογές που θέλουν να γράψουν δεδομένα στο δίσκο για ασφάλεια, τα γράφουν στη σταθερή μνήμη και συνεχίζουν την εκτέλεσή τους. Ταυτόχρονα, τα δεδομένα από τη σταθερή μνήμη γράφονται ασύγχρονα στο δίσκο. Σε περίπτωση που το σύστημα πέσει τα δεδομένα θα είναι είτε στο δίσκο είτε στη σταθερή μνήμη. Η σταθερή μνήμη έχει πολλές μορφές, αλλά πιο συχνά αποτελείται από μνήμη SRAM με εφεδρικό σύστημα μπαταρίας χαμηλής ισχύος.

Το πιο βασικό μειονέκτημα αυτών των προϊόντων είναι το υψηλό κόστος τους, κοστίζουν τέσσερις με δέκα φορές πιο ακριβά από το ίδιο μέγεθος κανονικής μνήμης DRAM [BAD+92]. Για να ελαττώσουν το κόστος, οι ερευνητές έχουν προτείνει τη χρήση μνήμης FLASH (EEPROM) για τη κατασκευή σταθερής μνήμης [WZ95]. Τα ολοκληρωμένα κυκλώματα της μνήμης FLASH διατηρούν τα δεδομένα τους ακόμα και στην περίπτωση πτώσης της τροφοδοσίας ρεύματος, έχουν συγκρίσιμη τιμή με τη μνήμη DRAM ίδιου μεγέθους, αλλά έχουν μεγάλη καθυστέρηση εγγραφής. Συστήματα τα οποία θέλουν σταθερή μνήμη, συνήθως έχουν μια μικρή ποσότητα από μνήμη SRAM με βοηθητικό σύστημα τροφοδοσίας από μπαταρία, για να έχουν γρήγορες εγγραφές και μία μεγάλη ποσότητα από μνήμη τύπου FLASH. Παρόλα αυτά τέτοια συστήματα παραμένουν ακριβά και χρησιμοποιούνται συνήθως από ακριβούς σταθμούς εργασίας.

1.3 Η Προτεινόμενη Λύση

Σε αυτή την εργασία μελετάμε έναν άλλο τρόπο με τον οποίο μπορούμε να προσφέρουμε αυξημένη απόδοση και αξιοπιστία σε συστήματα δοσοληψιών. Η ιδέα μας είναι να χρησιμοποιήσουμε το σύνολο της κύριας μνήμης των σταθμών εργασίας ενός τοπικού δικτύου σαν ένα νέο επίπεδο στην ιεραρχία της μνήμης, σχήμα 1.3, που στο εξής θα ονομάζεται *απομακρυσμένη μνήμη*, για να προσφέρουμε αξιοπιστία. Η μέθοδός μας θα είναι σε λογισμικό και δεν θα χρειάζεται επιπλέον υλικό. Για να προσφέρουμε



Σχήμα 1.3: **Η νέα δομή της ιεραρχίας μνήμης.**

σταθερότητα και αξιοπιστία στα δεδομένα μας, χρησιμοποιούμε μερικές ανεξάρτητες μη σταθερές αποθήκες δεδομένων (τις μνήμες των διαφόρων σταθμών εργασίας). Τα δεδομένα που κανονικά θα γράφαμε σε σταθερή μνήμη τώρα γράφονται σε όλες τις μη σταθερές μνήμες. Με αυτό τον τρόπο για να χάσουμε δεδομένα θα πρέπει να χάσουμε *όλα* τα αντίγραφα, πράγμα που είναι εξαιρετικά απίθανο. Μία εφαρμογή η οποία θέλει να γράψει δεδομένα σε σταθερή μνήμη, τα γράφει στη μνήμη του σταθμού εργασίας στον οποίο τρέχει και ταυτόχρονα στη μνήμη ενός άλλου σταθμού. Μόλις η εγγραφή γίνει στην απομακρυσμένη μνήμη η εφαρμογή μπορεί να συνεχίσει. Όσο η εφαρμογή τρέχει, τα δεδομένα γράφονται στο δίσκο ασύγχρονα. Αν κάποιος σταθμός πέσει πριν τα κρίσιμα δεδομένα γραφτούν στο δίσκο, τα δεδομένα δεν έχουν χαθεί γιατί βρίσκονται στην απομακρυσμένη μνήμη και μπορούμε να τα διαβάσουμε από εκεί όταν η εφαρμογή μας ξανατρέξει.

Αυτή η έννοια της παραπάνω πληροφορίας έχει εμφανιστεί και στο παρελθόν για να μας παρέχει αξιοπιστία διαφόρων βαθμών, από RAIDs μέχρι γεωγραφικά κατανεμημένες βάσεις δεδομένων. Η ιδέα όμως να χρησιμοποιήσουμε τέτοιες μεθόδους για αύξηση της απόδοσης και της αξιοπιστίας σε μία υλοποίηση μόνο από λογισμικό είναι ιδιαίτερα ενδιαφέρουσα σήμερα λόγω των σύγχρονων τάσεων στην αρχιτεκτονική των διασυνδετικών δικτύων υπολογιστών:

- *Ο ρυθμός μεταγωγής δεδομένων στα Τοπικά Δίκτυα έχει αυξηθεί με γρήγορους ρυθμούς:* τα σύγχρονα δίκτυα υπολογιστών τα τύπου FDDI [Ame87] και ATM [New94] δεν είναι πλέον σπάνια. Τέτοιες συνδέσεις προσφέρουν ρυθμούς μεταγωγής της τάξης των 100–155 Mbits το δευτερόλεπτο. Δίκτυα ATM των 622 Mbits το δευτερόλεπτο έχουν αρχίσει να εμφανίζονται, και ταυτόχρονα αναπτύσσονται δίκτυα της τάξης

των Gigabits το δευτερόλεπτο. Έτσι η μεταγωγή δεδομένων στο δίκτυο μπορεί πλέον να συγκριθεί με τη μεταγωγή δεδομένων στην κύρια μνήμη, πράγμα που σημαίνει ότι οι μεταφορές δεδομένων μεταξύ σταθμών εργασίας μπορούν να πραγματοποιηθούν με ρυθμούς αντίστοιχους των μεταφορών από μνήμη−σε−μνήμη μέσα στον ίδιο σταθμό εργασίας, και φυσικά πολύ πιο γρήγορους από τη μεταφορά από μνήμη−σε−δίσκο. Γενικά ο ρυθμός μεταγωγής δεδομένων των δικτύων αυξάνεται κατά 45% το χρόνο [Dah95].

- *Η καθυστέρηση προσπέλασης στα Τοπικά Δίκτυα έχει μειωθεί σημαντικά:* πολλά δίκτυα υπολογιστών προσφέρουν καθυστέρηση προσπέλασης μόνο μερικών μικροδευτερολέπτων [MK96, SL91, BJMW95, Del88, BCF+95, Gil95, BLA+94, JLGS90]. Βελτιστοποιημένα πρωτόκολλα επικοινωνίας [ABvE95] προσφέρουν καθυστέρηση προσπέλασης μερικών δεκάδων μικροδευτερολέπτων ακόμα και πάνω από δίκτυα όπως ATM και Fast Ethernet. Έτσι η μεταφορά μικρών ποσοτήτων πληροφορίας μεταξύ υπολογιστών χρειάζεται μόνο μερικές δεκάδες μικροδευτερόλεπτα, ενώ αντίθετα μεταφορές δίσκου−μνήμης χρειάζονται μερικά χιλιοστά του δευτερολέπτου, ακόμα και για πολύ μικρές ποσότητες πληροφορίας. Η καθυστέρηση προσπέλασης των δικτύων μειώνεται κατά 20% το χρόνο [Dah95].

Επίσης το ποσό της ελεύθερης μνήμης στο δίκτυο είναι αρκετό ώστε να μπορεί να καλύψει τις ανάγκες συστημάτων δοσοληψιών. Αυτή η εκτίμηση επιβεβαιώνεται από το σχήμα 1.4 που δείχνει το συνολικό ποσό της ελεύθερης κύριας μνήμης των μηχανημάτων του τοπικού δικτύου της ομάδας Αρχιτεκτονικής και Συστημάτων VLSI του ΙΠ−ΙΤΕ κατά τη διάρκεια μιας εβδομάδας. Όπως αναμενόταν το ποσό της ελεύθερης μνήμης είναι πολύ μεγάλο κατά τις μη εργάσιμες ώρες και το Σαββατοκύριακο. Ακόμη όμως και κατά τις εργάσιμες ώρες των ημερών της εβδομάδας, το ποσό της ελεύθερης μνήμης είναι αρκετά σημαντικό και σπάνια μικρότερο από 400 MB. Έτσι, αφενός μπορούμε να πετύχουμε μεγαλύτερη χρησιμοποίηση της *υπάρχουσας* μνήμης και αφετέρου μπορούμε πάντα να καταφεύγουμε στον τοπικό δίσκο όποτε η ελεύθερη μνήμη δεν επαρκεί.

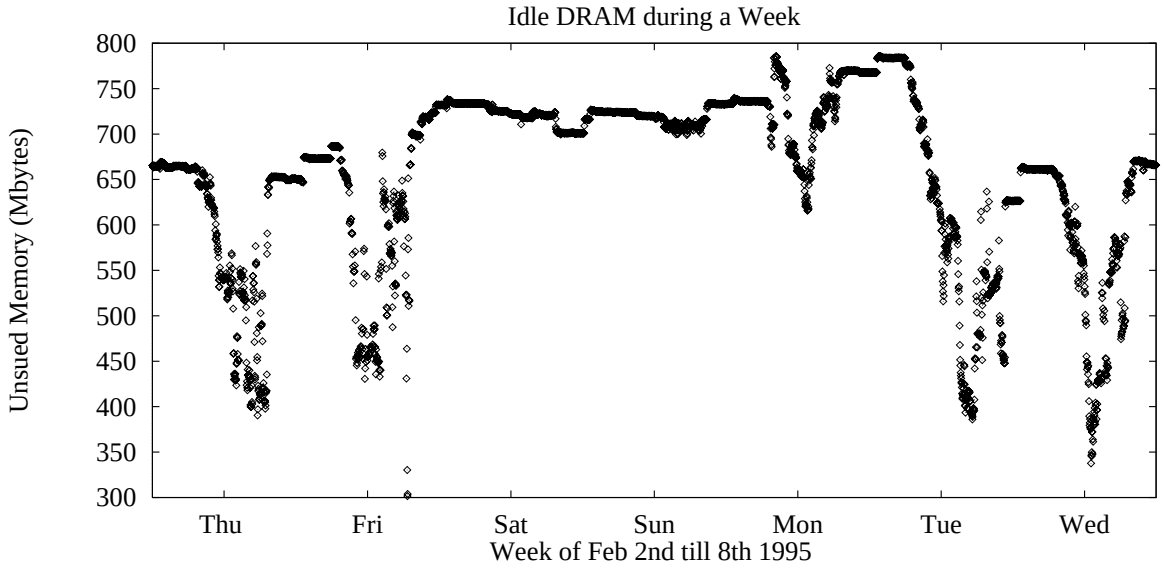
1.4 Η συνεισφορά αυτής της εργασίας

Η συνεισφορά της εργασίας αυτής σε σχέση με προηγούμενες εργασίες στον ίδιο τομέα συνίσταται στα παρακάτω:

- Παρουσιάζουμε τις μετατροπές που κάναμε σε υπάρχοντα συστήματα δοσοληψιών έτσι ώστε να χρησιμοποιούν απομακρυσμένη μνήμη για αποθήκευση δεδομένων. Παράλληλα με τις αλλαγές που κάναμε φροντίσαμε να περιλάβουμε αλγορίθμους για να πετύχουμε αξιοπιστία σε περίπτωση πτώσης ενός σταθμού εργασίας, καθώς αυτή είναι η πιο πιθανή περίπτωση.

- Παρουσιάζουμε μια βιβλιοθήκη υποστήριξης, με την οποία προσθέτουμε αξιοπιστία στην απομακρυσμένη μνήμη. Με αυτό τον τρόπο εφαρμογές μπορούν να την χρησιμοποιήσουν χωρίς τον φόβο απώλειας χρήσιμων δεδομένων. Φροντίσαμε η σχεδίαση μας να προσφέρει απλότητα και χαμηλό κόστος.

- Αποδεικνύουμε ότι η αποθήκευση δεδομένων στην απομακρυσμένη μνήμη από εφαρμογές που χρησιμοποιούν δοσοληψίες, προσφέρει σημαντική βελτίωση στο



Σχήμα 1.4: **Η ελεύθερη μνήμη ενός τοπικού δικτύου 16 μηχανών με ολική μνήμη 800 MB κατά τη διάρκεια μιας εβδομάδας.**

χρόνο εκτέλεσης τους χωρίς να χάνει σε αξιοπιστία σε σχέση με τη χρήση τοπικού δίσκου.

Τα υπόλοιπα κεφάλαια της εργασίας αυτής περιγράφουν τα διάφορα συστήματα δοσοληψιών που μετατρέψαμε, τη σχεδίαση, την υλοποίηση και την αξιολόγηση των συστημάτων που χρησιμοποιούν την ελεύθερη μνήμη του δικτύου σε σχέση με τα αρχικά σύστημα. Παρουσιάζεται μία βιβλιοθήκη υποστήριξης που αναπτύξαμε. Παρουσιάζονται διάφορα προβλήματα που προέκυψαν, όπως η ανάγκη ανθεκτικότητας του συστήματος σε βλάβες, και οι λύσεις που εφαρμόστηκαν. Επίσης παρουσιάζονται διάφορες πολιτικές αξιοπιστίας. Παρουσιάζεται η απόδοση των συστημάτων σε δίκτυα διαφόρων ρυθμών μεταγωγής δεδομένων. Τέλος, περιγράφονται διάφορα σχετικά συστήματα και παρουσιάζονται τα συμπεράσματα που προέκυψαν από την εργασία αυτή.

Κεφάλαιο 2

Σχεδίαση

2.1 Στόχοι Σχεδίασης

Για την υλοποίηση του συστήματός μας θέσαμε τους παρακάτω στόχους:

1. *Θέλουμε να έχουμε όσο το δυνατόν μεγαλύτερη ευελιξία.*
2. *Εφαρμογές που τρέχουν πάνω από το σύστημα που μετατρέπουμε πρέπει να συνεχίσουν να τρέχουν.*
3. *Είναι ανάγκη να δοκιμάσουμε τις ιδέες μας σε παραπάνω από ένα συστήματα.*
4. *Θέλουμε να κάνουμε όσο το δυνατόν λιγότερες και απλές αλλαγές στα συστήματα που μετατρέπουμε.*

Η ανάγκη για ευελιξία μας κάνει να δουλέψουμε σε επίπεδο χρήστη, έξω από το λειτουργικό σύστημα. Με αυτό τον τρόπο το σύστημά μας θα μπορεί να τρέχει πάνω από διάφορες μηχανές χωρίς να είναι ανάγκη να αλλάξουμε τον πυρήνα του λειτουργικού. Κάτι τέτοιο μας προσφέρει μεταφερσιμότητα.

Κάθε αλλαγή που θα κάνουμε σε υπάρχοντα συστήματα δεν πρέπει να επηρεάσει τις εφαρμογές που τα χρησιμοποιούν. Η ανάγκη για κάτι τέτοιο είναι αυτονόητη. Αν μετατρέποντας ένα σύστημα αλλάξουμε τη διασύνδεσή του με τα προγράμματα που το χρησιμοποιούν τότε θα ήταν ανάγκη να αλλάξουμε και όλες τις εφαρμογές. Κάτι τέτοιο θα ήταν αρκετά επίπονο, αν όχι αδύνατο !

Επειδή θέλουμε να έχουμε μία σφαιρική εικόνα για το πώς βελτιώνεται η απόδοση συστημάτων δοσοληψιών από τις ιδέες που θέλουμε να εφαρμόσουμε, κρίναμε σκόπιμο να τις υλοποιήσουμε σε περισσότερα από ένα συστήματα, τα οποία μάλιστα έχουν και διαφορετική αρχιτεκτονική. Με αυτό τον τρόπο θα μπορέσουμε να βγάλουμε καλύτερα συμπεράσματα για την εφαρμοσιμότητα των ιδεών μας.

Τέλος θέλουμε να κρατήσουμε τα συστήματά μας απλά. Αυτό το θέλουμε για δύο λόγους, πρώτον για να δείξουμε ότι με απλές αλλαγές σε υπάρχοντα συστήματα μπορούμε να πετύχουμε αυξημένη απόδοση, και δεύτερον γιατί δεν θέλουμε να περιπλέξουμε τον κώδικα των συστημάτων που μελετάμε.

Έχοντας όλα αυτά υπ'όψιν επιλέξαμε δύο συστήματα, τον Exodus [Cea90, Gro93], ένα διαχειριστή αποθήκευσης τύπου πελάτη−εξυπηρετητή, και το RVM [SMK+93], μία βιβλιοθήκη που παρέχει δυνατότητα δοσοληψιών. Πέρα από αυτά τα δύο συστή−ματα αναπτύξαμε και ένα δικό μας, το NVRAM, δηλαδή Network Volatile RAM, το οποίο είναι μία βιβλιοθήκη που προσφέρει σταθερή μνήμη.

Στα συστήματά μας λοιπόν πρέπει να δώσουμε τη δυνατότητα να χρησιμοποιούν την απομακρυσμένη μνήμη. Αυτό θα το καταφέρουμε αλλάζοντας τον κώδικά τους έτσι ώστε αντί να στέλνουν δεδομένα στο δίσκο να τα στέλνουν σε άλλους σταθμούς εργασίας πάνω από το δίκτυο. Σε αυτούς τους σταθμούς εργασίας θα τρέχει ένας δαίμονας, ο *εξυπηρετητής μνήμης*. Η λειτουργία του θα είναι να δέχεται δεδομένα από τα συστήματα δοσοληψιών και να τα αποθηκεύει στη μνήμη του.

Οι επόμενες παράγραφοι περιγράφουν τα συστήματα που χρησιμοποιήσαμε, τον εξυπηρετητή μνήμης και το περιβάλλον ανάπτυξης των συστημάτων μας.

2.2 Περιβάλλον Ανάπτυξης και Πειραμάτων

Για τα πειράματά μας χρησιμοποιήσαμε διάφορους σταθμούς εργασίας και διάφορα διασυνδετικά δίκτυα.

2.2.1 Exodus

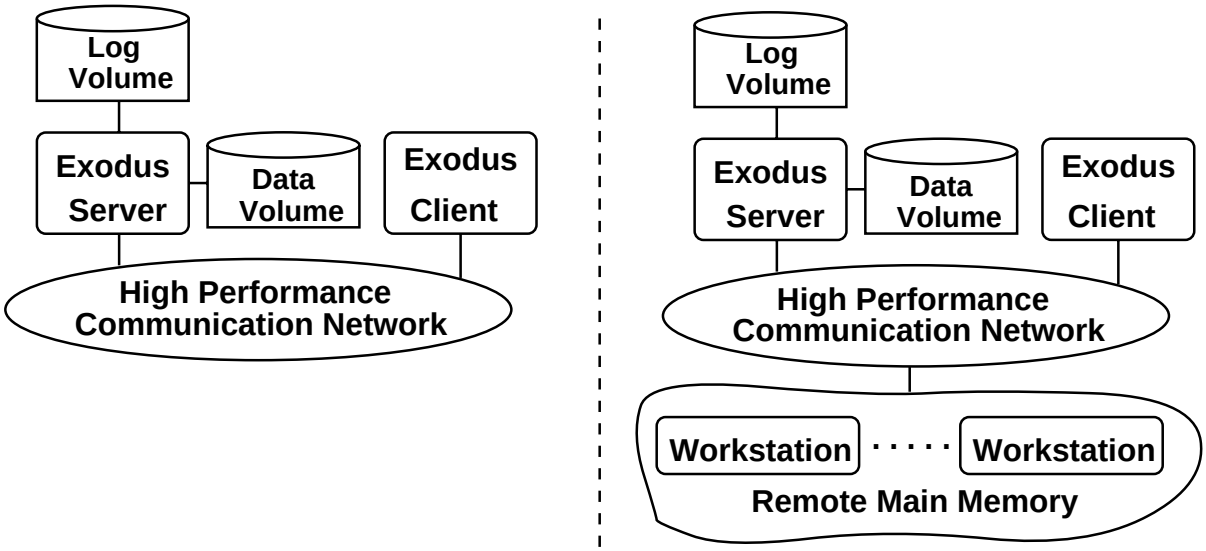
Οι μετρήσεις για την περίπτωση του αρχικού συστήματος που χρησιμοποιεί τον δίσκο για αξιοπιστία έγιναν σε ένα Sun SS10 με 32MB φυσικής μνήμης έχοντας ένα SUN0424 τοπικό δίσκο. Το ημερολόγιο και τα δεδομένα φυλάσσονται σαν αρχεία UNIX στο δίσκο. Οι μετρήσεις για το Ethernet [IEE85] και το FDDI έγιναν σε περιόδους χαμηλού φόρτου. Τα πειράματα στο Ethernet έγιναν σε τρία Sun SS10 με 32, 48 και 424 MB φυσικής μνήμης αντίστοιχα. Τα πειράματα στο FDDI έγιναν σε δύο Sun SparkServers 690MP με 128 MB φυσικής μνήμης το καθένα και σε ένα Sun SS10 με 64 MB φυσικής μνήμης. Ο Exodus έτρεχε στο ένα σταθμό και μία συλλογή από εξυπηρετητές μνήμης στα άλλα δύο.

2.2.2 RVM

Το πειραματικό μας περιβάλλον αυτή τη φορά αποτελείτο από οκτώ σταθμούς εργασίας DEC Alpha 2000 [Dig93] στα 233 MHz με 128 MB φυσικής μνήμης το καθένα. Οι σταθμοί συνδέονται με FDDI και με Ethernet. Επιπλέον ο κάθε σταθμός είχε και 6 GB τοπικό δίσκο. Η παραπάνω ομάδα υπολογιστών βρίσκεται στη Νορβηγία, στο Laboratorium for parallell prosessering – Parallab (http://www.ii.uib.no/plab/index.html).

2.2.3 NVRAM

Στην περίπτωση του NVRAM είχαμε το ίδιο πειραματικό περιβάλλον με το RVM. Επιπλέον κάναμε πειράματα και σε ένα δίκτυο από Sun SS10 στα 40 MHz. Οι σταθμοί ήταν συνδεδεμένοι με ένα SCI διασυνδετικό δίκτυο το οποίο χρησιμοποιεί SBUS−σε−SCI network interfaces (ναι δεν το μεταφράζω) της Dolphin [Dol94]. Το SCI (Scalable Coherent Interface) είναι ένα διασυνδετικό δίκτυο χαμηλής καθυστέρησης και υψηλού ρυθμού μεταγωγής δεδομένων το οποίο έχει αναπτυχθεί για ομάδες σταθμών εργασίας και πολυεπεξεργαστές [JLGS90]. Το SCI επιτρέπει σε εφαρμογές να έχουν απευθείας πρόσβαση στην κύρια μνήμη όλων των σταθμών εργασίας στην ομάδα χωρίς την παρέμβαση του λειτουργικού συστήματος, αποδοτικά, παρέχοντας πολύ χαμηλή καθυστέρηση επικοινωνίας.



Σχήμα 2.1: **Στην αριστερή πλευρά του σχήματος φαίνεται η αρχιτεκτονική του διαχειριστή αποθήκευσης αντικειμένων Exodus. Στην δεξιά πλευρά παρουσιάζεται η πρότασή μας για τη νέα αρχιτεκτονική που θα χρησιμοποιεί την απομακρυσμένη μνήμη για βελτίωση της απόδοσης και της αξιοπιστίας.**

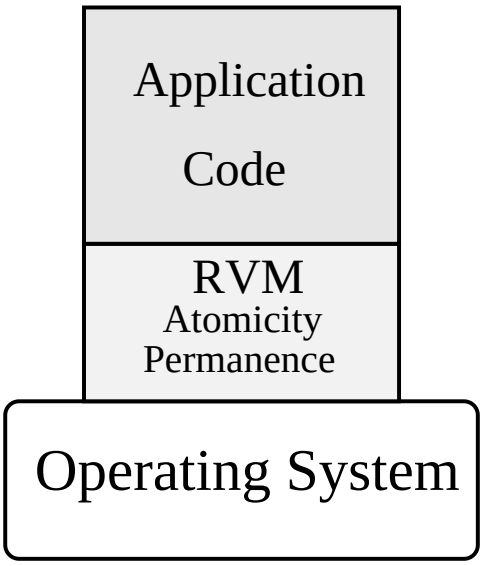
2.3 Συστήματα

2.3.1 Exodus

Ο Exodus είναι ένας διαχειριστής αποθήκευσης αντικειμένων, πολλών χρηστών. Υποστηρίζει δείκτες, δοσοληψίες, έλεγχο συγχρονισμού, ανάκτηση και έχει αρχιτε−κτονική πελάτη−εξυπηρετητή. Επιλέξαμε τον Exodus λόγο του ότι είναι ένας πολύ διαδεδομένος διαχειριστής αποθήκευσης. Η αρχιτεκτονική του φαίνεται στο σχήμα 2.1.

Στην αριστερή πλευρά του σχήματος φαίνεται η αρχιτεκτονική του διαχειριστή αποθήκευσης Exodus. Όπως είπαμε έχουμε να κάνουμε με μία αρχιτεκτονική πελάτη−εξυπηρετητή. Ο εξυπηρετητής τρέχει σε ένα σταθμό εργασίας και έχει πρόσβαση στο ημερολόγιο και τα δεδομένα της βάσης δεδομένων. Το ημερολόγιο και τα δεδομένα φυλάσσονται σε διαιρέσεις του δίσκου ή σε αρχεία UNIX. Οι πελάτες του Exodus, οι οποίοι μπορεί να τρέχουν σε άλλους σταθμούς εργασίας, κάνουν αιτήσεις ανάγνωσης και εγγραφής στον εξυπηρετητή. Αν οι σελίδες που ζήτησε για ανάγνωση ο πελάτης βρίσκονται στην κύρια μνήμη του εξυπηρετητή αυτός απαντά αμέσως, διαφορετικά πρέπει να τις ανακτήσει από το δίσκο. Στην περίπτωση της εγγραφής το ημερολόγιο ενημερώνεται και γράφεται στον δίσκο. Παρατηρούμε λοιπόν ότι η απόδοση του Exodus περιορίζεται από το ποσό της κύριας μνήμης του σταθμού εργασίας στον οποίο τρέχει ο εξυπηρετητής και από την απόδοση του δίσκου.

Στη δική μας σχεδίαση, η οποία φαίνεται στο δεξί μέρος του σχήματος 2.1, θα παρακάμψουμε αυτά τα δύο προβλήματα κατανέμοντας το ημερολόγιο και τα δεδομένα στην απομακρυσμένη μνήμη. Με αυτό τον τρόπο οι αιτήσεις για εγγραφή και ανάγνωση θα κατευθύνονται στην απομακρυσμένη μνήμη αντί για τον τοπικό δίσκο.



Σχήμα 2.2: **Το RVM είναι μία βιβλιοθήκη που χρησιμοποιείται από εφαρμογές που χρειάζονται δοσοληψίες για τη λειτουργία τους.**

Αυτό θα το καταφέρουμε τρέχοντας στους σταθμούς εργασίας της ομάδας κάποιους εξυπηρετητές μνήμης. Οι εξυπηρετητές θα φυλάσσουν τις σελίδες ημερολογίου και δεδομένων της βάσης δεδομένων.

2.3.2 RVM

Το RVM (Recoverable Virtual Memory), είναι ένα σύστημα που παρέχει μηχανισμούς για την υποστήριξη σταθερής ιδεατής μνήμης σε περίπτωση πτώσης του συστήματος. Η κατάσταση των πραγμάτων σήμερα είναι ότι το λογισμικό και όχι το υλικό είναι ο περιοριστικός παράγοντας στην αξιοπιστία των συστημάτων, και έτσι το RVM βοηθάει στη συγγραφή πιο ανθεκτικού λογισμικού. Το βασικό κριτήριο αξιοπιστίας στο RVM είναι η εξασφάλιση μονιμότητας σε δεσμευμένες αλλαγές και ακεραιότητας σε δεδομένα μετά από πτώσεις του συστήματος. Τέτοιες πτώσεις αφήνουν τις δομές δεδομένων της εφαρμορής σε μη συμβατή κατάσταση. Η ιδιότητα όλα−ή−τίποτα (ατομικότητα) των δοσοληψιών μπορεί να εγγυηθεί συμβατότητα των δεδομένων μετά από πτώσεις του συστήματος και με αυτό τον τρόπο να κάνει τις εφαρμογές πιο ανθεκτικές.

Το RVM, όπως φαίνεται στο σχήμα 2.2, είναι μία βιβλιοθήκη που προσφέρει απλές, μη−φωλιασμένες δοσοληψίες, για να πετύχει διάρκεια στα δεδομένα, τα οποία φυλάσσονται σε αρχεία UNIX ή διαιρέσεις του δίσκου.

Όταν μία δοσοληψία δεσμευτεί, όλα τα δεδομένα που έχουν μεταβληθεί από αυτή γράφονται σύγχρονα στο ημερολόγιο. Μόλις αυτό συμβεί η εφαρμογή που χρησιμοποιεί το RVM μπορεί να συνεχίσει. Το πρόβλημα λοιπόν είναι ότι οι σύγχρονες εγγραφές στο δίσκο είναι πολύ ακριβές. Με την σχεδίαση μας θέλουμε να βελτιώσουμε το χρόνο που χρειάζεται για να εκτελεστεί μία δοσοληψία αλλά ταυτόχρονα να μην θυσιάσουμε τίποτε από την αξιοπιστία της. Γι'αυτό αντικαθιστούμε

τις σύγχρονες εγγραφές στο δίσκο με *σύγχρονες εγγραφές στο δίκτυο*. Με κάθε σύστημα RVM (από εδώ και πέρα θα αναφερόμαστε σε αυτό σαν πελάτη) αντιστοιχούμε και ένα εξυπηρετητή μνήμης. Ο εξυπηρετητής μνήμης κρατάει στη μνήμη του ένα αντίγραφο του ημερολογίου του πελάτη. Σε περίπτωση πτώσης του πελάτη τα δεδομένα μπορούν να βρεθούν στη μνήμη του εξυπηρετητή.

2.3.3 NVRAM

Το NVRAM είναι ένα σύστημα λογισμικού−μόνο, το οποίο προσφέρει σταθερή μνήμη χωρίς χρήση επιπλέον υλικού. Για να μπορέσουμε να προσφέρουμε σταθερή μνήμη χρησιμοποιούμε μερικές ανεξάρτητες, μη−σταθερές αποθήκες δεδομένων. Έτσι, όποτε πρέπει να γράψουμε δεδομένα σε σταθερή μνήμη, τα γράφουμε σε *όλες* τις μη−σταθερές μνήμες. Χάσιμο δεδομένων θα έχουμε μόνο αν τα χάσουμε από *όλες* τις μνήμες, πράγμα που έχει πολύ μικρή πιθανότητα να συμβεί. Το σύστημά μας θα χρησιμοποιήσει τις κύριες μνήμες των σταθμών εργασίας σε μία ομάδα σταθμών, σαν ανεξάρτητες αποθήκες μνήμης. Μόλις τα δεδομένα γραφτούν στις κύριες μνήμες των άλλων σταθμών η διεργασία που χρησιμοποιεί την σταθερή μνήμη μπορεί να συνεχίσει. Όσο η διεργασία υπολογίζει, τα δεδομένα γράφονται ασύγχρονα στο δίσκο. Σε περίπτωση που η εφαρμογή πέσει πριν τα δεδομένα γραφτούν όλα στο δίσκο, τα δεδομένα δεν έχουν χαθεί, και αυτό γιατί μπορεί να τα ανακτήσει από την απομακρυσμένη μνήμη.

2.4 Ο Εξυπηρετητής Μνήμης

Ο εξυπηρετητής μνήμης είναι ένα πρόγραμμα−δαίμονας που εκτελείται σε κατάσταση χρήστη, η δουλειά του είναι να φυλάει δεδομένα στο πεδίο διευθύνσεών του. Αυτά τα δεδομένα προέρχονται από τα τρία συστήματα που περιγράψαμε. Ο Exodus, το RVM ή το NVRAM συνδέεται με έναν εξυπηρετητή μνήμης (RVM, NVRAM) ή με περισσότερους (Exodus), και του στέλνει αιτήσεις ανάγνωσης ή εγγραφής. Ο εξυπηρετητής μνήμης είναι υπεύθηνος να απαντήσει στις αιτήσεις αναγνώσεις ή να φυλάξει τα δεδομένα των αιτήσεων εγγραφών.

Κεφάλαιο 3

Υλοποίηση

3.1 Εισαγωγή

Σε ένα κατανεμημένο σύστημα, όπως είναι ένα τοπικό δίκτυο, ένα μηχάνημα μπορεί να πέσει ανά πάσα στιγμή. Αν στο μηχάνημα αυτό τρέχει ένας εξυπηρετητής μνήμης, αυτό σημαίνει ότι θα χαθούν όλα τα δεδομένα που είναι αποθηκευμένα σ' αυτόν. Είναι προφανές ότι οι εφαρμογές που εκτελούνται σε μια μηχανή του δικτύου δεν είναι επιτρεπτό να αποτύχουν εξαιτίας βλάβης σε κάποιο άλλο μηχανήμα στο κατανεμημένο σύστημα που τυχαίνει να αποθηκεύει κάποια δεδομένα τους. Γι' αυτό το λόγο το σύστημά μας θα πρέπει να είναι σε θέση να εγγυηθεί ότι η πιθανότητα να χαθούν κάποια δεδομένα λόγο πτώσης σε ένα μακρυνό μηχάνημα είναι μικρότερη ή ίση της πιθανότητας που έχει το ίδιο το μηχάνημα να υποστεί βλάβη.

Υπάρχουν τρία είδη βλαβών και αυτά εξετάζουμε (δεν θα ασχοληθούμε με κατα−στροφές όπως σεισμοί, πυρκαγιές κ.λ.π).

1. Καταρχήν, μπορεί να υπάρξει βλάβη λόγω πτώσης της τάσης του ηλεκτρικού ρεύματος. Μπορούμε να υποθέσουμε ότι οι σταθμοί εργασίας είναι συνδεδεμένοι σε ανεξάρτητες παροχές ηλεκτρικού ρεύματος, για παράδειγμα σε διαφορετικά UPS. Αν έχουμε πτώση τάσης σε κάποιο ή κάποιους σταθμούς οι άλλοι σταθμοί μπορούν να επιβιώσουν και έτσι τα δεδομένα μας δε θα επηρεαστούν.

2. Μια άλλη αιτία βλάβης είναι η βλάβη στο διασυνδετικό δίκτυο. Για παράδειγμα μία γέφυρα ή ένας μεταγωγέας παθαίνει βλάβη. Αν συμβεί κάτι τέτοιο τότε οι σταθμοί εργασίας δεν μπορούν να επικοινωνήσουν. Σε αυτή τη περίπτωση, ο πελάτης γενικά δεν μπορεί να ανακτήσει τα δεδομένα του αφού μπορεί να έχει αποκοπεί από όλα τα υπόλοιπα μηχανήματα. Η λύση λοιπόν που ακολουθείται είναι ότι ο πελάτης μπλοκάρει μέχρι να επανέλθει το δίκτυο. Αυτή τη λύση άλλωστε υιοθετούν και τα περισσότερα κατανεμημένα συστήματα αρχείων, όπως το NFS [SGK+85].

3. Η πιο πιθανή αιτία βλάβης είναι η πτώση ενός μηχανήματος η οποία οφείλεται είτε σε πρόβλημα του υλικού, είτε σε πρόβλημα του λογισμικού. Κάτι τέτοιο έχει σαν αποτέλεσμα την απώλεια των περιεχομένων της μνήμης που κρατούσε ο εξυπηρετητής μνήμης που έτρεχε σε εκείνο το μηχάνημα. Μια τέτοια περίπτωση λοιπόν θα πρέπει να αντιμετωπίζεται από το σύστημά μας.

Ο μέσος χρόνος μεταξύ βλαβών του συστήματός μας είναι περίπου ίσος με $\frac{MTTFw}{N}$, όπου N ο αριθμός των μηχανών μας και $MTTFw$ ο μέσος χρόνος μεταξύ βλαβών ενός

μηχανήματος, θεωρώντας ότι οι βλάβες είναι ανεξάρτητες και ακολουθούν εκθετική κατανομή. Αυτός ο χρόνος είναι σχετικά σύντομος άρα οι πτώσεις πολύ πιθανές και γι' αυτό πρέπει να αντιμετωπιστεί. Μια βλάβη μπορεί να αντιμετωπιστεί αν αποθηκεύουμε πλεονάζουσα πληροφορία ώστε να μπορούμε είτε να ανακτήσουμε, είτε να ανακατασκευάσουμε τις σελίδες που έχουν αποθηκευτεί στο μηχάνημα που έπαθε τη βλάβη. Έστω ότι η πλεονάζουσα πληροφορία που διατηρούμε μας επιτρέπει την ανακατασκευή των σελίδων που είναι αποθηκευμένες σε ένα υπολογιστή αν διαθέτουμε τις σελίδες που είναι αποθηκευμένες στους άλλους υπολογιστές του δικτύου, στις οποίες συμπεριλαμβάνονται και σελιδές πλεονάζουσας πληροφορίας. Έστω $MTTRw$ ο μέσος χρόνος επισκευής ενός μηχανήματος. Αν G είναι ο αριθμός των μηχανών από τις οποίες αν κάποια πάθει βλάβη μπορούμε να ανακατασκευάσουμε τις σελίδες της χρησιμοποιώντας την πληροφορία που είναι αποθηκευμένη στις υπόλοιπες $G-1$ μηχανές, τότε ο μέσος χρόνος μεταξύ βλαβών του συστήματός μας, όπως αποδεικνύεται στο [CLG+94] είναι ίσος με $\frac{MTTFw^2}{(N(G-1)MTTRw)}$. Για τυπικές τιμές των παραμέτρων, η τιμή της παράστασης αυτής είναι πολύ μεγάλη, π.χ. για $MTTFw$ ίσο με τρεις μήνες, $MTTRw$ δέκα λεπτά, N ίσο με 20 και G ίσο με 4, η τιμή της παράστασης είναι ίση περίπου με 54 χρόνια. Για τις ίδιες παραμέτρους με G ίσο με 16, η τιμή της είναι περίπου ίση με 10 χρόνια. Άρα η ικανότητα του συστήματος να αναρρώνει από βλάβες ενός μηχανήματος προσφέρει ικανοποιητική αξιοπιστία μια που το ενδεχόμενο ταυτόχρονης βλάβης περισσοτέρων του ενός μηχανήματος είναι σπάνιο.

3.2 Στόχοι των Πολιτικών Αξιοπιστίας

Στην υλοποίηση λοιπόν του συστήματός μας είναι απαραίτητο να ενσωματώσουμε κάποια τεχνική για την αντιμετώπιση βλαβών. Μια τέτοια τεχνική θα πρέπει να εκπληρώνει κάποιους βασικούς στόχους.

- *Η επιβάρυνση του χρόνου εκτέλεσης θα πρέπει να είναι όσο το δυνατόν μικρότερη, αφού η επιβάρυνση αυτή αποτελεί ένα κόστος το οποίο το πληρώνουμε σε κάθε μεταφορά δεδομένων.*
- *Η επιπλέον μνήμη που απαιτεί για να λειτουργήσει θα πρέπει επίσης να είναι όσο το δυνατόν μικρότερη, αφού η μνήμη που είναι δεσμευμένη για την εξασφάλιση αξιοπιστίας θα μπορούσε να χρησιμοποιηθεί για την αποθήκευση άλλων δεδομένων.*

Στις ενότητες 3.3.1, 3.4.1 και 3.5.1 παρουσιάζουμε τρεις διαφορετικές πολιτικές αξιοπιστίας και ελέγχουμε κατά πόσο ικανοποιούν τα παραπάνω κριτήρια.

3.3 Υλοποίηση στον Exodus

Για να μπορέσουμε να κάνουμε τον Exodus να χρησιμοποιήσει απομακρυσμένη μνήμη έπρεπε να προσθέσουμε κώδικα έτσι ώστε να μπορεί να στέλνει τις σελίδες της βάσης δεδομένων στους εξυπηρετητές μνήμης αντί για τον τοπικό δίσκο. Οι σελίδες γράφονται και διαβάζονται προς και από το δίσκο από τον εξυπηρετητή του Exodus με κώδικα που βρίσκεται στα αρχεία `openLocalDisk.c`, `closeLocalDisk.c`, `readLocalDisk.c`, `writeLocalDisk.c` και `fsyncLocalDisk.c`. Ο κώδικας σε αυτά τα αρχεία είναι υπεύθυνος για το άνοιγμα και το κλείσιμο του αρχείου

δεδομένων και του ημερολογίου, για τις αναγνώσεις και τις εγγραφές, καθώς και για τον συγχρονισμό των δεδομένων στο δίσκο (χρησιμοποιώντας τις κλήσεις συστήματος `open`, `close`, `read`, `write` και `fsync`).

Προσθέσαμε λοιπόν κώδικα ο οποίος ξεκινάει μία σύνδεση με έναν εξυπηρετητή μνήμης χρησιμοποιώντας υποδοχές τύπου TCP [Pos81b, Pos81a, Com91]. Με αυτό τον τρόπο (χρησιμοποιώντας τις κλήσεις συστήματος `socket`, `connect`, `bind`, `accept`, `read` και `write`) ο εξυπηρετητής Exodus μπορεί να στείλει τις σελίδες του στους μακρυνούς εξυπηρετητές μνήμης. Διαιρέσαμε το ημερολόγιο και το αρχείο δεδομένων σε ένα σύνολο από απομακρυσμένους εξυπηρετητές μνήμης και με αυτό τον τρόπο ο Exodus ήταν πλέον δυνατόν να κάνει τις αναγνώσεις και τις εγγραφές του στην απομακρυσμένη μνήμη.

3.3.1 Αξιοπιστία με τη μέθοδο της Ισοτιμίας

Δεδομένου ότι με τη διανομή του ημερολογίου και του αρχείου δεδομένων σε απομακρυσμένη μνήμη πάψαμε να έχουμε σταθερότητα στο μέσο αποθήκευσης, έπρεπε να υλοποιήσουμε μηχανισμούς για την ανάκτηση ή ανακατασκευή τους σε περίπτωση που είχαμε πτώση σε κάποιο σταθμό εργασίας. Γι' αυτό το λόγο χρησιμοποιήσαμε την τεχνική της ισοτιμίας, την οποία δανειστήκαμε από τα RAIDs [CLG+94, PGK88]. Η τεχνική της ισοτιμίας για κάθε N σελίδες δεδομένων υπολογίζει και μία σελίδα ισοτιμίας η οποία είναι το Αποκλειστικό Ή (XOR) των N σελίδων. Με αυτό τον τρόπο μπορούμε να υπολογίσουμε οποιαδήποτε σελίδα θέλουμε από τις N − 1 και τη σελίδα ισοτιμίας.

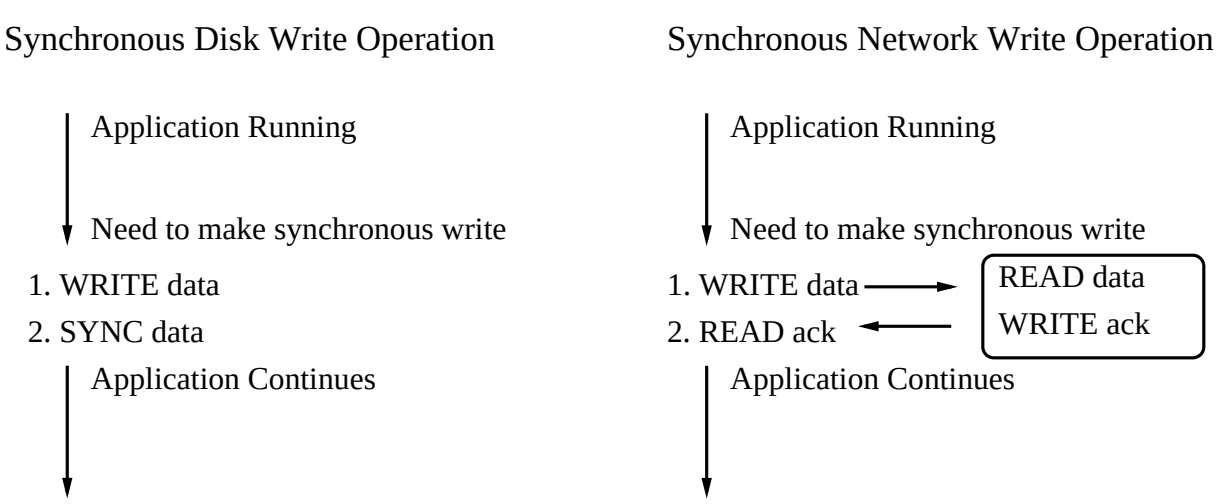
Όπως προαναφέραμε διαιρέσαμε το ημερολόγιο σε N κομμάτια και για κάθε Νάδα από σελίδες υπολογίζουμε την σελίδα ισοτιμίας. Το ίδιο κάνουμε και για το αρχείο δεδομένων. Τις σελίδες ισοτιμίας τις κρατάει ο εξυπηρετητής του Exodus είτε στη μνήμη του είτε στο δίσκο αν δεν έχει χώρο στην κύρια μνήμη. Αν κάποιος μακρυνός εξυπηρετητής μνήμης πέσει μπορούμε να ανακατασκευάσουμε τα δεδομένα του όπως αναφέραμε παραπάνω. Επιπλέον αν πέσει ο εξυπηρετητής του Exodus δεν χάνουμε δεδομένα γιατί αυτά είναι κατανεμημένα στους μακρυνούς εξυπηρετητές μνήμης.

Όταν ο εξυπηρετητής του Exodus θελήσει να γράψει μία σελίδα πρέπει να ενημερώσει την αντίστοιχη σελίδα ισοτιμίας. Αυτό γίνεται σε δύο στάδια:

1. Ο εξυπηρετητής του Exodus στέλνει τη νέα σελίδα στον κατάλληλο μακρυνό εξυπηρετητή μνήμης και ζητάει την παλιά του σελίδα.

2. Όταν παίρνει την παλιά σελίδα υπολογίζει το Αποκλειστικό Ή της σελίδας ισοτιμίας, της παλιάς σελίδας και της νέας σελίδας, και το αποτέλεσμα είναι η νέα σελίδα ισοτιμίας.

Η μέθοδος αυτή αυξάνει ελάχιστα το ποσό της απαιτούμενης μνήμης, κατά 1/N, αλλά έχει το μειονέκτημα ότι χρειάζεται μία επιπλέον μεταφορά σελίδας. Στην περίπτωση του ημερολογίου αυτό όμως *δεν* είναι πρόβλημα γιατί οι εγγραφές σε αυτό γίνονται πάντα στο τέλος του οπότε δεν χρειάζεται ποτέ να υπολογίσουμε νέες ισοτιμίες. Από την άλλη αυτό δεν είναι αλήθεια στο αρχείο δεδομένων. Σε αυτό όμως τις πιο πολλές φορές οι εγγραφές που κάνουμε είναι μεγαλύτερες από τις N σελίδες που χρειάζονται για τον υπολογισμό μίας σελίδας ισοτιμίας και έτσι δεν είμαστε αναγκασμένοι να ζητήσουμε όλες τις σελίδες που θεωρητικά θα έπρεπε από τους μακρυνούς εξυπηρετητές μνήμης.



Σχήμα 3.1: **Στα αριστερά παρουσιάζουμε μία σύγχρονη εγγραφή στο δίσκο και στα δεξιά μία σύγχρονη εγγραφή στο δίκτυο.**

3.4 Υλοποίηση στο RVM

Όπως και στον Exodus έτσι και στο RVM, για να μπορέσουμε να χρησιμοποιήσουμε την απομακρυσμένη μνήμη έπρεπε να προσθέσουμε κώδικα στα κατάλληλα σημεία στο RVM έτσι ώστε να είναι δυνατόν οι εγγραφές να πηγαίνουν εκεί και όχι στον τοπικό δίσκο. Στο αρχείο `rvm_io.c` βρίσκεται ο κώδικας που αναλαμβάνει την πρόσβαση στο δίσκο για το RVM. Σε αυτό το αρχείο προσθέσαμε κώδικα έτσι ώστε να δημιουργήσουμε μία υποδοχή τύπου TCP με ένα μακρυνό εξυπηρετητή μνήμης, έτσι ώστε να χρησιμοποιήσουμε την απομακρυσμένη μνήμη. Σε αντίθεση με τον Exodus, στο RVM θα χρησιμοποιήσουμε την απομακρυσμένη μνήμη μόνο για το ημερολόγιο και όχι για το αρχείο δεδομένων.

3.4.1 Αξιοπιστία με τη μέθοδο των Πολλαπλών Αντιγράφων

Η πολιτική αξιοπιστίας που υλοποιήσαμε για το RVM είναι διαφορετική από αυτή του Exodus. Η ιδέα είναι να αντικαταστήσουμε τις σύγχρονες εγγραφές στο δίσκο με σύγχρονες εγγραφές στο δίκτυο, όπως φαίνεται στο σχήμα 3.1. Έτσι λοιπόν κάθε σύγχρονη εγγραφή στο δίσκο αντικαθίσταται από την ακόλουθη διαδικασία:

1. Αποστολή των δεδομένων από τον πελάτη στον εξυπηρετητή μνήμης.

2. Ταυτόχρονη *ασύγχρονη* εγγραφή των δεδομένων στο δίσκο.

3. Όταν ο πελάτης δεχτεί επιβεβαίωση ότι ο εξυπηρετητής μνήμης δέχτηκε τα δεδομένα, η εφαρμογή συνεχίζει.

3.5 Υλοποίηση του NVRAM

Για την υλοποίηση του NVRAM γράψαμε μία βιβλιοθήκη την οποία μπορεί να χρησιμοποιήσει οποιαδήποτε εφαρμογή θέλει αξιόπιστες εγγραφές (όπως για παράδειγμα βάσεις δεδομένων). Η βιβλιοθήκη αποτελείται από τις ακόλουθες δύο συναρτήσεις:

- nv_init (int size, char * hostname, int port): αυτή η κλήση φροντίζει για την αρχικοποίηση του συστήματος NVRAM. Ξεκινάει ένα εξυπηρετητή μνήμης σε κάποιο μηχάνημα με μέγεθος μνήμης size.
- nv_commit (char * start, int length): η κλήση αυτή γίνεται όποτε θέλουμε να εγγυηθούμε ότι κάποια δεδομένα μας είναι γραμμένα με ασφάλεια σε σταθερή μνήμη.

3.5.1 Αξιοπιστία με τη μέθοδο των Πολλαπλών Αντιγράφων

Για να πετύχουμε αξιοπιστία στο NVRAM, χρησιμοποιούμε το ίδιο πρωτόκολλο με το RVM. Αυτή τη φορά όμως επειδή το μέγεθος της μνήμης που θέλουμε να καλύψουμε αξιόπιστα μπορεί να είναι μεγαλύτερο από το μέγεθος της απομακρυσμένης μνήμης που έχουμε ζητήσει, όταν αυτή γεμίσει η `nv_commit` δεν επιστρέφει μέχρι τα δεδομένα που έχουμε δώσει για ασύγχρονη εγγραφή στο δίσκο εγγραφούν σε αυτόν.

Κεφάλαιο 4

Πειραματικά Αποτελέσματα

4.1 Παράμετροι Πειραμάτων και Μετρικές Απόδοσης

Σε αυτό το κεφάλαιο θα παρουσιάσουμε τα πειραματικά μας αποτελέσματα. Για να γίνει πιο κατανοητή η παρουσίαση τους είναι ανάγκη να εξηγήσουμε τις παραμέτρους των πειραμάτων μας αλλά και τις μετρικές που χρησιμοποιούμε για να μετρήσουμε την απόδοση.

Στα πειράματά μας στον Exodus μετράμε την "επιτάχυνση" σε διάφορες δοκιμασίες όταν αυτές έγιναν με και χωρίς τη χρήση απομακρυσμένης μνήμης. Με τον όρο "επιτάχυνση" ορίζουμε το χρόνο που χρειάστηκε για να τερματίσει η δοκιμασία όταν έκανε χρήση του δίσκου, προς το χρόνο που χρειάστηκε για να τερματίσει η ίδια δοκιμασία όταν έκανε χρήση της απομακρυσμένης μνήμης.

Στα πειράματά μας στο RVM μεταβάλλουμε το μέγεθος του ημερολογίου και το μέγεθος του τμήματος εγγραφής από τις δοσοληψίες (I/O block size) και μετράμε πόσες τέτοιες δοσοληψίες έγιναν ανά δευτερόλεπτο.

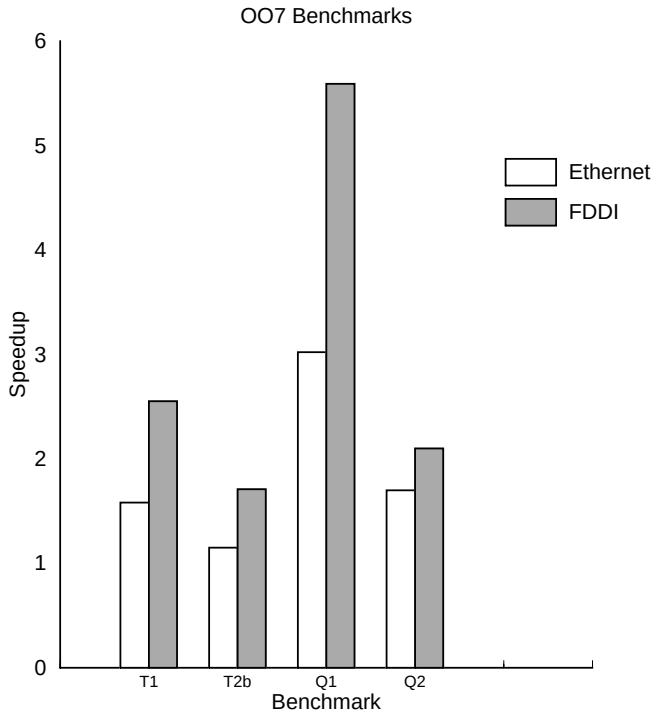
Στα πειράματά μας στο NVRAM μεταβάλλουμε το μέγεθος του τμήματος απομα−κρυσμένης μνήμης καθώς και της εγγραφής και μετράμε το πλήθος των σταθερών εγγραφών καθώς και το ρυθμό μεταγωγής δεδομένων.

4.2 Πειράματα στον Exodus

4.2.1 Σημείο Αναφοράς OO7

Το σημείο αναφοράς OO7 είναι ένα σύνολο από δοκιμασίες σχεδιασμένες για να αξιολογήσει αντικειμενοστρεφεις βάσεις δεδομένων. Αποτελείται από ένα σύνολο σαρώσεων, ενημερώσεων και αναζητήσεων. Χρησιμοποιήσαμε το OO7 για να πάρουμε ενδείξεις για το πόσο βελτιώνεται η απόδοση του Exodus με τη σχεδίασή μας. Γι' αυτό το λόγο φτιάξαμε μία βάση δεδομένων των 21 MB και κάναμε πειράματα χρησιμοποιώντας το OO7. Στο σχήμα 4.1 βλέπουμε την επιτάχυνση που είχαν οι διάφορες δοκιμασίες όταν έτρεξαν με αρκετή απομακρυσμένη μνήμη. Τα πειράματα έγιναν πάνω από Ethernet και FDDI χρησιμοποιώντας την τεχνική της ισοτιμίας για αξιοπιστία. Οι δοκιμασίες που παρουσιάζουμε είναι οι:

- **T1:** Διατρέχει τη βάση δεδομένων.
- **T2b:** Διατρέχει τη βάση δεδομένων και ταυτόχρονα ενημερώνει ένα πεδίο.



Σχήμα 4.1: **Επιτάχυνση στα σημεία αναφοράς OO7.**

- **Q1:** Κάνει ακριβή αναζήτηση στην βάση δεδομένων.
- **Q2:** Κάνει αναζήτηση πεδίου στην βάση δεδομένων.

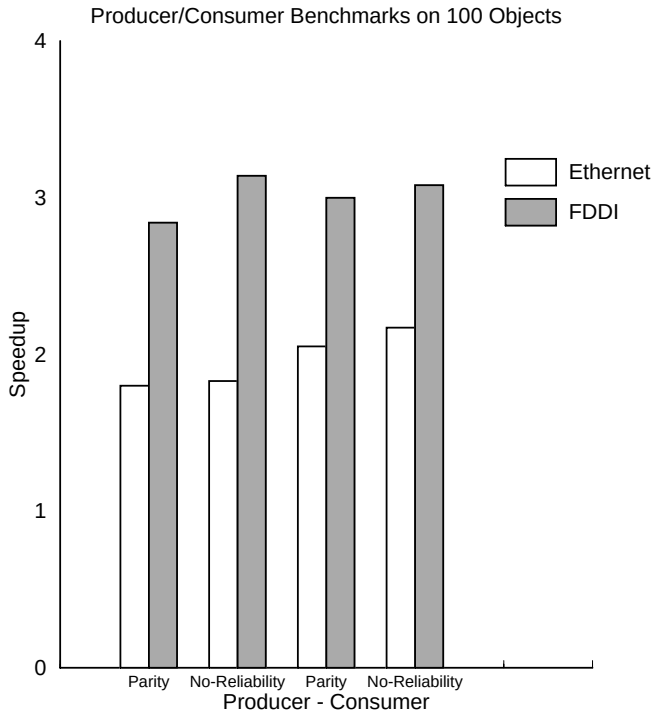
Στην εικόνα παρατηρούμε ότι το T2b έχει χειρότερη επιτάχυνση από το T1. Αυτή η συμπεριφορά εξηγείται από το διαφορετικό ποσοστό εργασίας/επικοινωνίας που έχουν οι δύο δοκιμασίες. Το T2b ξοδεύει συγκριτικά λιγότερο χρόνο σε επικοινωνία σε σχέση με το T1 και γι' αυτό ωφελείται λιγότερο από τις βελτιώσεις μας. Η ίδια συμπεριφορά παρατηρείται και στις δοκιμασίες Q1 και Q2.

Βλέπουμε λοιπόν ότι οι αναγνώσεις από την απομακρυσμένη μνήμη είναι ταχύτερες από τις αναγνώσεις από τον τοπικό δίσκο. Επιπλέον, ο συγχρονισμός των δεδομένων στο δίσκο, για σταθερότητα, είναι και αυτός πιο ακριβός από την αποστολή τους στην απομακρυσμένη μνήμη.

4.2.2 Σημείο Αναφοράς Παραγωγού−Καταναλωτή

Για να αξιολογήσουμε την πολιτική της ισοτιμίας κάναμε άλλη μία σειρά πειραμάτων. Αυτή τη φορά το σημείο αναφοράς είναι τύπου παραγωγού/καταναλωτή. Το πρώτο πρόγραμμα παράγει αντικείμενα κάποιου ορισμένου μεγέθους και τα αποθηκεύει στη βάση δεδομένων μας, το δεύτερο τα καταναλώνει. Στο σχήμα 4.2 φαίνεται η επιτάχυνση όταν τρέξαμε τα προγράμματα για 100 αντικείμενα μεγέθους 10000 bytes.

Στο αριστερό μέρος της εικόνας τα δύο πρώτα ζευγάρια στηλών αφορούν τον παραγωγό και στο δεξιό τα δύο δεύτερα ζευγάρια στηλών αφορούν τον καταναλωτή. Τα πειράματα έχουν γίνει σε Ethernet και FDDI. Η μέθοδος της ισοτιμίας συγκρίνεται



Σχήμα 4.2: **Επιτάχυνση στα σημεία αναφοράς Παραγωγού – Καταναλωτή.**

με την απουσία αξιοπιστίας, δηλαδή όταν δεν κρατάμε την επιπλέον πληροφορία της ισοτιμίας. Παρατηρούμε ότι παρόλο την παραπάνω εργασία που χρειάζεται για τον υπολογισμό της ισοτιμίας η μέθοδος δεν χάνει σχεδόν καθόλου σε απόδοση. Άρα ουσιαστικά το μόνο κόστος που μας προσδίδει είναι το κόστος σε χώρο αποθήκευσης.

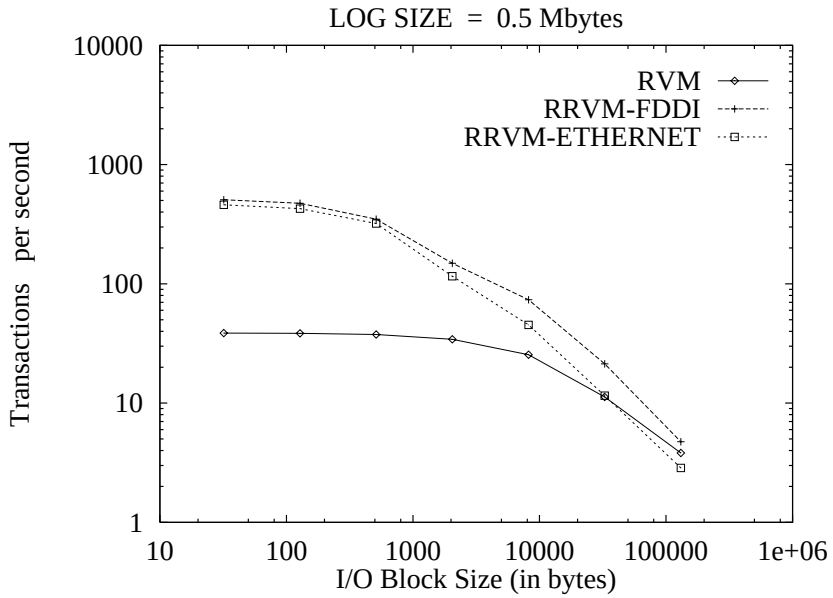
4.3 Πειράματα στο RVM

Με τα πειράματά μας στο RVM θέλουμε να ερευνήσουμε μία σειρά από παράγοντες που επηρεάζουν την απόδοση του συστήματός μας. Γι' αυτό το σκοπό σχεδιάσαμε ένα σύνολο από πειράματα που μετρούν διάφορες πτυχές του συστήματος. Με τον όρο RRVM αναφερόμαστε στο RVM που χρησιμοποιεί απομακρυσμένη μνήμη. Για τα πειράματα μας έχουμε τρεις διαμορφώσεις:

- `RVM`: Το αρχικό σύστημα RVM.
- `RRVM-FDDI`: Το RRVM όταν έτρεξε πάνω από διασυνδετικό δίκτυο τύπου FDDI.
- `RRVM-ETHERNET`: Το RRVM όταν έτρεξε πάνω από διασυνδετικό δίκτυο τύπου ETHERNET.

4.3.1 Μέγεθος της Δοσοληψίας

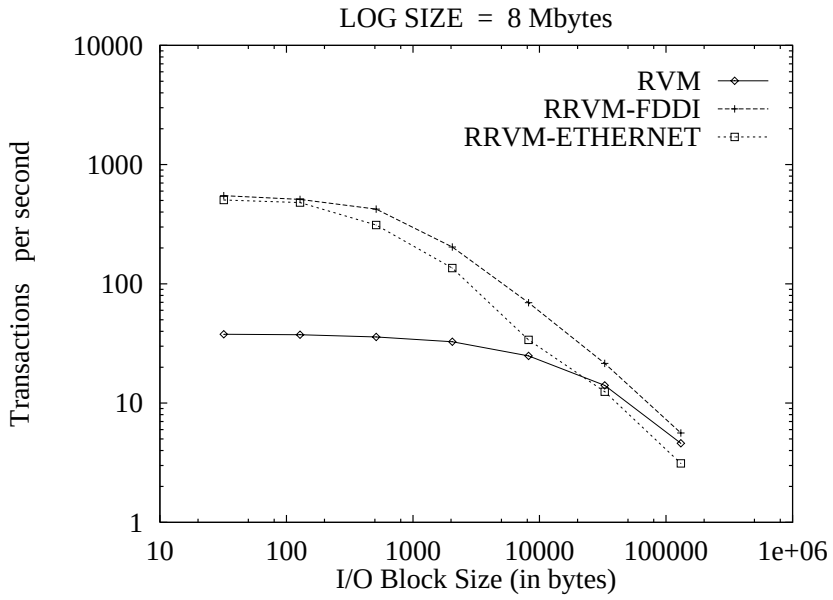
Πρώτα απ' όλα θα θέλαμε να μάθουμε πόσες δοσοληψίες το δευτερόλεπτο μπορεί να υποστηρίξει το RRVM σε σχέση με τις δοσοληψίες το δευτερόλεπτο που εκτελεί το αυθεντικό RVM. Γι αυτό τον σκοπό σχεδιάσαμε το ακόλουθο πείραμα:



Σχήμα 4.3: **Η απόδοση του RVM σαν συνάρτηση του μεγέθους της εγγραφής της κάθε δοσοληψίας. Μέγεθος ημερολογίου 512 KB – σειριακές προσπελάσεις.**

1. Δημιουργούμε ένα σχετικά μεγάλο αρχείο, μεγέθους 100 MB.
2. Σε αυτό ξεκινάμε μία ακολουθία από 10000 δοσοληψίες.
3. Η κάθε δοσοληψία γράφει ένα τμήμα του αρχείου, μετά δεσμεύει αυτό το τμήμα και τερματίζει.
4. Οι δοσοληψίες μεταβάλλουν το αρχείο με σειριακό τρόπο.
5. Για τις ανάγκες του πειράματος μεταβάλλουμε το μέγεθος του τμήματος που γράφουν οι δοσοληψίες.

Στο σχήμα 4.3 παρουσιάζονται τα αποτελέσματά μας, στον κάθετο άξονα βλέπουμε πόσες δοσοληψίες το δευτερόλεπτο επιτύχαμε με τις διάφορες διαμορφώσεις σαν συνάρτηση του μεγέθους της δοσοληψίας. Για το πείραμα έχουμε θέσει μέγεθος αρχείου δεδομένων 100 MB και μέγεθος ημερολογίου 512 KB. Το αρχικό `RVM` βλέπουμε ότι μπορεί να κάνει περίπου 40 δοσοληψίες το δευτερόλεπτο πράγμα που συμφωνεί με τα αποτελέσματα του [SMK+93], που παρουσιάζει το `RVM` να μην ξεπερνάει τις 50 δοσοληψίες το δευτερόλεπτο, (βλέπε σχήμα 8(b) στο [SMK+93]). Επιπλέον παρατηρούμε ότι καθώς το μέγεθος της δοσοληψίας μεγαλώνει το πλήθος των δοσοληψιών το δευτερόλεπτο πέφτει σε ακόμη χαμηλότερα επίπεδα. Αν τώρα κοιτάξουμε τις καμπύλες του `RRVM-FDDI` και του `RRVM-ETHERNET` βλέπουμε ότι η απόδοση βρίσκεται κοντά στις 500 δοσοληψίες το δευτερόλεπτο, δηλαδή περισσότερο από μία τάξη μεγέθους παραπάνω από το `RVM`. Αυτή η μεγάλη διαφορά στην απόδοση οφείλεται στον τρόπο που συγχρονίζουμε το ημερολόγιο. Το `RVM` για να επιτύχει αξιοπιστία αναγκάζει όλες τις ενημερώσεις που γίνονται στο ημερολόγιο να γράφονται στον τοπικό δίσκο. Από την άλλη το `RRVM-FDDI` και το `RRVM-ETHERNET` εκμεταλλεύονται την απομακρυσμένη



Σχήμα 4.4: **Η απόδοση του RVM σαν συνάρτηση του μεγέθους της εγγραφής της κάθε δοσοληψίας. Μέγεθος ημερολογίου 8 MB – σειριακές προσπελάσεις.**

μνήμη και έτσι επιτυγχάνουν αυτή την επιτάχυνση. Αυτό είναι αναμενόμενο γιατί στον δίσκο πληρώνουμε το κόστος της καθυστέρησης περιστροφής και αναζήτησης πράγμα που δεν έχουμε στα δίκτυα.

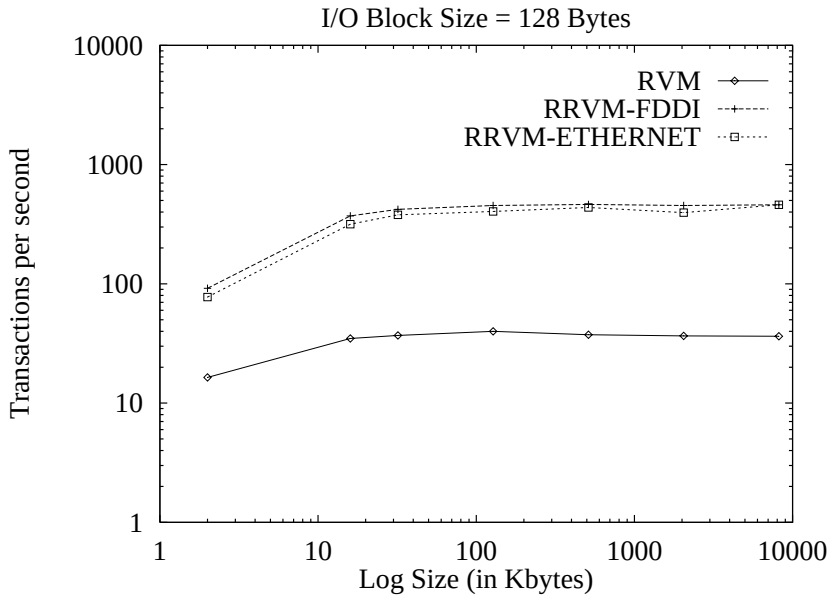
Ένα άλλο σημείο που αξίζει προσοχή είναι η απόδοση στο FDDI και στο Ethernet. Μελετώντας τις δύο αυτές καμπύλες παρατηρούμε ότι έχουν σχεδόν την ίδια απόδοση, για μικρές εγγραφές. Κάτι τέτοιο ίσως φαίνεται παράλογο αν σκεφτεί κανείς ότι το FDDI έχει θεωρητική παροχή δεκαπλάσια του Ethernet. Όμως στις μικρές δοσοληψίες που τα δεδομένα που θα μεταφέρουμε είναι πολύ λίγα, αυτό που επηρεάζει είναι η καθυστέρησή και όχι ο ρυθμός παροχής δεδομένων του δικτύου. Η καθυστέρηση όμως είναι σχεδόν ίδια στις δύο αυτές περιπτώσεις. Όταν οι δοσοληψίες μεγαλώνουν σε μέγεθος το χάσμα ανάμεσα στο `RRVM-FDDI` και το `RRVM-ETHERNET` μεγαλώνει, γιατί πλέον παίζει μεγαλύτερο ρόλο η παροχή του δικτύου.

Επαναλάβαμε το πείραμα, όμως αυτή τη φορά θέσαμε μέγεθος ημερολογίου ίσο με 8 MB. Τα αποτελέσματα, τα οποία παρουσιάζουμε στο σχήμα 4.4, έχουν την ίδια μορφή μόνο που τώρα η απόδοση σε όλες τις καμπύλες είναι καλύτερη. Αυτό συμβαίνει γιατί έχουμε μεγαλύτερο ημερολόγιο, και λόγω αυτού λιγότερες αναζητήσεις σε αυτό και λιγότερες αντιγραφές των δεδομένων μας στο αρχείο δεδομένων.

Τέλος κάτι που παρατηρούμε σε όλες τις καμπύλες είναι ότι όσο μεγαλώνει το μέγεθος της δοσοληψίας και τα τρία συστήματα συγκλίνουν αλλά με το `RRVM-FDDI` να διατηρεί την καλύτερη απόδοση.

4.3.2 Μέγεθος του Ημερολογίου

Σε αυτό το πείραμα αλλάζουμε το βήμα 5 του προηγούμενου πειράματος. Έτσι διατηρούμε το μέγεθος της δοσοληψίας σταθερό και μεταβάλλουμε το μέγεθος του



Σχήμα 4.5: **Η απόδοση του RVM σαν συνάρτηση του μεγέθους του ημερολογίου. Μέγεθος εγγραφής από κάθε δοσοληψία 128 Bytes – σειριακές προσπελάσεις.**

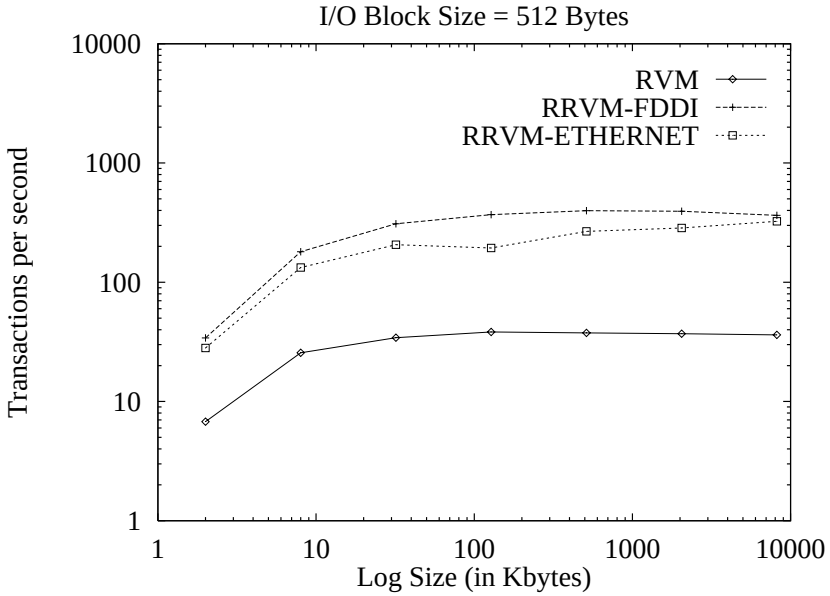
ημερολογίου. Δεδομένου ότι οι δοσοληψίες συνήθως έχουν μικρό μέγεθος παρουσιάζουμε τα αποτελέσματα μας, στα σχήματα 4.5 και 4.6 για δοσοληψίες μεγέθους 128 και 512 bytes αντίστοιχα.

Οι καμπύλες στα δύο σχήματα έχουν ένα βασικό κοινό στοιχείο, το οποίο είναι πιο εμφανές στη δεύτερη εικόνα. Και τα τρία συστήματα έχουν χαμηλή απόδοση όταν το μέγεθος του ημερολογίου είναι μικρό και βελτιώνονται σταδιακά καθώς το μέγεθος του μεγαλώνει. Η εξήγηση είναι ότι όταν έχουμε μικρό ημερολόγιο αναγκαζόμαστε να κάνουμε αρκετά συχνά δύο σύγχρονες εγγραφές, μία στο ημερολόγιο και μία στο αρχείο δεδομένων λόγω του ότι το ημερολόγιο γέμισε. Με ημερολόγια μεγέθους μεγαλύτερα από 32 KB αυτό το πρόβλημα αυτό δεν υφίσταται πλέον.

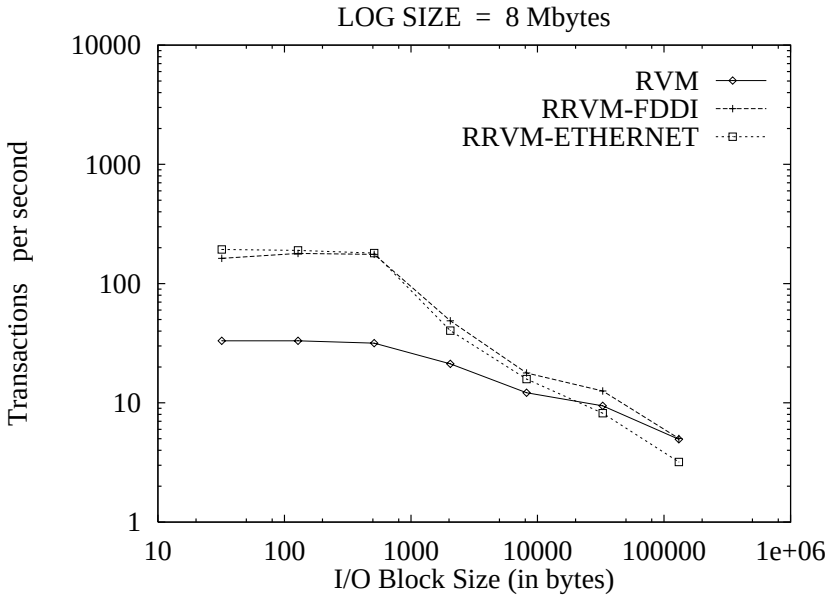
4.3.3 Τυχαίες Προσπελάσεις

Συνεχίζουμε να μεταβάλλουμε τις παραμέτρους του πειράματός μας για να έχουμε μια σφαιρική άποψη. Αυτή τη φορά αλλάζουμε το βήμα 4 και κάνουμε τις προσπελάσεις μας με τυχαίο τρόπο. Στο σχήμα 4.7 φαίνονται τα αποτελέσματα για μέγεθος ημερολογίου 8 MB.

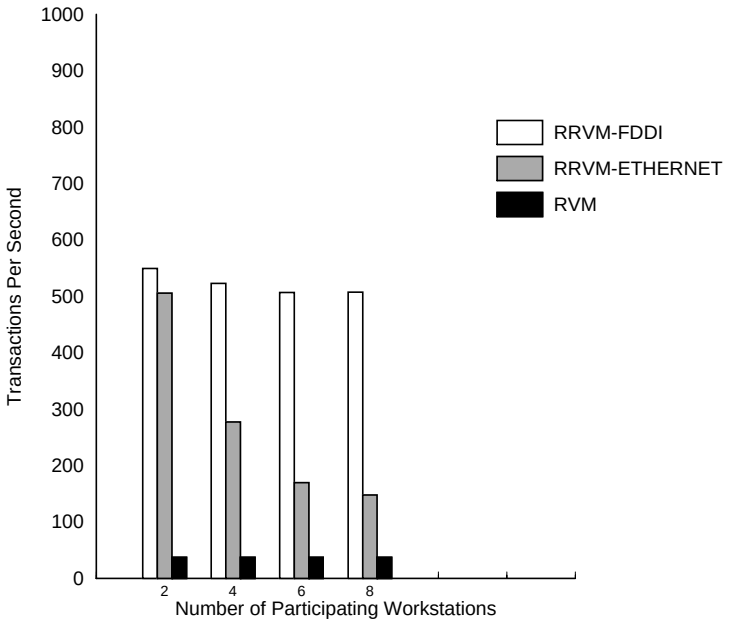
Παρόλο που συνεχίζουμε να έχουμε πολύ καλύτερη απόδοση στο `RRVM-FDDI` και στο `RRVM-Ethernet` από ότι στο `RVM` παρατηρούμε μία πτώση της απόδοσης. Αυτό συμβαίνει για δύο λόγους: αυξημένα λάθη σελίδας και πράξεις εγγραφής. Κάνοντας 10000 δοσοληψίες όπου η κάθε μία γράφει 32 bytes δεδομένων, τελικά προσπελαύνουμε 320 KB δεδομένων. Με σειριακή προσπέλαση και για σελίδες μεγέθους 8 KB αυτό είναι 40 σελίδες δεδομένων. Όταν όμως οι προσπελάσεις είναι τυχαίες τα λάθη σελίδας και οι προσπελάσεις στο δίσκο είναι πολύ περισσότερες.



Σχήμα 4.6: **Η απόδοση του RVM σαν συνάρτηση του μεγέθους του ημερολογίου. Μέγεθος εγγραφής από κάθε δοσοληψία 512 Bytes – σειριακές προσπελάσεις.**



Σχήμα 4.7: **Η απόδοση του RVM σαν συνάρτηση του μεγέθους της εγγραφής της κάθε δοσοληψίας – με τυχαία προσπέλαση.**



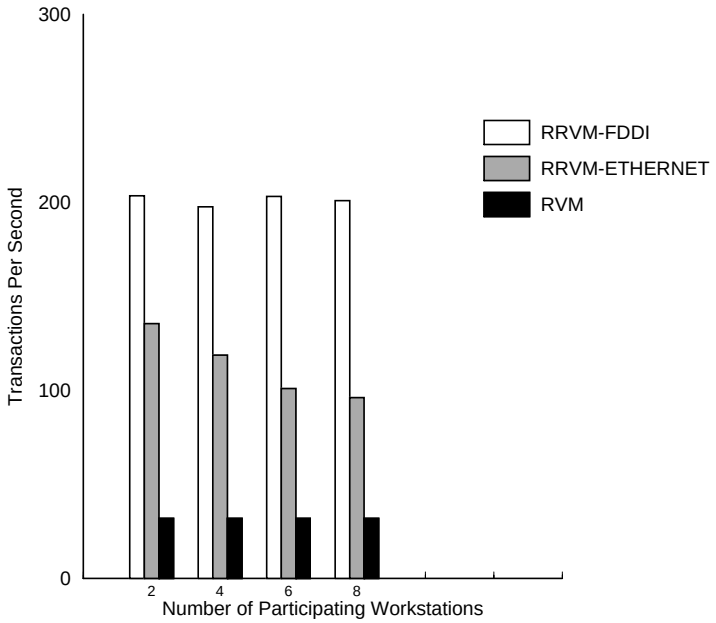
Σχήμα 4.8: **Φόρτος Δικτύου: Η απόδοση του RVM σαν συνάρτηση του φόρτου του δικτύου. Μέγεθος εγγραφής από κάθε δοσοληψία 32 bytes.**

4.3.4 Φόρτος Δικτύου

Αυτό το πείραμα έχει σκοπό να ερευνήσει πώς επηρεάζεται το `RVM` από τον φόρτο του δικτύου. Στα προηγούμενα πειράματά μας τρέχαμε ένα `RRVM` σε ένα μηχάνημα και τον εξυπηρετητή μνήμης του σε ένα άλλο, τώρα θα αυξήσουμε αυτά τα ζευγάρια για να μελετήσουμε την απόδοση του `RRVM`.

Παρουσιάζουμε τα αποτελέσματά μας στα σχήματα 4.8 και 4.9. Έχουμε σταθερό μέγεθος αρχείου δεδομένων 100 MB, ημερολόγιο 8 MB και μέγεθος δοσοληψίας 32 bytes και 2 Kbytes αντίστοιχα. Στο σχήμα 4.8 βλέπουμε το πλήθος των δοσοληψιών ανά δευτερόλεπτο για δύο, τέσσερις, έξι και οκτώ σταθμούς εργασίας που συμμετέχουν στο πείραμα.

Καταρχήν παρατηρούμε είναι ότι ο αριθμός των δοσοληψιών το δευτερόλεπτο που πετυχαίνει το `RVM` είναι σταθερός. Αυτό φυσικά ήταν αναμενόμενο γιατί δεν χρησιμοποιούμε καθόλου το δίκτυο. Κοιτώντας τώρα τα `RRVM-FDDI` και `RRVM-Ethernet` κάνουμε δύο παρατηρήσεις. Πρώτ' απ' όλα στην περίπτωση του FDDI η απόδοσή μας παραμένει σταθερή στις 500 δοσοληψίες ανά δευτερόλεπτο. Αντίθετα στην περίπτωση του Ethernet η απόδοση πέφτει αλλά ακόμα και στην χειρότερη περίπτωση πετυχαίνουμε 150 δοσοληψίες ανά δευτερόλεπτο, δηλαδή περισσότερο από τρεις φορές μεγαλύτερη απόδοση από το `RVM`. Η διαφορά μεταξύ του Ethernet και του FDDI ήταν αναμενόμενη δεδομένου ότι το FDDI έχει δέκα φορές μεγαλύτερο ρυθμό μεταγωγής δεδομένων από το Ethernet. Και στην δεύτερη εικόνα παρουσιάζονται αντίστοιχα αποτελέσματα με μειωμένη απόδοση όμως λόγω του ότι οι δοσοληψίες είναι μεγαλύτερες. Το συμπέρασμα που βγάζουμε είναι ότι τα σύγχρονα διασυνδετικά δίκτυα μπορούν να αντεπεξέλθουν στο φόρτο που τους βάζουμε.



Σχήμα 4.9: **Φόρτος Δικτύου: Η απόδοση του RVM σαν συνάρτηση του φόρτου του δικτύου. Μέγεθος εγγραφής από κάθε δοσοληψία 2 Kbytes.**

4.3.5 Φόρτος Εξυπηρετητή

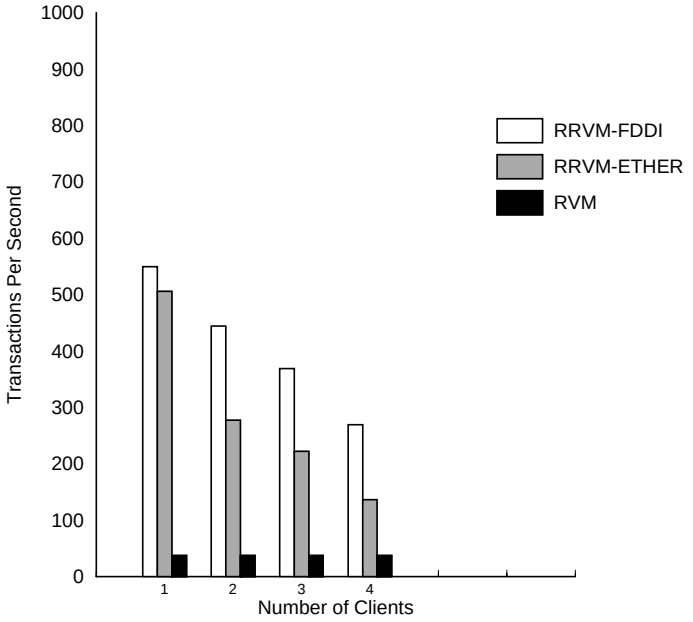
Το επόμενο και τελευταίο πείραμά μας φροντίζει να βάλει φόρτο όχι μόνο στο δίκτυο αλλά και σε κάποιο σταθμό εργασίας. Αυτή τη φορά λοιπόν τρέχουμε όλους τους εξυπηρετητές μνήμης σε ένα σταθμό εργασίας και τους πελάτες σε διαφορετικούς σταθμούς. Όλες οι υπόλοιπες παράμετροι είναι ίδιες με το προηγούμενο πείραμα.

Από τα σχήματα 4.10 και 4.11 βλέπουμε ότι η απόδοση των `RRVM-FDDI` και `RRVM-Ethernet` μειώνεται αλλά παραμένει αρκετά μεγαλύτερη από του `RVM` ακόμα και στην περίπτωση των τεσσάρων πελατών.

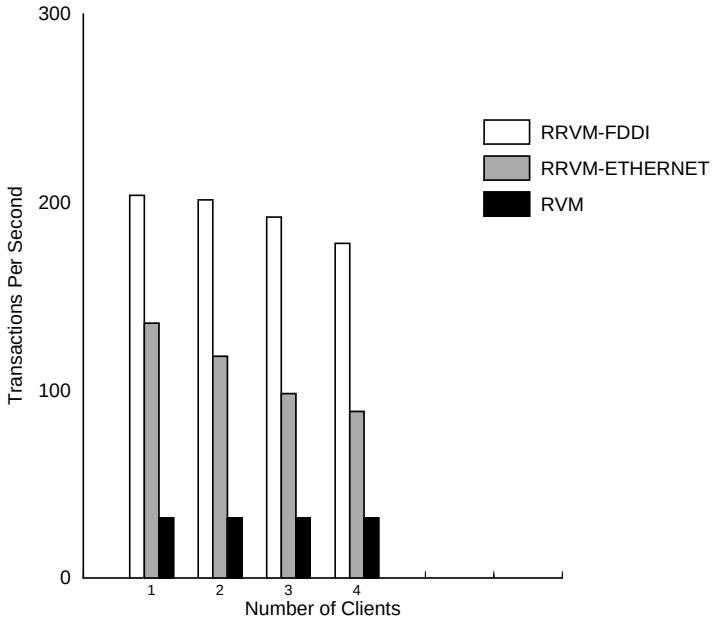
4.4 Πειράματα του NVRAM

Για να δοκιμάσουμε τη βιβλιοθήκη που σχεδιάσαμε και υλοποιήσαμε κάναμε μία σειρά πειραμάτων αντίστοιχα με αυτά του RVM. Για τα πειράματά μας είχαμε πάλι τις τρεις διαφορετικές αρχιτεκτονικές που είχαμε στο RVM και ακόμα μία πάνω από SCI:

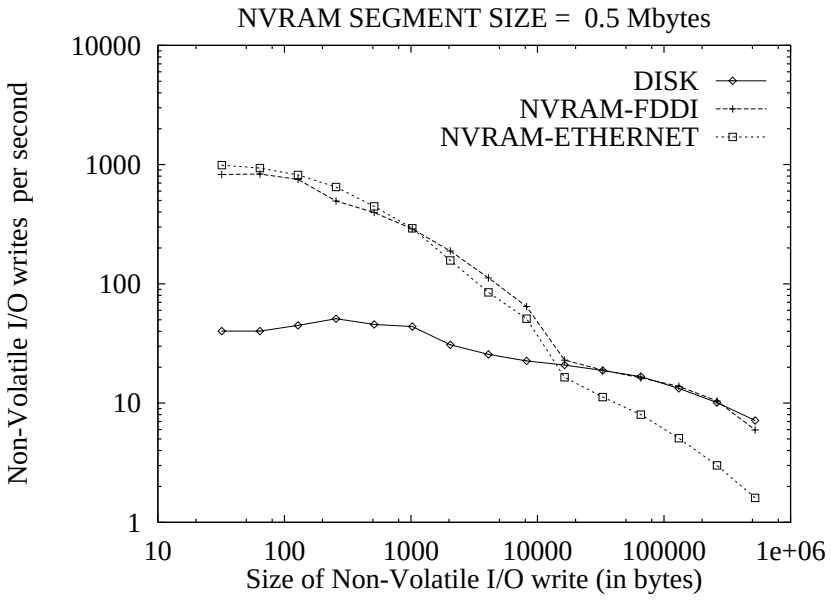
- `DISK`: Όλες οι λειτουργίες που χρειάζονται αξιοπιστία χρησιμοποιούν το δίσκο. Δηλαδή όλες οι εγγραφές γίνονται σύγχρονα σε αυτόν.
- `NVRAM-FDDI`: Την αξιοπιστία την προσφέρει η βιβλιοθήκη μας, NVRAM. Χρησιμοποιεί την απομακρυσμένη μνήμη και κάνει εγγραφές σε αυτή πάνω από διασυνδετικό δίκτυο τύπου FDDI.
- `NVRAM-ETHERNET`: Αυτή τη φορά το διασυνδετικό δίκτυο είναι τύπου ETHERNET.
- `NVRAM-SCI`: Το ίδιο με διασυνδετικό δίκτυο τύπου SCI.



Σχήμα 4.10: **Φόρτος Εξυπηρετητή: Η απόδοση του RVM σα συνάρτηση του φόρτου του εξυπηρετητή. Μέγεθος εγγραφής από κάθε δοσοληψία 32 bytes.**



Σχήμα 4.11: **Φόρτος Εξυπηρετητή: Η απόδοση του RVM σα συνάρτηση του φόρτου του εξυπηρετητή. Μέγεθος εγγραφής από κάθε δοσοληψία 2 Kbytes.**



Σχήμα 4.12: **Η απόδοση του NVRAM με Τμήμα Σταθερής Απομακρυσμένης Μνήμης 512 KB.**

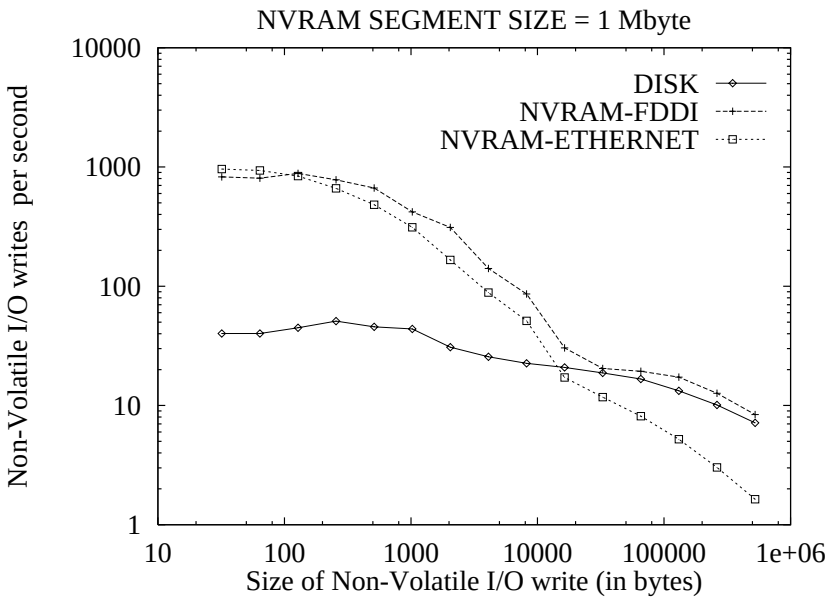
4.4.1 Μέγεθος της Εγγραφής

Το πρώτο μας πείραμα είναι όμοιο με το πρώτο πείραμα του RVM (βλέπε σχήματα 4.12, 4.13 και 4.14). Αυτό που είναι ξεκάθαρο είναι ότι το `NVRAM-ETHERNET` και `NVRAM-FDDI` ξεπερνούν κατά πολύ σε απόδοση το `DISK`. Ειδικότερα για μικρές εγγραφές μέχρι και 128 bytes η απόδοσή τους είναι περίπου δύο τάξεις μεγέθους μεγαλύτερη από αυτή του δίσκου. Ο λόγος είναι πάλι ο ίδιος, οι δίσκοι έχουν το επιπλέον κόστος περιστροφής και αναζήτησης που δεν έχουν τα δίκτυα. Καθώς μεγαλώνει το μέγεθος της εγγραφής η απόδοση των τριών συστημάτων συγκλίνει και ενώ στην περίπτωση του Ethernet η απόδοση γίνεται χειρότερη από αυτή του δίσκου, το FDDI παραμένει καλύτερο.

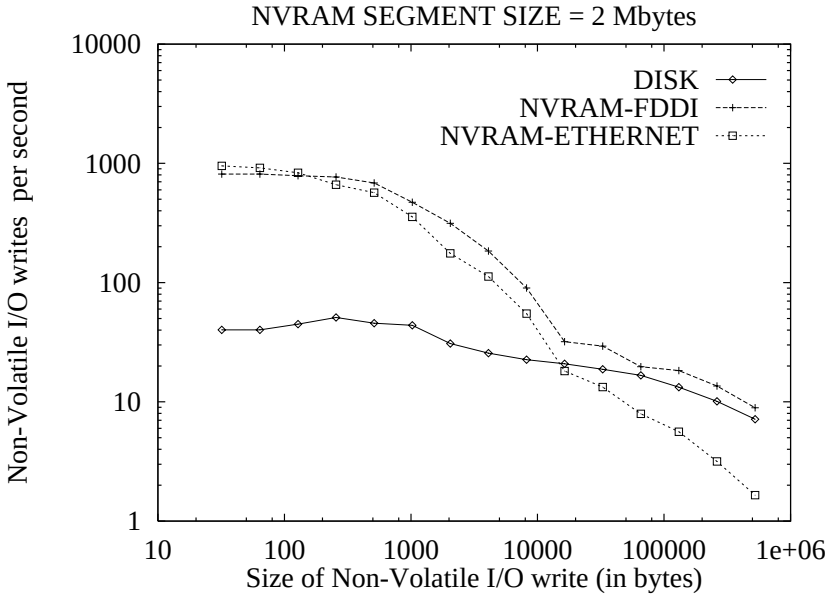
Ο λόγος που παρατηρείται το φαινόμενο αυτό είναι ότι στις μεγάλες πράξεις εγγραφής ο κύριος παράγοντας καθυστέρησης είναι ο ρυθμός μεταφοράς δεδομένων. Έτσι ενώ ο δίσκος και το FDDI έχουν συγκρίσιμους ρυθμούς μεταφοράς δεδομένων το Ethernet έχει πολύ χαμηλότερο γι' αυτό και η μεγάλη διαφορά στις μεγάλες εγγραφές.

Τα αποτελέσματά μας αποδεικνύουν ότι το NVRAM θα μπορούσε να βοηθήσει κατά πολύ την απόδοση σε συστήματα δοσοληψιών, και αυτό γιατί συνήθως οι δοσοληψίες γράφουν ένα μικρό ποσό πληροφορίας. Ένα ακόμα σημαντικό συμπέρασμα είναι ότι και δίκτυα χαμηλού ρυθμού μεταφοράς δεδομένων μπορούν να μας προσφέρουν τη βελτίωση αυτή.

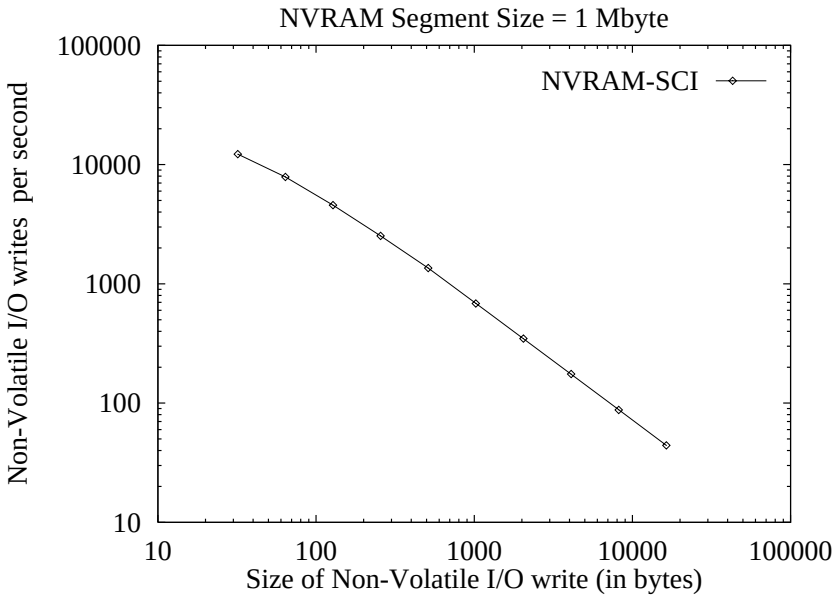
Είχαμε την ευκαιρία να χρησιμοποιήσουμε ακόμα ένα διασυνδετικό δίκτυο, το SCI, το οποίο προσφέρει καθυστέρηση δικτύου μικρότερη από Ethernet και FDDI. Υλοποιήσαμε λοιπόν την βιβλιοθήκη μας και πάνω από SCI. Πραγματοποιήσαμε το προηγούμενο πείραμα και η απόδοση που πήραμε ήταν μία τάξη μεγέθους καλύτερη από το Ethernet και το FDDI και τρεις τάξεις μεγέθους καλύτερη από το δίσκο. Το σύστημά μας μπόρεσε να κάνει πάνω από 12000 σταθερές εγγραφές ανά δευτερόλεπτο.



Σχήμα 4.13: **Η απόδοση του NVRAM με Τμήμα Σταθερής Απομακρυσμένης Μνήμης 1 MB.**



Σχήμα 4.14: **Η απόδοση του NVRAM με Τμήμα Σταθερής Απομακρυσμένης Μνήμης 2 MB.**



Σχήμα 4.15: **Η απόδοση του NVRAM με Τμήμα Σταθερής Απομακρυσμένης Μνήμης 1 MB, πάνω από το διασυνδετικό δίκτυο SCI.**

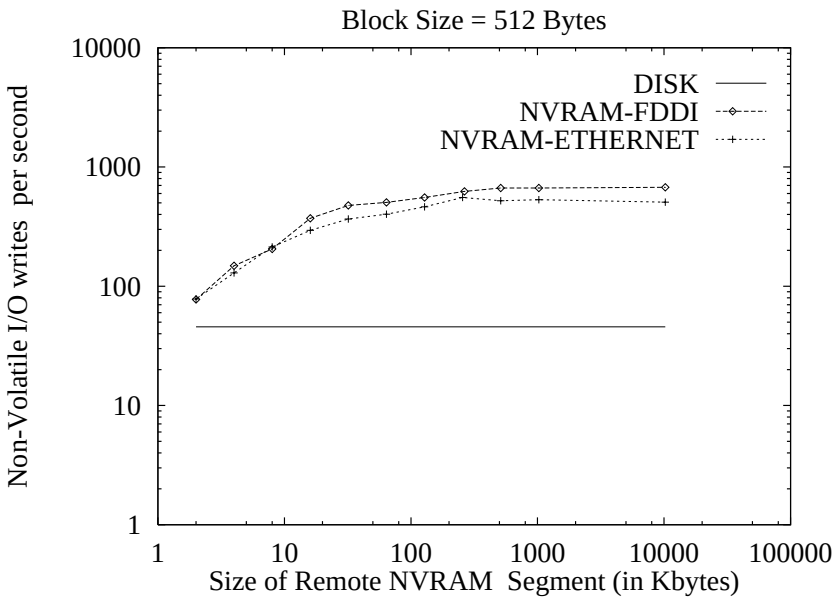
Αυτό είναι δυνατόν γιατί το SCI μπορεί να κάνει εγγραφές και αναγνώσεις σε μακρινές μνήμες μέσα σε μερικές δεκάδες μικροδευτερόλεπτα. Τα αποτελέσματα φαίνονται στο σχήμα 4.15.

4.4.2 Μέγεθος Τμήματος Απομακρυσμένης Σταθερής Μνήμης

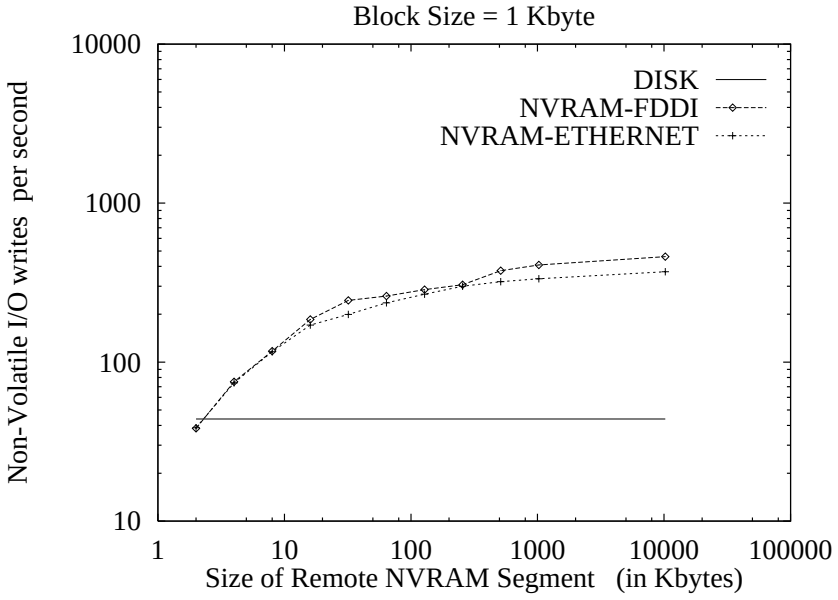
Κάναμε ακόμα ένα πείραμα στο NVRAM, αυτή τη φορά διατηρούμε σταθερή τη μονάδα εγραφής και μεταβάλουμε το μέγεθος του τμήματος απομακρυσμένης σταθερής μνήμης. Αυτό το κάνουμε γιατί θα θέλαμε να ξέρουμε πόση απομακρυσμένη μνήμη χρειαζόμαστε για να μπορέσουμε να αποφύγουμε την καθυστέρηση του δίσκου χωρίς να χάσουμε αξιοπιστία. Στο σχήμα 4.16 φαίνονται τα αποτελέσματά μας.

Φυσικά η απόδοση του δίσκου δεν επηρεάζεται από το ποσό της απομακρυσμένης μνήμης. Αντίθετα στην περίπτωση του `NVRAM-FDDI` και του `NVRAM-Ethernet` όσο αυξάνει το τμήμα, τόσο καλύτερη απόδοση έχουμε. Και αυτό ήταν κάτι αναμενόμενο αλλά όχι το ζητούμενο. Αυτό που θέλουμε να δούμε εμφανίζεται όταν θέσουμε ένα τμήμα απομακρυσμένης μνήμης ίσο με 512 KB. Μέχρι εκείνο το σημείο όσο αυξάναμε το τμήμα απομακρυσμένης μνήμης τόσο βελτιωνόταν η απόδοση, μετά από τα 512 KB η απόδοση παραμένει σχεδόν σταθερή.

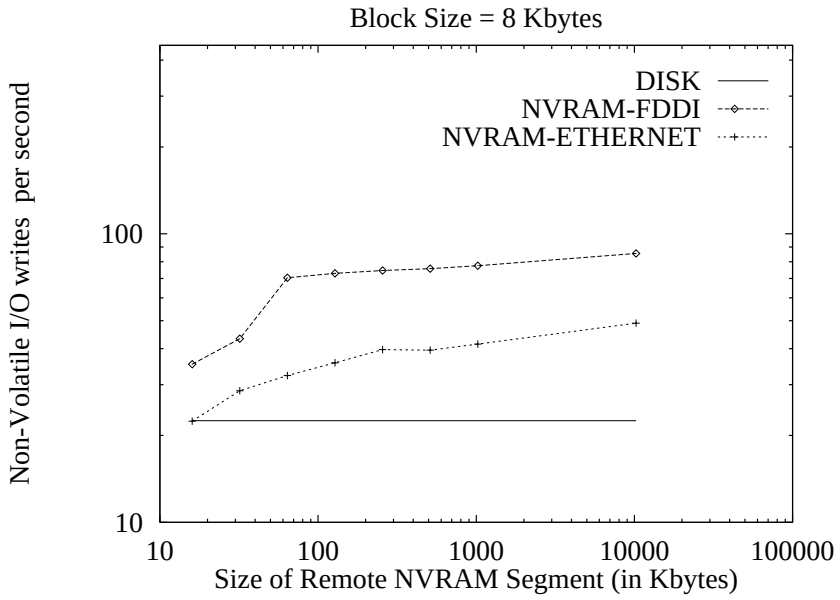
Στα σχήματα 4.17 και 4.18 βλέπουμε το ίδιο φαινόμενο. Συγκρίνοντας και τα τρία σχήματα μπορούμε να συμπεράνουμε ότι ένα τμήμα απομακρυσμένης μνήμης ίσο με ένα MB, μπορεί να δώσει πολύ καλά αποτελέσματα, γεγονός ενθαρρυντικό γιατί δείχνει ότι δεν μπορούμε να πετύχουμε αξιοπιστία δίνοντας ελάχιστους πόρους.



Σχήμα 4.16: **Η απόδοση του NVRAM όταν κάνουμε εγγραφές μεγέθους 512 bytes.**



Σχήμα 4.17: **Η απόδοση του NVRAM όταν κάνουμε εγγραφές μεγέθους 1KB.**



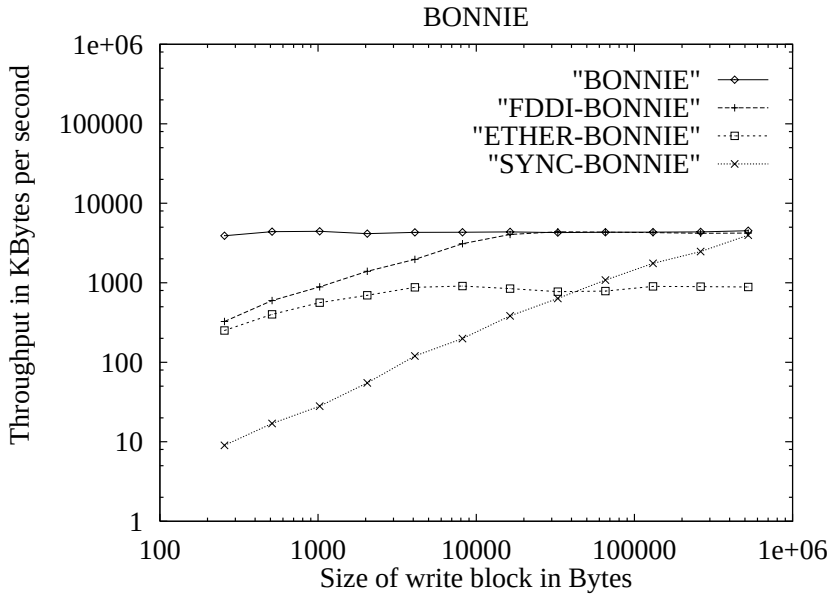
Σχήμα 4.18: **Η απόδοση του NVRAM όταν κάνουμε εγγραφές μεγέθους 8 KB.**

4.4.3 Σημείο αναφοράς Bonnie

Το Bonnie είναι ένα σημείο αναφοράς για το ρυθμό μεταγωγής δεδομένων του δίσκου Αποτελείται από μία σειρά εγγραφών, αναγνώσεων και αναζητήσεων μέσα σε ένα αρχείο. Οι εγγραφές στο Bonnie είναι ασύγχρονες. Για να πειραματιστούμε κάναμε το Bonnie να κάνει σύγχρονες εγγραφές και επιπλέον το κάναμε να χρησιμοποιεί και τη βιβλιοθήκη μας NVRAM. Για τις ανάγκες του πειράματος μεταβάλουμε το μέγεθος της εγγραφής και μετράμε το ρυθμό μεταγωγής δεδομένων.

- `BONNIE`: Το σημείο αναφοράς.
- `BONNIE-FDDI`: Το Bonnie με τη χρήση της NVRAM πάνω απο το διασυνδετικό δίκτυο FDDI.
- `BONNIE-ETHERNET`: Το ίδιο με παραπάνω αλλά αυτή τη φορά το διασυνδετικό δίκτυο είναι τύπου ETHERNET.
- `SYNC-BONNIE`: Το σημείο αναφοράς Bonnie όταν κάνει όλες τις εγγραφές σύγχρονα.

Η πρώτη παρατήρηση είναι ότι με τις ασύγχρονες εγγραφές το Bonnie έχει το μέγιστο ρυθμό μεταγωγής δεδομένων. Όταν κάνουμε τις εγγραφές σύγχρονα η απόδοση αρχίζει από πολύ χαμηλά αλλά καθώς το μέγεθος της εγγραφής μεγαλώνει η απόδοση φτάνει να είναι ίδια με τις ασύγχρονες εγγραφές. Η πιο ενδιαφέρουσα παρατήρηση όμως είναι ότι στην περίπτωση που χρησιμοποιήσαμε την βιβλιοθήκη NVRAM πάνω από FDDI επιτύχαμε και σταθερότητα στα δεδομένα μας αλλά και επίδοση παρόμοια με τις ασύγχρονες εγγραφές, όταν το μέγεθος της εγγραφής έφτασε τα 128 Kbytes. Επομένως συμπεραίνουμε ότι είναι δυνατό να επιτύχουμε σταθερότητα στα δεδομένα μας χωρίς να χάσουμε σχεδόν καθόλου σε απόδοση.



Σχήμα 4.19: **Σημείο αναφοράς Bonnie.**

Κεφάλαιο 5

Σχετική Εργασία

Στο κεφάλαιο αυτό παρουσιάζονται συστήματα τα οποία έχουν κάποια σχέση με την εργασία αυτή. Τα συστήματα αυτά παρουσιάζονται στις επόμενες ενότητες και εξετάζονται σε σχέση με το σύστημά μας.

5.1 Συστήματα που Βασίζονται σε Ειδικό Υλικό

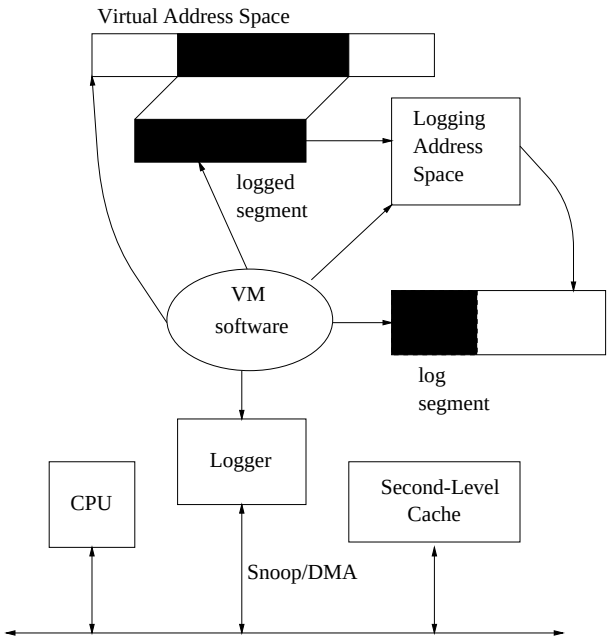
5.1.1 LVM

Η Καταγραφόμενη Ιδεατή Μνήμη ή Logged Vitrual Memory (LVM) [CD95], η οποία σχεδιάστηκε από τους Cheriton και Duda, παρέχει ένα ημερολόγιο των εγγραφών σε ένα ή περισσότερα δοσμένα τμήματα ιδεατής μνήμης. Η καταγραφή αυτή μπορεί να είναι χρήσιμη σε εφαρμογές που θέλουν επαναφορά και σταθερότητα. Πιο συγκεκριμένα η Καταγραφόμενη Ιδεατή Μνήμη είναι μία επέκταση του συστήματος ιδεατής μνήμης. Το πρωτότυπο αποτελείται από μία κάρτα, τον *καταγραφέα (logger)*, ο οποίος παρακολουθεί το δίαυλο του επεξεργαστή ParaDiGM, και καταγράφει τις εγγραφές που γίνονται, και μία επέκταση στο σύστημα ιδεατής μνήμης του πυρήνα V++ Cache Kernel για να συσχετίζει το τμήμα του ημερολογίου με ένα τμήμα μνήμης. Το σύστημα φαίνεται στο σχήμα 5.1.

Όπως αναφέραμε προηγουμένως, ο καταγραφέας παρακολουθεί τον δίαυλο για εγγραφές και τις καταγράφει στο ημερολόγιο (log segment). Όταν γεμίσει το ημερολόγιο τότε ο καταγραφέας το στέλνει στη μνήμη με DMA. Ένα σύστημα λοιπόν που θα ήθελε να ξέρει πια δεδομένα έχουν γραφτεί, όπως για παράδειγμα το RVM, μπορεί να χρησιμοποιήσει το LVM.

5.1.2 eNVy

Το eNVy είναι ένα σύστημα που αναπτύχθηκε στο πανεπιστήμιο Rice από τους Wu και Zwaenepoel, με στόχο την αποφυγή της χρήσης του δίσκου για την αποθήκευση δεδομένων [WZ95] ώστε να επιταχύνει εγγραφές που πρέπει να γίνουν σε σταθερό μέσο. Χρησιμοποιεί μνήμες ειδικού τύπου Flash, οι οποίες διατηρούν τα περιεχόμενά τους χωρίς την ανάγκη ύπαρξης οποιασδήποτε πηγής ενέργειας, είναι λίγο ακριβότερες από την κύρια μνήμη, και έχουν τον ίδιο χρόνο ανάγνωσης με αυτήν. Το μειονέκτημά τους είναι ότι δεν μπορεί να γίνει εγγραφή μιας λέξης μόνο, αλλά αντίθετα ένα ολόκληρο μπλοκ μνήμης πρέπει να σβηστεί και να επαναπρογραμματιστεί κάθε φορά, διαδικασία που διαρκεί αρκετά μs. Για να ξεπεράσουν αυτό το μειονέκτημα



Σχήμα 5.1: **Διάγραμμα του συστήματος LVM.**

χρησιμοποιούν και μια μικρή στατική μνήμη SRAM σαν ενταμιευτή εγγραφής. Με τα προσομοιομένα αποτελέσματά τους υποστηρίζουν ότι ένα σύστημα eNVy των 2 GB μπορεί να υποστηρίξει 30.000 δοσοληψίες ανά δευτερόλεπτο. Το όλο σύστημα φαίνεται στο σχήμα 5.2.

5.1.3 Σύγκριση

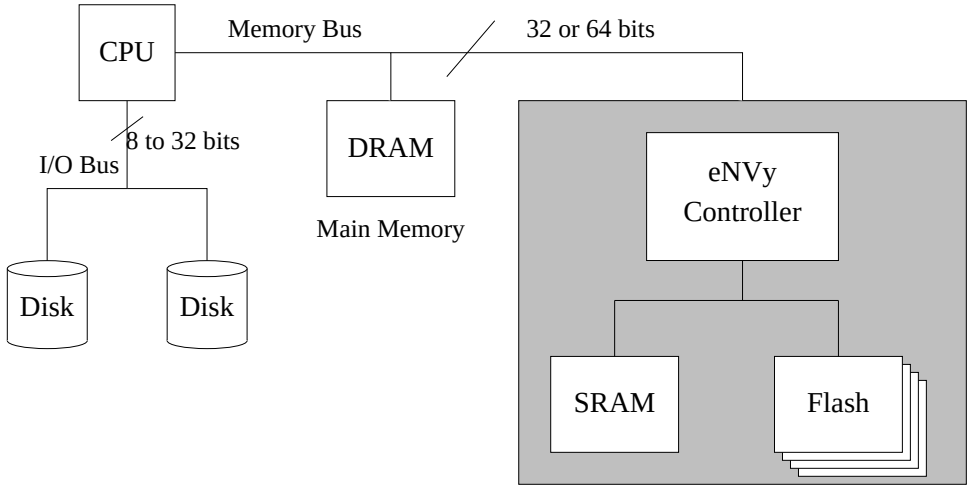
Τα συστήματα που παρουσιάσαμε αυξάνουν την κύρια μνήμη που διαθέτει ο σταθμός εργασίας, ή δίνουν τη δυνατότητα να παρακολουθήσουμε τις εγγραφές σε αυτή. Ο στόχος παραμένει ο ίδιος: να μειωθούν οι προσπελάσεις στο δίσκο. Το βασικό μειονέκτημα τους είναι οτι απαιτούν ειδικό υλικό, το οποίο αυξάνει το κόστος του υπολογιστή.

Με τη σχεδίαση μας εμείς προσπαθούμε να εκμεταλλευτούμε την απομακρυσμένη μνήμη χωρίς προσθήκη επιπλέον μνήμης ή ειδικού υλικού. Παρόλα αυτά, αν κάποιο μηχάνημα σε ένα τοπικό δίκτυο έχει τέτοιο ειδικό υλικό εμείς έχουμε τη δυνατότητα να το χρησιμοποιήσουμε προς όφελος όλων των σταθμών εργασίας.

5.2 Κατανεμημένα Συστήματα Αρχείων

5.2.1 HARP

Το HARP [LGG+91] (Highly Available, Reliable, Persistent file system), είναι ένα σύστημα αρχείων υψηλής διαθεσιμότητας και σταθερής φύλαξης αρχείων. Για να το πετύχει αυτό βασίζεται σε UPS έτσι ώστε να επιβιώνει πτώσεις τάσης και αντιγραφές των δεδομένων σε παραπάνω από ένα κόμβους του συστήματος για να επιτύχει υψηλή



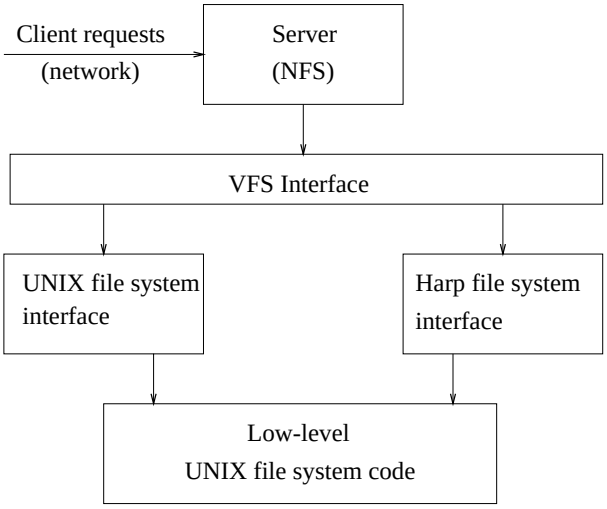
Σχήμα 5.2: **Διάγραμμα του συστήματος eNVy.**

διαθεσιμότητα. Όλες οι μεταβολές ενός αρχείου γράφονται σε μερικούς εξυπηρετητές που τρέχουν σε διαφορετικά μηχανήματα. Με αυτό τον τρόπο τα αρχεία μπορούν να αντέξουν καταστροφή των δεδομένων τους.

Η τεχνική που χρησιμοποιεί το HARP είναι αυτή του *πρωτεύοντος αντίγραφου*. Σύμφωνα με τη μέθοδο αυτή οι κλίσεις ενός πελάτη κατευθύνονται προς ένα μοναδικό *κύριο* εξυπηρετητή, ο οποίος επικοινωνεί με άλλους *δευτερεύοντες* εξυπηρετητές, και περιμένει τις δικές τους απαντήσεις πριν απαντήσει στον πελάτη. Με αυτό τον τρόπο το HARP βγάζει το δίσκο από το κρίσιμο μονοπάτι αντικαθιστώντας τις εγγραφές στο δίσκο με εγγραφές στους δευτερεύοντες εξυπηρετητές. Αυτή η τεχνική είναι παρόμοια με την δική μας, όμως έχουμε δύο πολύ βασικές διαφορές τις οποίες θα αναφέρουμε παρακάτω.

5.2.2 xFS

Το xFS είναι ένα κατανεμημένο σύστημα αρχείων στο οποίο δεν υπάρχει κάποιος κεντρικός εξυπηρετητής [ADN+95]. Ενώ τα καθιερωμένα συστήματα αρχείων δικτύου βασίζονται σε κεντρικούς σταθμούς εργασίας για την εξυπηρέτησή τους, στο xFS όλοι οι σταθμοί εργασίας ενός τοπικού δικτύου συνεργάζονται για να παρέχουν τις υπηρεσίες του συστήματος αρχείων. Οποιοδήποτε μηχάνημα στο δίκτυο μπορεί να φυλάξει και να ελέγξει τα δεδομένα των αρχείων. Δεδομένου ότι μπορούμε να έχουμε δεδομένα στη μνήμη οποιουδήποτε σταθμού εργασίας, οι αιτήσεις ανάγνωσης δε θα ικανοποιηθούν από το δίσκο παρά μόνο αν τα ζητούμενα δεδομένα δεν υπάρχουν στην κύρια μνήμη κανενός μηχανήματος του δικτύου. Επίσης επειδή τα δεδομένα είναι κατανεμημένα σε όλα τα μηχανήματα μπορούμε να έχουμε παράλληλες μεταφορές προς και από τους δίσκους. Τέλος, τεχνικές ισοτιμίας χρησιμοποιούνται για να έχουμε ομαλή λειτουργία στο σύστημα σε περίπτωση που κάποιος σταθμός πέσει.



Σχήμα 5.3: **Η δομή του συστήματος αρχείων Harp.**

5.2.3 Zebra

Το Zebra [HO93] είναι άλλο ένα σύστημα αρχείων δικτύου, το οποίο μοιράζει τα δεδομένα του σε πολλούς εξυπηρετητές προκειμένου να αυξήσει το ρυθμό μεταγωγής τους. Το σύστημα μπορεί να επιβιώσει τυχούσες βλάβες χάρη στην τεχνική της ισοτιμίας που χρησιμοποιεί. Η τεχνική αυτή μπορεί να επιβαρύνει το σύστημα και για να το αποφύγει αυτό, το Zebra βασίζεται σε ένα ημερολόγιο και έτσι όλες οι εγγραφές γίνονται σειριακά.

5.2.4 PACA

Το PACA είναι ένα άλλο σύστημα αρχείων το οποίο έχει υλοποιηθεί για χρήση από πολυεπεξεργαστές [CGL95]. Ο στόχος του είναι να χρησιμοποιήσει τη μνήμη όλων των επεξεργαστών σαν κρυφή μνήμη του συστήματος αρχείων. Με αυτό τον τρόπο προσπαθεί να αποφύγει λειτουργίες εισόδου/εξόδου προς και από το δίσκο οι οποίες αποτελούν κρίσιμο παράγοντα καθυστέρησης της εκτέλεσης παράλληλων προγραμμάτων. Η σχεδίαση αυτή έχει σαν σκοπό να είναι αποδοτική, απλή και κλιμακωτή.

5.2.5 Σύγκριση

Τα κατανεμημένα συστήματα αρχείων χρησιμοποιούν τη μνήμη όλων των κόμβων ενός τοπικού δικτύου ή όλων των επεξεργαστών της πολυεπεξεργαστικής μηχανής προκειμένου να αποφύγουν την προσπέλαση στο δίσκο. Αυτό το χαρακτηριστικό τους, είναι η βασική σχέση με το σύστημά μας. Βασικός τους στόχος είναι να αποσυμφορήσουν τον ένα κεντρικό εξυπηρετητή από το βάρος της εξυπηρέτησης των πολλών πελατών και ταυτόχρονα να βελτιώσουν την απόδοση των λειτουργιών εισόδου/εξόδου χρησιμοποιώντας δίκτυα μεγάλου ρυθμού μεταγωγής δεδομένων.

Αυτό που απασχολεί την δική μας εργασία είναι πώς θα πετύχουμε μεγαλύτερη απόδοση σε συστήματα δοσοληψιών. Οι διαφορές μας από τα παραπάνω συστήματα μπορούν να συμπυκνωθούν σε δύο βασικά σημεία:

1. *Μέγεθος δεδομένων:* Στα συστήματα δοσοληψιών με τα οποία ασχολούμαστε τα μεγέθη των αναγνώσεων και τον εγγραφών είναι μικρά. Αυτό συμβαίνει γιατί συνήθως οι δοσοληψίες μεταβάλλουν μικρά ποσά πληροφορίας. Στα συστήματα αρχείων το μέγεθος της ανάγνωσης και εγγραφής είναι η σελίδα. Κάθε φορά λοιπόν που θα θέλαμε να κάνουμε μία δοσοληψία μερικών bytes θα αναγκαζόμαστε να διαβάσουμε ή να γράψουμε ένα τμήμα δεδομένων μεγέθους σελίδας, δηλαδή 4 με 8 Kbytes. Κάτι τέτοιο όπως είναι φυσικό θα μείωνε την απόδοση του συστήματός μας.

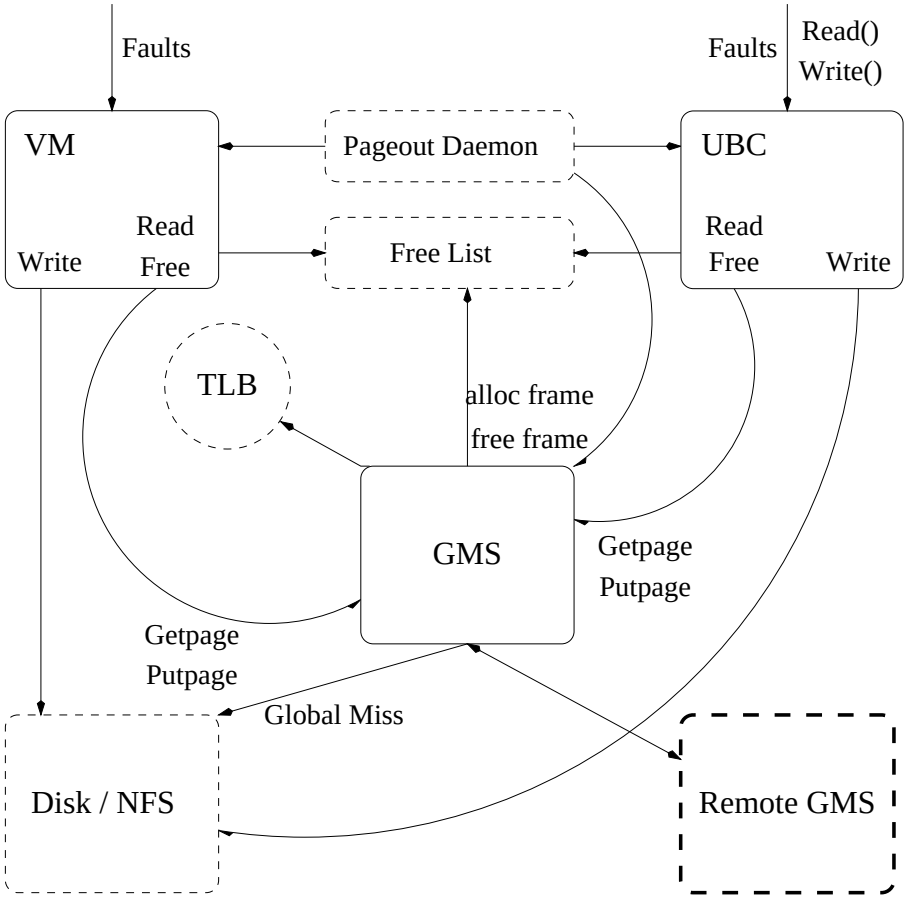
2. *Υλοποίηση σε επίπεδο χρήστη:* Το δεύτερο πολύ βασικό σημείο διαφοράς είναι το επίπεδο υλοποίησης. Τα συστήματά μας βρίσκονται όλα σε επίπεδο χρήστη, έξω από το λειτουργικό σύστημα. Αυτό τα κάνει μεταφέρσιμα και εύκολα στις αλλαγές. Δεν έχουμε αλλάξει καθόλου τον πυρήνα του λειτουργικού ούτε χρειάζεται να είμαστε υπερχρήστες για να τρέξουμε το σύστημά μας.

5.3 Υλοποίηση Κοινής Μνήμης σε ένα Δίκτυο Υπολογιστών

Στο πανεπιστήμιο της Washington υλοποίησαν ένα σύστημα, το GMS [FMP+95], που έχει τη δυνατότητα να χρησιμοποιεί την απομακρυσμένη μνήμη. Αυτό το πέτυχαν επανασχεδιάζοντας το σύστημα διαχείρισης μνήμης στο λειτουργικό σύστημα. Με αυτό τον τρόπο τα μηχανήματα ενός τοπικού δικτύου μπορούν να βλέπουν τη μνήμη όλων των μηχανών σαν ενιαία κοινή κατανεμημένη μνήμη. Το λειτουργικό σύστημα που άλλαξαν είναι το DEC OSF/1, το οποίο χρησιμοποιείται από τις μηχανές DEC Alpha. Η τροποποίηση του λειτουργικού έχει γίνει στο σύστημα διαχείρισης μνήμης όπως φαίνεται στο σχήμα 5.4. Σύμφωνα με τη νέα σχεδίαση η μνήμη κάθε κόμβου διακρίνεται σε *τοπική* που περιέχει δεδομένα ιδιωτικά του συγκεκριμένου κόμβου και *ολική* που περιέχει *καθαρές* σελίδες των υπόλοιπων κόμβων. Τα ποσοστά της τοπικής και της ολικής μνήμης διαμορφώνονται δυναμικά σύμφωνα με τις ανάγκες τόσο του τοπικού όσο και των υπολοίπων κόμβων. Με το νέο αυτό σύστημα μπόρεσαν και πέτυχαν επιταχύνσεις μέχρι και 3.5 φορές σε εφαρμογές που χρειάζονται μεγάλα ποσά μνήμης.

5.3.1 Σύγκριση

Και αυτό το σύστημα έχει κάποιες ομοιότητες με το σύστημά μας γιατί και αυτό χρησιμοποιεί την απομακρυσμένη μνήμη για να αποφύγει την προσπέλαση στο δίσκο. Το GMS δεν προσπαθεί να έχει σταθερότητα στα δεδομένα που διαχειρίζεται σε αντίθεση με το δικό μας. Επιπλέον, η βασική του λειτουργία είναι να βελτιστοποιήσει τις αναγνώσεις κρατώντας όσο το δυνατόν περισσότερα δεδομένα στις κύριες μνήμες, ενώ εμείς παρουσιάζουμε ένα τρόπο για γρήγορες και σταθερές εγγραφές.



Σχήμα 5.4: **Η τροποποίηση στο σύστημα διαχείρισης μνήμης του DEC OSF/1.**

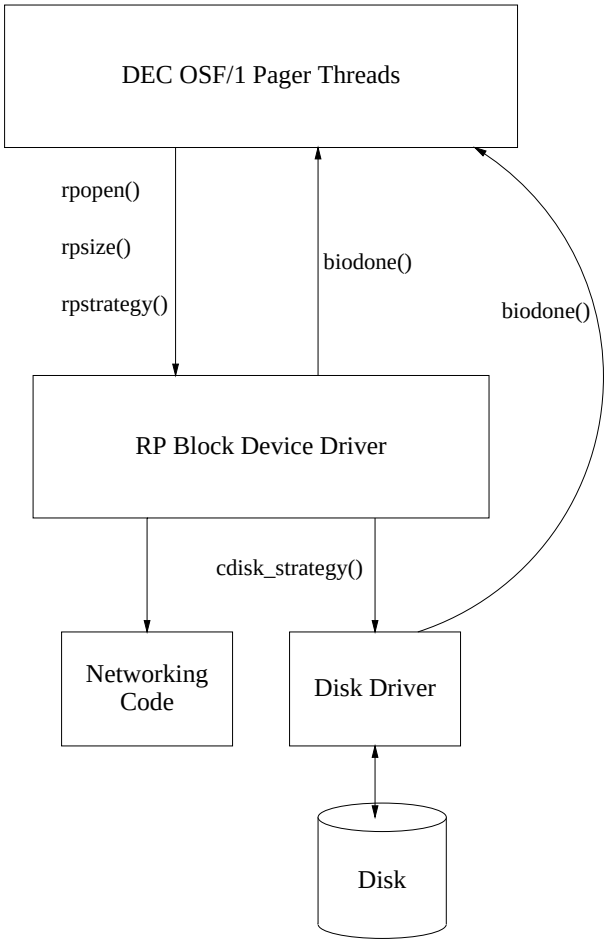
5.4 Σελιδοδιαχειριστές

5.4.1 Σελιδοδιαχείριση σε Απομακρυσμένη Μνήμη

Οι Δραμιτινός και Μαρκάτος στα [MD96, MD95, DM96] παρουσιάζουν και υλοποιούν ένα σύστημα το οποίο αποθηκεύει τις σελίδες του συστήματος ιδεατής μνήμης στην απομακρυσμένη μνήμη αντί για τον τοπικό δίσκο. Το σύστημά τους, σχήμα 5.5 έχει υλοποιηθεί σαν οδηγός συσκευής πάνω από το λειτουργικό σύστημα DEC OSF/1. Ο οδηγός δέχεται τις σελίδες από το σύστημα διαχείρισης μνήμης οι οποίες κατευθύνονται στο δίσκο και τις στέλνει σε εξυπηρετητές που τρέχουν σε μακρυνούς σταθμούς εργασίας. Το σύστημα χρησιμοποιεί μία πολιτική μεταβλητής ισοτιμίας για να εγγυηθεί την ακεραιότητα των δεδομένων σε περίπτωση πτώσης ενός σταθμού εργασίας.

5.4.2 Η Χρήση Απομακρυσμένης Μνήμης σε Φορητούς Υπολογιστές

Μια άλλη υλοποίηση συστήματος απομακρυσμένης μνήμης έχουμε στο πανεπιστήμιο Columbia. Η υλοποίηση έχει γίνει χρησιμοποιώντας εξωτερικούς εξυπηρετητές

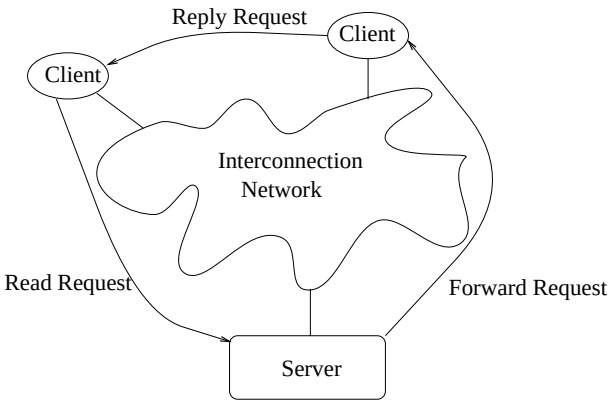


Σχήμα 5.5: **Η σύνδεση του σελιδοδιαχειριστή και η αλληλεπίδραση με το λειτουργικό σύστημα.**

μνήμης στο λειτουργικό σύστημα Mach 2.5 [SD91] και έχει στόχο τους φορητούς που δεν έχουν πολύ χώρο στο δίσκο. Η απόδοση του συστήματος αυτού είναι παραπλήσια με την απόδοση του τοπικού δίσκου. Λόγω της χρήσης εξωτερικού εξυπηρετητή μνήμης έχουμε μεγάλη επιβάρυνση από τα στρώματα λογισμικού, έτσι για τη μεταφορά μιας σελίδας μεγέθους 4 KB απαιτείται χρόνος ίσος με 45 ms.

5.4.3 Σύγκριση

Όπως βλέπουμε τα συστήματα που χρησιμοποιούν εξωτερικούς εξυπηρετητές μνήμης επιβαρύνονται πάρα πολύ σε κάθε μεταφορά σελίδας λόγω της αντιγραφής δεδομένων ανάμεσα σε πολλούς χώρους διευθύνσεων. Από την άλλη μία υλοποίηση σαν οδηγός συσκευής φαίνεται να έχει πολύ καλά αποτελέσματα. Το πρόβλημα που έχουν αυτές οι δύο υλοποιήσεις είναι καταρχήν ότι βρίσκονται μέσα στο λειτουργικό σύστημα και όχι όπως η δική μας σχεδίαση σε επίπεδο χρήστη. Επιπλέον, έχουμε πάλι μεταφορά και αποθήκευση δεδομένων μεγέθους σελίδας ενώ οι δοσοληψίες μπορεί να έχουν μέγεθος μόνο μερικών bytes.



Σχήμα 5.6: **Η ακολουθία μίας αίτησης ανάγνωσης.**

5.5 Καθολική Μνήμη σε Συστήματα Βάσεων Δεδομένων τύπου Πελάτη–Εξυπηρετητή

Οι Franklin, Carey και Livny στο πανεπιστημιο του Wisconsin–Madison εξετάζουν την χρήση της απομακρυσμένης μνήμης, σε ένα σύστημα βάσης δεδομένων (DBMS) τύπου πελάτη–εξυπηρετητή. Το σύστημα τους θεωρεί ένα κεντρικό εξυπηρετητή της βάσης ο οποίος περιέχει τους δίσκους για σταθερή αποθήκευση και αρκετή μνήμη για να φυλάσσει σελίδες της βάσης. Οι πελάτες αλληλεπιδρούν μεταξύ τους διαμέσου του κεντρικού εξυπηρετητή. Όταν ένα πελάτης κάνει μία αίτηση ανάγνωσης στον κεντρικό εξυπηρετητή και αυτός δεν έχει τη σελίδα στην κύρια μνήμη του, τότε ο κεντρικός εξυπηρετητής ερευνά αν κάποιος άλλος πελάτης έχει αυτή τη σελίδα στη κύρια μνήμη του. Αν αυτό είναι αλήθεια τότε ο κεντρικός εξυπηρετητής ζητάει από αυτόν τον πελάτη να προωθήσει τη σελίδα στον πελάτη που τη ζήτησε. Διαφορετικά ο κεντρικός εξυπηρετητής την διαβάζει από το δίσκο.

5.5.1 Σύγκριση

Με αυτή τη σχεδίαση του συστήματός τους προσπαθούν να ελαχιστοποιήσουν τα αντίγραφα των σελίδων της βάσης δεδομένων που βρίσκονται στην κύρια μνήμη του κεντρικού εξυπηρετητή και των πελατών. Με αυτό τον τρόπο επιτυγχάνουν καλύτερη χρήση της και έτσι βελτιώνουν την απόκριση των αιτήσεων ανάγνωσης. Εμείς στη εργασία μας βελτιώνουμε και τις αιτήσεις εγγραφής

5.6 RAIDs

Τα RAIDs είναι συστήματα τα οποία αποτελούνται από πολλούς δίσκους στους οποίους αποθηκεύονται παράλληλα δεδομένα [CLG+94]. Με τη χρήση πολλών δίσκων παράλληλα αυξάνουμε σημαντικά το ρυθμό μεταγωγής δεδομένων και έτσι μειώνεται ο χρόνος που χρειάζονται οι λειτουργίες εισόδου/εξόδου. Το πρόβλημα που έχουν τα Raids είναι ότι υπάρχει μεγάλη πιθανότητα βλάβης σε κάποιο από τους δίσκους

που αποτελούνται. Για να αντιμετωπίσουν αυτό το πρόβλημα χρησιμοποιούν διάφορες πολιτικές ισοτιμίας οι οποίες έχουν υλοποιηθεί σε υλικό.

5.6.1 Σύγκριση

Στη εργασία μας χρησιμοποιούμε και εμείς μία πολιτική ισοτιμίας για την αντιμετώπιση τυχών βλαβών. Αυτό που κυρίως μας ενδιαφέρει είναι πώς θα επιτύχουμε σταθερότητα στη μνήμη και όχι στα δεδομένα που έχουν ήδη πάει στο δίσκο.

5.7 Συστήματα Κατανεμημένης Μοιραζόμενης Μνήμης

5.7.1 Συνδυάζοντας Συνέπεια και Ανάκτηση

Στο πανεπιστήμιο της Washington έχει γίνει επίσης έρευνα πάνω σε θέματα συνέπειας και ανάκτησης δεδομένων σε συστήματα κατανεμημένης μνήμης [FCanHML94, Cha95]. Έχουν υλοποιήσει ένα σύστημα κατανεμημένης μνήμης που βασίζεται πάνω στο RVM. Η συνέπεια επιτυγχάνεται χρησιμοποιώντας δοσοληψίες πάνω στα δεδομένα τα οποία πρόκειται να αλλαχτούν. Με αυτό τον τρόπο πριν μία δοσοληψία δεσμευθεί, τα δεδομένα του ημερολογίου προωθούνται σε όλους τους πελάτες και μετά δεσμεύεται.

5.7.2 Αντοχή σε Σφάλματα με τη βοήθεια Συντρόφου

Στο πανεπιστήμιο του Rochester οι Hunt και Scott θέλοντας να παρέχουν αντοχή σε σφάλματα σε ένα σύστημα κατανεμημένης διαμοιραζόμενης μνήμης ανέπτυξαν ένα σύστημα που χρησιμοποιεί απομακρυσμένη μνήμη γι' αυτό το σκοπό [HS96]. Η σχεδίασή τους είναι παρόμοια με τη δική μας. Για κάθε κόμβο υπάρχει ένας κόμβος–σύντροφος, όταν ο κόμβος θέλει να γράψει αξιόπιστα κάποια δεδομένα τα στέλνει στον κόμβο–σύντροφο του, και περιμένει επιβεβαίωση. Τώρα που τα δεδομένα βρίσκονται σε δύο μέρη ο κόμβος μπορεί να συνεχίζει την εργασία του, ταυτόχρονα ο κόμβος–σύντροφος στέλνει τα δεδομένα στο δίσκο του ασύγχρονα. Στην εργασία τους παρουσιάζουν συγκριτικά αποτελέσματα της χρήσης δίσκου για αντοχή σε σφάλματα και της χρήσης απομακρυσμένης μνήμης. Όπως είναι αναμενόμενο η απομακρυσμένη μνήμη δίνει πολύ καλύτερα αποτελέσματα.

5.7.3 Σύγκριση

Το πρώτο σύστημα είναι ένα σύστημα κατανεμημένης διαμοιραζόμενης μνήμης, το οποίο δεν την χρησιμοποιεί για αξιοπιστία ή για βελτίωση της απόδοσης της εφαρμογής. Το δεύτερο σύστημα είναι σχεδόν όμοιο με την υλοποίηση που έχουμε για το RVM και το NVRAM. Η μόνη διαφορά είναι ότι σε εμάς αυτός που στέλνει τα δεδομένα ασύγχρονα στο δίσκο είναι η εφαρμογή και όχι ο μακρυνός εξυπηρετητής.

Κεφάλαιο 6

Συμπεράσματα

Στην εργασία αυτή εξετάζουμε τη χρήση της απομακρυσμένης μνήμης για την βελτίωση της απόδοσης συστημάτων δοσοληψιών χωρίς όμως να χάσουμε την αξιοπιστία που μας δίνει η σταθερότητα του δίσκου. Περιγράφουμε τις αλλαγές που κάναμε σε δύο υπάρχοντα συστήματα καθώς και μία δική μας υλοποίηση. Τα συστήματα δοκιμάστηκαν πάνω στα λειτουργικά συστήματα Solaris, SunOS και Digital Unix, και για την υλοποίησή τους δεν χρειάστηκε να αλλαχθεί ούτε μια γραμμή από τον κώδικα των λειτουργικών συστημάτων. Για να εξετάσουμε την απόδοση των συστημάτων χρησιμοποιήσαμε συνθετικές εφαρμογές και κάναμε πειράματα πάνω σε τοπικά δίκτυα διαφόρων αρχιτεκτονικών. Γενικά στην εργασία αυτή

- Μελετήσαμε και μετατρέψαμε υπάρχοντα συστήματα δοσοληψιών έτσι ώστε να χρησιμοποιούν απομακρυσμένη μνήμη για αποθήκευση δεδομένων. Παράλληλα φροντίσαμε να ενσωματώσουμε στις αλλαγές μας αλγορίθμους για να πετύχουμε αξιοπιστία σε περίπτωση πτώσης ενός σταθμού εργασίας.

- Αναπτύξαμε μια βιβλιοθήκη υποστήριξης, με την οποία προσθέτουμε αξιοπιστία στην απομακρυσμένη μνήμη. Με αυτό τον τρόπο εφαρμογές μπορούν να την χρησιμοποιήσουν χωρίς τον φόβο απώλειας χρήσιμων δεδομένων. Φροντίσαμε η σχεδίαση μας να προσφέρει απλότητα και χαμηλό κόστος.

- Αποδεικνύουμε ότι η αποθήκευση δεδομένων στην απομακρυσμένη μνήμη από εφαρμογές που χρησιμοποιούν δοσοληψίες, προσφέρει σημαντική βελτίωση στο χρόνο εκτέλεσης τους χωρίς να χάνει σε αξιοπιστία σε σχέση με τη χρήση τοπικού δίσκου.

Βασιζόμενοι στις υλοποιήσεις μας και στα πειραματικά αποτελέσματα που προέκυψαν, μπορούμε να συμπεράνουμε τα εξής

- *Η χρήση της απομακρυσμένης μνήμης για την αποθήκευση δεδομένων δοσοληψιών δίνει καλύτερη απόδοση από τη χρήση του τοπικού δίσκου.* Τα πειράματά μας έδειξαν ότι μετά τη μετατροπή τα συστήματα βελτίωσαν την απόδοσή τους μία με δύο τάξεις μεγέθους, ακόμα και πάνω από δίκτυα χαμηλού ρυθμού μεταγωγής δεδομένων, όπως το Ethernet. Στην περίπτωση μάλιστα του SCI, η βιβλιοθήκη μας έδωσε μέχρι και τρεις τάξεις μεγέθους μεγαλύτερη απόδοση από τον τοπικό δίσκο.

- *Η αλλαγή των υπαρχόντων συστημάτων καθώς και η εξαρχής ανάπτυξη συστήματων που χρησιμοποιούν την απομακρυσμένη μνήμη είναι απλή.* Στην πρώτη περίπτωση

οι αλλαγές στο κώδικα είναι περιορισμένες και στη δεύτερη ο κώδικας που χρειάζεται είναι αρκετά μικρός.

- *Η εξασφάλιση αξιοπιστίας είναι πολύ φθηνή.* Η τεχνικές που χρησιμοποιήσαμε για αξιοπιστία προσθέτουν μία πολύ μικρή επιβάρυνση σε χρόνο εκτέλεσης. Επιπλέον ακόμα και στην περίπτωση των διπλών αντιγράφων, ο επιπλέον χώρος που χρειαζόμαστε είναι πολύ μικρότερος από το συνολικό χώρο των δεδομένων μας.

- *Η βελτίωση στην απόδοση είναι εμφανής ακόμα και σε "αργά" δίκτυα όπως το Ethernet.* Τα πειράματά μας έδειξαν ότι ακόμα και δίκτυα χαμηλής παροχής δεδομένων παρόλο το φόρτο τους έδωσαν αρκετά καλύτερα αποτελέσματα από τον τοπικό δίσκο, ιδιαίτερα για μικρές δοσοληψίες.

- *Τα οφέλη από τη χρήση της απομακρυσμένης μνήμης για την αποθήκευση δεδομένων δοσοληψιών θα αυξάνουν με το πέρασμα του χρόνου.* Οι σημερινές τάσεις στην αρχιτεκτονική υπολογιστών δείχνουν ότι ο χρόνος προσπέλασης του δίσκου μετρημένος σε κύκλους του επεξεργαστή συνεχίζει να αυξάνει με ταχείς ρυθμούς. Οι δίσκοι λοιπόν δεν αναμένεται να είναι σε θέση να προσφέρουν το ρυθμό μεταγωγής δεδομένων αλλά και ούτε το μικρό χρόνο αναζήτησης που απαιτείται για την γρήγορη και αξιόπιστη περάτωση των δοσοληψιών. Αντίθετα τα δίκτυα διασύνδεσης προσφέρουν ένα ολοένα αυξανόμενο ρυθμό μεταγωγής δεδομένων και συνεχώς χαμηλότερο χρόνο απόκρισης, αυξάνοντας έτσι το πλεονέκτημά τους έναντι του δίσκου.

Σύμφωνα με τα πειραματικά μας αποτελέσματα, η χρήση της απομακρυσμένης μνήμης, από συστήματα δοσοληψιών, είναι ένας αποδοτικός και φθηνός τρόπος με τον οποίο μπορούν να αυξήσουν την απόδοση τους χωρίς να συμβιβάσουν την ανάγκη τους για αξιοπιστία.

Παράρτημα Α

Ορολογία

ανενεργός	idle
ανενεργός	inactive
αξιοπιστία	reliability
απεικονισμένος στη μνήμη	memory mapped
αποκλειστικό Ή	xor
απομακρυσμένη ανάγνωση	remote read
απομακρυσμένη εγγραφή	remote write
απομακρυσμένη κλήση διαδικασίας	remote procedure call
απομακρυσμένη μνήμη	remote memory
ανάκτηση	recovery
ανάκτηση	rollback
γέφυρα	bridge
δέσμευση	commit
διαίρεση δίσκου	disk partition
διαφανής	transparent
διασύνδεση	interface
δίαυλος	bus
διεγγραφή	write through
διεργασία	process
δοκιμασία	test
δοσοληψία	transaction
έγκυρος	valid
ενεργός	busy
ενταμιευτής	buffer
ενταμιευτής εγγραφής	write buffer
εξυπηρετητής	server
εξυπηρετητής ισοτιμίας	parity server
επεκταμένος	extended
επιβεβαίωση	acknowlegement
επιτάχυνση	speedup
ημερολόγιο	log
ιδεατή μνήμη	virtual memory
ισοτιμία	parity
ίχνος	track
κατανεμημένος	distributed

κατάρρευση	collapse
κατάσταση πυρήνα	kernel mode
κατάσταση χρήστη	user mode
κατάτμηση	fragmentation
κλήση συστήματος	system call
πτώση	crash
κρυφή μνήμη	cache
λάθος σελίδας	page fault
λειτουργικό σύστημα	operating system
λέξη	word
λογισμικό	software
μεταβλητή ισοτιμία	parity logging
μεταγλώττιση	compilation
μεταφορά	porting
μεταφέρσιμος	portable
μη σταθερή μνήμη	volatile memory
μοιραζόμενος	shared
νήμα	thread
οδηγός συσκευής	device driver
παλινδρόμηση	thrashing
πελάτης	client
πηγαίος κώδικας	source code
πλακέτα διασύνδεσης δικτύου	network interface
πολυνηματικός	multithreaded
πόρτα	port
υποδοχή	socket
προνομιούχος	privileged
πρώιμη μεταφορά	prefetching
πυρήνας	kernel
ρυθμός μεταγωγής δεδομένων	bandwidth
σελιδοδιαχειριστής	pager
σημείο αναφοράς	benchmark
σταθερή μνήμη	non−volatile memory
στρώμα διασύνδεσης δεδομένων	data link layer
στρώμα πρίζας	socket layer
σύγκρουση	collision
σύνδεση	linking
σύνολο εργασίας	working set
συσκευή μπλοκ	block device
συσκευή χαρακτήρων	character device
σύστημα αρχείων	file system
σύστημα αρχείων δικτύου	network file system
τεχνική πολλαπλών αντιγράφων	mirroring
τομέας	sector
τοπικό δίκτυο	local area network
τυχαία πρόσβαση	random access
υλικό	hardware
υπερχρήστης	super user

φόρτος μνήμης	memory load
χρόνος αναζήτησης	seek time
χρόνος περιστροφής	rotational time
χρόνος συστήματος	system time
χρόνος χρήστη	user time
χώρος διευθύνσεων	address space

Βιβλιογραφία

[ABvE95] Werner Vogels Anindya Basu, Vineet Buch and Thorsten von Eicken. U−Net: A User−Level Network Interface for Parallel and Distributed Computing. In *Proceedings of the 15th ACM Symposium on Operating System Principles*, Copper Mountain Resort, Colorado, USA, December 1995.

[ADN+95] Thomas E. Anderson, Michael D. Dahlin, Jeanna M. Neefe, David A. Patterson, Drew S. Roseli, and Randolph Y. Wang. Serverless Network File Systems. In *Proceedings of the 15th ACM Symposium on Operating System Principles*, pages 109−−126, Copper Mountain Resort, Colorado, USA, December 1995.

[Ame87] American National Standards Institute. *FDDI Media Access Control. American National Standard X3.139*, 1987.

[BAD+92] M. Baker, S. Asami, E. Deprit, J. Ousterhout, and M. Se ltzer. Non−volatile Memory for Fast, Reliable File Systems. In *Proc. of the 5−th International Conference on Architectura l Support for Programming Languages and Operating Systems.*, pages 10−−22, Boston, MA, October 1992.

[BCF+95] Nanette J. Boden, Danny Cohen, Robert E. Felderman, Alan E. Kulawic, Charles L. Seitz, Jakov N. Seizovic, and Wen−King Su. Myrinet − A Gigabit−per−Second Local−Area Network. *IEEE Micro*, 15(1):29−−36, February 1995.

[BHK+91] M. G. Baker, J. H. Hartman, M. D. Kupfer, K. W. Shirri ff, and J. K. Ousterhout. Measurements of a Distributed File System. In *Proc. 13−th Symposium on Operating Systems Principles*, pages 198−−212, October 1991.

[BJMW95] Greg Buzzard, David Jacobson, Scott Marovich, and John Wilkes. Hamlyn: a High−Performance Network Interface with Sender−Based Memory Management. In *Proceedings of the Hot Interconnects III Symposium*, August 1995.

[BLA+94] Matthias A. Blumrich, Kai Li, Richard Alpert, Cezary Dubnicki, Edward W. Felten, and Jonathan Sandberg. Virtual Memory Mapped Network Interface for the SHRIMP Multicomputer. In *Proceedings of the 21st International Symposium on Computer Architecture*, pages 142−−153, Chicago, IL, USA, April 1994.

[CD95] David R. Cheriton and Kenneth J. Duda. Logged Virtual Memory. In *Proceedings of the 15th ACM Symposium on Operating System Principles*, Copper Mountain Resort, Colorado, USA, December 1995.

[Cea90] M. Carey and D. DeWitt et. al. The EXODUS Extensible DBMS Project: An Overview. In S.Zdonik and D.Maie, editors, *Readings in Object−Oriented Database Systems*. Morgan Kaufman, 1990.

[CGL95] Toni Cortes, Sergi Girona, and Jesus Labarta. PACA: A Distributed File System Cache for Parallel Machines. Performance under Unix−like Workload. Tech report UPC−DAC−1995−20, Universitat Politechnica de Catalunya, Department d' Arquitectura de computadors, June 1995.

[Cha95] Jeff Chase. A Network Virtual Store, 1995. Duke University. Talk given at the SOSP 95.

[CLG+94] Peter M. Chen, Edward K. Lee, Garth A. Gibson, Randy H. Katz, and David A. Patterson. RAID: High−Performance, Reliable Secondary Storage. *ACM Computing Surveys*, 26(2):145−−185, June 1994.

[Com91] Douglas Comer. *Internetworking with TCP/IP: Principles, Protocols and Architecture*. Prentice−Hall, Inc, 1991.

[Dah95] M. Dahlin. *Serverless Network File Systems*. PhD thesis, University of Berkley, 1995.

[DCK+94] Fred Douglis, Ramon Caceres, Frans Kaashoek, Kai Li an d Brian Marsh, and Joshua Tauber. Storage Alternatives for Mobile Computers. *First USENIX Symposium on Operating System Design and Implementation*, November 1994.

[Del88] G. Delp. *The Architecture and Implementation of Memnet: A High−Speed Shared Memory Computer Communication Network*. PhD thesis, University of Delaware, 1988.

[Dig93] Digital Equipment Corporation. *DEC 3000 300/400/500/600/800 Models System Programmer's Manual*, 1993.

[DM96] George Dramitinos and Evangelos P. Markatos. Reliable Paging to Remote Main Memory in a Workstation Cluster. In R.H. Campbell and Nayeem Islam, editors, *Modern Operating Systems*. IEEE Computer Society Press, 1996.

[Dol94] Dolphin Interconnect Solutions AS, Oslo, Norway. *DIS301 Sbus−to−SCI Adapter User's Guide*, April 1994. e−mail scimktg@dolphin.no.

[FCanHML94] Michael J. Feeley, Jeffrey S. Chase, and Vivek R. Narasayya a nd Henry M. Levy. Integrating Coherency and Recovery in Distributed Systems. *First USENIX Symposium on Operating System Design and Implementation*, November 1994.

[FMP+95] Michael J. Feeley, William E. Morgan, Frederic H. Pighin, Anna R. Karlin, Henry M. Levy, and Chandramohan A. Thekkath. Implementing Global Memory Management in a Workstation Cluster. In *Proceedings*

of the 15th ACM Symposium on Operating System Principles, pages 201−−212, Copper Mountain Resort, Colorado, USA, December 1995.

[Gil95] R. Gillet. Memory Channel. In *Proceedings of the Hot Interconnects III Symposium*, August 1995.

[Gro93] The EXODUS Group. Using the EXODUS Storage Manager V3.1, November 1993.

[HO93] John H. Hartman and John K. Ousterhout. The Zebra Striped Network File System. In *Proceedings of the 14th ACM Symposium on Operating System Principles*, pages 29−−43, Asheville, NC, USA, December 1993.

[HP90] John L. Hennessy and David A. Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann Publishers, Inc., 1990.

[HS96] Galen C. Hunt and Michael L. Scott. Tech report, University of Rochester, July 1996.

[IEE85] IEEE. *802.3: Carrier Sense Multiple Access with Collision Detection*, 1985.

[JLGS90] D. V. James, A. T. Laundrie, S. Gjessing, and G. S. Sohi. Scalable Coherent Interface. *IEEE Computer*, 23(6):74−−77, June 1990.

[LGG+91] B. Liskov, S. Ghemawat, R. Gruber, P. Johnson, L. S hrira, and M. Williams. Replication in the Harp File System. *Proc. 13−th Symposium on Operating Systems Principles*, pages 226−−238, October 1991.

[MD95] Evangelos P. Markatos and George Dramitinos. Adding Flexibility to a Remote Memory Pager. In *Proceedings of the Fourth International Workshop on Object Orientation in Operating Systems*, pages 183−−186, Lund, Sweeden, August 1995.

[MD96] Evangelos P. Markatos and George Dramitinos. Implementation of a Reliable Remote Memory Pager. In *Proceedings of the 1996 Usenix Technical Conference*, pages 177−−190, San Diego, CA, USA, January 1996.

[MK96] Evangelos P. Markatos and Manolis G.H. Katevenis. Telegraphos: High−Performance Networking for Parallel Processing on Workstation Clusters. In *Proceedings of the Second International Symposium on High−Performance Computer Architecture (HPCA)*, San Jose, CA, USA, February 1996.

[New94] Peter Newman. ATM Local Area Networks. *IEEE Communications Magazine*, 32(3):86−−98, March 1994.

[NWO88] M. Nelson, B. Welch, and J. Ousterhout. Caching in the Sprite Network File System. *ACM Transactions on Computer Systems*, 6(1):134−−154, February 1988.

[PGK88] David Patterson, Garth Gibson, and Randy Katz. A case for redundant arrays of inexpensive disks (RAID). In *ACM SIGMOD Conference*, pages 109−−116, June 1988.

[Pos81a] J. Postel. *Internet Protocol–DARPA Internet Program Protocol Specification. Internet Network Working Group, RFC 791.* USC Information Sciences Institute, September 1981.

[Pos81b] J. Postel. *Transmission Control Protocol. Internet Network Working Group, RFC 793.* USC Information Sciences Institute, September 1981.

[SD91] Bill N. Schilit and Dan Duchamp. Adaptive Remote Paging for Mobile Computers. Technical Report CUCS–004–91, Columbia University, Department of Computer Science, February 1991.

[SGK+85] R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh, and B. Lyon. Design and Implementation of the Sun Network File System. In *Proceedings of the 1985 Summer Usenix Conference*, Portland, OR, USA, June 1985.

[SL91] Dimitrios N. Serpanos and Richard J. Lipton. The Design of a High–Speed Network. In *Proceedings of the Third IFIP WG 6.4 Conference on High Speed Networking*, Berlin, Germany, May 1991.

[SMK+93] M. Stayanarayanan, Henry H Mashburn, Puneet Kumar, Dav id C. Steere, and James J. Kistler. Lightweight Recoverable Virtual Memory. *ACM Transactions on Computer Systems*, 12(1), February 1993.

[WZ95] Michael Wu and Willy Zwaenepoel. eNVy: A Non–Volatile, Main Memory Storage System. In *Proceedings of the 6th Symposium on Architectural Support for Programming Languages and Operating Systems*, 1995.