ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΡΟΣΟΜΟΙΩΣΗ ΣΥΣΤΗΜΑΤΩΝ ΕΠΕΞΕΡΓΑΣΙΑΣ ΔΟΣΟΛΗΨΙΩΝ ΚΑΙ ΜΕΛΕΤΗ ΜΕΘΟΔΩΝ ΓΙΑ ΤΗΝ ΙΚΑΝΟΠΟΙΗΣΗ ΣΤΟΧΩΝ ΕΠΙΔΟΣΗΣ

Μανόλης Μαραζάκης

Μεταπτυχιακή Εργασία

Ηράκλειο, Σεπτέμβριος 1995

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΡΟΣΟΜΟΙΩΣΗ ΣΥΣΤΗΜΑΤΩΝ ΕΠΕΞΕΡΓΑΣΙΑΣ ΔΟΣΟΛΗΨΙΩΝ ΚΑΙ ΜΕΛΕΤΗ ΜΕΘΟΔΩΝ ΓΙΑ ΤΗΝ ΙΚΑΝΟΠΟΙΗΣΗ ΣΤΟΧΩΝ ΕΠΙΔΟΣΗΣ

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

Συγγραφέας:

_____
Μανόλης Μαραζάκης
Τμήμα Επιστήμης Υπολογιστών

Εισηγητική Επιτροπή:

_____
Χρήστος Νικολάου, Αναπληρωτής Καθηγητής, Επόπτης

_____
Βαγγέλης Μαρκάτος, Επίκουρος Καθηγητής, Μέλος

_____
Δημήτρης Σερπάνος, Επίκουρος Καθηγητής, Μέλος

Δεκτή:

_____
Πάνος Κωνσταντόπουλος, Αναπληρωτής Καθηγητής
Πρόεδρος Επιτροπής Μεταπτυχιακών Σπουδών

Ηράκλειο, Σεπτέμβριος 1995

Προσομοίωση Συστημάτων Επεξεργασίας Δοσοληψιών και Μελέτη Μεθόδων για την Ικανοποίηση Στόχων Επίδοσης

Μανόλης Μαραζάκης

Μεταπτυχιακή Εργασία

Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

ΠΕΡΙΛΗΨΗ

Η έννοια της δοσοληψίας (transaction), βασισμένη στην νομική έννοια της σύμβασης, υποδηλώνει τις ιδιότητες της ατομικότητας, συνέπειας, απομόνωσης, και μονιμότητας για μια σειρά πράξεων. Οι ιδιότητες αυτές, που αναφέρονται στην βιβλιογραφία με την ακροστοιχίδα ACID (atomicity, consistency, isolation, durability), είναι απαραίτητες για την υποστήριξη ταυτόχρονης προσπέλασης σε κοινόχρηστα δεδομένα και την αντιμετώπιση βλαβών, ειδικά σε κατανεμημένα περιβάλλοντα.

Στα πλαίσια της εργασίας αυτής αναπτύχθηκε ένα ολοκληρωμένο περιβάλλον για την προσομοίωση συστημάτων επεξεργασίας δοσοληψιών, με σκοπό την μελέτη της δυναμικής συμπεριφοράς και επίδοσης τέτοιων συστημάτων. Το περιβάλλον περιλαμβάνει τον προσομοιωτή TPsim και ένα μηχανισμό που υποστηρίζει την διαχείριση πειραμάτων προσομοίωσης, ώστε να αυτοματοποιηθεί κατά το δυνατόν η διαδικασία της περιγραφής πειραμάτων, του συντονισμού της εκτέλεσης των πειραματικών βημάτων, και της συλλογής μετρήσεων. Ο προσομοιωτής μοντελοποιεί με σημαντική λεπτομέρεια όλα τα υποσυστήματα ενός τυπικού συστήματος επεξεργασίας δοσοληψιών, και επιτρέπει την εκτίμηση ποικιλίας μεταβλητών σχετικών με την κατανάλωση πόρων και την επίδοση του συστήματος.

Το περιβάλλον προσομοίωσης χρησιμοποιήθηκε για την πειραματική αξιολόγηση μιας σειράς αλγορίθμων χρονοπρογραμματισμού για σύνθετες μονάδες φόρτου εξυπηρέτησης, που εκτελούνται ως σειρά από δοσοληψίες. Αυτού του είδους οι μονάδες φόρτου εμφανίζονται συχνά στην πράξη, και αποτελούν μια περίπτωση της κατηγορίας μονάδων φόρτου που αναφέρεται με τον όρο workflows. Η μελέτη αυτής της κατηγορίας έχει αρχίσει να συγκεντρώνει σημαντικό ερευνητικό ενδιαφέρον και στην εργασία αυτή γίνεται μια πρώτη μελέτη επίδοσης. Οι προτεινόμενοι αλγόριθμοι χρονοπρογραμματισμού είναι προσανατολισμένοι στην ικανοποίηση στόχων επίδοσης, που διατυπώνονται ως απαιτήσεις για τον μέσο χρόνο απόκρισης ανά κλάση μονάδων φόρτου.

Λέξεις–Κλειδιά: Επεξεργασία Δοσοληψιών, Προσομοίωση Συστημάτων, Διαχείριση Πειραμάτων, Σύνθετες Μονάδες Φόρτου, Χρονοπρογραμματισμός, Στόχοι Επίδοσης.

Επόπτης : Χρήστος Νικολάου, Αναπληρωτής Καθηγητής

Simulation of Transaction Processing Systems and A Study of Methods for Performance Goal Satisfaction

Manolis Marazakis

Master of Science Thesis

Department of Computer Science
University of Crete

ABSTRACT

The transaction concept, based on the concept of contract law, signifies the properties of atomicity, consistency, isolation, and durability for a sequence of actions. These properties, widely known with the acronym ACID, are essential for supporting concurrent access to shared data and for failure handling, especially in distributed environments.

The object of this work is the development of an integrated environment for simulation of trancaction processing systems, with the aim to study the operation and performance of such systems. The environment incorporates the TPsim simulator and a mechanism that supports experiment management, so as to automate the process of specifying simulation experiments, conducting the experimental steps, and collecting measurements. The simulator models with considerable detail the operation of a typical trancaction processing system, and enables the estimation of a variety of variables related to resource consumption and system performance.

The simulation environment was used for the experimental evaluation of a series of scheduling algorithms for complex units of work, that consist of multiple transactions. Workload units of this type occur often in practice, and represent a special class of workflows. The study of this workload class has started to draw considerable research interest, and this work presents a first performance study. The proposed scheduling algorithms are oriented towards satisfying performance goals, which are specified as requirements about the average response time per workload class.

Keywords: Transaction Processing, System Simulation, Experiment Management, Complex Units of Work, Scheduling, Performance Goals.

Supervisor : Christos Nikolaou, Associate Professor

Ευχαριστίες

Ευχαριστίες

Θεωρώ χρέος μου να ευχαριστήσω όλους τους ανθρώπους που με διάφορους τρόπους με βοήθησαν στην εκπόνηση αυτής της εργασίας.

Ο επόπτης καθηγητής κ. Χρήστος Νικολάου μου έδωσε την δυνατότητα να ασχοληθώ με το θέμα της διαχείρισης πόρων σε συστήματα επεξεργασίας δοσοληψιών και με καθοδήγησε σε όλη την διάρκεια της εργασίας μου. Τον ευχαριστώ ιδιαίτερα για την εμπιστοσύνη που έδειξε στο πρόσωπό μου. Ο κ. Βαγγέλης Μαρκάτος και ο κ. Δημήτρης Σερπάνος, μέλη της εξεταστικής επιτροπής, έκαναν εκτενή και ουσιώδη σχόλια σχετικά με την αρχική έκδοση του κειμένου της εργασίας. Οι υποδείξεις τους συνέβαλαν ουσιαστικά στην βελτίωση της παρουσίασης. Ευχαριστώ επίσης τον κ. Σαράντο Καπιδάκη για τις πολύτιμες υποδείξεις του για την προετοιμασία της παρουσίασης της εργασίας.

Ο Κοσμάς Παπαχρήστος βοήθησε αποφασιστικά στην αντιμετώπιση "ιδιοσυγκρασιών" των συστημάτων σε κρίσιμες φάσεις της εργασίας μου. Η Μαρία Καραβασίλη και ο Γιώργος Γεωργιανάκης βοήθησαν στην χρήση του συστήματος επεξεργασίας κειμένου *LaTeX*, καθώς και στην κατασκευή γραφικών παραστάσεων με το πρόγραμμα *EXCEL*. Ο Αλέξανδρος Λαμπρινίδης βοήθησε στην μετάφραση αγγλικών τεχνικών όρων από το πεδίο της επεξεργασίας δοσοληψιών. Οι προαναφερθέντες καθώς και οι Πηνελόπη Κωνσταντά–Φανουράκη, Τίτος Σαρειδάκης και Γιώργος Δραμιτινός βοήθησαν στην αντιμετώπιση δύσκολων στιγμών με χαμόγελο. Ευχαριστώ επίσης τον Σωτήρη Τερζή για τις παρατηρήσεις του πάνω στο κείμενο της εργασίας.

Επίσης θα ήθελα 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.

Αφιερώνω την εργασία αυτή στους γονείς μου, Λάμπρο και Καλλιόπη.

Περιεχόμενα

Περίληψη **i**

Ευχαριστίες **iii**

Περιεχόμενα **v**

Κατάλογος Πινάκων **ix**

Κατάλογος Σχημάτων **xi**

1 Εισαγωγή **1**
1.1 Αντικείμενο της Εργασίας . . . 2
1.2 Δοσοληψίες και Βάσεις Δεδομένων . . . 2
1.2.1 Βασικές Ιδιότητες Δοσοληψιών . . . 3
1.2.2 Παραδείγματα Εφαρμογών και Benchmarks . . . 6
1.3 Οργάνωση της Εργασίας . . . 6

2 Επεξεργασία Δοσοληψιών **9**
2.1 Συστήματα Επεξεργασίας Δοσοληψιών . . . 9
2.1.1 Χαρακτηρισμός του Φόρτου Εξυπηρέτησης σε ένα Περιβάλλον Επεξεργασίας Δοσοληψιών . . . 10
2.1.2 Τύποι Δοσοληψιών . . . 12
2.1.3 Μοντέλο Προγραμματισμού . . . 12
2.1.4 Δομή Συστημάτων Επεξεργασίας Δοσοληψιών . . . 13
2.2 Επόπτες Επεξεργασίας Δοσοληψιών: Στόχοι και Χαρακτηριστικά . . . 13
2.2.1 Βασικές Υπηρεσίες . . . 13
2.2.2 Δομή του Επόπτη Επεξεργασίας Δοσοληψιών . . . 15
2.2.3 Διαχείριση Ουρών . . . 16
2.3 Επισκόπηση της Σχετικής Βιβλιογραφίας . . . 17

3 Διαχείριση Πόρων για την Ικανοποίηση Στόχων Επίδοσης **19**
3.1 Πόροι Συστήματος και Διαχειριστές Πόρων . . . 19
3.2 Διαχείριση συστημάτων από απόψεως επίδοσης . . . 20
3.3 Διατύπωση Στόχων Επίδοσης . . . 22
3.4 Η Εννοια του Δείκτη Επίδοσης . . . 23
3.5 Δυναμική Διαχείριση Πόρων . . . 24
3.6 Επισκόπηση της Σχετικής Βιβλιογραφίας . . . 25
3.6.1 Διαχείριση Πόρων στο Λειτουργικό Σύστημα MVS/ESA . . . 25
3.6.2 Δρομολόγηση Δοσοληψιών . . . 26
3.6.3 Διαχείριση Ενταμιευτών . . . 27
3.6.4 Στόχοι Επίδοσης για Σύνθετες Μονάδες Φόρτου . . . 29

4 Ο Προσομοιωτής TPsim: Σχεδίαση **31**
4.1 Αρχές Σχεδίασης . . . 31
4.2 Το Μοντέλο Συστήματος . . . 33
4.3 Η Γλώσσα Προδιαγραφής . . . 35
4.3.1 Περιγραφή Συσκευών Αποθήκευσης . . . 36
4.3.2 Περιγραφή Κόμβων και Δικτύου Κόμβων . . . 37
4.3.3 Περιγραφή Βάσης Δεδομένων . . . 40
4.3.4 Περιγραφή Φόρτου Εξυπηρέτησης . . . 41
4.4 Διαχείριση Πειραμάτων Προσομοίωσης . . . 44
4.5 Επισκόπιση της Σχετικής Βιβλιογραφίας . . . 45

5 Ο Προσομοιωτής TPsim: Υλοποίηση **47**
5.1 Η Βιβλιοθήκη Υποστήριξης Προσομοίωσης . . . 47
5.1.1 Δομικά Στοιχεία Μοντέλων Προσομοίωσης . . . 48
5.1.2 Νήματα Ελέγχου . . . 48
5.1.3 Μέθοδος Προσομοίωσης . . . 50
5.1.4 Οργάνωση του Περιβάλλοντος Εκτέλεσης . . . 52
5.1.5 Ο Μηχανισμός Ελέγχου της Προσομοίωσης . . . 53
5.1.6 Σύνδεση με τον Μεταφραστή της Γλώσσας Προδιαγραφής . . . 53
5.2 Προσομοίωση Υλικού Συστήματος . . . 55
5.3 Προσομοίωση Υποσυστημάτων Λογισμικού . . . 56
5.3.1 Κατανεμημένη Εκτέλεση Δοσοληψιών . . . 56
5.3.2 Διαχείριση Δοσοληψιών . . . 56
5.3.3 'Ελεγχος Ταυτόχρονης Πρόσβασης . . . 56
5.3.4 Διαχείριση Ενταμιευτών . . . 57
5.3.5 Μηχανισμός Καταγραφής Μεταβολών . . . 60
5.3.6 Διαχείριση Περιφερειακής Μνήμης . . . 63
5.3.7 Διαχείριση Ουρών . . . 64
5.3.8 Δρομολόγηση και Χρονοπρογραμματισμός Δοσοληψιών . . . 65
5.4 Προσομοίωση Φόρτου Εξυπηρέτησης . . . 65
5.4.1 Απλές Μονάδες Φόρτου . . . 65
5.4.2 Σύνθετες Μονάδες Φόρτου . . . 66
5.4.3 Κλειστό Μοντέλο Φόρτου Εξυπηρέτησης . . . 67
5.4.4 Ανοικτό Μοντέλο Φόρτου Εξυπηρέτησης . . . 68
5.4.5 Συλλογή Μετρήσεων . . . 68

6 Περιβάλλον Υποστήριξης Πειραμάτων **69**
6.1 Η Διαδικασία Διεξαγωγής Πειραμάτων Προσομοίωσης . . . 69
6.2 Προδιαγραφές για το Περιβάλλον Υποστήριξης . . . 70
6.3 Υλοποίηση του Περιβάλλοντος Υποστήριξης . . . 72
6.4 Επισκόπηση της Σχετικής Βιβλιογραφίας . . . 75

7 Αλγόριθμοι Χρονοπρογραμματισμού για Σύνθετες Μονάδες Φόρτου **77**
7.1 Σύνθετες Μονάδες Φόρτου . . . 77
7.2 Υποστήριξη Σύνθετων Μονάδων Φόρτου . . . 78
7.3 Αλγόριθμοι Χρονοπρογραμματισμού . . . 79
7.3.1 Επιμέρους Στόχοι Επίδοσης . . . 80
7.3.2 Περιγραφή των Αλγόριθμων . . . 81
7.4 Πειραματική Αξιολόγηση των Αλγορίθμων . . . 82
7.4.1 Παράμετροι Μοντέλου Προσομοίωσης . . . 83
7.4.2 Προδιαγραφή Πειραμάτων . . . 84
7.5 Πειραματικά Αποτελέσματα . . . 85
7.6 Επισκόπιση της Σχετικής Βιβλιογραφίας . . . 87

8 Συμπεράσματα και Ερευνητικές Κατευθύνσεις **101**
8.1 Σύνοψη Αποτελεσμάτων 101
8.2 Επεκτάσεις και Ερευνητικές Κατευθύνσεις 102
8.2.1 Επεκτάσεις του Προσομοιωτή TPsim 102
8.2.2 Ικανοποίηση Στόχων Επίδοσης 103
8.2.3 Διαχείριση Σύνθετων Μονάδων Φόρτου 105

Βιβλιογραφία **106**

Παράρτημα Α **113**

Παράρτημα Β **121**

Παράρτημα Γ **125**

Κατάλογος Πινάκων

7.1 Profiles για τις κλάσεις δοσοληψιών 84
7.2 Στατιστικές Ιδιότητες των Μετρήσεων για την Κλάση WC1 89
7.3 Στατιστικές Ιδιότητες των Μετρήσεων για την Κλάση WC2 90
7.4 Σύνοψη Μετρήσεων 91
7.5 Απόσταση Παραβίασης Στόχων και Λόγος min/max Δείκτη Επίδοσης 91
7.6 Μελέτη Ευαισθησίας ως προς τον Καθορισμό Στόχων: Στατιστικές Ιδιότητες των Μετρήσεων για την Κλάση WC1. Ο στόχος για την κλάση WC2 είναι 5.0 sec, και ο αριθμός των τερματικών ίσος με 150. 94
7.7 Μελέτη Ευαισθησίας ως προς τον Καθορισμό Στόχων : Στατιστικές Ιδιότητες των Μετρήσεων για την Κλάση WC2. Ο στόχος για την κλάση WC2 είναι 5.0 sec, και ο αριθμός των τερματικών ίσος με 150 95
7.8 Μεταβολή Δεικτών Επίδοσης και Μέτρου VD ως προς τον στόχο G_{WC1}. Ο στόχος G_{WC2} είναι 5.0 sec, και ο αριθμός των τερματικών ίσος με 150. 95
7.9 Δυναμική Συμπεριφορά των Αλγορίθμων WPI και WT: Μεταβολή του ποσοστού των βημάτων των οποίων η προτεραιότητα τροποποιείται ως προς τον στόχο G_{WC1}. Ο στόχος G_{WC2} είναι 5.0 sec, και ο αριθμός των τερματικών ίσος με 150. Σημειώνεται το ποσοστό των βημάτων των οποίων η προτεραιότητα τροποποιείται ξεχωριστά για κάθε κλάση, και συνολικά για όλες τις κλάσεις. Υπενθυμίζεται ότι οι μονάδες φόρτου της κλάσης WC1 αποτελούνται από 3 βήματα, ενώ αυτές της κλάσης WC2 από 5 βήματα. 98

Κατάλογος Σχημάτων

1.1 Η Δοσοληψία ως Μετασχηματισμός μεταξύ Καταστάσεων. 3
1.2 Διάγραμμα Καταστάσεων−Μεταβάσεων για μια Δοσοληψία. 5

2.1 Το $X/Open$ Μοντέλο Επεξεργασίας Δοσοληψιών. 14
2.2 Το $X/Open$ Μοντέλο Επεξεργασίας Δοσοληψιών για Κατανεμημένα Συστήματα. 14
2.3 Ροή Ελέγχου ενός Επόπτη Επεξεργασίας Δοσοληψιών. 15
2.4 Συνεργασία ανάμεσα στα Δομικά Στοιχεία ενός Συστήματος Επεξεργασίας Δοσοληψιών. 16

3.1 Δυναμικός Ελεγχος Συστήματος για την Ικανοποίηση Στόχων Επίδοσης. 25
3.2 Ενδεικτικό Διάγραμμα Ροής Διαχειριστή Πόρων Προσανατολισμένου στην Ικανοποίηση Στόχων Επίδοσης. 26

4.1 Αρχιτεκτονική του Προσομοιωτή TPsim. 33
4.2 Δομή του μοντέλου προσομοίωσης για ένα κόμβο σε σύστημα επεξεργασίας δοσοληψιών. 34
4.3 Αρχιτεκτονική Shared−Nothing για κατανεμημένα συστήματα επεξεργασίας δοσοληψιών. 35
4.4 Φάσεις μιας Μελέτης Προσομοίωσης. 44

5.1 Πρόγραμμα Εφαρμογής με ένα Νήμα Ελέγχου. 49
5.2 Πρόγραμμα Εφαρμογής με Πολλαπλά Νήματα Ελέγχου. 50
5.3 Ιεραρχία Νημάτων Ελέγχου. 52
5.4 Λειτουργία του Μηχανισμού Προσομοίωσης. 54
5.5 Η τρέχουσα κατάσταση του διαχειριστή δοσοληψιών περιλαμβάνει την κατάσταση όλων των δοσοληψιών που βρίσκονται σε εξέλιξη στον κόμβο. 57
5.6 Δομές Δεδομένων για την Υλοποίηση Ελέγχου Ταυτόχρονης Πρόσβασης. 58
5.7 Λειτουργία του Διαχειριστή Ενταμιευτών. 59
5.8 Αλληλεπιδράσεις μεταξύ Διαχειριστών Πόρων από την σκοπιά του Διαχειριστή του Μηχανισμού Καταγραφής Μεταβολών. 61
5.9 Διαχείριση Περιφερειακής Μνήμης. 64

6.1 Οργάνωση του Περιβάλλοντος Υποστήριξης Πειραμάτων. 73

7.1 Χρήση Ουρών για την Υποστήριξη Σύνθετων Μονάδων Φόρτου. 78
7.2 Δυναμική Αναπροσαρμογή των Συντελεστών Βάρους w_t 83
7.3 Μεταβολή του Μέγιστου Δείκτη Επίδοσης ως προς τον Αριθμό Τερματικών. 92
7.4 Μεταβολή του Μέτρου VD ως προς τον Αριθμό Τερματικών. 93
7.5 Μεταβολή του Μέγιστου Δείκτη Επίδοσης ως προς τον Στόχο G_{WC1}. Ο στόχος G_{WC2} είναι 5.0 sec, και ο αριθμός των τερματικών ίσος με 150. 96
7.6 Μεταβολή του Μέτρου VD ως προς τον Στόχο G_{WC1}. Ο στόχος G_{WC2} είναι 5.0 sec, και ο αριθμός των τερματικών ίσος με 150. 97
7.7 Μεταβολή του ποσοστού των βημάτων των οποίων τροποποιείται η προτεραιότητα ως προς τον στόχο G_{WC1}. Ο στόχος G_{WC2} είναι 5.0 sec, και ο αριθμός των τερματικών ίσος με 150. Ο αλγόριθμος WPI λειτουργεί πιο "επιλεκτικά' από τον WT, καθώς τροποποιεί την προτεραιότητα μικρότερου ποσοστού βημάτων από σύνθετες μονάδες φόρτου. Το ποσοστό γενικά μειώνεται όσο ο στόχος G_{WC1} γίνεται μεγαλύτερος, καθώς το σύστημα πιέζεται λιγότερο για να ικανοποιήσει τους στόχους επίδοσης. 99

8.1 Αρχιτεκτονική Data Sharing για κατανεμημένα συστήματα επεξεργασίας δοσοληψιών. . 103

Στους γονείς μου, Λάμπρο και Καλλιόπη.

ΡΗΜΑ Δ'ΕΡΓΜΑΤΩΝ ΧΡΟΝΙΩΤΕΡΟΝ ΒΙΟΤΕΥΕΙ.

Κεφάλαιο 1

Εισαγωγή

Η έννοια της δοσοληψίας (transaction) προέρχεται από την νομική έννοια της σύμβασης [Gra81]. Σε μια σύμβαση, δύο ή περισσότερα συμβαλλόμενα μέρη διαπραγματεύονται για ένα διάστημα και τελικά καταλήγουν σε μια συμφωνία. Η συμφωνία αποκτά υπόσταση και δεσμεύει τα συμβαλλόμενα μέρη μόλις αυτά από κοινού υπογράψουν το σχετικό νομικό κείμενο. Εάν τα συμβαλλόμενα μέρη δεν έχουν αμοιβαία εμπιστοσύνη, ή απλά επιθυμούν να καλύψουν τυχόν απρόβλεπτες περιπλοκές κατά την εκτέλεση της συμφωνίας, ορίζουν από κοινού κάποιον ενδιάμεσο, ο οποίος αναλαμβάνει να συντονίσει τις ενέργειές τους για την επιτυχή κατάληξη της σύμβασης.

Μια σύμβαση είναι απλά μια συμφωνία, συνεπώς υπάρχει πάντα το ενδεχόμενο να παραβιαστεί από τα συμβαλλόμενα μέρη. Η παραβίαση μιας σύμβασης αποτελεί παραβίαση του νόμου. Μια σύμβαση δεν μπορεί να ακυρωθεί, παρά μόνο εάν δεν ήταν σύμφωνη με τους ισχύοντες νόμους. Από την στιγμή που μια σύμβαση (δοσοληψία) αποκτά δεσμευτική ισχύ για τα συμβαλλόμενα μέρη μπορεί μόνο να τροποποιηθεί, μέσω νέων επανορθωτικών δοσοληψιών.

Η μελέτη της νομικής έννοιας της σύμβασης αναδεικνύει τις παρακάτω βασικές ιδιότητες των δοσοληψίων:

- Συνέπεια: Μια δοσοληψία πρέπει να είναι σύμφωνη με τους ισχύοντες νόμους.
- Ατομικότητα: Μια δοσοληψία είτε διεκπεραιώνεται εξ'ολοκλήρου, δεσμεύοντας όλα τα συμβαλλόμενα μέρη με τους όρους της σχετικής σύμβασης, είτε δεν υλοποιείται, οπότε κανένα από τα συμβαλλόμενα μέρη δεν δεσμεύεται κατά οποιοδήποτε τρόπο.
- Μονιμότητα (Διάρκεια): Μόλις μια δοσοληψία διεκπεραιωθεί, οι δεσμεύσεις που επιβάλλει δεν μπορούν να αγνοηθούν από τα συμβαλλόμενα μέρη, ανεξάρτητα των περιστάσεων.

Μεταφέροντας την έννοια της δοσοληψίας στο πεδίο της επιστήμης υπολογιστών, παρατηρεί κανείς ότι σχεδόν όλες οι δοσοληψίες που πραγματοποιούνται καθημερινά γύρω μας μπορούν να παρασταθούν αφαιρετικά ως μετασχηματισμοί κατάστασης σε ένα σύστημα. Αυτή η ερμηνεία της δοσοληψίας, σε συνδυασμό με τις βασικές ιδιότητες που χαρακτηρίζουν μια δοσοληψία, καθιστούν την έννοια της δοσοληψίας ιδιαίτερα σημαντική για την οργάνωση εφαρμογών που διαχειρίζονται υψηλό όγκο πληροφορίας (data−intensive applications). Τα συστήματα επεξεργασίας δοσοληψιών παρέχουν την υποστήριξη για τέτοιου είδους εφαρμογές, υλοποιώντας τις εγγυήσεις που συνδέονται με την έννοια της δοσοληψίας.

1.1 Αντικείμενο της Εργασίας

Αντικείμενο της παρούσας εργασίας είναι η ανάπτυξη ενός ολοκληρωμένου περιβάλλοντος προσομοίωσης για την μελέτη της δυναμικής συμπεριφοράς και επίδοσης συστημάτων επεξεργασίας δοσοληψιών. Εμφαση δίδεται στην μελέτη τεχνικών για την διαχείριση των πόρων του συστήματος, ειδικά για σύνθετες μονάδες φόρτου, με τρόπο ώστε το σύστημα, πέρα από τις προδιαγραφές που θέτει ο ορισμός της δοσοληψίας, να είναι σε θέση να ικανοποιήσει και προδιαγραφές επίδοσης.

Ο προσομοιωτής TPsim που αναπτύχθηκε στα πλαίσια αυτής της εργασίας επιτρέπει την περιγραφή της διαμόρφωσης του προς μελέτη συστήματος και του φόρτου εξυπηρέτησης μέσω μιας τυπικής γλώσσας προδιαγραφής. Η δυνατότητα αυτή, που δεν είναι ιδιαίτερα διαδεδομένη, διευκολύνει την προσαρμογή του προσομοιωτή στις απαιτήσεις διαφορετικών μελετών εφαρμογής. Ο προσομοιωτής μοντελοποιεί τα κύρια υποσυστήματα ενός τυπικού συστήματος επεξεργασίας δοσοληψιών με σημαντική λεπτομέρεια, και επιτρέπει την μελέτη επίδοσης εναλλακτικών πολιτικών διαχείρισης πόρων.

Ο προσομοιωτής εντάσσεται σε ένα περιβάλλον υποστήριξης πειραμάτων προσομοίωσης που παρέχει ένα μηχανισμό για την προδιαγραφή πειραμάτων προσομοίωσης, τον συντονισμό της εκτέλεσης των πειραματικών βημάτων σε ένα κατανεμημένο περιβάλλον, και την συλλογή των παραγόμενων μετρήσεων. Η διαχείριση πειραμάτων αποτελεί ένα πολύ σημαντικό πρόβλημα για την μελέτη συστημάτων μέσω προσομοίωσης.

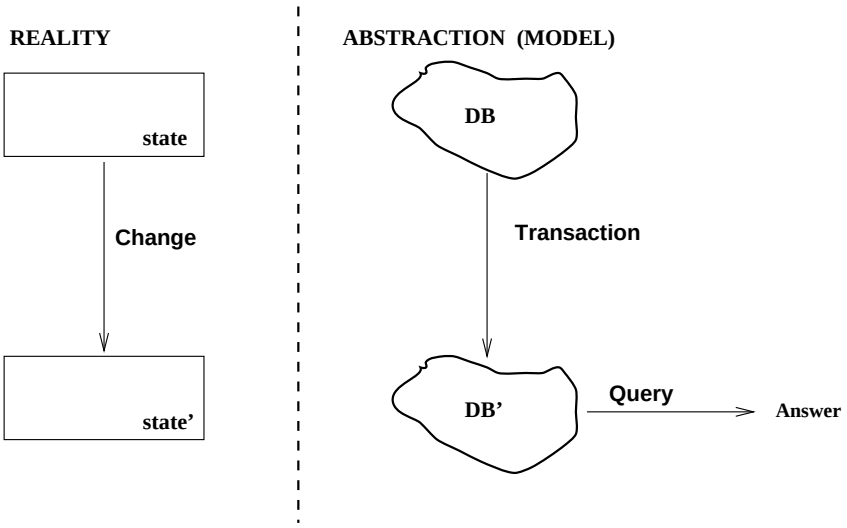
Το περιβάλλον υποστήριξης πειραμάτων προσομοίωσης χρησιμοποιήθηκε για μια μελέτη επίδοσης τεχνικών χρονοπρογραμματισμού (scheduling). Ξεκινώντας από την μελέτη τεχνικών διαχείρισης πόρων που είναι προσανατολισμένες στην ικανοποίηση στόχων επίδοσης για κλάσεις δοσοληψιών, η εργασία αυτή προτείνει τεχνικές κατάλληλες για μονάδες φόρτου που αποτελούνται από σειρά δοσοληψιών διαφόρων τύπων. Οι τεχνικές αυτές ελέγχουν τον χρονοπρογραμματισμό των δοσοληψιών, και κατ'αυτόν τον τρόπο ρυθμίζουν δυναμικά την κατανομή του χρόνου του επεξεργαστή ανά κλάση μονάδων φόρτου, λαμβάνοντας υπ'όψιν τους στόχους επίδοσης.

1.2 Δοσοληψίες και Βάσεις Δεδομένων

Η παράσταση όψεων του πραγματικού κόσμου με χρήση αφαιρετικών μοντέλων (abstraction models) είναι μια θεμελειώδης αρχή για την λειτουργία κάθε οργανωμένης ανθρώπινης δραστηριότητας. Η παράσταση αυτή είναι αφαιρετική, καθώς εστιάζει σε εκείνα μόνο τα δομικά και χαρακτηριστικά στοιχεία του περιβάλλοντος που επηρεάζουν άμεσα την κάθε δραστηριότητα, και αυτά στον βαθμό λεπτομέρειας που επιβάλλουν οι λειτουργικές ανάγκες για το δοθέν πεδίο εφαρμογών. Μια βάση δεδομένων (database) αποθηκεύει πληροφορίες σχετικές με το αφαιρετικό μοντέλο για τις όψεις του πραγματικού κόσμου που αφορούν άμεσα το συγκεκριμένο πεδίο εφαρμογών. Η παράσταση των πληροφοριών γίνεται με χρήση των συντακτικών δομών που υποστηρίζει το χρησιμοποιούμενο μοντέλο δεδομένων (data model), με την διατύπωση, σε τυπική γλώσσα, του σχήματος της βάσης δεδομένων (database schema). Σκοπός της ύπαρξης μιας βάσης δεδομένων είναι να παρέχει στους χρήστες τις πληροφορίες που διατηρεί, με τρόπο αποδοτικό και εύχρηστο, ως υποστήριξη για τις εργασίες που αυτοί έχουν να διεκπεραιώσουν στα πλαίσια μιας δραστηριότητας στην

οποία μετέχουν. Οι χρήστες ζητούν τις πληροφορίες που χρειάζονται διατυπώνοντας επερωτήσεις (queries), μέσω μιας τυπικής γλώσσας που επιτρέπει τον προσδιορισμό των ζητούμενων στοιχείων από την βάση δεδομένων.

Είναι σαφές ότι μια βάση δεδομένων εκφράζει ένα στατικό στιγμιότυπο της όψης του πραγματικού κόσμου που παριστάνει το σχήμα βάσης δεδομένων, συνεπώς είναι αναγκαία η ενημέρωση της βάσης δεδομένων για τυχόν μεταβολές που σημειώνονται, ώστε να διατηρείται η εγκυρότητα της πληροφορίας. Η ενημέρωση μπορεί να γίνει με την εισαγωγή ή διαγραφή στοιχείων, ή με την μεταβολή τιμών. Μια δοσοληψία εκφράζει μια μεταβολή κατάστασης στα πλαίσια ενός αφαιρετικού μοντέλου για μια όψη του πραγματικού κόσμου. Οι παραπάνω έννοιες και οι μεταξύ τους σχέσεις παρουσιάζονται συνοπτικά στο σχήμα 1.1.



Σχήμα 1.1: Η Δοσοληψία ως Μετασχηματισμός μεταξύ Καταστάσεων.

Μια βάση δεδομένων (database – DB) αποτελεί αφαιρετικό μοντέλο μιας άποψης του πραγματικού κόσμου, στα πλαίσια των εφαρμογών που αυτή υποστηρίζει. Οι καταστάσεις της βάσης δεδομένων αντιστοιχούν σε καταστάσεις του πραγματικού κόσμου. Μια δοσοληψία (transaction) αντιστοιχεί σε μια μεταβολή κατάστασης, ενώ μια επερώτηση (query) δίδει μια εικόνα της τρέχουσας κατάστασης.

1.2.1 Βασικές Ιδιότητες Δοσοληψιών

Με τον όρο *δοσοληψία* δηλώνεται μια ακολουθία από πράξεις που επιδρούν στην κατάσταση μιας εφαρμογής. Η εκτέλεση μιας δοσοληψίας αντιπροσωπεύει μια αλλαγή κατάστασης στο περιβάλλον για το οποίο το σύστημα αποτελεί αφαιρετικό μοντέλο, και πληρεί μια σειρά από συνθήκες που αναφέρονται στην βιβλιογραφία με την ακροστοιχίδα ACID:

1. Ατομικότητα (Atomicity). Οι αλλαγές στην κατάσταση του συστήματος που επιφέρει μια δοσοληψία είναι μια *ατομική* πράξη: είτε όλες πραγματοποιούνται, είτε καμία δεν πραγματοποιείται. Οι αλλαγές που θα επιφέρει μια δοσοληψία μπορεί να έχουν την μορφή μιας ενημέρωσης σε μια βάση δεδομένων, ή της αποστο–

λής/λήψης ενός μηνύματος, ή μιας αλληλεπίδρασης με τον πραγματικό κόσμο μέσω μιας συσκευής ελέγχου. Η ιδιότητα αυτή αναφέρεται στην βιβλιογραφία και ως ιδιότητα "all−or−nothing".

2. Συνέπεια (**C**onsistency). Μια δοσοληψία επιφέρει μια *ορθή* μεταβολή κατάστασης. Οι πράξεις μιας δοσοληψίας, αν θεωρηθούν ως μια ενότητα, δεν παραβιάζουν κανένα από τους περιορισμούς ακεραιότητας (integrity constraints) που σχετίζονται με την κατάσταση του συστήματος. Η απαίτηση αυτή ουσιαστικά σημαίνει ότι η δοσοληψία πρέπει να είναι ένα ορθό πρόγραμμα.

3. Απομόνωση (**I**solation). Αν και πολλές δοσοληψίες μπορεί να εκτελούνται ταυτόχρονα, κάθε δοσοληψία, T, έχει την εντύπωση ότι οι υπόλοιπες δοσοληψίες εκτελούνται είτε πριν την T είτε μετά την T. Συνεπώς, τα αποτελέσματα πολλών δοσοληψιών που συμβαίνουν ταυτόχρονα είναι ίδια με τα αποτελέσματα των ίδιων δοσοληψιών αν αυτές εκτελεστούν σειριακά (με κάποια αυθαίρετη σειρά) και εξασφαλίζεται ότι τα αποτελέσματα κάθε δοσοληψίας είναι ανεξάρτητα από τα αποτελέσματα οποιασδήποτε άλλης δοσοληψίας. Η ιδιότητα αυτή αναφέρεται και ως *σειριοποιησιμότητα* (serializability).

4. Μονιμότητα (**D**urability). Οταν μια δοσοληψία τερματιστεί επιτυχώς, οι αλλαγές που αυτή επέφερε στην κατάσταση του συστήματος δεν πρόκειται να ακυρωθούν σαν συνέπεια κάποιας βλάβης του συστήματος. Η κατάσταση την οποία ανακτά το σύστημα μετά από μια βλάβη (failure) αντικατοπτρίζει όλες τις αλλαγές που έγιναν από δοσοληψίες που είχαν τερματίσει επιτυχώς πριν να σημειωθεί η βλάβη.

Μια δοσοληψία μπορεί να θεωρηθεί ως η μονάδα ταυτόχρονης προσπέλασης (unit of concurrency) και η μονάδα επανάκτησης (unit of recovery). Ως μονάδα ταυτόχρονης προσπέλασης, λόγω της ιδιότητας της απομόνωσης, τα βήματα από διάφορες δοσοληψίες μπορούν να αναμειχθούν μεταξύ τους χωρίς να επιδρούν το ένα στο άλλο, ενώ ως μονάδα επανάκτησης του συστήματος, μια δοσοληψία, λόγω της ιδιότητας της ατομικότητας, είτε επιτυγχάνει πλήρως, οπότε όλες οι αλλαγές που πραγματοποιεί καθίστανται μόνιμες, είτε αποτυγχάνει πλήρως χωρίς να επιφέρει καμιά μόνιμη αλλαγή.

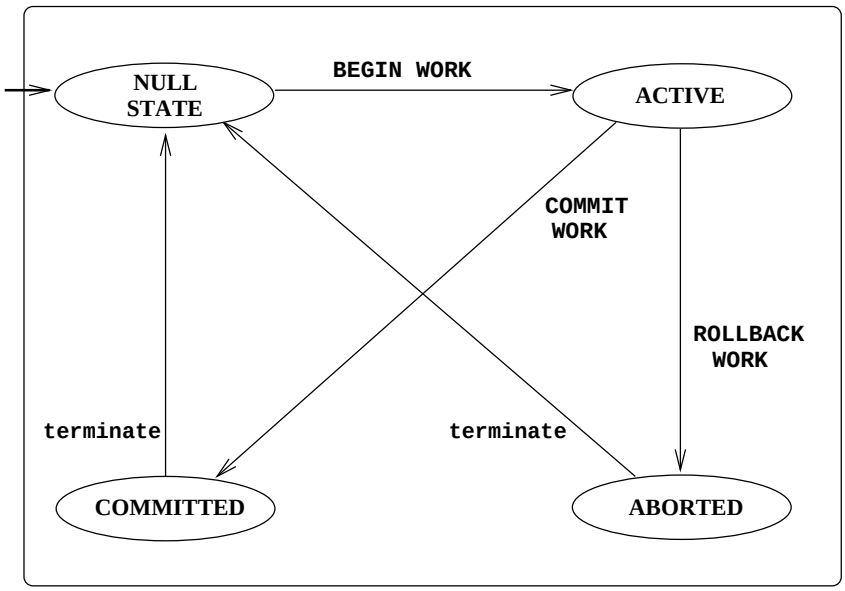
Μια δοσοληψία, λόγω της ιδιότητας της ατομικότητας, μπορεί να έχει ακριβώς μια από τις παρακάτω εκβάσεις:

- Commit: Επιτυχής τερματισμός. Στην περίπτωση αυτή τα αποτελέσματα όλων των πράξεων που εκτέλεσε η δοσοληψία γίνονται μόνιμα. Οι αλλαγές που επιφέρει μια δοσοληψία που τερματίζει επιτυχώς στην κατάσταση του συστήματος μπορούν να ακυρωθούν μόνο με την (επιτυχή) εκτέλεση άλλων επανορθωτικών δοσοληψιών (compensating transactions).

- Abort: Αποτυχία. Στην περίπτωση αυτή τα αποτελέσματα των πράξεων που εκτέλεσε η δοσοληψία δεν γίνονται ποτέ ορατά σε άλλες δοσοληψίες.

Τα παραπάνω συνοψίζονται στο σχήμα 1.2, που παρουσιάζει το μοντέλο δοσοληψιών ως διάγραμμα καταστάσεων−μεταβάσεων.

Οι ιδιότητες ACID δίνουν ένα ιδιαίτερα απλό μοντέλο βλαβών. Κάθε δομικό στοιχείο (module) ενός συστήματος χαρακτηρίζεται από την παρατηρούμενη και την αναμενόμενη (βάσει των προδιαγραφών σχεδίασης) συμπεριφορά του. Μια βλάβη (failure) εμφανίζεται όταν η παρατηρούμενη συμπεριφορά αποκλίνει από την



Σχήμα 1.2: Διάγραμμα Καταστάσεων–Μεταβάσεων για μια Δοσοληψία.

Με την πράξη BEGIN_WORK η δοσοληψία μεταβαίνει από την αρχική κατάσταση NULL στην κατάσταση ACTIVE, από την οποία, μετά την εκτέλεση ενός αριθμού πράξεων, μπορεί να μεταβεί είτε στην κατάσταση COMMITTED (πράξη COMMIT_WORK), είτε στην κατάσταση ABORTED (πράξη ROLLBACK_WORK). Οι καταστάσεις αυτές είναι τελικές, αντιπροσωπεύοντας τις δυο δυνατές εκβάσεις μιας δοσοληψίας.

αναμενόμενη. Συχνά χρησιμοποιείται και ο όρος "εξαίρεση" (exception) για να δηλωθεί μια τέτοια απόκλιση. Με βάση τις ιδιότητες ACID, η έννοια της δοσοληψίας μπορεί να αποτελέσει την βάση για την σχεδίαση και υλοποίηση συστημάτων που οφείλουν να αντιμετωπίζουν το ενδεχόμενο βλαβών. Είναι σημαντικό, ιδίως για σύνθετα συστήματα μεγάλης κλίμακας, ότι *μια* θεμελιώδης έννοια παρέχει το λογικό πλαίσιο/πρότυπο για την αντιμετώπιση εξαιρέσεων, κατά τρόπο εννιαίο.

Από την άποψη αυτή, η έννοια της δοσοληψίας παρέχει έναν απλό και καλά δομημένο τρόπο για την υλοποίηση εφαρμογών, ιδίως σε κατανεμημένα συστήματα. Κάθε αλληλεπίδραση με το σύστημα, που έχει την μορφή μιας αίτησης εξυπηρέτησης, οργανώνεται ως μια δοσοληψία. Εάν δεν εμφανιστεί κάποιο πρόβλημα κατά την εξυπηρέτηση, τότε η δοσοληψία τερματίζεται επιτυχώς, και *όλα* τα λειτουργικά μέρη της εφαρμογής μεταβαίνουν στην νέα συνεπή και μόνιμη κατάσταση που προέκυψε από την εκτέλεση της δοσοληψίας. Εάν όμως εμφανιστεί οποιοδήποτε πρόβλημα σε κάποιο στάδιο της εξυπηρέτησης, *όλα* τα λειτουργικά μέρη της εφαρμογής, χωρίς επέμβαση του χρήστη, επανέρχονται στην συνεπή κατάσταση πριν την έναρξη της δοσοληψίας. Καθώς η διαδικασία αυτή είναι διαφανής στον χρήστη, είναι δυνατή η σχεδίαση λειτουργικών μερών με πολύ απλό μοντέλο βλαβών (κυρίως λόγω της ιδιότητας της ατομικότητας).

1.2.2 Παραδείγματα Εφαρμογών και Benchmarks

Το κλασσικό παράδειγμα δοσοληψίας [Ano85] είναι η δοσοληψία Πίστωσης/Χρέωσης (Debit/Credit Transaction), όπου ένα χρηματικό ποσό αφαιρείται από έναν τραπεζικό λογαριασμό και προστίθεται σε έναν άλλο. Η δοσοληψία αυτή είναι ενδεικτική αρκετών πεδίων εφαρμογών, και αποτελεί την βάση των benchmarks TPC−A και TPC−B [Gra91].

Στο παράδειγμα αυτό, για να εξασφαλίζεται η ατομικότητα μιας δοσοληψίας, θα πρέπει είτε να γίνουν και οι δύο πράξεις ταυτόχρονα είτε καμία από τις δύο, ενώ για να υπάρχει συνέπεια θα πρέπει να μην γίνεται χρέωση σε λογαριασμό με μηδενικό υπόλοιπο. Επιπλέον, για να υπάρχει απομόνωση, δεν θα πρέπει η οποιαδήποτε δοσοληψία χρέωσης/πίστωσης να μπορεί να διακρίνει ότι ενδεχομένως κάποια άλλη εφαρμογή διαβάζει ή γράφει στον ίδιο τραπεζικό λογαριασμό, ενώ για να εξασφαλίζεται μονιμότητα, θα πρέπει όταν μεταφερθούν χρήματα από ένα λογαριασμό σε έναν άλλο, τότε, παρόλο που το σύστημα μπορεί να πάθει κάποια βλάβη, η μεταβίβαση αυτή των χρημάτων θα πρέπει να υπάρχει και μετά την επαναφορά του συστήματος σε ομαλή λειτουργία.

Συστήματα επεξεργασίας δοσοληψιών στηρίζουν την λειτουργία οργανισμών όπως τράπεζες, ασφαλιστικές εταιρίες, μονάδες παραγωγής, αεροπορικές εταιρίες, και οργανισμοί επικοινωνιών, υποστηρίζοντας εφαρμογές όπως τήρηση λογιστικών, καταγραφή συναλλαγών, επεξεργασία και διεκπεραίωση παραγγελιών, έκδοση λογαριασμών χρέωσης για παροχή υπηρεσιών, έλεγχος διεργασιών παραγωγής, και αυτοματισμός γραφείου.

Τα benchmarks TPC−A, TPC−B, TPC−C, TPC−D [Cou90a, Cou90b, Cou93, Cou95] του consortium Transaction Processing Council, που έχει ως μέλη πολλούς κατασκευαστές συστημάτων, έχουν ως στόχο να μοντελοποιήσουν την "συμπεριφορά" τυπικών εφαρμογών επεξεργασίας δοσοληψιών, όπως αυτές που προαναφέρθηκαν, ώστε να είναι δυνατή μια αξιόπιστη σύγκριση της επίδοσης διαφορετικών συστημάτων για *συγκεκριμένα* πεδία εφαρμογών. Οι προδιαγραφές των benchmarks αυτών είναι πολύ λεπτομερείς ώστε τα αποτελέσματα να δίδουν μια αξιόπιστη εικόνα της επίδοσης, για δεδομένο φόρτο εξυπηρέτησης και διαμόρφωση συστήματος.

Η μελέτη των προδιαγραφών αυτών υπήρξε πολύ σημαντική για την ανάπτυξη του προσομοιωτή που αναπτύχθηκε στα πλαίσια αυτής της εργασίας, καθώς έδωσε το κίνητρο για ορισμένες βασικές σχεδιαστικές επιλογές.

1.3 Οργάνωση της Εργασίας

Το υπόλοιπο αυτής της αναφοράς οργανώνεται ως εξής: Στο κεφάλαιο 2 γίνεται μια αρκετά εκτεταμένη παρουσίαση βασικών αρχών στην σχεδίαση συστημάτων επεξεργασίας δοσοληψιών, ώστε να τεθούν οι βάσεις για την περιγραφή του προσομοιωτή που αναπτύχθηκε. Στο κεφάλαιο 3 εισάγεται η έννοια του στόχου επίδοσης, και το πρόβλημα της διαχείρισης συστημάτων από απόψεως επίδοσης. Παρουσιάζεται ένα γενικό σχήμα τεχνικών διαχείρισης πόρων για την ικανοποίηση στόχων επίδοσης που ορίζονται για κλάσεις μονάδων φόρτου, και γίνεται μια επισκόπιση της σχετικής βιβλιογραφίας. Στα κεφάλαια 4 και 5 παρουσιάζεται λεπτομερώς η σχεδίαση και υλοποίηση του προσομοιωτή TPsim που αναπτύχθηκε στα πλαίσια αυτής της εργασίας. Ο προσομοιωτής εντάσσεται σε ένα ολοκληρωμένο περιβάλλον για την υποστήριξη πειραμάτων προσομοίωσης. Το περιβάλλον αυτό, που παρουσιάζεται στο κεφάλαιο

6, χρησιμοποιήθηκε για την πειραματική αξιολόγηση μιας σειράς τεχνικών χρονο–προγραμματισμού δοσοληψιών για σύνθετες μονάδες φόρτου, στο κεφάλαιο 7. Το θέμα των σύνθετων μονάδων φόρτου έχει αρχίσει να συγκεντρώνει έντονο ερευνητικό ενδιαφέρον. Η εργασία αυτή εξετάζει σχετικά θέματα επίδοσης, για τα οποία δεν υπάρχει μέχρι στιγμής κάλυψη στην βιβλιογραφία. Οι προτεινόμενες τεχνικές είναι προσανατολισμένες στην ικανοποίηση στόχων επίδοσης για σύνθετες μονάδες φόρτου. Στο κεφάλαιο 8 συνοψίζονται τα αποτελέσματα της εργασίας, και προτείνονται επεκτάσεις της και ερευνητικές κατευθύνσεις.

Κεφάλαιο 2

Επεξεργασία Δοσοληψιών

Στο κεφάλαιο αυτό, με βάση την επισκόπιση της βασικής έννοιας της δοσοληψίας που δόθηκε στο κεφάλαιο 1, παρουσιάζονται τα κύρια χαρακτηριστικά των συστημάτων επεξεργασίας δοσοληψιών (transaction processing systems). Διακρίνονται τα βασικά δομικά στοιχεία ενός συστήματος επεξεργασίας δοσοληψιών, και οι μεταξύ τους αλληλεπιδράσεις που είναι απαραίτητες για την εκτέλεση δοσοληψιών. Τα διάφορα τμήματα του συστήματος πρέπει να συνεργαστούν για να παρέχουν στον χρήστη τις εγγυήσεις που απαιτεί ο ορισμός της έννοιας της δοσοληψίας. Στο κεφάλαιο επίσης παρουσιάζεται λεπτομερώς η δομή και οργάνωση του επόπτη επεξεργασίας δοσοληψιών (transaction processing monitor), που αποτελεί τον κύριο κορμό ενός συστήματος επεξεργασίας δοσοληψιών. Η παρουσίαση στηρίζεται στον καθορισμό βασικών υπηρεσιών που πρέπει να παρέχει ο επόπτης επεξεργασίας δοσοληψιών. Η παρουσίαση αυτή δίδει ουσιαστικά τις προδιαγραφές για την σχεδίαση του προσομοιωτή συστημάτων επεξεργασίας δοσοληψιών TPsim που αναπτύχθηκε στα πλαίσια αυτής της εργασίας.

2.1 Συστήματα Επεξεργασίας Δοσοληψιών

Ενα σύστημα επεξεργασίας δοσοληψιών (transaction processing system), κατά την αναφορά [GR93], παρέχει εργαλεία για να διευκολύνει τον προγραμματισμό, την εκτέλεση και τη διαχείριση εφαρμογών. Οι εφαρμογές διατηρούν βάσεις δεδομένων που αναπαριστούν όψεις του πραγματικού κόσμου, και υποστηρίζουν ένα δίκτυο από συσκευές που υποβάλλουν επερωτήσεις και κάνουν ενημερώσεις. Οι απαντήσεις που δίνουν οι εφαρμογές μπορούν να καθοδηγούν συσκευές ελέγχου ώστε να μεταβάλλουν την κατάσταση. Οι εφαρμογές, οι βάσεις δεδομένων και τα δίκτυα εξελίσσονται με τον χρόνο. Τα συστήματα καταλήγουν να είναι όλο και περισσότερο γεωγραφικά κατανεμημένα, ετερογενή (δηλαδή περιλαμβάνουν εξοπλισμό και προγράμματα από πολλούς κατασκευαστές), ενώ είναι απαραίτητο να είναι συνεχώς διαθέσιμα (δηλαδή δεν υπάρχει προγραμματισμένος χρόνος παύσης της λειτουργίας των συστημάτων), και να παρέχουν προκαθορισμένες εγγυήσεις επίδοσης.

Είναι απαραίτητο ο προγραμματιστής εφαρμογών να καθορίζει επακριβώς τα όρια (αρχή και τέλος) κάθε δοσοληψίας. Η οριοθέτηση αυτή εν γένει υλοποιείται με την οργάνωση του κώδικα που εκτελεί τις πράξεις της δοσοληψίας σε μια συντακτική ενότητα (block) η αρχή και το τέλος της οποίας σημειώνεται με τις πράξεις BEGIN_WORK και COMMIT_WORK αντίστοιχα (βλ. σχήμα 1.2). Ο προγραμματιστής μπορεί επίσης

να ακυρώσει τα αποτελέσματα μιας δοσοληψίας που είναι σε εξέλιξη με την πράξη ROLLBACK_WORK.

Τα συστήματα διαχείρισης βάσεων δεδομένων παρέχουν τις εγγυήσεις που απαιτούν οι ιδιότητες ACID για δοσοληψίες που προσπελαύνουν τα δεδομένα που αυτά διαχειρίζονται. Για να εξασφαλιστούν οι ACID ιδιότητες χρησιμοποιούνται δύο βασικοί μηχανισμοί:

- Έλεγχος ταυτόχρονης προσπέλασης (concurrency control), συνήθως βάσει ενός πρωτοκόλλου κλειδώματος (locking), για να εξασφαλιστεί η απομόνωση των δοσοληψιών.
- Μηχανισμός επαναφοράς (recovery), βάσει ενός μηχανισμού καταγραφής μεταβολών (logging), για να εξασφαλιστεί η μονιμότητα και η ατομικότητα των δοσοληψιών, αφού όλες οι αλλαγές καταγράφονται στην μόνιμη (περιφερειακή) μνήμη του συστήματος και έτσι μπορούν να διατηρηθούν, ακόμα και μετά από βλάβες του συστήματος, αλλά και να αναιρεθούν (στην περίπτωση που μια δοσοληψία τερματίζει ανεπιτυχώς).

Ενα σύστημα διαχείρισης βάσεων δεδομένων παρέχει μηχανισμούς για τον έλεγχο ταυτόχρονης προσπέλασης σε κοινόχρηστα δεδομένα, και την επαναφορά τους σε συνεπή κατάσταση μετά από μια βλάβη του συστήματος ή μια αποτυχημένη δοσοληψία. Ομως η εμβέλεια των μηχανισμών περιορίζεται στα δεδομένα που ένα τέτοιο σύστημα διαχειρίζεται. Αν μια δοσοληψία χρειάζεται να προσπελάσει δεδομένα από διάφορες βάσεις δεδομένων, ή γενικότερα να χρησιμοποιήσει πόρους τους οποίους διαχειρίζονται περισσότεροι από ένας διαχειριστές πόρων, τότε για την εξασφάλιση της ατομικότητας απαιτείται συντονισμός (coordination) των ενεργειών των διαχειριστών πόρων όταν ολοκληρωθεί η εκτέλεση των πράξεων της δοσοληψίας. Ο συντονισμός αυτός υλοποιείται με ένα πρωτόκολλο δέσμευσης (commit protocol), το οποίο υποστηρίζεται από το σύστημα επεξεργασίας δοσοληψιών. Επιπλέον, τα συστήματα διαχείρισης βάσεων δεδομένων δεν παρέχουν επαρκή υποστήριξη για την εκτέλεση δοσοληψιών όπου οι αιτήσεις εξυπηρέτησης ακολουθούν το γενικό μοντέλο client−server, ζητώντας από το σύστημα κάποιες από τις διαθέσιμες υπηρεσίες, και όπου η εκτέλεση μιας δοσοληψίας μπορεί να περιλαμβάνει πολλαπλές αλληλεπιδράσεις τύπου client−server μεταξύ διαφόρων πελατών (clients) και μονάδων εξυπηρέτησης (servers).

Βασικό δομικό στοιχείο ενός συστήματος επεξεργασίας δοσοληψιών είναι ο επόπτης επεξεργασίας δοσοληψιών (transaction processing monitor). Η κύρια λειτουργία του επόπτη επεξεργασίας δοσοληψιών είναι η ενσωμάτωση πολλών διαφορετικών υποσυστημάτων σε ένα ενιαίο πλαίσιο, ώστε οι χρήστες και οι υπεύθυνοι διαχείρισης του συστήματος να έχουν στην διάθεσή τους μια ομοιόμορφη επαφή χρήσης που να υποστηρίζει ένα *εννιαίο* μοντέλο βλαβών, όπως αυτό ορίζεται από τις ιδιότητες ACID, και η αποτελεσματική διαχείριση όλων των πόρων του συστήματος.

2.1.1 Χαρακτηρισμός του Φόρτου Εξυπηρέτησης σε ένα Περιβάλλον Επεξεργασίας Δοσοληψιών

Είναι απαραίτητη μια περιγραφή της φύσης του φόρτου εξυπηρέτησης που έχει να διεκπεραιώσει ένα σύστημα επεξεργασίας δοσοληψιών. Η ποιοτική και ποσοτική περιγραφή του φόρτου εργασίας ενός συστήματος, που ονομάζεται *χαρακτηρισμός φόρτου εργασίας/εξυπηρέτησης* (workload characterization), είναι θεμελιώδης για την

κατανόηση της σχεδίασης και λειτουργίας του, καθώς για την μελέτη της επίδοσής του.

Οπως επισημαίνεται στην αναφορά [Ber90], συνήθως δίδεται έμφαση στο τμήμα της επεξεργασίας δοσοληψιών που έχει να κάνει με την διαχείριση βάσεων δεδομένων. Αυτή η προσέγγιση, αν και σημαντική, είναι ατελής αφού η ευχρηστία, αλλά και η επίδοση, ενός συστήματος επεξεργασίας δοσοληψιών εξαρτώνται σε μεγάλο βαθμό από το λειτουργικό σύστημα, που παρέχει το περιβάλλον εκτέλεσης, και τους διαθέσιμους μηχανισμούς επικοινωνίας. Ενα σύστημα επεξεργασίας δοσοληψιών υλοποιείται πάνω από μια ''πλατφόρμα'' (computing platform), που περιλαμβάνει κάποιο υλικό (hardware), ένα λειτουργικό σύστημα, και λογισμικό συστήματος για την υποστήριξη βασικών λειτουργιών, όπως η επικοινωνία μέσω ενός δικτύου επικοινωνίας. Οι διαθέσιμες ''πλατφόρμες'' εν γένει δεν υποστηρίζουν επαρκώς τις ανάγκες εφαρμογών επεξεργασίας δοσοληψιών. Ο λόγος που προβάλλεται στις αναφορές [Ber90] και [GR93] είναι το γεγονός ότι η επεξεργασία δοσοληψιών διαφέρει σημαντικά ως προς το μοντέλο υπολογισμού από τα άλλα γνωστά μοντέλα, που είναι το μοντέλο μαζικής επεξεργασίας (batch processing), το μοντέλο διαμοιρασμού χρόνου (time−sharing), και το μοντέλο πραγματικού χρόνου (real−time). Η ειδοποιός διαφορά εντοπίζεται στο γεγονός ότι τα συστήματα επεξεργασίας δοσοληψιών υποστηρίζουν ως αφαιρετικό μοντέλο της μονάδας φόρτου την δοσοληψία, η έννοια της οποίας δεν εμφανίζεται στα άλλα μοντέλα. Η δοσοληψία ως μονάδα φόρτου (unit of work) διαφέρει σημαντικά από την έννοια της διεργασίας (process/task) που υποστηρίζει ένα λειτουργικό σύστημα, κυρίως λόγω των ιδιοτήτων ACID που πρέπει να εξασφαλίζονται από το σύστημα.

Στην επεξεργασία δοσοληψιών, οι μονάδες φόρτου προσπελαύνουν κοινόχρηστα δεδομένα, με αυστηρούς περιορισμούς συνέπειας και διαθεσιμότητας. Ο συγχρονισμός των ταυτόχρονων προσπελάσεων σε δεδομένα, καθώς και η επαναφορά συνεπούς κατάστασης μετά μια αποτυχία ή βλάβη είναι ευθύνη του συστήματος, ώστε να εξασφαλίζεται η ικανοποίηση των ιδιοτήτων ACID, μια και το σύστημα χρησιμοποιεί υψηλός αριθμός χρηστών. Σε συστήματα ευρείας κλίμακας, όπως τα συστήματα κράτησης θέσεων που χρησιμοποιούν οι αεροπορικές εταιρίες (airline reservation systems) ο αριθμός των τερματικών μπορεί να είναι της τάξης των εκατοντάδων χιλιάδων. Ο φόρτος εξυπηρέτησης συγκροτείται από μονάδες φόρτου που έχουν μεταβλητές απαιτήσεις σε πόρους συστήματος. Οι αφίξεις αιτήσεων εξυπηρέτησης είναι τυχαίες. Είναι δυνατή η ταξινόμηση των μονάδων φόρτου σε κλάσεις, με βάση στατιστικές τεχνικές [Lab95] που λαμβάνουν υπ'όψιν ομοιότητες στις απαιτήσεις πόρων και στην κατανομή των προσπελάσεων σε δεδομένα. Για την εξυπηρέτηση μιας μονάδας φόρτου το σύστημα πρέπει να εκτελέσει ορισμένες λειτουργίες, ενεργοποιώντας το κατάλληλο πρόγραμμα εφαρμογής. Γενικά ο αριθμός των διαφορετικών προγραμμάτων εφαρμογής είναι της τάξης μερικών εκατοντάδων. Κάθε εκτέλεση ενός πρόγραμματος εφαρμογής συνήθως απαιτεί μερικές εκατοντάδες χιλιάδες εντολών μηχανής, και μια ή δυο δεκάδες προσπελάσεις στην περιφερειακή μνήμη. Αυτό ισχύει για on−line εφαρμογές (OLTP − On−Line Transaction Processing), όμως το σύστημα έχει να εξυπηρετήσει και μονάδες φόρτου μακράς διάρκειας με υψηλότερες απαιτήσεις σε πόρους. Οι μονάδες αυτές μπορεί να είναι είτε δοσοληψίες με απαιτήσεις παρόμοιες με αυτές των μονάδων φόρτου στο μοντέλο μαζικής επεξεργασίας (batch transactions) είτε επερωτήσεις (queries) για την στήριξη αποφάσεων (decision support). Το σύστημα οφείλει να υποστηρίξει με τρόπο αποτελεσματικό και τους δυο τύπους μονάδων φόρτου, και να παρέχει κάποιας μορφής εγγυήσεις επίδοσης, τόσο για τον αναμενόμενο χρόνο απόκρισης όσο και για τον ρυθμό εξυπηρέτησης μονάδων

φόρτου. Η βασική μονάδα φόρτου είναι η δοσοληψία, που αποτελεί επίσης την μονάδα ταυτόχρονης προσπέλασης αλλά και επανάκτησης.

Από την περιγραφή αυτή φαίνεται ότι ένα σύστημα επεξεργασίας δοσοληψιών είναι ουσιαστικά ένα σύστημα πραγματικού χρόνου που παρέχει τις ιδιότητες ACID για μονάδες φόρτου που προσπελαύνουν κοινόχρηστα δεδομένα. Οι περιορισμοί επίδοσης είναι μάλλον πιο χαλαροί από αυτούς που συνηθίζονται για συστήματα πραγματικού χρόνου που αλληλεπιδρούν άμεσα με τον πραγματικό κόσμο, γι'αυτό και χρησιμοποιείται συχνά ο όρος soft real−time.

2.1.2 Τύποι Δοσοληψιών

Διαφορετικοί τύποι δοσοληψιών απαιτούν διαφορετικού τύπου εξυπηρέτηση από το σύστημα [GR93]. Μια δοσοληψία μπορεί να είναι *άμεσα εκτελέσιμη* (direct), οπότε η αίτηση εξυπηρέτησης μεταβιβάζεται από το τερματικό του χρήστη στον κόμβο εξυπηρέτησης. Εάν κατά την διάρκεια της εκτέλεσης της δοσοληψίας χρειάζεται παραπέρα αλληλεπίδραση με τον χρήστη, τότε η δοσοληψία χαρακτηρίζεται *διαλογική* (conversational). Από την άλλη πλευρά, είναι δυνατόν η αίτηση εξυπηρέτησης να αποθηκευθεί από το σύστημα σε μια ουρά, η οποία να είναι μόνιμη (durable), μέχρι, βάσει μιας πολιτικής χρονοπρογραμματισμού, το σύστημα να ανακτήσει αυτή την αίτηση και να την εξυπηρετήσει. Μια δοσολοψία αυτού του τύπου χαρακτηρίζεται *μη διαλογική* (queued). Μια άλλη διάσταση στον χαρακτηρισμό των δοσοληψιών είναι η "πολυπλοκότητα" που τις χαρακτηρίζει. Η πολυπλοκότητα αυτή μπορεί να μετρηθεί είτε βάσει του αριθμού των μηνυμάτων που το σύστημα χρειάζεται να ανταλλάξει με τον χρήστη ή γενικότερα του αριθμού των βημάτων για την ολοκλήρωση της δοσο−ληψίας, είτε βάσει του αριθμού των προσπελάσεων σε δεδομένα. Ο χαρακτηρισμός αυτός εστιάζει στην διάρκεια της παραμονής της δοσοληψίας, ως μονάδας φόρτου, στο σύστημα, και, σύμφωνα με τα παραπάνω, λαμβάνει υπ'όψιν είτε την δομή της ροής ελέγχου είτε τον όγκο των απαιτούμενων δεδομένων. Τέλος, η εκτέλεση της δοσοληψίας μπορεί να διεκπεραιωθεί τοπικά, σε ένα κόμβο, ή να είναι κατανεμημένη.

2.1.3 Μοντέλο Προγραμματισμού

Ενα σύστημα επεξεργασίας δοσοληψιών παρέχει ένα περιβάλλον για την ανά−πτυξη και εκτέλεση εφαρμογών με την δυνατότητα για επικοινωνία με συσκευές εισόδου/εξόδου και βάσεις δεδομένων. Υποστηρίζεται ένας μηχανισμός για κλήση μακρινής διαδικασίας (RPC − remote procedure call) που επιτρέπει την ανάπτυξη κατανεμημένων εφαρογών που οργανώνονται κατά το μοντέλο client−server. Ο μηχα−νισμός αυτός ονομάζεται TRPC (transactional RPC), διότι μεταφέρει τον κωδικό της δοσοληψίας που εκτελεί ο client στον server, ώστε ο server να καταγραφεί, από τον διαχειριστή δοσοληψιών στον κόμβο όπου αυτός εκτελείται, ως μέρος της δοσοληψίας που εκτελεί ο client.

Το μοντέλο προγραμματισμού στηρίζεται στην έννοια του διαχειριστή πόρων (resource manager). Με τον όρο αυτό εννοείται ένα υποσύστημα, αποτελούμενο από διεργασίες, κώδικα, και δεδομένα, που επιτρέπει προσπέλαση σε κοινόχρη−στα δεδομένα και υπηρεσίες. Παραδείγματα διαχειριστών πόρων είναι ένα σύστημα διαχείρισης δοσοληψιών, και ένα σύστημα διαχείρισης επικοινωνίας. Ενα σύστημα επεξεργασίας δοσοληψιών παρέχει μια σειρά από διαχειριστές πόρων, και ένα βασικό υποσύστημα που ονομάζεται *επόπτης επεξεργασίας δοσοληψιών* (transaction processing

monitor) και έχει την ευθύνη να παρέχει το περιβάλλον εκτέλεσης για τους διαχειριστές πόρων και τις δοσοληψίες, και να εγγυηθεί τις ιδιότητες ACID. Η κατάσταση των εφαρμογών παριστάνεται ως μόνιμα (durable) δεδομένα που διατηρούνται από τους διαχειριστές πόρων. Οι μετασχηματισμοί μεταξύ καταστάσεων, καθώς και οι επερωτήσεις για την ανάκτηση στοιχείων σχετικών με την τρέχουσα κατάσταση (βλ. σχήμα 1.1), υλοποιούνται ως δοσοληψίες, με τις ιδιότητες ACID.

2.1.4 Δομή Συστημάτων Επεξεργασίας Δοσοληψιών

Ο επόπτης επεξεργασίας δοσοληψιών επιβάλλει μια συγκεκριμένη δομή στις εφαρμογές επεξεργασίας δοσοληψιών που καλείται να εξυπηρετήσει. Η μεγάλη πλειοψηφία των εφαρμογών είναι δομημένες ώστε να επιτελούν τις ακόλουθες λειτουργίες:

1. Αλληλεπίδραση με το τερματικό, ώστε να συλλεχθεί η είσοδος.
2. Μετατροπή της εισόδου σε μια τυποποιημένη μορφή αιτήσεων.
3. Εναρξη δοσοληψίας.
4. Εξακρίβωση του τύπου της δοσοληψίας.
5. Εκτέλεση του αντίστοιχου προγράμματος εφαρμογής.
6. Επιτυχής τερματισμός της δοσοληψίας.
7. Αποστολή των αποτελεσμάτων στο τερματικό.

Το μοντέλο αυτό περιγράφεται στην αναφορά [Ber90].

Ενα σύστημα επεξεργασίας δοσοληψιών παρέχει τρόπους διασύνδεσης μεταξύ εφαρμογών και διαχειριστών πόρων (βλ. σχήμα 2.1), εξασφαλίζοντας τις ιδιότητες ACID. Η βάση για την δυνατότητα συνεργασίας μεταξύ διαχειριστών πόρων στα πλαίσια της εκτέλεσης ενός προγράμματος εφαρμογής είναι η ύπαρξη ενός standard μηχανισμού για την κλήση υπηρεσιών από προγράμματα εφαρμογών και διαχειριστές πόρων, που να μπορεί να λειτουργήσει και σε ένα κατανεμημένο περιβάλλον επεξεργασίας (βλ. σχήμα 2.2). Ο μηχανισμός αυτός είναι ο μηχανισμός TRPC.

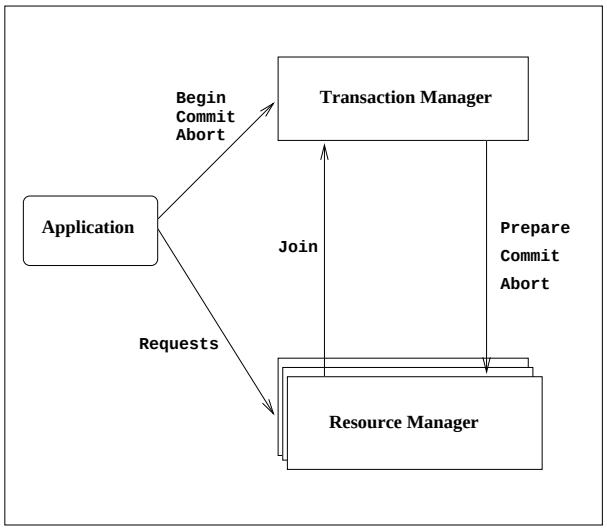
2.2 Επόπτες Επεξεργασίας Δοσοληψιών: Στόχοι και Χαρακτηριστικά

Στην ενότητα αυτή παρουσιάζεται λεπτομερώς η δομή και οργάνωση του επόπτη επεξεργασίας δοσοληψιών, που αποτελεί τον συνδετικό ιστό ενός συστήματος επεξεργασίας δοσοληψιών. Η παρουσίαση στηρίζεται στον καθορισμό βασικών υπηρεσιών που πρέπει να παρέχει ο επόπτης επεξεργασίας δοσοληψιών.

2.2.1 Βασικές Υπηρεσίες

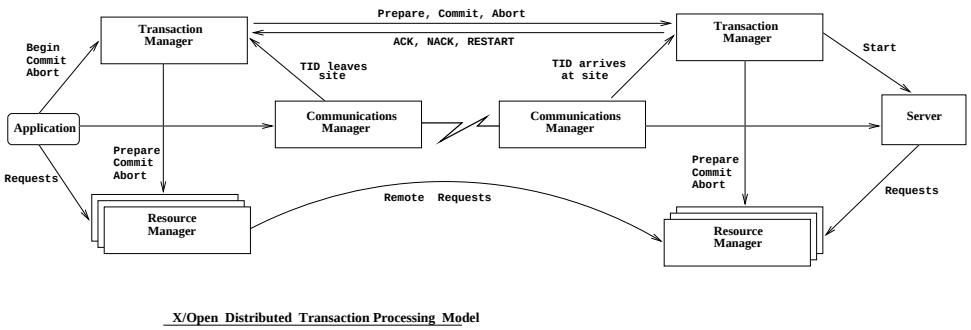
Οι βασικές υπηρεσίες που παρέχει ένας επόπτης επεξεργασίας δοσοληψιών είναι οι εξής:

- Μηχανισμός TRPC. Επιτρέπει την κλήση υπηρεσιών, ενδεχομένως από ‘‘μακρινές’’ μονάδες εξυπηρέτησης (servers), και επιβάλλει ελέγχους πρόσβασης (authentication/authorization).



Σχήμα 2.1: Το $X/Open$ Μοντέλο Επεξεργασίας Δοσοληψιών.

- Διαχείριση Δοσοληψιών. Συντονίζει τις ενέργειες για τον τερματισμό μιας δοσοληψίας, ώστε να εξασφαλίζονται οι ιδιότητες της ατομικότητας και της μονιμότητας. Επίσης, έχει την ευθύνη για τον συντονισμό των ενεργειών για επαναφορά του συστήματος σε μια συνεπή κατάσταση μετά από μια βλάβη.
- Μηχανισμός Καταγραφής Μεταβολών. Επιτρέπει την καταγραφή μεταβολών στην κατάσταση του συστήματος, ώστε σε περίπτωση βλάβης ή αποτυχίας να είναι δυνατή η επαναφορά σε μια συνεπή κατάσταση.
- Μηχανισμός Ελέγχου Ταυτόχρονης Προσπέλασης. Επιτρέπει τον συντονισμό ταυτόχρονων προσπελάσεων σε κοινόχρηστους πόρους ώστε οι διαχειριστές πόρων να είναι σε θέση να ικανοποιούν την ιδιότητα της απομόνωσης.

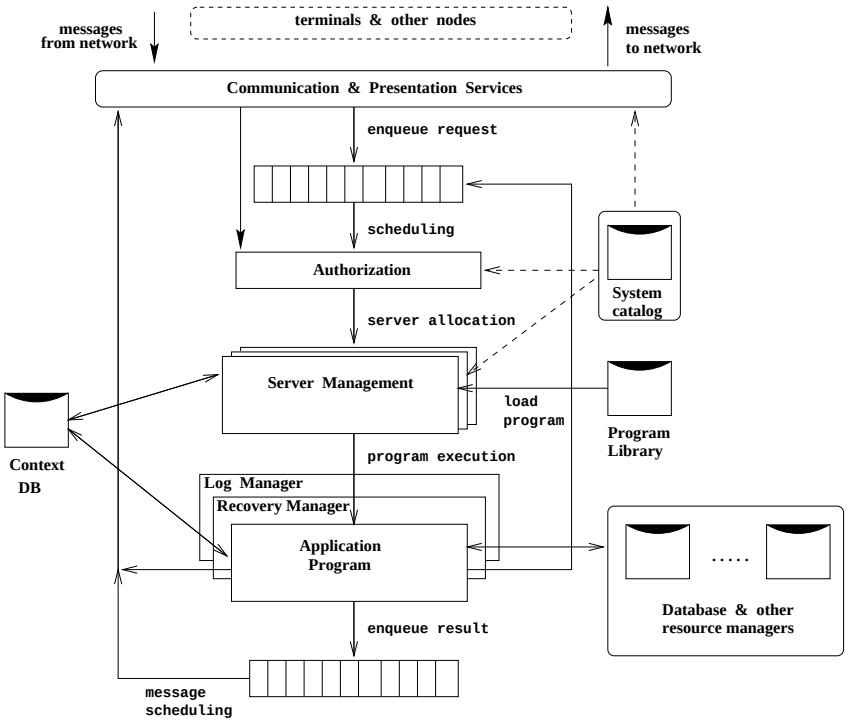


Σχήμα 2.2: Το $X/Open$ Μοντέλο Επεξεργασίας Δοσοληψιών για Κατανεμημένα Συστήματα.

Οι διαχειριστές πόρων που εντάσσονται στο σύστημα, όπως συστήματα διαχείρισης βάσεων δεδομένων και συστήματα επικοινωνιών, κάνουν χρήση αυτών των υπηρεσιών, και κατ'αυτόν τον τρόπο επεκτείνουν το σύστημα επεξεργασίας δοσοληψιών. Σε ένα κατανεμημένο περιβάλλον εργασίας, υπάρχει ένα ξεχωριστό σύστημα επεξεργασίας δοσοληψιών σε κάθε κόμβο του δικτύου διασύνδεσης. Τα συστήματα αυτά συνεργάζονται ώστε να παρέχουν στον χρήστη ένα κατανεμημένο περιβάλλον εκτέλεσης δοσοληψιών.

2.2.2 Δομή του Επόπτη Επεξεργασίας Δοσοληψιών

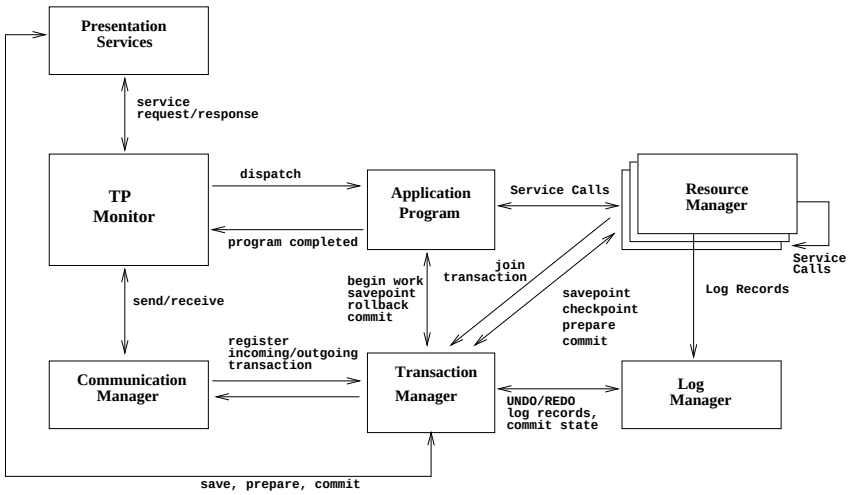
Βασική λειτουργία του επόπτη επεξεργασίας δοσοληψιών είναι ο συντονισμός της ροής αιτήσεων μέσα σε ένα σύστημα επεξεργασίας δοσοληψιών, μεταξύ τερματικών ή άλλων συσκευών εισόδου και προγραμμάτων εφαρμογών που επεξεργάζονται τις αιτήσεις αυτές. Στο σχήμα 2.3 απεικονίζεται η ροή ελέγχου του επόπτη επεξεργασίας δοσοληψιών, και οι βασικές λειτουργίες που είναι απαραίτητες για την διαχείριση δοσοληψιών.



Σχήμα 2.3: Ροή Ελέγχου ενός Επόπτη Επεξεργασίας Δοσοληψιών.

Στο σχήμα 2.4 φαίνεται η συνεργασία ανάμεσα στα τμήματα που συγκροτούν ένα σύστημα επεξεργασίας δοσοληψιών για την εκτέλεση μιας δοσοληψίας, μέσα στο περιβάλλον εκτέλεσης που παρέχει ο επόπτης επεξεργασίας δοσοληψιών. Ο επόπτης επεξεργασίας δοσοληψιών συντονίζει την ροή αιτήσεων εξυπηρέτησης, και παρέχει το περιβάλλον εκτέλεσης και τους πόρους για την εκτέλεσή τους. Ο διαχειριστής δοσοληψιών έχει την ευθύνη να κατευθύνει την εκτέλεση ώστε να

εξασφαλιστούν οι ιδιότητες της ατομικότητας και της μονιμότητας. Ο μηχανισμός καταγραφής αλλαγών χρησιμοποιείται από τον διαχειριστή δοσοληψιών και τους διαχειριστές πόρων. Ο διαχειριστής επικοινωνιών επιτρέπει την κατανομή της εκτέλεσης της δοσοληψίας σε πολλαπλούς κόμβους, και ενημερώνει σχετικά τον διαχειριστή δοσοληψιών. Ο μηχανισμός ελέγχου ταυτόχρονης προσπέλασης (που δεν φαίνεται στο σχήμα) εξασφαλίζει την ιδιότητα της απομόνωσης.



Σχήμα 2.4: Συνεργασία ανάμεσα στα Δομικά Στοιχεία ενός Συστήματος Επεξεργασίας Δοσοληψιών.

2.2.3 Διαχείριση Ουρών

Μια αίτηση εξυπηρέτησης (request) παριστάνεται με μια εγγραφή (record) η οποία περιγράφει κάποια εργασία που το σύστημα πρέπει να εκτελέσει για λογαριασμό ενός χρήστη. Μια ουρά (queue) σε ένα σύστημα επεξεργασίας δοσοληψιών λειτουργεί ως διάμεσος ανάμεσα σε ένα client και ένα server [BHM90]. Ο client εισάγει μια αίτηση εξυπηρέτησης στην ουρά που εξυπηρετείται από τον server, ο οποίος εξάγει την αίτηση αυτή από την ουρά και εκτελεί την ζητούμενη υπηρεσία. Η διαχείριση ουρών αποτελεί μια από τις βασικές υπηρεσίες που υποστηρίζουν τα περισσότερα συστήματα επεξεργασίας δοσοληψιών. Η υπηρεσία αυτή υλοποιείται από ένα ειδικό υποσύστημα που ονομάζεται *διαχειριστής ουρών* (queue manager, ή queueing facility). Ο διαχειριστής ουρών συνδιάζει στοιχεία από συστήματα βάσεων δεδομένων και συστήματα επικοινωνίας. Αποθηκεύει εγγραφές, και επιτρέπει την ανάκλησή τους, και επίσης μπορεί να μεταφέρει μηνύματα μεταξύ διεργασιών ή δοσοληψιών κατά τρόπο ασύγχρονο. Οι πράξεις πάνω στα στοιχεία μιας ουράς μπορούν να εκτελεστούν μέσα στα πλαίσια μιας δοσοληψίας, συνεπώς πρέπει να είναι δυνατή η επανάκτηση της αρχικής κατάστασης της ουράς εάν μια τέτοια δοσοληψία αποτύχει. Μέσω ουρών είναι δυνατή η επικοινωνία μεταξύ διεργασιών κατά τρόπο έμμεσο, χωρίς να χρειάζεται η μια διεργασία να συνδεθεί (bind) άμεσα με την άλλη. Οι διεργασίες που επικοινωνούν με τον τρόπο αυτό αρκεί να συνδεθούν με την μια ουρά. Αυτός ο τύπος επικοινωνίας προστατεύει από αποτυχίες διεργασιών (process failures), καθώς

η έμμεση σύνδεση επιτρέπει στις διεργασίες να λειτουργούν ανεξάρτητα, χωρίς συγχρονισμό, καθώς και να αποτυγχάνουν ανεξάρτητα. Για παράδειγμα, μια διεργασία μπορεί να εισάγει μια αίτηση εξυπηρέτησης σε μια ουρά για να εκτελεστεί από μια άλλη διεργασία, χωρίς να είναι απαραίτητο οι δυο διεργασίες να είναι ταυτόχρονα διαθέσιμες. Είναι σαφές ότι η ασύγχρονη επικοινωνία μέσω ουρών είναι επίσης σε θέση να καλύψει αποτυχίες του συστήματος επικοινωνίας (communication failures). Για παράδειγμα, μια διεργασία μπορεί να συγκεντρώνει αιτήσεις εξυπηρέτησης σε μια τοπική ουρά, και κατά περιόδους να μεταφέρει τις σχετικές εγγραφές στην ουρά που εξυπηρετείται από μια "μακρυνή" διεργασία. Για την διεργασία−πελάτη η διεργασία εξυπηρέτησης φαίνεται να παρέχει μια αξιόπιστη υπηρεσία, ακόμα και όταν η μεταξύ τους επικοινωνία είναι αδύνατη λόγω προβλημάτων στο δίκτυο επικοινωνίας. Η δυνατότητα μεταφοράς εγγραφών μέσω ουρών μεταξύ δοσοληψιών είναι ο βασικός μηχανισμός που παρέχει ένας επόπτης επεξεργασίας δοσοληψιών για την υποστήριξη σύνθετων μονάδων φόρτου.

Η χρήση ουρών διευκολύνει την επεξεργασία κατά ομάδες (batch processing), καθώς οι αιτήσεις εξυπηρέτησης μπορούν να συγκεντρώνονται, αξιόπιστα, σε μια ουρά, απ'όπου μια διεργασία εξυπηρέτησης μπορεί αργότερα να τις ανακτήσει. Είναι δυνατόν περισσότερες από μια διεργασίες να αντλούν εγγραφές από μια ουρά, συνεπώς είναι δυνατός ο καταμερισμός του φόρτου εξυπηρέτησης (load sharing), κατά τρόπο διαφανή στον χρήστη. Επιπλέον, η χρήση ουρών για να κρατηθούν οι αιτήσεις εξυπηρέτησης που έρχονται σε ένα σύστημα μέχρι αυτές να ανακτηθούν από μια διεργασία εξυπηρέτησης επιτρέπει στο σύστημα να ανταπεξέλθει σε απότομες παροδικές αυξήσεις του ρυθμού αφίξεων, καθώς ανεξάρτητα του ρυθμού αφίξεων οι διεργασίες εξυπηρέτησης "βλέπουν" τις εισερχόμενες αιτήσεις εξυπηρέτησης με τον ρυθμό που μπορούν να τις χειριστούν.

2.3 Επισκόπηση της Σχετικής Βιβλιογραφίας

Αν και τα συστήματα επεξεργασίας δοσοληψιών χρησιμοποιούνται ευρύτατα, σε πληθώρα πεδίων εφαρμογών, υπάρχουν λίγες σχετικές αναφορές στην ανοικτή βιβλιογραφία, αν και υπάρχει αρκετή βιβλιογραφία για επιμέρους υποσυστήματα, όπως για τον έλεγχο ταυτόχρονης προσπέλασης και την διαχείριση μνήμης. Ο λόγος είναι το γεγονός ότι η σχεδίαση και ανάπτυξη συστημάτων επεξεργασίας δοσοληψιών, και ειδικά εποπτών επεξεργασίας δοσοληψιών, διεξάγεται κυρίως σε ερευνητικά κέντρα εταιριών, απ'όπου πολύ λίγα στοιχεία δημοσιοποιούνται. Μια σημαντική πηγή πληροφοριών είναι πάντως τα εγχειρίδια χρήσης που παρέχουν οι διάφοροι κατασκευαστές.

Η πλέον κλασσική αναφορά σχετικά με θέματα επεξεργασίας δοσοληψιών είναι η [Gra78], όπου παρουσιάζονται με σημαντική λεπτομέρεια πρωτόκολλα ελέγχου ταυτόχρονης προσπέλασης (hierarchical two−phase locking), καταγραφής μεταβολών, και δέσμευσης, για την ικανοποίηση των ιδιοτήτων ACID. Γενικά στοιχεία για τα θέματα αυτά περιέχονται και στις αναφορές [CP84] και [OV91]. Η αναφορά [Ber90] εστιάζει στην παρουσίαση της δομής και λειτουργίας εποπτών επεξεργασίας δοσοληψιών, με έμφαση στην λειτουργία του περιβάλλοντος εκτέλεσης. Τα θέματα αυτά παρουσιάζονται με μεγάλη λεπτομέρεια στην αναφορά [GR93], μαζί με στοιχεία για την υλοποίηση βασικών υπηρεσιών.

Υπάρχουν λίγες αναφορές σχετικά με υλοποιήσεις ερευνητικών πρότυπων συστη−

μάτων. Ενδιαφέρον παρουσιάζει το περιβάλλον ανάπτυξης κατανεμημένων εφαρμογών Argus [Lis88], όπου η έννοια της δοσοληψίας χρησιμοποιείται ως βασικός μηχανισμός για τον χειρισμό εξαιρέσεων (exception handling). Πλήρη υποστήριξη για εκτέλεση δοσοληψιών παρέχει το σύστημα Camelot [EMS91], το οποίο στη συνέχεια αποτέλεσε την βάση για το εμπορικό σύστημα Encina. Το σύστημα αυτό υποστηρίζει nested transactions [Mos85], και παρέχει μια γλώσσα, επέκταση της C, για την ανάπτυξη εφαρμογών. Το σύστημα αναπτύχθηκε πάνω από το λειτουργικό σύστημα Mach. Στις αναφορές [SO92, Sel92] παρουσιάζεται η υλοποίηση υπηρεσιών για την υποστήριξη δοσοληψιών πάνω από το λειτουργικό σύστημα UNIX, τόσο σε επίπεδο χρήστη (user level) όσο και ως επέκταση του πυρήνα (kernel) του λειτουργικού συστήματος.

Υπάρχει πληθώρα από επόπτες επεξεργασίας δοσοληψιών. Μπορεί κανείς να ξεχωρίσει τα συστήματα IMS και CICS της IBM [Kag89], το σύστημα TUXEDO [Nov93] της Novell, το σύστημα Encina της Transarc [She93, YTJ91], και τα συστήματα ACMS και DECintact της DEC [SS91], που εντάσσονται στην αρχιτεκτονική DECdta [BET91] για τα λειτουργικά συστήματα VMS και UNIX. Ιδιαίτερο ενδιαφέρον παρουσιάζει η αρχιτεκτονική Guardian 90 της Tandem [GR93], που παρέχει μια ολοκληρωμένη λύση στο πρόβλημα της επεξεργασίας δοσοληψιών, καθώς περιλαμβάνει ένα λειτουργικό σύστημα (Guardian) για πολυπεξεργαστές με fault−tolerant αρχιτεκτονική, ένα επόπτη επεξεργασίας δοσοληψιών (Pathway/TMF), ένα σύστημα διαχείρισης βάσεων δεδομένων (NonStop SQL), και λογισμικό επικοινωνιών (Expand) για WAN δίκτυα.

Σε εξέλιξη βρίσκονται προσπάθειες για την ανάπτυξη και καθιέρωση διεθνών προτύπων για να είναι δυνατή η συνεργασία μεταξύ ετερογενών κατανεμημένων συστημάτων επεξεργασίας δοσοληψιών από τους οργανισμούς ISO και OSI. Σε ανάπτυξη είναι τα πρότυπα X/Open DTP και OSI TP, που περιλαμβάνουν αντίστοιχα προδιαγραφές μιας κοινής επαφής χρήσης (API − application programming interface) για τους διαχειριστές πόρων και το σύστημα διαχείρισης δοσοληψιών, και προδιαγραφές για τα μηνύματα με τα οποία επικοινωνούν τα διάφορα τμήματα ενός συστήματος επεξεργασίας δοσοληψιών, τόσο μεταξύ τους όσο και με τα προγράμματα εφαρμογών. Τα σχήματα 2.1 και 2.2 δίδουν μια εικόνα ενός συστήματος επεξεργασίας δοσοληψιών κατά το πρότυπο X/Open DTP.

Κεφάλαιο 3

Διαχείριση Πόρων για την Ικανοποίηση Στόχων Επίδοσης

Στο κεφάλαιο αυτό παρουσιάζεται το πρόβλημα της δυναμικής διαχείρισης πόρων σε συστήματα επεξεργασίας δοσοληψιών, και εισάγεται η έννοια της διαχείρισης συστημάτων από απόψεως επίδοσης, με σκοπό την παροχή εγγυήσεων στους χρήστες για το επίπεδο παροχής υπηρεσιών από το σύστημα. Για την επίτευξη αυτού του σκοπού έχουν προταθεί στην βιβλιογραφία τεχνικές που βασίζονται στην διατύπωση στόχων επίδοσης για κάθε κατηγορία μονάδων φόρτου που το σύστημα έχει να εξυπηρετήσει. Οι τεχνικές αυτές επιχειρούν να ικανοποιήσουν προδιαγραφές επίδοσης για τις κλάσεις μονάδων φόρτου ασκώντας έλεγχο σε κρίσιμες παραμέτρους του συστήματος, που επηρεάζουν την ανάθεση πόρων σε μονάδες φόρτου εξυπηρέτησης. Παρουσιάζεται μια επισκόπηση της σχετικής βιβλιογραφίας, και επισημαίνεται η ανάγκη για δυναμική διαχείριση σύνθετων μονάδων φόρτου, όπως τα workflows (κεφάλαιο 7), από απόψεως επίδοσης. Το θέμα αυτό άρχισε πρόσφατα να συγκεντρώνει ενδιαφέρον. Στα πλαίσια της εργασίας αυτής σχεδιάστηκαν και μελετήθηκαν τεχνικές για την ικανοποίηση στόχων επίδοσης σύνθετων μονάδων φόρτου. Το παρόν κεφάλαιο σκοπεύει να χρησιμεύσει ως βάση για την παρουσίαση των τεχνικών αυτών στο κεφάλαιο 7.

3.1 Πόροι Συστήματος και Διαχειριστές Πόρων

Ο όρος ‘‘πόρος συστήματος’’ (system resource) αναφέρεται σε κάθε αντικείμενο που χρησιμοποιείται από μια μονάδα φόρτου για την εκτέλεσή της, και για το οποίο μπορεί να υπάρξει ανταγωνισμός. Ο όρος καλύπτει και το υλικό (hardware) και το λογισμικό (software) του συστήματος. Παραδείγματα πόρων συστήματος είναι ο επεξεργαστής, η μνήμη (σε όλα τα επίπεδα ιεραρχίας), το διαθέσιμο εύρος ζώνης για επικοινωνία (communication bandwidth), αλλά και κάθε είδους δεδομένα και κώδικας στα οποία η ταυτόχρονη προσπέλαση υπόκειται σε περιορισμούς (όπως για παράδειγμα μια κρίσιμη περιοχή κώδικα (critical region), την οποία μόνο μια διεργασία μπορεί να εκτελεί ανά πάσα στιγμή, ή ακόμα κάποια δεδομένα στα οποία επιτρέπεται ταυτόχρονη προσπέλαση για ανάγνωση αλλά απαιτείται αποκλειστική πρόσβαση για εγγραφή). Το ενδεχόμενο ανταγωνισμού ανάμεσα σε μονάδες φόρτου για την χρήση ενός πόρου, που προκύπτει ως συνέπεια της ελεγχόμενης πρόσβασης στον πόρο, δημιουργεί την ανάγκη για ανάπτυξη και εφαρμογή πολιτικών διαχείρισης πόρων. Οι διαχειριστές πόρων (resource managers) είναι οι οντότητες που έχουν την ευθύνη της εφαρμογής

της πολιτικής ανάθεσης για ένα πόρο.

Μονάδες φόρτου που δεν μπορούν αμέσως να προσπελάσουν ένα πόρο συστήματος τον οποίο χρειάζονται, περιμένουν σε μια ουρά αναμονής που σχετίζεται με τον πόρο. Μια μονάδα φόρτου μπορεί να περιμένει, κατά την διάρκεια της εκτέλεσής της, για πολλούς πόρους. Για παράδειγμα, μια δοσοληψία, που αποτελεί την μονάδα φόρτου που απασχολεί αυτή την εργασία, περιμένει αρχικά στο υποσύστημα δρομολόγησης δοσοληψιών (transaction router) για να σταλεί για εξυπηρέτηση σε κάποιο κόμβο. Οταν το μήνυμα έναρξης εκτέλεσης της δοσοληψίας φτάσει στον κόμβο, η δοσοληψία (ακριβέστερα το νήμα ελέγχου ή η διεργασία που την εκτελεί) περιμένει για εξυπηρέτηση από τον επεξεργαστή, για να αρχίσει να εκτελεί προσπελάσεις στην βάση δεδομένων. Κάθε προσπέλαση σε δεδομένα αναλύεται σε αιτήσεις εξυπηρέτησης στον ελεγκτή ταυτόχρονης προσπέλασης (concurrency control manager), που επίσης είναι ένας διαχειριστής πόρων, και στον διαχειριστή του ενταμιευτή δεδομένων (database buffer manager). Μετά την επικύρωση του δικαιώματος πρόσβασης στα ζητούμενα δεδομένα, εάν αυτά δεν βρίσκονται στον ενταμιευτή, ο διαχειριστής του ενταμιευτή χρειάζεται να ζητήσει, για λογαριασμό της δοσοληψίας, εξυπηρέτηση από το υποσύστημα διαχείρισης αρχείων δεδομένων. Από το παράδειγμα είναι σαφές ότι ένας διαχειριστής πόρων λειτουργεί και ως εξυπηρετητής (server) παρέχοντας υπηρεσίες σε μονάδες φορτίου, αλλά και ως πελάτης (client) άλλων διαχειριστών πόρων.

Για να εφαρμοστεί μια πολιτική διαχείρισης πόρων είναι απαραίτητο ο αντίστοιχος διαχειριστής να μπορεί να ελέγξει παραμέτρους ελέγχου που επηρεάζουν την ανάθεση πόρων σε μονάδες φόρτου και να έχει πληροφόρηση για την χρήση του πόρου. Γενικά, οι πληροφορίες που διατηρεί ένας διαχειριστής πόρων είναι καθαρά τοπικού χαρακτήρα, δηλαδή αφορούν μόνο τους πόρους που αυτός ελέγχει, και περιλαμβάνει στοιχεία όπως ο χρόνος άφιξης και αποχώρησης κάθε μονάδας φόρτου, και οι απαιτήσεις κάθε μονάδας φόρτου. Αν και η χρήση πληροφορίας για ολόκληρο το σύστημα είναι κατ'αρχήν δυνατή, η ανταλλαγή πληροφοριών ανάμεσα σε διαχειριστές πόρων, ειδικά σε κατανεμημένα συστήματα, θέτει σημαντικά προβλήματα υλοποίησης, και επιβαρύνει την λειτουργία του συστήματος με το κόστος της επικοινωνίας για την συγκέντρωση της πληροφορίας. Επίσης τίθεται θέμα αξιοπιστίας της πληροφορίας που συλλέγεται από "μακρινούς" κόμβους, λόγω της μη αμελητέας καθυστέρησης στην μετάδοση της πληροφορίας. Αποφάσεις ανάθεσης πόρων που βασίζονται σε πληροφορία που δεν είναι πλέον έγκυρη είναι πολύ πιθανό να είναι επιζήμιες για την επίδοση του συστήματος.

3.2 Διαχείριση συστημάτων από απόψεως επίδοσης

Η διαχείριση συστημάτων περιλαμβάνει, ως υποπρόβλημα, την διαδικασία της παρακολούθησης και ανάλυσης της επίδοσης, καθώς και την λήψη μέτρων για την αντιμετώπιση προβλημάτων επίδοσης για τις διάφορες κλάσεις μονάδων φόρτου που το σύστημα έχει να εξυπηρετήσει. Η διαχείριση συστημάτων από απόψεως επίδοσης αποτελεί σημαντική τεχνική πρόκληση λόγω της πολυπλοκότητας των προγραμμάτων εφαρμογής και της ποικιλίας των απαιτήσεων σε πόρους. Η δυσκολία του προβλήματος είναι ιδιαίτερα αυξημένη σε κατανεμημένα συστήματα.

Ακόμα και όταν, υπό κανονικές συνθήκες λειτουργίας, το σύστημα είναι σε θέση να καλύψει πλήρως τις απαιτήσεις πόρων όλων των κλάσεων μονάδων φόρτου, υπάρχει πάντα το ενδεχόμενο να εμφανιστεί υπερφόρτωση λόγω μιας παροδικής αύξησης

του προσφερόμενου φόρτου εξυπηρέτησης (load upsurge). Τέτοιες καταστάσεις πρέπει να αντιμετωπίζονται, με τρόπο αποφασιστικό, σύντομα αφού εμφανιστούν. Η πιθανότητα εμφάνισης του προβλήματος της υπερφόρτωσης εν γένει αυξάνει με την πάροδο του χρόνου, όσο οι εφαρμογές, και οι απαιτήσεις τους σε πόρους, γίνονται πιο σύνθετες. Καθώς το σύστημα έχει να εξυπηρετήσει πολλαπλές κλάσεις εφαρμογών, είναι απαραίτητο να επιλυθούν προβλήματα ανάθεσης πόρων κάνοντας συμβιβασμούς ανάμεσα στις συγκρουόμενες απαιτήσεις των εφαρμογών. Είναι επίσης σημαντικό να λαμβάνονται μέτρα για να προστατευθούν, από απόψεως κατανομής πόρων, οι κρίσιμης σημασίας εφαρμογές από την ‘‘παρενόχληση’’ από λιγότερο σημαντικές εφαρμογές.

Αποτελεί κοινή πρακτική οργανισμοί να ζητούν από τους κατασκευαστές/προμηθευτές υπολογιστικών συστημάτων και λογισμικού να υπογράψουν μια σύμβαση ως δέσμευση για το επίπεδο παροχής υπηρεσιών (quality of service). Μια σύμβαση αυτού του είδους, που αναφέρεται με τον όρο service level agreement − SLA [Noo89] ουσιαστικά συμπληρώνει τις λειτουργικές προδιαγραφές του συστήματος με την διατύπωση προδιαγραφών επίδοσης. Το πρόβλημα της διαχείρισης συστημάτων από απόψεως επίδοσης ανάγεται στην υλοποίηση των εγγυήσεων που υπόσχεται η σύμβαση SLA, με την εφαρμογή πολιτικών ανάθεσης πόρων.

Είναι απαραίτητο το ίδιο το σύστημα να μετέχει ενεργά στην διαχείρισή του από πλευράς επίδοσης, καθώς δεν είναι πρακτικό [BCL93a, WHMZ93a, WHMZ93b] ο υπεύθυνος για την διαχείριση του συστήματος να τροποποιήσει τις παραμέτρους ελέγχου για κάθε πόρο του συστήματος ξεχωριστά, και να λάβει υπ’όψιν όλες τις μεταξύ τους αλληλεπιδράσεις. Σε κάθε σύστημα, υπάρχουν πολλαπλά επίπεδα ελέγχου, τα οποία αλληλεπιδρούν. Το πρόβλημα γίνεται ιδιαίτερα σύνθετο όταν πρέπει να γίνουν ρυθμίσεις και σε παραμέτρους του λειτουργικού συστήματος σε κάθε κόμβο του συστήματος αλλά και σε παραμέτρους ελέγχου συστημάτων λογισμικού, όπως ένα σύστημα διαχείρισης βάσεων δεδομένων ή ένας επόπτης επεξεργασίας δοσοληψιών, που υλοποιούν τις δικές τους πολιτικές διαχείρισης πόρων πάνω από τις βασικές πολιτικές διαχείρισης που επιβάλλει το λειτουργικό σύστημα. Για παράδειγμα, ένα σύστημα διαχείρισης βάσεων δεδομένων μπορεί να ελέγχει το μέγεθος του ενταμιευτή δεδομένων, ζητώντας επιπλέον μνήμη από το διαχειριστή μνήμης του λειτουργικού συστήματος, και να εφαρμόζει την δική του πολιτική αντικατάστασης σελίδων, ενώ ένας επόπτης επεξεργασίας δοσοληψιών μπορεί να διαχειρίζεται νήματα ελέγχου για την εξυπηρέτηση δοσοληψιών, και να ελέγχει την εκτέλεσή τους υλοποιώντας μια πολιτική χρονοπρογραμματισμού βάσει προτεραιοτήτων. Η ρύθμιση παραμέτρων ελέγχου από κάποιον ειδικό είναι δύσκολη λόγω του χαμηλού επιπέδου αφαίρεσης που χαρακτηρίζει τις παραμέτρους αυτές. Παραδείγματα παραμέτρων ελέγχου σε ένα σύστημα διαχείρισης βάσεων δεδομένων είναι το μέγεθος των περιοχών ενταμιευτών (buffer pools), και ο βαθμός πολυπρογραμματισμού. Επιπλέον, η φύση πολλών ρυθμίσεων σε παραμέτρους ελέγχου είναι στατική. Για παράδειγμα, ο βαθμός πολυπρογραμματισμού ενός συστήματος διαχείρισης βάσεων δεδομένων καθορίζεται στατικά, κατά την εκκίνηση του συστήματος, και δεν γίνεται διάκριση μεταξύ κλάσεων φόρτου. Αντίστοιχα, ο καθορισμός του μεγέθους των περιοχών ενταμιευτών γίνεται στατικά.

3.3 Διατύπωση Στόχων Επίδοσης

Η διαχείριση συστημάτων από απόψεως επίδοσης οφείλει να παρέχει στους υπεύθυνους για το σύστημα μηχανισμούς που να επιτρέπουν να καθοριστούν προδιαγραφές για την επίδοση κάθε κλάσης μονάδων φόρτου που το σύστημα εξυπηρετεί, *χωρίς* να χρειάζεται να καθοριστεί επακριβώς ο τρόπος με τον οποίο θα επιτευχθεί η επιθυμητή επίδοση κατά την λειτουργία του συστήματος. Εισάγεται κατ'αυτόν τον τρόπο η έννοια του στόχου επίδοσης (performance goal).

Μπορεί κανείς να καθορίσει ποικιλία στόχων επίδοσης [NFC92]:

- Στόχοι προθεσμίας (deadline): ορίζεται μια προθεσμία για την περάτωση της εξυπηρέτησης κάθε μονάδας φόρτου.
- Στόχοι σχετικοί με τον μέσο χρόνο απόκρισης (average response time): ορίζεται ένα επιθυμητό άνω όριο για τον μέσο χρόνο απόκρισης για μια κλάση μονάδων φόρτου.
- Στόχοι σχετικοί με την κατανομή των χρόνων απόκρισης (percentile response time): ορίζεται ότι ο χρόνος απόκρισης μονάδων φόρτου μιας κλάσης πρέπει να είναι μικρότερος από κάποιο όριο, τουλάχιστον για ένα ποσοστό $\alpha\%$ των μονάδων φόρτου αυτής της κλάσης.
- Στόχοι σχετικοί με τον ρυθμό εξυπηρέτησης (service rate): ορίζεται ότι σε μονάδες φόρτου μιας κλάσης πρέπει να δίδεται ένα συγκεκριμένο ποσοστό των πόρων (όπως οι διαθέσιμοι κύκλοι CPU). Στόχοι αυτού του τύπου είναι βολικό να ορίζονται για κλάσεις μονάδων φόρτου για τις οποίες είναι δύσκολο να προσδιοριστεί κάποιος "τυπικός" χρόνος απόκρισης ή "μέσες" απαιτήσεις σε πόρους. Σε τέτοιες περιπτώσεις (όπως για παράδειγμα κάποιες σύνθετες επερωτήσεις σε βάσεις δεδομένων για την στήριξη αποφάσεων) είναι επιθυμητό να μπορεί το σύστημα να εγγυηθεί ότι προχωρούν με "ικανοποιητικό" ρυθμό προς την περάτωσή τους.

Οι στόχοι επίδοσης μπορεί να διαφέρουν σημαντικά μεταξύ των κλάσεων φόρτου. Η διαφοροποίηση αυτή μπορεί να αντανακλά τις απαιτήσεις σε πόρους για την κάθε κλάση, ή την σημασία που δίδεται σ'αυτή από τους τελικούς χρήστες.

Για την ικανοποίηση στόχων επίδοσης για κάθε κλάση φόρτου το σύστημα μπορεί να κάνει χρήση μηχανισμών διαχείρισης πόρων. Παραδείγματα τέτοιων μηχανισμών για ένα σύστημα διαχείρισης βάσεων δεδομένων είναι ο έλεγχος φόρτου (load control) για να περιοριστεί ο ανταγωνισμός για μνήμη και προσπέλαση σε κοινόχρηστα δεδομένα, η δρομολόγηση δοσοληψιών (transaction routing) για την ισοκατανομή του φόρτου, η διαχείριση μνήμης και ενταμιευτών, ο χρονοπρογραμματισμός του επεξεργαστή και των δίσκων, η χωροθέτηση δεδομένων (data placement), καθώς και η βελτιστοποίηση επερωτήσεων (query processing), και η ανάθεση επεξεργαστών για την εκτέλεση επερωτήσεων (processor allocation). Καθένας από αυτούς τους μηχανισμούς ελέγχου μπορεί να λαμβάνει υπ'όψιν τους στόχους επίδοσης. Στην ανοικτή βιβλιογραφία έχουν μέχρι στιγμής προταθεί τέτοιοι μηχανισμοί για την δρομολόγηση δοσοληψιών [FNGD93], την διαχείριση ενταμιευτών [CFW+95, BCL93b], και τον έλεγχο φόρτου [BMCL94].

Το σύστημα έχει την ευθύνη να ικανοποιήσει τους (εν γένει συγκρουόμενους λόγω του ανταγωνισμού για κοινόχρηστους πόρους) στόχους επίδοσης πολλαπλών κλάσεων

μονάδων φόρτου. Είναι σημαντικό να τονιστεί ότι τα χαρακτηριστικά του φόρτου εξυπηρέτησης ενός συστήματος (όπως ο ρυθμός άφιξης μονάδων φόρτου και οι απαιτήσεις σε πόρους) δεν μένουν σταθερά κατά την λειτουργία του συστήματος, συνεπώς υπάρχει πάντα το ενδεχόμενο το σύστημα να βρεθεί σε κατάσταση υπερφόρτωσης για ένα χρονικό διάστημα. Στην κατάσταση αυτή, οι στόχοι επίδοσης είναι πιθανό να μην μπορούν να ικανοποιηθούν. Στην περίπτωση αυτή, έμφαση πρέπει να δοθεί στην εξυπηρέτηση κλάσεων μονάδων φόρτου οι οποίες έχουν οριστεί ως *κρίσιμες* για το σύστημα.

Στην εργασία αυτή η προσοχή εστιάζεται σε στόχους σχετικούς με τον μέσο χρόνο απόκρισης για κλάσεις μονάδων φόρτου.

3.4 Η Εννοια του Δείκτη Επίδοσης

Για την δυναμική διαχείριση πόρων, ώστε η πολιτική ανάθεσης να προσαρμόζεται στην τρέχουσα κατάσταση του συστήματος, είναι απαραίτητο να υπάρχει ένας μηχανισμός που να παρακολουθεί τον βαθμό ικανοποίησης των στόχων επίδοσης για τις κλάσεις μονάδων φόρτου που εξυπηρετεί το σύστημα. Βάση αυτού του μηχανισμού αποτελεί μια μορφή σημείου αναφοράς για να προσδιοριστεί η κατάσταση κάθε κλάσης μονάδων φόρτου ως προς την ικανοποίηση του στόχου επίδοσης. Το σημείο αναφοράς παρέχεται από μια συνάρτηση του μετρούμενου (ή εκτιμούμενου) χρόνου απόκρισης για την κλάση και του στόχου επίδοσης, που ονομάζεται "δείκτης επίδοσης" (performance index).

Εχοντας εστιάσει την προσοχή σε στόχους επίδοσης που αφορούν τον μέσο χρόνο απόκρισης για κλάσεις μονάδων φόρτου, ο δείκτης επίδοσης P_i μιας κλάσης i για την οποία έχει καθοριστεί στόχος επίδοσης g_i, ορίζεται [NFC92, FNGD93] ως εξής:

$$P_i \;=\; \frac{RT_i}{g_i}, \tag{3.1}$$

όπου RT_i είναι μια εκτίμηση του μέσου χρόνου απόκρισης των μονάδων φόρτου της κλάσης i. Ακολουθώντας την αναφορά [FNGD93], χρησιμοποιείται μια εκτίμηση του μέσου χρόνου απόκρισης που λαμβάνει υπ'όψιν την πρόσφατη ιστορία του συστήματος, αντί για τον άμεσα μετρούμενο χρόνο απόκρισης, ώστε να αποσβεστεί η επίδραση παροδικών μεταβολών στην εν γένει ευσταθή κατάσταση (steady−state) του συστήματος, για να επιτευχθεί μια πιο αξιόπιστη εκτίμηση για το αν ικανοποιείται ο στόχος επίδοσης μέσα στο χρονικό διάστημα που εξετάζεται.

Η ερμηνεία του δείκτη επίδοσης όπως ορίστηκε στην εξίσωση (3.1), είναι ότι, όταν ικανοποιείται ο στόχος επίδοσης για μια κλάση, ο αντίστοιχος δείκτης έχει τιμή το πολύ ίση με την μονάδα. Με βάση την ερμηνεία αυτή, το πρόβλημα της διαχείρισης πόρων εκφράζεται ως το πρόβλημα του προσδιορισμού μιας πολιτικής π με την ιδιότητα $P_i(\pi) \leq 1$, για κάθε κλάση i. Με βάση την διατύπωση αυτή, το πρόβλημα της διαχείρισης πόρων επιλύεται με την εύρεση λύσης για το πρόβλημα $\min \max P_i(\pi)$. Με τον τρόπο αυτό, το πρόβλημα της εύρεσης πολιτικής ανάθεσης πόρων εκφράζεται ως πρόβλημα βελτιστοποίησης στο οποίο η αντικειμενική συνάρτηση είναι ο μέγιστος, μεταξύ όλων των κλάσεων φόρτου, δείκτης επίδοσης. Ας σημειωθεί ότι υπάρχει το ενδεχόμενο να μην υπάρχει λύση που να δίνει $\max P_i \leq 1$, δηλαδή πολιτική που να ικανοποιεί τους στόχους επίδοσης όλων των κλάσεων.

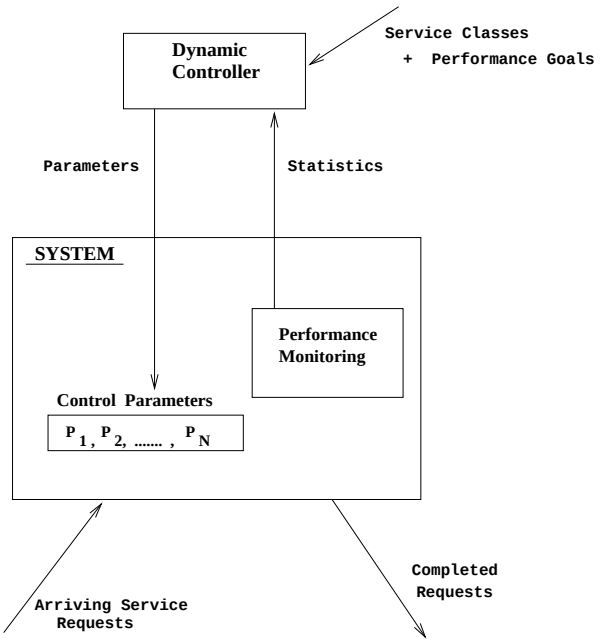
Ας σημειωθεί ότι εναλλακτικοί ορισμοί του δείκτη επίδοσης, αλλά και της αντικειμενικής συνάρτησης στο πρόβλημα βελτιστοποίησης, επιτρέπουν την διατύπωση

εναλλακτικών τύπων στόχων επίδοσης. Για παράδειγμα, η ανάθεση πόρων μπορεί να βασιστεί στην λύση ενός προβλήματος της μορφής $\min\max\{W_i \cdot P_i\}$, όπου W_i είναι συντελεστές βάρους για τις κλάσεις φόρτου, ώστε να γίνεται διάκρισή τους βάσει της σημασίας που έχουν για τους τελικούς χρήστες. Παρόμοια, ορίζοντας τον δείκτη επίδοσης για μια κλάση φόρτου i ως την πιθανότητα ο χρόνος απόκρισης για μονάδες φόρτου αυτής της κλάσης να είναι μεγαλύτερος από το όριο g_i που έχει οριστεί ως στόχος για την κλάση, η λύση του προβλήματος $\min\max P_i$ δίδει μια πολιτική για την ικανοποίηση στόχων σχετικών με την κατανομή του χρόνου απόκρισης για μονάδες φόρτου από τις διάφορες κλάσεις. Η δυσκολία στην επίλυση του προβλήματος έγγειται σε μεγάλο βαθμό στην εκτίμηση της επίδρασης που θα έχει μια μεταβολή της ανάθεσης πόρων ανά κλάση φόρτου στην ικανοποίηση των στόχων επίδοσης.

3.5 Δυναμική Διαχείριση Πόρων

Η έννοια του δείκτη επίδοσης είναι θεμελειώδης για την υλοποίηση του ελέγχου επίδοσης ενός συστήματος που να λαμβάνει υπ'όψιν προδιαγραφές επίδοσης για τις διάφορες κλάσεις φόρτου. Στο σχήμα 3.1 παρουσιάζεται η λογική οργάνωση ενός συστήματος που παρακολουθεί δυναμικά την επίδοσή του στην εξυπηρέτηση κλάσεων μονάδων φόρτου, και λαμβάνει δυναμικά μέτρα για να αντιμετωπίσει αποκλίσεις από τους στόχους επίδοσης που έχουν προδιαγραφεί για τις κλάσεις αυτές. Για να ανιχνευθούν τέτοιες αποκλίσεις, το σύστημα χρειάζεται να παρακολουθεί τον χρόνο εξυπηρέτησης κάθε μονάδας φόρτου που εξυπηρετεί ώστε να γνωρίζει, για κάθε κλάση μονάδων φόρτου, την τιμή του δείκτη επίδοσης. Η τιμή αυτή λαμβάνεται υπ'όψιν όταν το σύστημα ελέγχει εάν χρειάζεται αναπροσαρμογή της πολιτικής ανάθεσης πόρων. Ο έλεγχος αυτός πραγματοποιείται σε καθορισμένες χρονικές στιγμές (decision instants), όπως για παράδειγμα σε κάθε άφιξη μονάδας φόρτου για εξυπηρέτηση ή περιοδικά, όπου η περίοδος ορίζεται είτε σαν απόλυτη χρονική περίοδος είτε ως το διάστημα για την περάτωση της εξυπηρέτησης ενός προκαθορισμένου αριθμού μονάδων φόρτου (συνολικά για μονάδες φόρτου όλων των κλάσεων ή ξεχωριστά για κάθε κλάση). Λαμβάνοντας υπ'όψιν τους δείκτες επίδοσης, το σύστημα προσαρμόζει τις τιμές παραμέτρων ελέγχου, με σκοπό να βελτιώσει την επίδοση κλάσεων φόρτου για τις οποίες δεν ικανοποιείται ο στόχος επίδοσης.

Στο σχήμα 3.2 παρουσιάζεται ένα ενδεικτικό διάγραμμα ροής για το σύστημα ελέγχου επίδοσης του σχήματος 3.1. Οι περισσότερες τεχνικές ικανοποίησης στόχων επίδοσης που έχουν προταθεί στην βιβλιογραφία ακολουθούν αυτό το οργανωτικό σχήμα, το οποίο ουσιαστικά είναι ένα σχήμα ανάδρασης (feedback). Το σύστημα παρακολουθεί την επίδοσή του για κάθε κλάση φόρτου, και προβαίνει σε διορθωτικές για την επίδοση ενέργειες, που υλοποιούνται με ρυθμίσεις παραμέτρων ελέγχου, όταν βρει ότι αποκλίνει από τους στόχους επίδοσης. Εν γένει το σύστημα ελέγχου επίδοσης ενεργοποιείται σε καθορισμένες χρονικές στιγμές (όπως ορίστηκαν παραπάνω), και, με βάση δεδομένα που έχουν συλλεχθεί για την κατανάλωση πόρων, καθώς και την ικανοποίηση μέχρι στιγμής των στόχων επίδοσης (ουσιαστικά τους δείκτες επίδοσης P_i), επιχειρεί να λύσει ένα πρόβλημα βελτιστοποίησης τύπου $\min\max P_i$, για να καθορίσει τις τιμές των παραμέτρων ελέγχου που θα ισχύουν μέχρι την επόμενη ενεργοποίησή του.



Σχήμα 3.1: Δυναμικός Ελεγχος Συστήματος για την Ικανοποίηση Στόχων Επίδοσης.

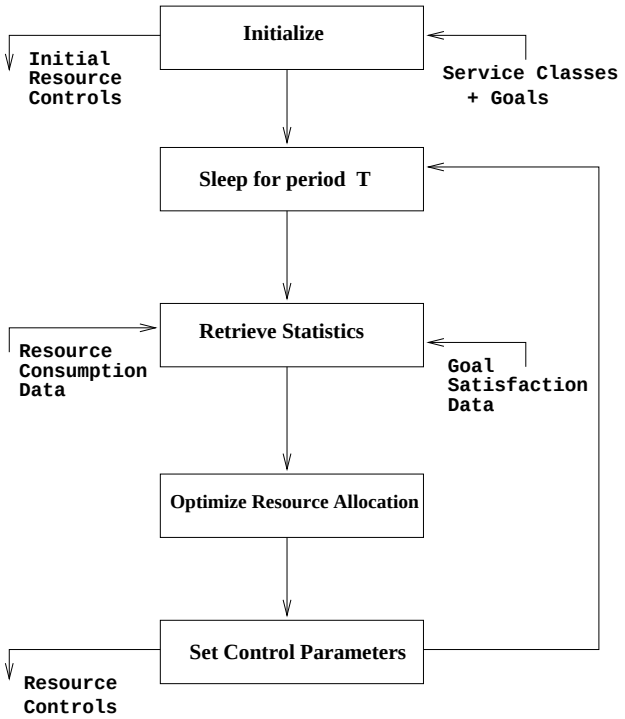
Ο υπεύθυνος για την διαχείριση του συστήματος παρέχει στο σύστημα ελέγχου επίδοσης μια περιγραφή των απαιτήσεων πόρων για κάθε κλάση μονάδων φόρτου που ενδιαφέρει τους χρήστες, καθώς και τις προδιαγραφές επίδοσης του συστήματος, διατυπωμένες ως στόχους επίδοσης για τις κλάσεις μονάδων φόρτου που εξυπηρετεί το σύστημα. Ο έλεγχος επίδοσης υλοποιείται με την μεταβολή παραμέτρων ελέγχου που επηρεάζουν την ανάθεση των πόρων σε μονάδες φόρτου. Το σύστημα παρακολουθεί τα αποτελέσματα των μεταβολών αυτών στην εξυπηρέτηση μονάδων φόρτου, ώστε το σύστημα ελέγχου επίδοσης να μπορεί να αναπροσαρμόσει τις παραμέτρους ελέγχου όταν κάποια κλάση μονάδων φόρτου αποκλίνει από τον στόχο επίδοσης που έχει διατυπωθεί γι'αυτήν.

3.6 Επισκόπηση της Σχετικής Βιβλιογραφίας

Στην βιβλιογραφία έχουν προταθεί κάποιες τεχνικές για την ικανοποίηση στόχων επίδοσης. Οι τεχνικές αυτές, εκτός από ελάχιστες εξαιρέσεις, επιχειρούν να ελέγξουν την επίδοση του συστήματος εφαρμόζοντας μια πολιτική διαχείρισης για ένα μόνο πόρο του συστήματος.

3.6.1 Διαχείριση Πόρων στο Λειτουργικό Σύστημα MVS/ESA

Το λειτουργικό σύστημα MVS/ESA για την σειρά υπολογιστών $S/390$ της IBM περιλαμβάνει ένα υποσύστημα για την διαχείριση κλάσεων φόρτου από απόψεως επίδοσης [IBM94, IBM93], με στόχο την ικανοποίηση προδιαγραφών επίδοσης. Το υποσύστημα αυτό, βάσει ενός μηχανισμού ανάδρασης που παρακολουθεί τον χρόνο απόκρισης ανά κλάση μονάδων φόρτου, τροποποιεί τον βαθμό πολυπρογραμματισμού και το μέγεθος της μνήμης που διατίθεται για την κλάση. Επίσης, η ανάθεση των διαθέσιμων κύκλων του επεξεργαστή (ακριβέστερα των CPU quantums) λαμβάνει υπ'όψιν την ικανοποίηση στόχων επίδοσης. Είναι εξαιρετικά ενδιαφέρον ότι το σύστημα ελέγχου επίδοσης μπορεί να ρυθμίσει παραμέτρους πολλαπλών πόρων, και μάλιστα σε ένα σύστημα που απευθύνεται σε χρήστες με υψηλές απαιτήσεις επίδο–



Σχήμα 3.2: Ενδεικτικό Διάγραμμα Ροής Διαχειριστή Πόρων Προσανατολισμένου στην Ικανοποίηση Στόχων Επίδοσης.

Το σύστημα ελέγχου επίδοσης έχει γνώση των αναμενόμενων απαιτήσεων πόρων των διαφόρων κλάσεων φόρτου, καθώς και των αντίστοιχων στόχων επίδοσης. Το σύστημα ελέγχου ενεργοποιείται περιοδικά για να αξιολογήσει κατά πόσο η πολιτική ανάθεσης πόρων που εφαρμόστηκε από την τελευταία του ενεργοποίηση φαίνεται να ικανοποιεί τους στόχους επίδοσης. Για την αξιολόγηση αυτή χρησιμοποιεί δεδομένα που συλλέγει το ίδιο το σύστημα για τις πραγματικές απαιτήσεις πόρων των μονάδων φόρτου που εξυπηρετήθηκαν στο διάστημα αυτό, καθώς και δεδομένα για την ικανοποίηση ή όχι των στόχων επίδοσης. Εάν υπάρχει απόκλιση από τους στόχους, τροποποιούνται παράμετροι που επηρεάζουν την πολιτική ανάθεσης πόρων.

σης και διαθεσιμότητας πόρων. Δυστυχώς, επειδή αποτελεί μέρος ενός εμπορικού συστήματος, δεν υπάρχουν στην ανοικτή βιβλιογραφία λεπτομερείς αναφορές για την σχεδίαση και λειτουργία του.

3.6.2 Δρομολόγηση Δοσοληψιών

Η δρομολόγηση δοσοληψιών σε ένα κατανεμημένο/παράλληλο σύστημα επεξεργασίας δοσοληψιών έχει σκοπό την ισοκατανομή του φόρτου εξυπηρέτησης στο σύστημα, λαμβάνοντας όμως υπ'όψιν και την σημαντική τοπικότητα αναφορών που χαρακτηρίζει τις δοσοληψίες και που έχει ως συνέπεια οι κλάσεις δοσοληψιών να εμφανίζουν "συγγένεια" (affinity) προς συγκεκριμένα δεδομένα.

Εχουν προταθεί [FNGD92, FNGD93] αλγόριθμοι δρομολόγησης δοσοληψιών που λαμβάνουν υπ'όψιν στόχους επίδοσης για κλάσεις δοσοληψιών. Οι αλγόριθμοι αυτοί εκτιμούν την επίδραση κάθε δυνατής επιλογής του κόμβου εκτέλεσης μιας δοσοληψίας στην ικανοποίηση των στόχων επίδοσης *όλων* των κλάσεων δοσοληψιών. Η εκτίμηση αυτή στηρίζεται σε ένα μοντέλο για τον υπολογισμό του χρόνου απόκρισης, δηλαδή

το άθροισμα του χρόνου εξυπηρέτησης (service time) και του χρόνου αναμονής για τον επεξεργαστή (CPU queueing delay). Για τον υπολογισμό χρησιμοποιούνται πληροφορίες για την μέση κατανάλωση πόρων ανά κλάση δοσοληψιών (μέσες απαιτήσεις για χρόνο του επεξεργαστή, δεδομένα από δίσκους, μηνύματα για επικοινωνία μεταξύ κόμβων), και πληροφορίες για την τρέχουσα κατάσταση του συστήματος, όπως ο αριθμός των δοσοληψιών από κάθε κλάση που είναι ενεργές σε κάθε κόμβο του συστήματος. Εκτιμάται, για κάθε δυνατή απόφαση δρομολόγησης, ο χρόνος απόκρισης της δοσοληψίας, καθώς και η μεταβολή του δείκτη επίδοσης για κάθε κλάση δοσοληψιών. Για την εκτίμηση αυτή το σύστημα χρειάζεται να παρακολουθεί τον μέσο χρόνο απόκρισης ανά κλάση δοσοληψιών. Τελικά επιλέγεται η απόφαση δρομολόγησης που δίδει την ελάχιστη τιμή για την εκτιμούμενη μέγιστη τιμή δείκτη επίδοσης μεταξύ όλων των κλάσεων. Συνεπώς, η δρομολόγηση μιας δοσοληψίας ανάγεται σε ένα πρόβλημα βελτιστοποίησης του τύπου $\min \max P_i$. Οι εκτιμούμενοι δείκτες επίδοσης, για μια από τις δυνατές αποφάσεις δρομολόγησης, σχηματίζουν ένα διάνυσμα, με τόσα στοιχεία όσες οι κλάσεις δοσοληψιών. Το ζητούμενο είναι τελικά η απόφαση δρομολόγησης που δίδει το μικρότερο κατά λεξικογραφική διάταξη διάνυσμα, ώστε να επιτευχθεί και μια κατά το δυνατόν εξισορρόπιση των τιμών των δεικτών επίδοσης μεταξύ των κλάσεων.

'Ενα μειονέκτημα των αλγορίθμων αυτών είναι ότι έχουν χρονική πολυπλοκότητα $O(K \cdot N^3)$, όπου K το πλήθος των κλάσεων δοσοληψιών και N το πλήθος των κόμβων, καθώς η τεχνική που χρησιμοποιούν για την εκτίμηση του χρόνου αναμονής για τον επεξεργαστή έχει πολυπλοκότητα $O(N^2)$.

3.6.3 Διαχείριση Ενταμιευτών

Η διαχείριση ενταμιευτών σε ένα σύστημα διαχείρισης βάσεων δεδομένων έχει σκοπό να περιορίσει την καθυστέρηση προσπέλασης σε μονάδες αποθήκευσης, τον ανταγωνισμό για τις μονάδες αποθήκευσης (λόγω του περιορισμού των πράξεων προσπέλασης), και την καθυστέρηση της επεξεργασίας πράξεων προσπέλασης. Η ανάθεση ενταμιευτών σε δοσοληψίες αποτελεί συνεπώς μια πολύ σημαντική τεχνική για τον έλεγχο του χρόνου απόκρισης. Ας σημειωθεί όμως ότι η τεχνική αυτή δεν μπορεί να είναι αρκετά αποτελεσματική για την ικανοποίηση των στόχων επίδοσης μονάδων φόρτου για τις οποίες η πιθανότητα η σελίδα που χρειάζεται μια προσπέλαση να βρεθεί σε ενταμιευτή (hit ratio) είναι πολύ χαμηλή. Παράδειγμα τέτοιων μονάδων φόρτου αποτελεί μια επερώτηση για την εκτέλεση της οποίας απαιτείται σειριακή σάρωση ενός τμήματος της βάσης δεδομένων.

Στην αναφορά [CFW+95] παρουσιάζεται ένας αλγόριθμος για την ικανοποίηση στόχων επίδοσης που ορίζονται για τον μέσο χρόνο προσπέλασης σε δεδομένα ανά περιοχή ενταμιευτών (buffer pool). Ο ορισμός αυτός είναι διαφορετικός από τον ορισμό στόχων επίδοσης για τον μέσο χρόνο απόκρισης ανά κλάση δοσοληψιών. Ο δείκτης επίδοσης ορίζεται ανά περιοχή ενταμιευτών, ως ο λόγος του μέσου χρόνου προσπέλασης στην περιοχή προς τον αντίστοιχο στόχο. Ο αλγόριθμος αυτός ενεργοποιείται περιοδικά και ρυθμίζει το μέγεθος της κάθε περιοχής ενταμιευτών, διατηρώντας σταθερό τον συνολικό αριθμό ενταμιευτών. Για τον σκοπό αυτό, το σύστημα παρακολουθεί κατά πόσο ικανοποιούνται οι στόχοι επίδοσης για τις περιοχές ενταμιευτών. Βάσει μιας εκτίμησης της πιθανότητας η σελίδα που χρειάζεται μια προσπέλαση να βρεθεί σε ενταμιευτή, συναρτήσει του μεγέθους της περιοχής ενταμιευτών, υπολογίζεται ο μέσος χρόνος για μια προσπέλαση στην περιοχή αυτή.

Κατόπιν, προσδιορίζεται η επίδραση που έχει μια μεταβολή στο μέγεθος μιας περιοχής ενταμιευτών. Ο αλγόριθμος επιχειρεί να αυξήσει το μέγεθος της περιοχής με τον χειρότερο δείκτη επίδοσης, μειώνοντας αντίστοιχα το μέγεθος περιοχών με καλύτερους δείκτες επίδοσης. Ο αλγόριθμος έχει στόχο να εντοπίσει το ελάχιστο κατά λεξικογραφική διάταξη διάνυσμα δεικτών επίδοσης για τις διάφορες περιοχές ενταμιευτών, όπου οι δείκτες επίδοσης υπολογίζονται βάσει της προαναφερθείσας προσέγγισης της σχέσης ανάμεσα στο μέγεθος μιας περιοχής ενταμιευτών με τον μέσο χρόνο προσπέλασης στην περιοχή.

Στην αναφορά [BCL93b] παρουσιάζεται ο δυναμικός αλγόριθμος fragment fencing για την διαχείριση ενταμιευτών δεδομένων λαμβάνοντας υπ'όψιν στόχους επίδοσης για πολλαπλές κλάσεις φόρτου σε ένα σύστημα διαχείρισης βάσεων δεδομένων. Η βάση δεδομένων θεωρείται ότι αποτελείται από τμήματα (fragments) με κατά προσέγγιση ομοιόμορφη πιθανότητα προσπέλασης. Ο αλγόριθμος αυτός (fragment fencing) επιχειρεί να ελέγξει την πιθανότητα η σελίδα που χρειάζεται μια προσπέλαση να βρεθεί σε ενταμιευτή, ορίζοντας, βάσει των στόχων επίδοσης που έχουν οριστεί για τις κλάσεις φόρτου, κάτω φράγματα για το πλήθος των σελίδων από κάθε τμήμα της βάσης δεδομένων που βρίσκονται σε ενταμιευτές. Μια βασική υπόθεση του αλγορίθμου είναι ότι ο χρόνος απόκρισης μιας δοσοληψίας είναι ευθέως ανάλογος του προσπελάσεων περιφερειακής μνήμης που πραγματοποιούνται για την εκτέλεσή της (I/O dominance assumption). Το όριο για κάθε τμήμα (που ονομάζεται target residency) επιβάλλεται από το σύστημα τροποποιώντας την πολιτική αντικατάστασης σελίδων που εφαρμόζεται για τους ενταμιευτές, ώστε να αποφεύγεται η αντικατάσταση μιας σελίδας όταν αυτή κάνει τον αριθμό των σελίδων από το τμήμα που βρίσκονται σε ενταμιευτές μικρότερο από το όριο. Το σύστημα παρακολουθεί, για κάθε κλάση φόρτου, τον χρόνο απόκρισης και την συχνότητα προσπέλασης σε κάθε τμήμα, καθώς και τον αριθμό των προσπελάσεων που ικανοποιούνται με δεδομένα από κάποιο ενταμιευτή, για κάθε τμήμα. Βάσει της προαναφερθείσας υπόθεσης για τον χρόνο απόκρισης, ο αλγόριθμος υπολογίζει, για κάθε κλάση φόρτου, την επιθυμητή μεταβολή στην πιθανότητα να βρεθεί σε ενταμιευτή η σελίδα που χρειάζεται μια προσπέλαση. Με βάση την υπόθεση ότι η πιθανότητα προσπέλασης σε κάθε τμήμα είναι κατά προσέγγιση ομοιόμορφη, προσδιορίζει για κάθε τμήμα το κάτω όριο για τον αριθμό των σελίδων που κρατούνται σε ενταμιευτές. Ο αλγόριθμος ενεργοποιείται περιοδικά για να ελέγξει την ικανοποίηση των στόχων επίδοσης για κάθε κλάση. Η περίοδος ορίζεται ξεχωριστά για κάθε κλάση.

Στην αναφορά [BMCL94] παρουσιάζεται, ως συνέχεια της εργασίας [BCL93b], ένας αλγόριθμος που επιχειρεί να ικανοποιήσει τους στόχους επίδοσης ελέγχοντας ταυτόχρονα την ανάθεση ενταμιευτών και τον βαθμό πολυπρογραμματισμού σε ένα σύστημα διαχείρισης βάσεων δεδομένων, ξεχωριστά για κάθε κλάση μονάδων φόρτου. Γίνεται διάκριση των κλάσεων φόρτου σε δύο κατηγορίες, ανάλογα με τον τρόπο με τον οποίο χρησιμοποιούν την μνήμη, και προτείνεται ένας αλγόριθμος που ακολουθεί το μοντέλο ανάδρασης που παρουσιάστηκε στα σχήματα 3.1 και 3.2. Η πρώτη κατηγορία (disk-buffer classes) περιλαμβάνει τις κλάσεις για τις οποίες ο χρόνος απόκρισης εξαρτάται άμεσα από το μέγεθος της περιοχής ενταμιευτών (όπως οι συνήθεις δοσοληψίες), ενώ η δεύτερη κατηγορία (working storage classes) περιλαμβάνει τις κλάσεις που χρειάζονται μνήμη κυρίως για επεξεργασία δεδομένων (όπως επερωτήσεις που περιλαμβάνουν πράξεις σύνδεσης – join), ώστε να αποφύγουν προσπελάσεις περιφερειακής μνήμης. Είναι σημαντικό να τονιστεί ότι ο προτεινόμενος αλγόριθμος αντιμετωπίζει κάθε κλάση φόρτου ξεχωριστά, αγνοώντας εντελώς τις αλληλεπιδράσεις μεταξύ κλάσεων,

που πηγάζουν από το γεγονός ότι ο χρόνος απόκρισης μιας κλάσης δεν εξαρτάται μόνο από τον βαθμό πολυπρογραμματισμού και την μνήμη που έχει διατεθεί για την κλάση, αλλά επηρεάζεται και από τον βαθμό του ανταγωνισμού για προσπέλαση σε κοινόχρηστους πόρους, όπως ο επεξεργαστής, και οι δίσκοι δεδομένων. Η επιλογή αυτή απλοποιεί σημαντικά την σχεδίαση και υλοποίηση του αλγορίθμου.

3.6.4 Στόχοι Επίδοσης για Σύνθετες Μονάδες Φόρτου

Είναι σαφές ότι η έννοια του στόχου επίδοσης καλύπτει και περιπτώσεις σύνθετων μονάδων φόρτου που εκτελούνται ως σειρά από δοσοληψίες, όπως περιγράφεται στο κεφάλαιο 7. Η ικανοποίηση όμως στόχων επίδοσης δυσχεραίνεται από τις εξαρτήσεις (ελέγχου και δεδομένων) μεταξύ των βημάτων που συγκροτούν μια σύνθετη μονάδα φόρτου. Στην αναφορά [MN95] παρουσιάζεται μια πρώτη προσπάθεια προς την κατεύθυνση της ικανοποίησης στόχων επίδοσης για σύνθετες μονάδες φόρτου σε συστήματα επεξεργασίας δοσοληψιών, όπου κάθε βήμα της σύνθετης μονάδας φόρτου εκτελείται ως μια δοσοληψία, με τον δυναμικό έλεγχο της προτεραιότητας (dispatching priority) που δίδεται σε κάθε βήμα. Το κεφάλαιο 7 αυτής της εργασίας μπορεί να θεωρηθεί συνέχεια της δουλειάς στην αναφορά [MN95], καθώς περιλαμβάνει λεπτομερή πειραματική μελέτη ενός ευρέος φάσματος τεχνικών χρονοπρογραμματισμού για σύνθετες μονάδες φόρτου, με κριτήριο αξιολόγησης την ικανοποίηση στόχων επίδοσης για τις κλάσεις φόρτου.

Κεφάλαιο 4

Ο Προσομοιωτής TPsim: Σχεδίαση

Στο κεφάλαιο αυτό παρουσιάζεται αναλυτικά η σχεδίαση του προσομοιωτή κατανεμημένων συστημάτων επεξεργασίας δοσοληψιών TPsim. Ο προσομοιωτής περιλαμβάνει μοντέλα για τους υπολογιστικούς κόμβους, τις συσκευές αποθήκευσης δεδομένων, και το δίκτυο επικοινωνίας, καθώς και για τα βασικά υποσυστήματα λογισμικού που απαρτίζουν ένα σύστημα επεξεργασίας δοσοληψιών, συμπεριλαμβανομένων και των διαχειριστών πόρων του συστήματος. Κεντρικό ρόλο στην αρχιτεκτονική του προσομοιωτή κατέχει ένας μεταφραστής για μια γλώσσα προδιαγραφής, με την οποία είναι δυνατή η περιγραφή της διαμόρφωσης του συστήματος προς προσομοίωση και του προσφερόμενου φόρτου εξυπηρέτησης για το σύστημα. Ο μεταφραστής αυτός συνθέτει τελικά το μοντέλο του προσομοιούμενου συστήματος κάνοντας χρήση των μηχανισμών που παρέχει μια βιβλιοθήκη για την υποστήριξη προσομοίωσης. Ο προσομοιωτής εξομοιώνει συμβάντα που ανακύπτουν κατά την λειτουργία του υπό μελέτη σύστηματος, και παράγει μια εκτίμηση για τις τιμές διαφόρων μεταβλητών του μοντέλου προσομοίωσης συστήματος και μιας σειράς από μέτρα επίδοσης. Ο προσομοιωτής εντάσσεται σε ένα ολοκληρωμένο περιβάλλον υποστήριξης πειραμάτων που αναπτύχθηκε στα πλαίσια αυτής της εργασίας, για την αυτοματοποίηση της διαδικασίας διεξαγωγής πειραμάτων προσομοίωσης και συλλογής μετρήσεων.

4.1 Αρχές Σχεδίασης

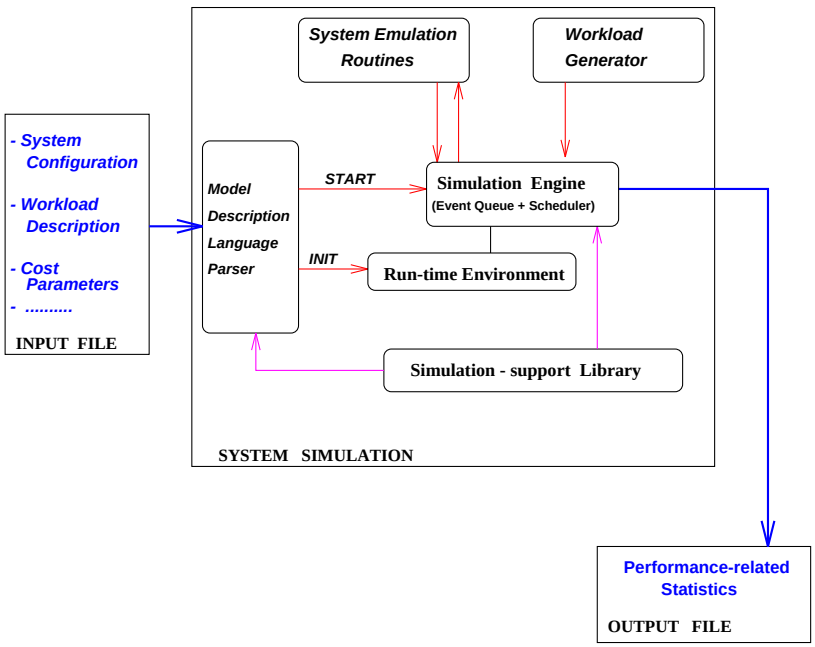
Η προσομοίωση ενός συστήματος στηρίζεται σε ένα μοντέλο του συστήματος που περιγράφει τα κύρια δομικά στοιχεία, τις μεταξύ τους εξαρτήσεις και αλληλεπιδράσεις, και τις συνθήκες του περιβάλλοντος μέσα στο οποίο το σύστημα λειτουργεί. Η περιγραφή αυτή βασίζεται στον προσδιορισμό ενός αριθμού από μεταβλητές που χαρακτηρίζουν σημαντικές (για τους σκοπούς της συγκεκριμένης μελέτης) παραμέτρους των δομικών στοιχείων του συστήματος και των μεταξύ τους σχέσεων, καθώς και παράγοντες σχετικούς με τις συνθήκες λειτουργίας. Βάσει ενός τέτοιου μοντέλου, η προσομοίωση ενός συστήματος έχει στόχο να δώσει μια εικόνα της δυναμικής συμπεριφοράς του συστήματος υπό τις δοθείσες συνθήκες λειτουργίας, και να υπολογίσει μια σειρά από μέτρα επίδοσης. Η εκτίμηση ποικίλων μέτρων επίδοσης για ευρύ φάσμα συνθηκών λειτουργίας είναι σημαντική για την αξιολόγηση εναλλακτικών σχεδιαστικών επιλογών και τον έλεγχο της ορθότητας υποθέσεων σχετικών με την επίδραση διαφόρων παραγόντων στην δυναμική συμπεριφορά και απόδοση του συστήματος. Ο προσομοιωτής TPsim αναπτύχθηκε στα πλαίσια αυτής της εργασίας ειδικά για την

μελέτη συστημάτων επεξεργασίας δοσοληψιών.

Ενας βασικός στόχος στην σχεδίαση του προσομοιωτή είναι να επιτρέπει την μεταβολή τόσο των παραμέτρων του μοντέλου, όσο και της ίδιας της διαμόρφωσης (configuration) του υπό μελέτη συστήματος, χωρίς να απαιτείται επέμβαση στον κώδικα του προγράμματος προσομοίωσης και εκ νέου μεταγλώττισή του. Για την επίτευξη αυτού του στόχου, σχεδιάστηκε μια γλώσσα προδιαγραφής (specification language) για την περιγραφή του συστήματος και του φόρτου εξυπηρέτησης. Η γλώσσα αυτή παρέχει συντακτικές δομές για την περιγραφή της διαμόρφωσης των κόμβων επεξεργασίας (processing nodes) του συστήματος και την τοπολογία διασύνδεσης των κόμβων, τον καθορισμό παραμέτρων επίδοσης του δικτύου επικοινωνίας, την περιγραφή του σχήματος της προσπελαυνόμενης βάσης δεδομένων, τον ορισμό της ανάθεσης αρχείων δεδομένων που υλοποιούν την βάση δεδομένων σε συσκευές αποθήκευσης δεδομένων των κόμβων του συστήματος, και την προδιαγραφή του φόρτου εξυπηρέτησης, με τον καθορισμό απλών στατιστικών παραμέτρων.

Η περιγραφή του συστήματος και του φόρτου εξυπηρέτησης στην γλώσσα προδιαγραφής είναι αρκετά υψηλού επιπέδου ώστε να μπορεί να χρησιμεύσει και ως τεκμηρίωση του μοντέλου προσομοίωσης. Η χρήση μιας τυπικής γλώσσας αίρει πολλές ασάφειες και παραλείψεις που είναι αναπόφευκτες όταν το μοντέλο προσομοίωσης περιγράφεται στη φυσική γλώσσα. Τέτοιες ασάφειες και παραλείψεις καθιστούν δύσκολη την ανεξάρτητη αναπαραγωγή των πειραματικών αποτελεσμάτων μιας μελέτης προσομοίωσης. Συνεπώς, ο περιορισμός τους οδηγεί στην βελτίωση της ποιότητας παρουσίασης των αποτελεσμάτων.

Για την υλοποίηση του προσομοιωτή κρίθηκε σκόπιμο να υιοθετηθεί ένα μοντέλο εκτέλεσης που υποστηρίζει πολλαπλά "νήματα ελέγχου" (threads of control) μέσα σε ένα πεδίο διευθύνσεων (address space). Αυτό το μοντέλο παρέχει σημαντική ευελιξία στην προσομοίωση σύνθετων συστημάτων αποτελούμενων από πολλά τμήματα που αλληλεπιδρούν ασύγχρονα. Μια βιβλιοθήκη υποστήριξης παρέχει αυτό το περιβάλλον εκτέλεσης (execution/run−time environment), στο οποίο είναι δυνατή η δημιουργία ανεξάρτητων νημάτων ελέγχου, καθένα από τα οποία εκτελεί τον δικό του κώδικα και μπορεί να επικοινωνήσει με άλλα νήματα ελέγχου είτε με πέρασμα μηνυμάτων είτε μέσω της κοινόχρηστης μνήμης, καθώς όλα τα νήματα ελέγχου χρησιμοποιούν το ίδιο πεδίο διευθύνσεων. Αυτό το μοντέλο εκτέλεσης είναι ιδιαίτερα κατάλληλο για την προσομοίωση παράλληλων και κατανεμημένων συστημάτων σε μονοεπεξεργαστές (uniprocessors). Ο προσομοιωτής στηρίζεται σε ένα μεταφραστή για την γλώσσα προδιαγραφής του συστήματος και του φόρτου. Ο χρήστης αρκεί να δώσει μια περιγραφή του προς μελέτη συστήματος βάσει της γραμματικής της γλώσσας προδιαγραφής, ώστε ο μεταφραστής να συνθέσει το μοντέλο του συστήματος, υλοποιώντας εμφανίσεις (instances) ορισμένων αντικειμένων που αποτελούν μοντέλα για δομικά στοιχεία του προσομοιούμενου συστήματος, όπως οι κόμβοι επεξεργασίας. Με τον τρόπο αυτό, το περιβάλλον εκτέλεσης προσαρμόζεται στο μοντέλο του προσομοιούμενου συστήματος. Η προσομοίωση του συστήματος, που ουσιαστικά συνίσταται στην εκτέλεση, από πολλαπλά νήματα ελέγχου, του κώδικα που εξομοιώνει τις λειτουργίες των διαφόρων υποσυστημάτων που απαρτίζουν το υπό μελέτη σύστημα, και τις μεταξύ τους αλληλεπιδράσεις, συντονίζεται από ένα ειδικό νήμα ελέγχου που ελέγχει τον μηχανισμό της προσομοίωσης (simulation engine). Η βιβλιοθήκη υποστήριξης προσομοίωσης παρέχει επίσης ένα μηχανισμό για την συλλογή μετρήσεων για μεταβλητές που ορίζονται από τον χρήστη. Στο σχήμα 4.1 συνοψίζεται η αρχιτεκτονική του προσομοιωτή, ενώ στο κεφάλαιο 5 παρουσιάζεται



Σχήμα 4.1: Αρχιτεκτονική του Προσομοιωτή TPsim.

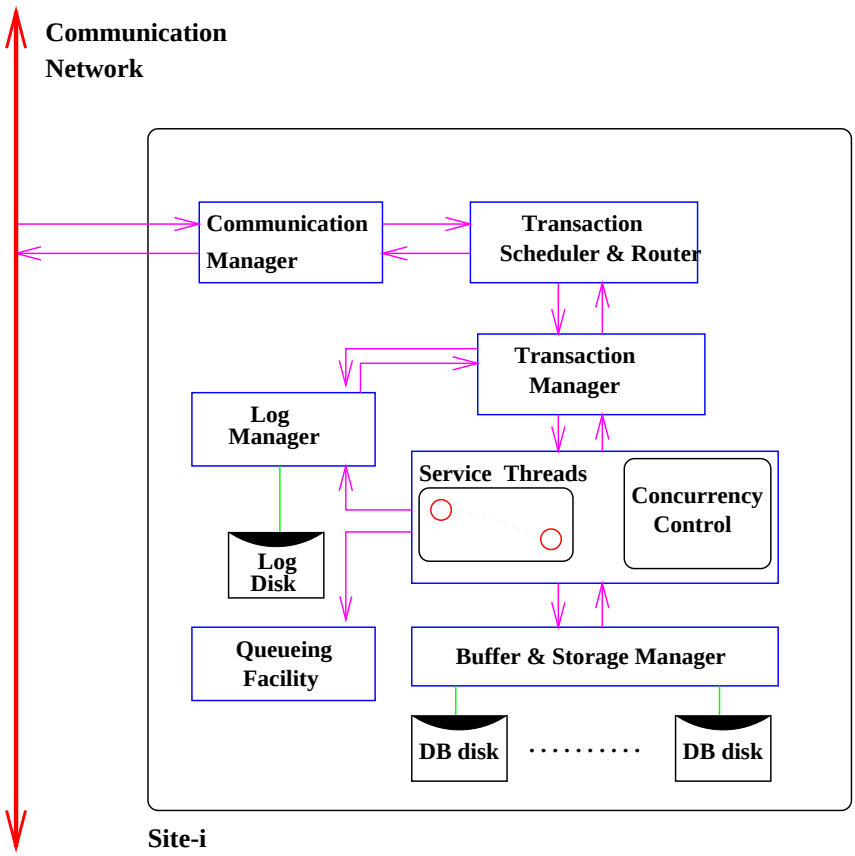
Ο χρήστης περιγράφει το σύστημα και τον φόρτο εξυπηρέτησης βάσει μιας γλώσσας προδιαγραφής. Ενας μεταφραστής για την γλώσσα αυτή προσαρμόζει το περιβάλλον εκτέλεσης που παρέχει μια βιβλιοθήκη υποστήριξης προσομοίωσης. Ο μηχανισμός προσομοίωσης συντονίζει πολλαπλά νήματα ελέγχου σε ένα πεδίο διευθύνσεων για την εκτέλεση κώδικα που εξομοιώνει τις λειτουργίες και αλληλεπιδράσεις των τμημάτων του συστήματος, και συλλέγει μετρήσεις των τιμών διαφόρων μεταβλητών (κυρίως μέτρων επίδοσης).

αναλυτικά η βιβλιοθήκη υποστήριξης προσομοίωσης και η υλοποίηση των βασικών υποσυστημάτων του προσομοιωτή.

4.2 Το Μοντέλο Συστήματος

Ο προσομοιωτής TPsim υλοποιεί μοντέλα για συστήματα επεξεργασίας δοσοληψιών που έχουν την δομή που παρουσιάστηκε στο κεφάλαιο 2. Ο προσομοιωτής περιλαμβάνει υποσυστήματα που μοντελοποιούν το υλικό (hardware) και το λογισμικό (software) ενός συστήματος επεξεργασίας δοσοληψιών. Το μοντέλο συστήματος περιλαμβάνει γενικά ένα επόπτη επεξεργασίας δοσοληψιών (transaction processing monitor) και ένα αριθμό από διαχειριστές πόρων (resource managers). Οι διαχειριστές πόρων που μοντελοποιούνται σε κάθε κόμβο του προσομοιούμενου συστήματος είναι ένα σύστημα διαχείρισης βάσης δεδομένων (DBMS − database management system), ένα υποσύστημα επικοινωνίας (communications manager), και ένα σύστημα διαχείρισης ουρών (queueing system).

Το μοντέλο συστήματος αναλύεται σε ένα σύνολο από υποσυστήματα. Το σχήμα 4.2 απεικονίζει το μοντέλο οργάνωσης ενός κόμβου επεξεργασίας. Στο κεφάλαιο 5 περιγράφεται αναλυτικά η υλοποίηση κάθε υποσυστήματος στον προσομοιωτή, βάσει της παρουσίασης των λειτουργικών χαρακτηριστικών τους στο κεφάλαιο 2.

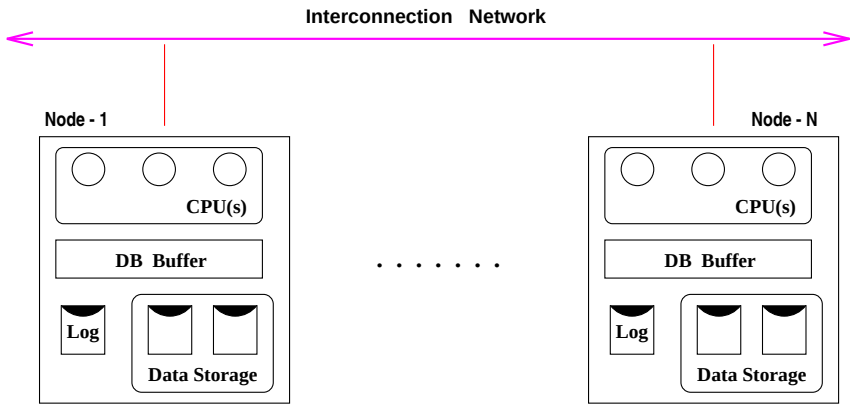


Σχήμα 4.2: Δομή του μοντέλου προσομοίωσης για ένα κόμβο σε σύστημα επεξεργασίας δοσοληψιών.

Το σχήμα δείχνει τα βασικά δομικά στοιχεία του μοντέλου συστήματος επεξεργασίας δοσοληψιών (κόμβοι επεξεργασίας, συσκευές αποθήκευσης για δεδομένα και καταγραφή αλλαγών, δίκτυο επικοινωνίας μεταξύ των κόμβων), καθώς και τα βασικά υποσυστήματα λογισμικού που συγκροτούν το μοντέλο ενός κόμβου επεξεργασίας.

Για κατανεμημένα συστήματα επεξεργασίας δοσοληψιών ακολουθείται η αρχιτεκτονική Shared−Nothing [DG92]. Οπως φαίνεται στο σχήμα 4.3, τα δεδομένα που προσπελάζονται από τις δοσοληψίες που υποβάλλονται από τα προγράμματα εφαρμογών είναι διαμερισμένα (partitioned) μεταξύ των κόμβων, βάσει ενός σχήματος ανάθεσης. Μια αίτηση προσπέλασης στην βάση δεδομένων από μια δοσοληψία που εκτελείται σε ένα κόμβο εξυπηρετείται τοπικά, εάν το ζητούμενο αντικείμενο δεδομένων έχει ανατεθεί στον κόμβο αυτό, διαφορετικά η αίτηση προσπέλασης μεταβιβάζεται για εξυπηρέτηση στον κόμβο που έχει το ζητούμενο αντικείμενο δεδομένων. Η μεταβίβαση αίτησης προσπέλασης (function request shipping) [CDY86, Kag89] είναι ο βασικός μηχανισμός για την κατανεμημένη εκτέλεση δοσοληψιών. Ο μηχανισμός αυτός επιτρέπει σε μια δοσοληψία που εκτελείται σε ένα κόμβο να προσπελάσει πόρους ενός άλλου κόμβου. Η προσπέλαση γίνεται στον κόμβο που διαχειρίζεται τον ζητούμενο πόρο με την εκκίνηση μιας δοσοληψίας στον κόμβο αυτό. Η δοσοληψία αυτή ονομάζεται "δευτερεύουσα" (secondary), σε αντιπαράθεση με την δοσοληψία που διατύπωσε την αίτηση προσπέλασης, που ονομάζεται "πρωτεύουσα" (primary). Μια δοσοληψία μπορεί να ξεκινήσει δευτερεύουσες δοσοληψίες σε πολλαπλούς κόμβους.

Οι δοσοληψίες αυτές παραμένουν ενεργές μέχρι τον τερματισμό της πρωτεύουσας δοσοληψίας. Για την εξασφάλιση της ιδιότητας της ατομικότητας είναι απαραίτητο να εκτελεστεί το πρωτόκολλο δέσμευσης δύο φάσεων (two−phase commit) [Gra78]. Στο πρωτόκολλο αυτό ο διαχειριστής δοσοληψιών (transaction manager) του κόμβου όπου ξεκίνησε η primary δοσοληψία λειτουργεί ως συντονιστής.



Σχήμα 4.3: Αρχιτεκτονική Shared−Nothing για κατανεμημένα συστήματα επεξεργασίας δοσοληψιών.

Κάθε κόμβος περιλαμβάνει ένα ή περισσότερους επεξεργαστές, και διαχειρίζεται μέρος των δεδομένων που αποτελούν την κατανεμημένη βάση δεδομένων. Το σύστημα υποστηρίζει τον μηχανισμό *function request shipping* για να μπορεί μια δοσοληψία που εκτελείται σε ένα κόμβο να προσπελάσει δεδομένα που αποθηκεύονται σε έναν άλλο.

4.3 Η Γλώσσα Προδιαγραφής

Η γλώσσα προδιαγραφής έχει ως στόχο να περιγράψει την διαμόρφωση του συστήματος, καθώς και τον φόρτο εξυπηρέτησης που θα προσομοιωθεί, ώστε το περιβάλλον εκτέλεσης που παρέχει ο προσομοιωτής να προσαρμοστεί στις προδιαγραφές της συγκεκριμένης μελέτης προσομοίωσης. Ο μεταφραστής αυτής της γλώσσας επεξεργάζεται αρχείο εισόδου που περιγράφει την διαμόρφωση του υπό μελέτη συστήματος, το σχήμα της κατανεμημένης βάσης δεδομένων, την ανάθεση τμημάτων της βάσης σε περιφερειακή μνήμη κόμβων του κατανεμημένου συστήματος, ποικιλία παραμέτρων κόστους για το σύστημα, και επιπλέον μια περιγραφή του προσφερόμενου στο σύστημα φόρτου, μέσω του καθορισμού των τύπων δοσοληψιών που μπορούν να εμφανιστούν και της διατύπωσης απλών στατιστικών προδιαγραφών για την τοπικότητα των αναφορών σε δεδομένα. Η πληροφορία που παρέχεται από τον χρήστη του προσομοιωτή μέσω αυτού του αρχείου χρησιμοποιείται για την υλοποίηση των δομικών στοιχείων του μοντέλου (όπως οι κόμβοι και το δίκτυο επικοινωνίας) με αντικείμενα που υποστηρίζονται από την βιβλιοθήκη υποστήριξης προσομοίωσης που ο χρήστης έχει στην διάθεσή του. Επίσης, τίθεται σε κίνηση ο μηχανισμός της προσομοίωσης, με το να δοθούν οι απαραίτητες παράμετροι στις διαδικασίες που υλοποιούν τους αλγόριθμους που χρησιμοποιούνται στο υπό μελέτη σύστημα, και να ενεργοποιηθεί ο μηχανισμός χειρισμού συμβάντων προσομοίωσης που υλοποιείται από την βιβλιοθήκη υποστήριξης. Τέλος, ο μεταφραστής παράγει μια δομή δεδομένων που παίζει κατά την

διάρκεια της προσομοίωσης τον ρόλο του καθολικού καταλόγου συστήματος (global directory/dictionary) για το υπό μελέτη σύστημα. Η δομή αυτή είναι ουσιαστικά ο πίνακας συμβόλων (symbol table) που χρησιμοποιείται από τον μεταφραστή για την επεξεργασία του αρχείου εισόδου.

Για την περιγραφή της διαμόρφωσης του συστήματος, η γλώσσα περιγραφής επιτρέπει τον ορισμό τύπων συσκευών περιφερειακής μνήμης, δικτύων επικοινωνίας, υπολογιστικών κόμβων, και ομάδων κόμβων (clusters). Γίνεται διάκριση ανάμεσα στον τύπο (class) ενός αντικειμένου και σε συγκεκριμένες εμφανίσεις (instances) αυτού, γεγονός που επιτρέπει οικονομία στον ορισμό ενός πλήθους από αντικείμενα του ίδιου τύπου. Τα αντικείμενα αυτά αποτελούν τα δομικά στοιχεία του μοντέλου προσομοίωσης. Παρακάτω περιγράφονται οι κύριες συντακτικές δομές της γλώσσας, και δίδονται παραδείγματα χρήσης τους. Η γραμματική της γλώσσας παρουσιάζεται στο παράρτημα Α, ενώ στο παράρτημα Β δίδεται ένα πλήρες μοντέλο προσομοίωσης, διατυπωμένο στην γλώσσα προδιαγραφής. Στα παρακάτω παραδείγματα με κεφαλαία σημειώνονται τα τελικά σύμβολα (tokens) της γλώσσας.

4.3.1 Περιγραφή Συσκευών Αποθήκευσης

Για την περιγραφή των συσκευών αποθήκευσης που χρησιμοποιούνται σε ένα σύστημα επεξεργασίας δοσοληψιών για την αποθήκευση δεδομένων και την ασφαλή καταγραφή αλλαγών (logging) στην βάση δεδομένων, η γλώσσα προδιαγραφής παρέχει την συντακτική δομή `DEFINE IO_DEVICE` για τον καθορισμό παραμέτρων που επηρεάζουν την επίδοση. Το μοντέλο συστήματος που υποστηρίζεται καθορίζει ότι κάθε κόμβος έχει ακριβώς μία συσκευή αποθήκευσης για καταγραφή αλλαγών, ενώ μπορεί να έχει οποιοδήποτε πλήθος συσκευών αποθήκευσης δεδομένων.

Η κύρια κατηγορία συσκευών αποθήκευσης που χρησιμοποιείται είναι οι μαγνητικοί δίσκοι. Τον χρόνο περάτωσης μιας προσπέλασης καθορίζουν τρείς παράγοντες: η καθυστέρηση αναζήτησης (seek delay), η καθυστέρηση λόγω περιστροφής (rotational delay), και η καθυστέρηση μεταφοράς δεδομένων από και προς την κύρια μνήμη (I/O transfer delay). Η καθυστέρηση αναζήτησης είναι ο χρόνος μέχρι η κεφαλή ανάγνωσης και εγγραφής του δίσκου να μετακινηθεί στον κύλινδρο που καθορίζει η διεύθυνση του προσπελαζόμενου block. Λόγω της περιστροφής του μαγνητικού δίσκου, κάθε προσπέλαση επιβαρύνεται με την καθυστέρηση της αναμονής για να έρθει το επιθυμητό τμήμα του κυλίνδρου κάτω από την κεφαλή ανάγνωσης και εγγραφής. Η καθυστέρηση αυτή έχει κατά μέσο όρο διάρκεια το ήμισυ του χρόνου για μια πλήρη περιστροφή του δίσκου. Μόλις η κεφαλή βρεθεί στην επιθυμητή θέση, ο ελεγκτής του δίσκου μπορεί να αρχίσει να μεταφέρει τα δεδομένα του block, με ρυθμό μεταφοράς που ορίζεται ως ιδιότητα της συσκευής. Ακολουθεί παράδειγμα περιγραφής ενός τύπου δίσκων:

```
DEFINE IO_DEVICE DataDisk WITH {
   IO_COPY_DELAY: 0.001;   % time to transfer a block to/from memory
   IO_LOAD_DELAY: 0.00075; % time to "load" disk arm
   IO_SEEK_DELAY: 0.00075; % time to move disk head from track to track
   IO_ROTATIONAL_DELAY: 0.008333; % time for full disk rotation
   NUM_HEADS: 8;        % number of disk heads
   NUM_SECTORS: 250000; % number of blocks (total: around 2 GBytes)
}
```

Η γλώσσα υποστηρίζει και μια απλουστευμένη μορφή περιγραφής ενός τύπου συσκευών, που απαιτεί μόνο τον καθορισμό της (αναμενόμενης) ελάχιστης και μέγιστης καθυστέρησης προσπέλασης στην συσκευή. Κατά την προσομοίωση του συστήματος, η καθυστέρηση για μια προσπέλαση λαμβάνεται να είναι μια τυχαία μεταβλητή με

ομοιόμορφη κατανομή στο διάστημα που ορίζεται από τις ακραίες τιμές που καθορίστηκαν κατά την δήλωση του τύπου της συσκευής. Οι τιμές αυτές μετρώνται σε seconds. Ακολουθεί παράδειγμα τέτοιας περιγραφής:

```
DEFINE IO_DEVICE LogDisk WITH { % type of device used for logging
   IO_DELAY_MIN:  0.015;   % minimum I/O delay:  15 msec
   IO_DELAY_MAX:  0.020;   % maximum I/O delay:  20 msec
}
```

Αυτή η περιγραφή αγνοεί εντελώς λεπτομέρειες σχετικές με την λειτουργία της συσκευής, παρουσιάζοντάς την ουσιαστικά σαν ένα "μαύρο κουτί" που εκτελεί μια συγκεκριμένη λειτουργία με σαφώς καθορισμένους περιορισμούς επίδοσης. Αυτού του είδους η μοντελοποίηση είναι αρκετά συνηθισμένη στην προσομοίωση σύνθετων συστημάτων, συνεπώς είναι επιθυμητό ο προσομοιωτής TPsim να υποστηρίζει και τέτοιου είδους μοντέλα. Η χρήση τους συχνά είναι δικαιολογημένη όταν η μελέτη προσομοίωσης εστιάζει σε θέματα σχετικά με την επίδοση του συστήματος που δεν επηρεάζονται σημαντικά από την λεπτομέρεια στην περιγραφή κάποιων υποσυστημάτων, αρκεί βέβαια στο μοντέλο προσομοίωσης να έχει παρασταθεί με ικανοποιητική ακρίβεια κάθε αλληλεπίδραση τέτοιων υποσυστημάτων με άλλα τμήματα του μοντέλου.

4.3.2 Περιγραφή Κόμβων και Δικτύου Κόμβων

Για την περιγραφή ενός κόμβου, η γλώσσα απαιτεί να έχει οριστεί ένας τύπος κόμβων, του οποίου ο κόμβος αποτελεί εμφάνιση. Ο ορισμός ενός τύπου κόμβων περιλαμβάνει τον καθορισμό παραμέτρων όπως ο βαθμός πολυπρογραμματισμού (multi−programming level), το μέγεθος του ενταμιευτή που χρησιμοποιεί ο κόμβος για να επιταχύνει τις προσπελάσεις του στην βάση δεδομένων, ο τύπος των συσκευών αποθήκευσης που χρησιμοποιεί ο κόμβος για την αποθήκευση δεδομένων, και μια σειρά παραμέτρων που περιγράφουν το κόστος εκτέλεσης βασικών λειτουργιών που είναι απαραίτητες για την εξυπηρέτηση δοσοληψιών.

Στα πλαίσια αυτής της εργασίας μελετήθηκαν αποκλειστικά συστήματα με μονοεπεξεργαστικούς (uniprocessor) κόμβους, αν και η γλώσσα προδιαγραφής επιτρέπει τον ορισμό κόμβων με περισσότερους από ένα επεξεργαστές οι οποίοι οργανώνονται σε μια αρχιτεκτονική τύπου UMA (Uniform Memory Access), όπου όλοι οι επεξεργαστές έχουν άμεση πρόσβαση σε μια κοινόχρηστη μνήμη και σε όλους τους δίσκους του κόμβου. Δεν υποστηρίζονται κόμβοι με περισσότερους από ένα επεξεργαστές οργανωμένους σε αρχιτεκτονική NUMA (Non−Uniform Memory Access), όπου υπάρχουν οι έννοιες της "μακρινής" και της "τοπικής" μνήμης.

Τύποι Κόμβων

Ο κάθε επεξεργαστής χαρακτηρίζεται από τον (μέσο) ρυθμό με τον οποίο εκτελεί εντολές, ο οποίος μετράται σε εκατομμύρια εντολών ανά second. Ο βαθμός πολυπρογραμματισμού καθορίζει το μέγιστο πλήθος δοσοληψιών που μπορούν να είναι ταυτόχρονα ενεργές σε ένα κόμβο. Το πλήθος αυτό επηρεάζει άμεσα τον βαθμό χρησιμοποίησης (utilization) του επεξεργαστή, καθώς και τον βαθμό του ανταγωνισμού λόγω ταυτόχρονης προσπέλασης σε δεδομένα (data contention). Ενα μέρος της μνήμης κάθε κόμβου δεσμεύεται για να χρησιμοποιηθεί ως ενταμιευτής, με σκοπό να μειώσει τις προσπελάσεις στις συσκευές αποθήκευσης που κρατούν τα δεδομένα,

αξιοποιώντας την τοπικότητα των αναφορών που χαρακτηρίζει τις προσπελάσεις. Η γλώσσα προδιαγραφής επιτρέπει να καθοριστεί το μέγεθος αυτού του ενταμιευτή.

Κάθε κόμβος έχει μια συσκευή αποθήκευσης για την καταγραφή αλλαγών, και ένα αριθμό από συσκευές αποθήκευσης δεδομένων. Η γλώσσα προδιαγραφής επιτρέπει τον καθορισμό του τύπου της συσκευής αποθήκευσης για την καταγραφή αλλαγών, και του τύπου των συσκευών αποθήκευσης δεδομένων. Ολες οι συσκευές αποθήκευσης δεδομένων είναι πανομοιότυπες στο θεωρούμενο μοντέλο προσομοίωσης. Η γλώσσα προδιαγραφής επιτρέπει τον καθορισμό του πλήθους τους. Η ανάθεση των φυσικών σελίδων δεδομένων στις συσκευές αυτές θεωρείται ότι καθορίζεται από την φυσική σχεδίαση της βάσης δεδομένων, και λαμβάνεται υπ'όψιν στο μοντέλο προσομοίωσης. Την ευθύνη για την ανάθεση αυτή, και την μοντελοποίηση της επίδρασης που έχει στην εκτέλεση δοσοληψιών, έχουν άλλα τμήματα του προσομοιωτή.

Την περιγραφή ενός τύπου κόμβων συμπληρώνει ένας αριθμός από παραμέτρους που εκφράζουν το κόστος εκτέλεσης βασικών λειτουργιών του συστήματος επεξεργασίας δοσοληψιών. Το κόστος μιας λειτουργίας εκφράζεται ως το πλήθος των εντολών (instruction pathlength) που απαιτούνται για την εκτέλεσή της. Το κόστος αυτό μπορεί να εκτιμηθεί σε ένα πραγματικό σύστημα. Εχοντας καθορίσει τον ρυθμό εκτέλεσης εντολών, είναι δυνατή η μετατροπή του αριθμού εντολών σε χρόνο του επεξεργαστή, ώστε το κόστος αυτό να ληφθεί υπ'όψιν στην εκτίμηση του χρόνου εξυπηρέτησης των δοσοληψιών που εκτελούνται στο προσομοιούμενο σύστημα. Οι λειτουργίες για τις οποίες καθορίζεται το κόστος εκτέλεσης κατά την δήλωση ενός τύπου κόμβων είναι η σύνδεση και αποσύνδεση μιας δοσοληψίας με ένα νήμα ελέγχου, η εκτέλεση μιας προσπέλασης στην βάση δεδομένων, τα βήματα του πρωτοκόλλου δέσμευσης, και η εκκίνηση μιας προσπέλασης σε μια συσκευή αποθήκευσης.

Ακολουθεί παράδειγμα δήλωσης ενός τύπου κόμβων:

```
DEFINE NODE_CLASS nodeType WITH {
   CPUcnt: 1;      % number of CPU's
   MPL: 50;        % multi-programming level
   CPUrate: 50.0;  % CPU capacity, measured in MIPS
   % The following costs are expressed as instruction counts (pathlengths).
   ATTACH_TASK_COST: 15100.0;  % cost of attaching a transaction with a thread
   DM_INTERFACE_COST: 2000.0;  % fixed cost for access to the DB
   DM_CALL_COST: 4000.0;       % average cost for executing a DB access call
   DM_IO_COST: 10000.0;        % fixed cost for access to an I/O device
   FUNCTION_SHIP_SEND_COST: 12600.0; % cost of sending a remote request
   FUNCTION_SHIP_RECV_COST: 12600.0; % cost of receiving a remote response
   % The following four parameters define the cost of the
   % 2-phase commit protocol (coordinator side).
   PRIMARY_PREPARE_COST: 7000.0;
   SEND_PREPARE_COST: 12600.0;
   SEND_COMMIT_COST: 12600.0;
   PRIMARY_COMMIT_COST: 14000.0;
   % The following four parameters define the cost of the
   % 2-phase commit protocol (participant side).
   RECV_PREPARE_COST: 12600.0;
   RECV_COMMIT_COST: 12600.0;
   SECONDARY_PREPARE_COST: 12000.0;
   SECONDARY_COMMIT_COST: 12000.0;
   LOG_IO_COST: 5000.0;       % cost of logging I/O
   DETACH_TASK_COST: 15100.0; % cost of detaching a transaction from a thread
   DM_BUFFER_SIZE: 20000;   % size of database buffer: 160 MBytes (8 KB pages)
   LOG_IO_DEVICE: LogDisk;  % type of I/O device used for logging
   DM_IO_DEVICE:  DataDisk; % type of I/O device(s) used for data storage
   numDisks: 2;             % number of disks with data files
```

```
}
```

Δήλωση Κόμβων

Εχοντας ορίσει ένα τύπο κόμβων, είναι δυνατή η δήλωση ενός συγκεκριμένου κόμβου, ως εμφάνιση (instance) του τύπου. Η δήλωση γίνεται όπως δείχνει το παρακάτω παράδειγμα:

```
DEFINE NODE ProcessingNode OF CLASS nodeType;
```

Μια παραλλαγή αυτής της δήλωσης επιτρέπει την δήλωση κόμβων που δεν εξυπηρετούν δοσοληψίες οι ίδιοι, παρά μόνο δρομολογούν δοσοληψίες σε άλλους κόμβους. Η δήλωση ενός τέτοιου κόμβου – front−end – φαίνεται στο παρακάτω παράδειγμα:

```
DEFINE NODE FrontEndNode OF CLASS nodeType (FRONT_END) ;
```

Δήλωση Ομάδων Κόμβων

Η γλώσσα προδιαγραφής επιτρέπει την δήλωση ομάδων κόμβων, όπου κάθε ομάδα κόμβων αποτελείται από ένα αριθμό μελών που διασυνδέονται μέσω ενός τοπικού δικτύου επικοινωνίας. Ομάδες κόμβων μπορούν να διασυνδεθούν μέσω ενός δικτύου κορμού (back−bone), ώστε να σχηματιστεί μια ιεραρχική δομή διασύνδεσης. Μια ομάδα κόμβων μπορεί να είναι ομοιογενής, δηλαδή να αποτελείται από ένα αριθμό κόμβων του ίδιου τύπου, ή ετερογενής. Μια ετερογενής ομάδα κόμβων δηλώνεται με απαρίθμηση των μελών της, τα οποία μπορεί να είναι διαφορετικών τύπων. Ολόκληρες ομάδες κόμβων μπορούν να δηλωθούν ως μέλη μιας άλλης (υπερ)ομάδας (super−cluster), επιτρέποντας έτσι τον αναδρομικό, ‘‘από κάτω προς τα πάνω’’ (bottom−up) ορισμό ιεραρχικών δομών διασύνδεσης. Το δίκτυο διασύνδεσης, είτε τοπικό είτε δίκτυο κορμού, που δηλώνεται με αυτόν τον τρόπο, αντιμετωπίζεται σαν μαύρο κουτί. Η γλώσσα προδιαγραφής επιτρέπει τον καθορισμό του ρυθμού μετάδοσης, και του μεγέθους των πακέτων με τα οποία μεταδίδονται τα μηνύματα, αλλά δεν παρέχει την δυνατότητα καθορισμού κάποιας συγκεκριμένης τοπολογίας. Παρέχεται όμως η δυνατότητα να οριστούν σύνδεσμοι επικοινωνίας (communication links), για την απευθείας σύνδεση δύο κόμβων.

Το υποσύστημα του προσομοιωτή που μοντελοποιεί την μετάδοση μηνυμάτων δρομολογεί τα πακέτα που απαρτίζουν κάθε μήνυμα, ενδεχομένως μεταδίδοντάς τα μέσω κάποιου δικτύου κορμού εάν ο κόμβος προορισμού δεν βρίσκεται στο ίδιο τοπικό δίκτυο με τον κόμβο αφετηρίας του μηνύματος. Για να μπορεί να μεταδοθεί ένα μήνυμα από ένα τοπικό δίκτυο σε ένα άλλο διαμέσω δικτύου κορμού, πρέπει να οριστεί ένας κόμβος στο καθένα από τα δύο τοπικά δίκτυα ως gateway και εν συνεχεία να οριστεί ένας σύνδεσμος επικοινωνίας που να συνδέει τους δύο κόμβους. Για να μοντελοποιηθεί μια συγκεκριμένη τοπολογία διασύνδεσης κόμβων, χρειάζεται ο χρήστης να τροποποιήσει κατάλληλα το σχετικό υποσύστημα του προσομοιωτή.

Στα παρακάτω παραδείγματα φαίνεται η δήλωση μιας ομοιογενούς ομάδας κόμβων (με 8 μέλη) και μιας ετερογενούς ομάδας κόμβων (με 2 μέλη):

```
DEFINE NODE_CLUSTER HomogeneousCluster OF CLASS nodeType WITH {
   numNodes: 8;          % number of nodes in cluster
   packetSize: 1024;     % packet size (in bytes)
   transferRate: 500000; % measured in bytes per sec (4 MBit).
}
% Τα μέλη της ομάδας HomogeneousCluster έχουν τα ονόματα
% HomogeneousCluster_node_1, ... , HomogeneousCluster_8.
```

```
DEFINE NODE NodeA OF CLASS nodeTypeA;
DEFINE NODE NodeB OF CLASS nodeTypeB;
DEFINE NODE_CLUSTER HeterogeneousCluster WITH {
   numNodes: 2;          % number of nodes in cluster
   packetSize: 1024;     % packet size (in bytes)
   transferRate: 500000; % measured in bytes per sec (4 MBit).
   MEMBER_NODES: { NodeA, NodeB }
}
```

4.3.3 Περιγραφή Βάσης Δεδομένων

Η γλώσσα προδιαγραφής επιτρέπει την περιγραφή της βάσης δεδομένων που προσπελαύνεται από τις δοσοληψίες στο προσομοιούμενο σύστημα, με τον ορισμό του σχήματος της βάσης δεδομένων (database schema) κατά το σχεσιακό μοντέλο, του σχήματος τμηματοποίησης (fragmentation schema), και του σχήματος ανάθεσης (allocation schema). Το σχήμα της βάσης δεδομένων ορίζεται καθορίζοντας το σχήμα κάθε σχέσης. Παράδειγμα ορισμού σχήματος σχέσης δίδεται παρακάτω. Απαριθμούνται τα πεδία–ιδιότητες (attributes) της σχέσης, καθορίζεται ο τύπος για το καθένα από αυτά, και ορίζεται ποιός συνδυασμός από ιδιότητες συνιστά το πρωτεύον κλειδί για την σχέση.

```
DEFINE RELATION Account WITH {
   ATTRIBUTES {
           AccountID: SYMBOLIC;
      AccountBalance: NUMERIC;
   }
   tuplesPerPage: 100.0;
   KEY = {AccountID};
   INDEX ON {AccountID};
}
```

Για κάθε σχέση ορίζεται επίσης ο αριθμός των πλειάδων (tuples) που χωρούν σε μια σελίδα της μνήμης. Είναι ακόμα δυνατός ο προσδιορισμός πεδίων (ή συνδυασμών πεδίων) βάσει των οποίων το σύστημα διαχείρισης δεδομένων μπορεί να κατασκευάσει δομή δεικτοδότησης (index) για να επιταχύνει την εκτέλεση ορισμένων προσπελάσεων.

Για κάθε σχέση επιτρέπεται ο καθορισμός τμηματοποίησης, που μπορεί να είναι είτε οριζόντια (horizontal fragmentation), οπότε η σχέση χωρίζεται σε ένα αριθμό τμημάτων βάσει ενός κατηγορήματος επιλογής (selection predicate), είτε κατακόρυφος (vertical fragmentation), οπότε η σχέση χωρίζεται σε τμήματα που το καθένα περιλαμβάνει ένα υποσύνολο των πεδίων της αρχικής σχέσης μαζί με το πρωτεύον κλειδί (primary key). Επιτρέπεται και ο καθορισμός υβριδικής τμηματοποίησης (hybrid fragmentation) με συνδυασμό (αναδρομικά) οριζόντιας και κατακόρυφης τμηματοποίησης.

Με τον ορισμό της τμηματοποίησης, μια σχέση αντιστοιχεί σε ένα ή περισσότερα τμήματα. Κάθε τμήμα υλοποιείται ως ένα αρχείο δεδομένων, έτσι η ανάθεση δεδομένων ορίζεται στο επίπεδο των τμημάτων, προσδιορίζοντας τους κόμβους στους οποίους αποθηκεύονται τα αντίστοιχα αρχεία δεδομένων. Είναι δυνατό να οριστούν πολλαπλά αντίγραφα (replicas) για ένα τμήμα μιας σχέσης. Ο καθορισμός της ανάθεσης αρχείων δεδομένων σε κόμβους περιλαμβάνει και ορισμό του μεγέθους των αρχείων. Το μέγεθος αυτό μετράται ως αριθμός σελίδων.

Το παρακάτω παράδειγμα ορίζει το σχήμα ανάθεσης για δύο σχέσεις, R και S, όπου για την R ορίζεται οριζόντια τμηματοποίηση και ένα αντίγραφο για κάθε τμήμα, ενώ για την S ορίζονται δύο αντίγραφα, χωρίς να έχει οριστεί τμηματοποίηση:

```
DEFINE RELATION R WITH {
   ATTRIBUTES {
      rA: SYMBOLIC;
      rB: NUMBERIC;
   }
   tuplesPerPage: 20.0;
   KEY = { rA };
   INDEX ON { rA };
}

DEFINE RELATION S WITH {
   ATTRIBUTES {
      sA: SYMBOLIC;
      sB: NUMBERIC;
   }
   tuplesPerPage: 50.0;
   KEY = { sA };
}

DEFINE HORIZONTAL FRAGMENTATION OF R AS {
   TblH1 WITH ((rB >= 128) AND (rB < 512)),
              tuplesPerPage: 20.0;
   TblH2 WITH (NOT((rB >= 128)) OR (rB >= 512)),
              tuplesPerPage: 20.0;
}

% Allocation of fragments of R
DEFINE INSTANCE OF TblH1 AT { Node1 };
DEFINE INSTANCE OF TblH2 AT { Node2 };

% Allocation of S
DEFINE INSTANCE OF S AT { Node1, Node2} WITH numItems: 1000;
```

4.3.4 Περιγραφή Φόρτου Εξυπηρέτησης

Για την περιγραφή του φόρτου εξυπηρέτησης (workload), η γλώσσα προδιαγραφής παρέχει συντακτικές δομές για τον ορισμό κλάσεων μονάδων φόρτου εξυπηρέτησης (classes of units of work), και κλάσεων χρηστών. Μια κλάση χρηστών υποβάλλει στο σύστημα αιτήσεις εξυπηρέτησης για συγκεκριμένες κλάσεις μονάδων φόρτου, όπου μια κλάση μονάδων φόρτου χαρακτηρίζεται από τις (αναμενόμενες) απαιτήσεις πόρων. Ο φόρτος εξυπηρέτησης για το σύστημα περιγράφεται με τον καθορισμό των κλάσεων χρηστών που είναι ενεργές σε κάθε κόμβο του συστήματος.

Η γλώσσα προδιαγραφής επιτρέπει τον ορισμό δύο τύπων κλάσεων μονάδων φόρτου. Οι απλές μονάδες φόρτου είναι δοσοληψίες που διαβάζουν και ενημερώνουν δεδομένα. Για την περιγραφή μιας απλής κλάσης μονάδων φόρτου καθορίζεται το (αναμενόμενο) πλήθος εντολών που εκτελεί το αντίστοιχο πρόγραμμα εφαρμογής (εκτός των εντολών που εκτελούνται για την υλοποίηση λειτουργιών του συστήματος), και η κατανομή των προσπελάσεων από δοσοληψίες αυτής της κλάσης στα αρχεία που υλοποιούν στο φυσικό επίπεδο την βάση δεδομένων. Καθορίζεται το ελάχιστο και το μέγιστο πλήθος προσπελάσεων, καθώς και η πιθανότητα προσπέλασης για κάθε αρχείο δεδομένων που οι δοσοληψίες αυτής της κλάσης χρειάζονται. Με τον τρόπο αυτό καθορίζεται η "συγγένεια" (affinity) μεταξύ αρχείων δεδομένων και κλάσεων δοσοληψιών, καθώς κάθε κλάση δεδομένων εμφανίζεται να έχει "προτίμηση" σε συγκεκριμένα δεδομένα. Για κάθε προσπελαζόμενο αρχείο δεδομένων καθορίζεται επίσης η πιθανότητα η

προσπέλαση να είναι για εγγραφή. Στην περίπτωση που έχουν οριστεί πολλαπλά αντίγραφα για μια σχέση ή ένα τμήμα σχέσης, είναι ευθύνη του υποσυστήματος του προσομοιωτή που μοντελοποιεί τις προσπελάσεις δεδομένων στο φυσικό επίπεδο να επιλέξει τα αντίγραφα που θα προσπελαστούν.

Εάν μια σχέση έχει τμηματοποιηθεί, είναι δυνατόν να περιγραφεί μια μη ομοιόμορφη κατανομή προσπελάσεων στα τμήματα (skewed access pattern). Παράδειγμα τέτοιας κατανομής των προσπελάσεων είναι η κατανομή 80−20, όπου 80% των προσπελάσεων αφορούν 20% των δεδομένων. Τέτοιου είδους κατανομές είναι πολύ συνηθισμένες σε πραγματικές εφαρμογές. Με την χρήση ενταμιευτή ικανού μεγέθους, το σύστημα επεξεργασίας δοσοληψιών μπορεί να αξιοποιήσει την τοπικότητα των αναφορών, περιορίζοντας τον αριθμό των προσπελάσεων σε συσκευές αποθήκευσης δεδομένων.

Μια σύνθετη μονάδα φόρτου (workflow) ορίζεται ως σύνθεση απλών μονάδων φόρτου (δοσοληψιών). Η γλώσσα προδιαγραφής, στην παρούσα της μορφή, υποστηρίζει τον ορισμό αλυσίδων από δοσοληψίες. Η υλοποίηση της ροής δεδομένων και ελέγχου μιας σύνθετης δοσοληψίας είναι ευθύνη του υποσυστήματος του προσομοιωτή που ασχολείται με την δρομολόγηση και τον χρονοπρογραμματισμό δοσοληψιών. Η περιγραφή μιας σύνθετης κλάσης φόρτου περιλαμβάνει τον καθορισμό του "προγράμματος" (ουσιαστικά μιας ρουτίνας σε γλώσσα C) που υλοποιεί ακριβώς την ροή δεδομένων και ελέγχου, με το να υποβάλλει στο σύστημα αιτήσεις εξυπηρέτησης για τις απλές δοσοληψίες που συγκροτούν την σύνθετη μονάδα φόρτου.

Μια κλάση χρηστών χαρακτηρίζεται από τις κλάσεις μονάδων φόρτου που υποβάλλει στο σύστημα προς εξυπηρέτηση. Για την περιγραφή της, η γλώσσα προδιαγραφής επιτρέπει να καθοριστεί η σχετική συχνότητα με την οποία χρήστες αυτής της κλάσης ζητούν την εκτέλεση μονάδων φόρτου για κάθε μια από τις κλάσεις μονάδων φόρτου που ενδιαφέρουν τους χρήστες αυτούς.

Το παρακάτω παράδειγμα δείχνει τον ορισμό μιας κλάσης χρηστών που υποβάλλει, με την ίδια πιθανότητα, αιτήσεις εξυπηρέτησης για απλές δοσοληψίες της κλάσης classA1 και για σύνθετες δοσοληψίες της κλάσης classB. Οι δοσοληψίες της κλάσης classA1 εκτελούν κατά μέσο όρο 80000 εντολές (εκτός από τις εντολές για την εκτέλεση λειτουργιών του συστήματος), και από 4 έως 12 προσπελάσεις, οι οποίες με σχετική συχνότητα 20% αφορούν αρχείο δεδομένων που υλοποιεί την σχέση (ή το τμήμα σχέσης) rA, ενώ με σχετική συχνότητα 80% αφορούν αρχείο δεδομένων που υλοποιεί την rB. Η πιθανότητα μια προσπέλαση, είτε στην rA είτε στην rB, να είναι για εγγραφή/μεταβολή δεδομένων ορίζεται να είναι 80%. Θεωρώντας ότι αντίστοιχα έχουν οριστεί οι κλάσεις δοσοληψιών classA2, classA3, η σύνθετη κλάση μονάδων φόρτου classB ορίζεται ως μια αλυσίδα 3 δοσοληψιών, των κλάσεων classA1, classA2, classA3 αντίστοιχα. Στο παράδειγμα καθορίζεται ότι η σύνθετη δοσοληψία υλοποιείται με το πρόγραμμα classB_Program.

```
DEFINE TRANSACTION_CLASS  classA1 AS {
   applicationBurst: 80000.0;
   blocksAccessed_min: 4;
   blocksAccessed_max: 12;
   accessProbability OF INSTANCE rA: 0.2;
   writeProbability OF INSTANCE  rA: 0.8;
   accessProbability OF INSTANCE rB: 0.8;
   writeProbability OF INSTANCE  rB: 0.8;
}

DEFINE WORKFLOW_CLASS WC1 AS {
   program: classB_Program;
   chain :  { classA1, classA2, classA3 };
```

```
}

DEFINE CLIENT_CLASS clientClass AS {
  arrivalRate: 5.0;
  SUBMIT classA1 WITH PROBABILITY 0.5;
  SUBMIT classB  WITH PROBABILITY 0.5;
}
```

Ο προσομοιωτής TPsim υποστηρίζει δύο είδη μοντέλων φόρτου εξυπηρέτησης. Στο κλειστό μοντέλο, υπάρχει ένας αριθμός από χρήστες, διαφόρων κλάσεων, που εκτελούν τον εξής κύκλο:

1. Υποβολή αίτησης εξυπηρέτησης για μια μονάδα φόρτου μιας από τις κλάσεις που ενδιαφέρουν την κλάση του χρήστη. Η αίτηση φτάνει ως μήνυμα στο υποσύστημα επικοινωνίας του κόμβου στον οποίο συνδέεται το τερματικό του χρήστη, και διαβιβάζεται στο υποσύστημα που έχει την ευθύνη για την δρομολόγηση και τον χρονοπρογραμματισμό μονάδων φόρτου.
2. Ο χρήστης περιμένει μήνυμα απάντησης στην αίτηση εξυπηρέτησης.
3. Μετά την λήψη της απάντησης, ο χρήστης στέλνει την επόμενη αίτηση εξυπηρέτησης (βήμα 1 του κύκλου), μετά από αναμονή της οποίας η μέση διάρκεια είναι παράμετρος στον ορισμό της κλάσης στην οποία ανήκει ο χρήστης. Η αναμονή αυτή αναφέρεται στην βιβλιογραφία με τον όρο think time.

Σύμφωνα με το μοντέλο αυτό, ο φόρτος εξυπηρέτησης περιγράφεται στην γλώσσα προδιαγραφής ορίζοντας, για κάθε κλάση χρηστών, την μέση διάρκεια αναμονής από την περάτωση της εξυπηρέτησης μιας μονάδας φόρτου έως την αποστολή της επόμενης αίτησης εξυπηρέτησης, καθώς και τον αριθμό των χρηστών κάθε κλάσης που είναι ενεργοί σε κάθε κόμβο. Στον ορισμό απαριθμούνται οι κόμβοι στους οποίους υπάρχουν ενεργοί χρήστες. Η γλώσσα προδιαγραφής παρέχει, για διευκόλυνση, το τελικό σύμβολο ALL_SITES ως συντομογραφία που δηλώνει το σύνολο των κόμβων του συστήματος. Η διάρκεια αναμονής θεωρείται ότι ακολουθεί εκθετική κατανομή, με την καθορισμένη για την κλάση μέση τιμή. Ως συνέχεια του προηγούμενου παραδείγματος, ο φόρτος εξυπηρέτησης για το σύστημα μπορεί να οριστεί ως εξής:

```
DEFINE SYNTHETIC WORKLOAD wrkLoad AS {
   numClients OF CLASS clientClass:  50 AT { Node1, Node2 };
}
% 50 χρήστες της κλάσης clientClass είναι ενεργοί στον καθένα από τους
% κόμβους Node1, Node2. Η παράμετρος think-time για την κλάση χρηστών
% clientClass είναι ίση με 5 seconds.
```

Ο προσομοιωτής TPsim υποστηρίζει επίσης ανοικτό μοντέλο φόρτου, στο οποίο θεωρείται ότι υπάρχει, σε κάθε κόμβο του συστήματος και για κάθε κλάση χρηστών που είναι ενεργή στον κόμβο, μια "πηγή φόρτου" που στέλνει αιτήσεις εξυπηρέτησης, με ρυθμό που καθορίζεται ως παράμετρος στον ορισμό της κλάσης χρηστών. Η γλώσσα προδιαγραφής επιτρέπει τον καθορισμό του μέσου χρόνου μεταξύ διαδοχικών αιτήσεων για κάθε κλάση χρηστών. Θεωρείται ότι ο χρόνος αυτός είναι τυχαία μεταβλητή που ακολουθεί εκθετική κατανομή. Το μοντέλο αυτό για τον φόρτο εξυπηρέτησης διαφέρει από το κλειστό μοντέλο στο γεγονός ότι η πηγή φόρτου δεν περιμένει να λάβει ειδοποίηση για την περάτωση της εξυπηρέτησης μιας μονάδας φόρτου για να προχωρήσει στην αποστολή της επόμενης αίτησης.

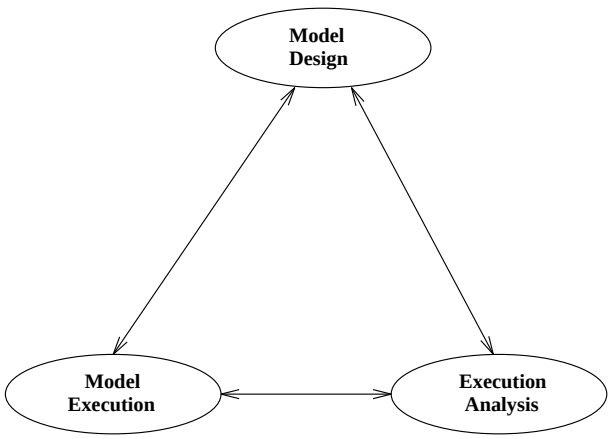
Τέλος, η γλώσσα προδιαγραφής επιτρέπει στον χρήστη του προσομοιωτή να καθορίσει στόχους επίδοσης, με την μορφή απαιτήσεων για τον μέσο χρόνο εξυπηρέτησης μονάδων φόρτου, για κάθε κλάση μονάδων φόρτου. Ως συνέχεια του προηγούμενου

παραδείγματος, παρακάτω καθορίζεται στόχος επίδοσης ίσος με 1 second για την κλάση μονάδων φόρτου classA, και στόχος επίδοσης ίσος με 2.5 second για την κλάση classB. Οπως φαίνεται από το παράδειγμα, επιτρέπεται ο καθορισμός στόχων επίδοσης και για απλές και για σύνθετες μονάδες φόρτου.

```
DEFINE PERFORMANCE_GOAL OF CLASS classA :  1.0;
DEFINE PERFORMANCE_GOAL OF CLASS classB :  2.5;
```

4.4 Διαχείριση Πειραμάτων Προσομοίωσης

Πέρα από την περιγραφή του μοντέλου συστήματος, μια μελέτη προσομοίωσης απαιτεί και τον καθορισμό μιας σειράς πειραμάτων, που υλοποιείται με διαδοχικές εκτελέσεις του προγράμματος προσομοίωσης. Ο όρος "διαχείριση πειραμάτων" (experiment management) αναφέρεται στον συντονισμό της διεξαγωγής μιας σειράς πειραμάτων, και στην συλλογή στατιστικών μετρήσεων από αυτά. Η διαχείριση πειραμάτων αποτελεί αναπόσπαστο στοιχείο κάθε μελέτης προσομοίωσης, όπως σημειώνεται χαρακτηριστικά στο σχήμα 4.4, που δείχνει ότι μια μελέτη προσομοίωσης μπορεί να χωριστεί σε τρεις φάσεις με σημαντικές αλληλεπιδράσεις μεταξύ τους. Οι φάσεις αυτές είναι η σχεδίαση του μοντέλου του υπό μελέτη συστήματος, η εκτέλεση του προγράμματος που προσομοιώνει το σύστημα, και η συλλογή και ανάλυση των αποτελεσμάτων από την προσομοίωση. Η διαχείριση πειραμάτων εντάσσεται στην τελευταία φάση.



Σχήμα 4.4: Φάσεις μιας Μελέτης Προσομοίωσης.

Μια μελέτη προσομοίωσης μπορεί να χωριστεί σε τρεις φάσεις με σημαντικές αλληλεπιδράσεις: σχεδίαση μοντέλου προσομοιωτή, υλοποίηση και εκτέλεση προσομοιωτή, ανάλυση και αξιολόγηση αποτελεσμάτων.

Μια περιγραφή συστήματος και φόρτου εξυπηρέτησης με την γλώσσα προδιαγραφής αντιστοιχεί ουσιαστικά σε *μια* εκτέλεση του προσομοιωτή TPsim. Συνεπώς, για να καθοριστεί μια σειρά πειραμάτων, ακόμα και αν αυτά διαφοροποιούνται σε μια μόνο παράμετρο του μοντέλου, είναι κατ'αρχήν απαραίτητο ο χρήστης να δώσει μια περιγραφή του μοντέλου προσομοίωσης για κάθε εκτέλεση του προγράμματος προσομοίωσης που απαιτείται στα πλαίσια του πειράματος προσομοίωσης. Επίσης, υπάρχει το πρόβλημα της συλλογής των μετρήσεων που προκύπτουν από την εκτέλεση του

προγράμματος προσομοίωσης για κάθε σημείο, στον χώρο που ορίζουν οι μεταβλητές του μοντέλου προσομοίωσης, που έχει καθοριστεί για το δοθέν πείραμα. Η διαδικασία αυτή είναι επίπονη και χρονοβόρα, συνεπώς είναι επιθυμητό να αυτοματοποιηθεί.

Η περιγραφή του μοντέλου συστήματος και του φόρτου εξυπηρέτησης μπορεί να θεωρηθεί ανεξάρτητη από την περιγραφή πειραμάτων προσομοίωσης. Αν και είναι δυνατή η επέκταση της γλώσσας προδιαγραφής ώστε να μπορεί να περιγράψει και πειράματα προσομοίωσης, στα πλαίσια της εργασίας αυτής προτιμήθηκε να σχεδιαστεί ένας ανεξάρτητος μηχανισμός διαχείρισης πειραμάτων. Η επιλογή αυτή απλοποιεί την σχεδίαση και της γλώσσας προδιαγραφής, καθώς αυτή δεν χρειάζεται να περιλαμβάνει συντακτικές δομές για την περιγραφή πειραμάτων, αλλά και του μηχανισμού διαχείρισης πειραμάτων, καθώς αυτός μπορεί να αναπτυχθεί χωρίς να εξαρτάται από την περιγραφή του μοντέλου προσομοίωσης, και κατά συνέπεια η σχεδίασή του μπορεί να αποδεσμευθεί από την σχεδίαση του μεταφραστή της γλώσσας περιγραφής αλλά και από τον ίδιο τον προσομοιωτή, ώστε να είναι σε θέση να χειριστεί γενικά πειράματα προσομοίωσης, πέρα από το συγκεκριμένο πεδίο εφαρμογής του προσομοιωτή TPsim. Επιπλέον, η επιλογή αυτή επιτρέπει την ανεξάρτητη εξέλιξη της γλώσσας περιγραφής και του μηχανισμού διαχείρισης, παρέχοντας έτσι μεγαλύτερη ευελιξία σε σχέση με την επιλογή της σύνθεσης των δύο. Το αποτέλεσμα είναι ένα γενικής εφαρμογής περιβάλλον υποστήριξης πειραμάτων προσομοίωσης, που μπορεί να συντονίσει την διεξαγωγή περισσότερων από ένα πειραμάτων παράλληλα, και αυτοματοποιεί την διαδικασία συλλογής στατιστικά αξιόπιστων μετρήσεων. Η σχεδίαση και υλοποίηση αυτού του περιβάλλοντος υποστήριξης πειραμάτων προσομοίωσης που αναπτύχθηκε στα πλαίσια αυτής της εργασίας περιγράφεται λεπτομερώς στο κεφάλαιο 6. Βάσει του σχήματος 4.4, η περιγραφή αυτή συμπληρώνει την παρουσίαση της σχεδίασης και υλοποίησης του περιβάλλοντος προσομοίωσης συστημάτων επεξεργασίας δοσοληψιών που αποτελεί το κύριο αντικείμενο αυτής της εργασίας.

4.5 Επισκόπιση της Σχετικής Βιβλιογραφίας

Στην βιβλιογραφία δεν έχουν μέχρι στιγμής αναφερθεί πολλές προσπάθειες για την συστηματική περιγραφή του μοντέλου προσομοίωσης, αν και η προσομοίωση χρησιμοποιείται ευρύτατα για την μελέτη υπολογιστικών συστημάτων, και ειδικά συστημάτων διαχείρισης βάσεων δεδομένων και επεξεργασίας δοσοληψιών. Ακόμα και ευρύτατα διαδεδομένα περιβάλλοντα προσομοίωσης, όπως το CSIM [Sch94], δεν παρέχουν κάποιο έτοιμο μηχανισμό για την δυναμική περιγραφή του μοντέλου προσομοίωσης, ούτε βοηθήματα για την διαχείριση πειραμάτων.

Η πλησιέστερη στην σχεδίαση της γλώσσας προδιαγραφής του προσομοιωτή TPsim προσπάθεια είναι η γλώσσα PRPL (Parallel Relational Plan Language). Η γλώσσα αυτή [Bro94] έχει ως αντικείμενο την περιγραφή συστημάτων διαχείρισης βάσεων δεδομένων. Παρέχει συντακτικές δομές για την περιγραφή των κόμβων επεξεργασίας, του δικτύου επικοινωνίας, και των συσκευών αποθήκευσης δεδομένων, καθώς και της βάσης δεδομένων (ως σύνολο αρχείων) και του φόρτου εξυπηρέτησης. Στην σχεδίασή της έχει δοθεί έμφαση στην περιγραφή διαγραμμάτων εκτέλεσης (execution plans), διατυπωμένων σε σχεσιακή άλγεβρα, που προκύπτουν από την μετάφραση επερωτήσεων διατυπωμένων σε μια γλώσσα επερωτήσεων όπως η SQL. Τα διαγράμματα αυτά περιγράφονται από ένα δένδρο από τελεστές της σχεσιακής άλγεβρας. Παρέχεται η δυνατότητα καθορισμού του κόμβου εκτέλεσης, και των

απαιτήσεων μνήμης, για κάθε κόμβο του διαγράμματος εκτέλεσης. Η δυνατότητα αυτή λείπει από την γλώσσα που δέχεται ο προσομοιωτής TPsim. Η γλώσσα PRPL δεν παρέχει όμως τρόπο για την περιγραφή ετερογενών συστημάτων, καθώς επιβάλλει το μοντέλο προσομοίωσης να έχει πανομοιότυπους κόμβους αλλά και συσκευές αποθήκευσης. Αυτή η επιλογή εν μέρει δικαιολογείται λόγω της έμφασης που δίδεται στην περιγραφή παράλληλων συστημάτων διαχείρισης βάσεων δεδομένων. Επίσης, δεν παρέχεται υποστήριξη για την περιγραφή πειραμάτων.

Αξίζει να γίνει αναφορά και στην γλώσσα προσομοίωσης DeNet [Liv89], μια επέκταση της Modula−2 για την προσομοίωση συστημάτων. Η γλώσσα αυτή επιτρέπει την προσομοίωση συστημάτων που περιγράφονται ως συλλογές από αντικείμενα που ονομάζονται Discrete Event Modules (DEVMs) και υλοποιούνται ως modules στην γλώσσα Modula−2. Τα αντικείμενα αυτά χαρακτηρίζονται από το σύνολο των εισόδων και των εξόδων τους, και από το σύνολο των ‘‘συμβάντων’’ (events) στα οποία ανταποκρίνονται. Ενα μοντέλο προσομοίωσης κατασκευάζεται διασυνδέοντας DEVMs. Το μοντέλο περιγράφεται ως προσανατολισμένος γράφος, με κόμβους τα DEVMs και ακμές που δηλώνουν ροή συμβάντων μεταξύ των συνδεόμενων κόμβων. Ο ορισμός αυτού του γράφου δίδεται από τον χρήστη σε ένα αρχείο (που ονομάζεται topology file) μαζί με αρχικές τιμές για παραμέτρους του κάθε DEVM. Η γλώσσα περιγραφής που δέχεται ο προσομοιωτής TPsim επιτρέπει υψηλότερου επιπέδου περιγραφή, καθώς απευθύνεται σε ένα συγκεκριμένο πεδίο εφαρμογής, αντίθετα με την γλώσσα DeNet που είναι γενικής εφαρμογής.

Κεφάλαιο 5

Ο Προσομοιωτής TPsim: Υλοποίηση

Στο κεφάλαιο αυτό δίδονται στοιχεία για την υλοποίηση του προσομοιωτή TPsim. Περιγράφονται αναλυτικά τα τμήματα που υλοποιούν τα μοντέλα για τους υπολογιστικούς κόμβους, τις συσκευές αποθήκευσης δεδομένων, και το δίκτυο επικοινωνίας, καθώς και τα τμήματα που εξομοιώνουν τα βασικά υποστήματα λογισμικού που απαρτίζουν ένα σύστημα επεξεργασίας δοσοληψιών, συμπεριλαμβανομένων και των διαχειριστών πόρων του συστήματος. Η παρουσίαση αυτή συμπληρώνει την παρουσίαση της σχεδίασης του προσομοιωτή στο κεφάλαιο 4. Επίσης, αναλύεται ο μηχανισμός ελέγχου της προσομοίωσης, που εξομοιώνει συμβάντα που ανακύπτουν κατά την λειτουργία του υπό μελέτη σύστηματος, και ο μηχανισμός συλλογής μετρήσεων για μεταβλητές του μοντέλου προσομοίωσης.

Η ανάπτυξη του προσομοιωτή έγινε σε DEC/3000 workstations, με το λειτουργικό σύστημα Digital UNIX.

5.1 Η Βιβλιοθήκη Υποστήριξης Προσομοίωσης

Ο προσομοιωτής TPsim στηρίζεται στην υλοποίηση ενός περιβάλλοντος εκτέλεσης από μια βιβλιοθήκη υποστήριξης, που επιτρέπει την κατασκευή μοντέλων προσομοίωσης με πολλαπλά νήματα ελέγχου. Η βιβλιοθήκη αυτή χρειάζεται να παρέχει ένα αριθμό από υπηρεσίες που είναι απαραίτητες για την προσομοίωση. Από την άποψη αυτή, η βιβλιοθήκη υποστήριξης μπορεί να θεωρηθεί ένας πυρήνας (kernel) με λειτουργίες παραπλήσιες αυτών του πυρήνα ενός λειτουργικού συστήματος. Οπως ο πυρήνας του λειτουργικού συστήματος παρέχει βασικές υπηρεσίες σε διεργασίες που εκτελούν προγράμματα εφαρμογών, έτσι και η βιβλιοθήκη υποστήριξης προσομοίωσης παρέχει κάποιες υπηρεσίες στον κώδικα που γράφει ένας χρήστης της βιβλιοθήκης για να κατασκευάσει μοντέλο ενός συστήματος. Ο πυρήνας του λειτουργικού συστήματος επιτρέπει πρόσβαση σε πόρους συστήματος, κρύβοντας από τον χρήστη πολλές τεχνικές λεπτομέρειες που έχουν να κάνουν με χαρακτηριστικά των πόρων αυτών, και υλοποιεί την έννοια της διεργασίας, ως αφαιρετικό μοντέλο για το υπολογιστικό σύστημα, παρέχοντας ένα περιβάλλον στο οποίο οι χρήστες μπορούν να χειριστούν διεργασίες. Ανάλογα, η βιβλιοθήκη υποστήριξης προσομοίωσης παρέχει ένα περιβάλλον εκτέλεσης στο οποίο μια σειρά από βασικά αντικείμενα μπορούν να συνδυαστούν για να κατασκευαστεί το μοντέλο συστήματος προσομοίωσης, ενώ ένας αριθμός από νήματα ελέγχου εκτελεί κώδικα που παρέχει ο χρήστης της βιβλιοθήκης για την εξομοίωση του συστήματος, χωρίς ο χρήστης να χρειάζεται να

ασχοληθεί με λεπτομέρειες σχετικές με τον έλεγχο και συντονισμό της διαδικασίας της προσομοίωσης.

Η παρούσα υλοποίηση του προσομοιωτή TPsim στηρίζεται στην βιβλιοθήκη υποστήριξης προσομοίωσης PARASOL [Nei91, Nei94]. Η βιβλιοθήκη αυτή επιτρέπει την ανάπτυξη προσομοιωτών που εκτελούνται ως μια διεργασία με πολλαπλά νήματα ελέγχου που εξομοιώνουν το οριζόμενο από τον χρήστη περιβάλλον λειτουργίας. Παρέχει μια σειρά από ρουτίνες για την δυναμική διαχείριση νημάτων ελέγχου, την υποστήριξη επικοινωνίας και συγχρονισμού μεταξύ τους, και την συλλογή στατιστικών μετρήσεων.

5.1.1 Δομικά Στοιχεία Μοντέλων Προσομοίωσης

Η βασική οντότητα που παρέχει η βιβλιοθήκη PARASOL για την μοντελοποίηση συστημάτων είναι ο ''κόμβος επεξεργασίας'' (processing node). Ενας κόμβος επεξεργασίας ουσιαστικά αποτελεί μοντέλο ενός πόρου συστήματος, όπως για παράδειγμα ένας επεξεργαστής ή ένας δίσκος δεδομένων. Κάθε κόμβος επεξεργασίας περιλαμβάνει μια ή περισσότερες μονάδες εξυπηρέτησης (server). Για παράδειγμα, ένας κόμβος επεξεργασίας για την μοντελοποίηση ενός υπολογιστικού συστήματος μπορεί να περιλαμβάνει πολλαπλές μονάδες εξυπηρέτησης που αντιστοιχούν σε πολλαπλούς επεξεργαστές του συστήματος.

Κάθε νήμα ελέγχου στο συνολικό μοντέλο συστήματος σχετίζεται με ένα κόμβο επεξεργασίας. Η βιβλιοθήκη περιλαμβάνει ρουτίνες που σημειώνουν την κατανάλωση αυτών των πόρων, ώστε η εκτέλεση κώδικα από ένα νήμα ελέγχου που εξομοιώνει λειτουργίες του συστήματος να ''χρεώνεται'' με την ανάλογη κατανάλωση πόρων συστήματος. Αυτή είναι μια πάγια τακτική για την μοντελοποίηση συστημάτων από πλευράς επίδοσης.

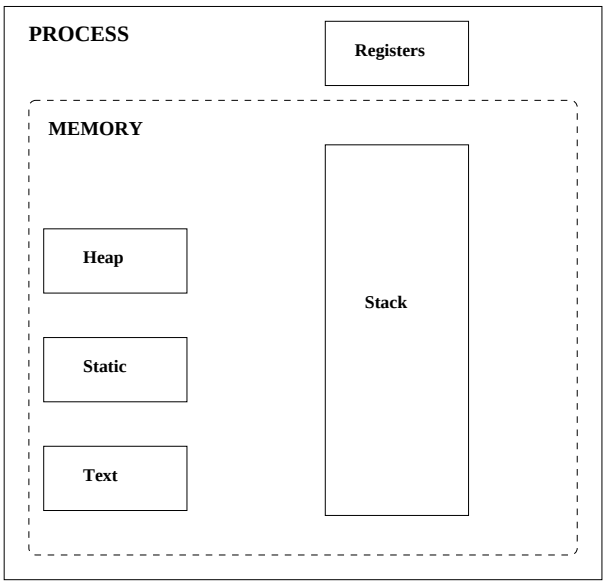
Καθώς με κάθε κόμβο επεξεργασίας μπορούν να συνδέονται πολλαπλά νήματα ελέγχου, είναι απαραίτητο κάθε κόμβος να εφαρμόζει μια πολιτική χρονοπρογραμματισμού (scheduling). Η βιβλιοθήκη PARASOL υποστηρίζει τις πολιτικές FCFS, HOL (head−of−line nonpreemtive priority), και PR (preemptive priority), με την επιπλέον επιλογή κυκλικής εξυπηρέτησης (round−robin) για νήματα ελέγχου με την ίδια προτεραιότητα. Κάθε κόμβος έχει μια ουρά αναμονής για τα νήματα ελέγχου που περιμένουν να εξυπηρετηθούν, ανεξάρτητα από το πλήθος των μονάδων εξυπηρέτησης που διαθέτει.

Η επικοινωνία μεταξύ νημάτων ελέγχου βασίζεται στην έννοια της θύρας επικοινωνίας (communication port). Αυτό σημαίνει ότι τα μηνύματα απευθύνονται αντί σε συγκεκριμένα νήματα ελέγχου σε διευθύνσεις θυρών επικοινωνίας. Κάθε νήμα ελέγχου συνδέεται με την εκκίνησή του με μια θύρα επικοινωνίας, ενώ κατά την διάρκεια της ζωής του μπορεί να δημιουργήσει δυναμικά και άλλες θύρες επικοινωνίας. Η ανταλλαγή μηνυμάτων είναι ο βασικός μηχανισμός για την επικοινωνία μεταξύ νημάτων ελέγχου, αν και είναι δυνατή η από κοινού προσπέλαση σε δομές δεδομένων, καθώς όλα τα νήματα ελέγχου μοιράζονται το πεδίο διευθύνσεων μιας διεργασίας. Κάθε μήνυμα φέρει μια χρονοσφραγίδα (timestamp).

5.1.2 Νήματα Ελέγχου

Από τα παραπάνω γίνεται σαφές ότι η έννοια του νήματος ελέγχου είναι βασική για την λειτουργία της βιβλιοθήκης υποστήριξης προσομοίωσης, και κατά συνέπεια του

προσομοιωτή TPsim. Ενα νήμα ελέγχου (thread of control) αντιπροσωπεύει μια σειριακή ροή ελέγχου μέσα σε ένα πρόγραμμα. Ενα πρόγραμμα μπορεί να περιλαμβάνει πολλαπλά νήματα ελέγχου, τα οποία χρησιμοποιούν από κοινού το πεδίο διευθύνσεων της διεργασίας (process) του λειτουργικού συστήματος που εκτελεί το πρόγραμμα. Πολλαπλά νήματα ελέγχου σε μια διεργασία μπορούν να θεωρηθούν ότι εκτελούνται ταυτόχρονα, όμως δεν μπορεί κανείς να κάνει με ασφάλεια υποθέσεις για την σειρά εκτέλεσής τους, μια και αυτή εξαρτάται από την πολιτική χρονοπρογραμματισμού (scheduling) που εφαρμόζεται για την εξυπηρέτηση νημάτων ελέγχου μιας διεργασίας. Οταν για την ορθότητα του προγράμματος απαιτούνται κάποιες εγγυήσεις για την σειρά εκτέλεσης, ειδικά όταν απαιτείται από κοινού προσπέλαση στην μνήμη χωρίς ανεπιθύμητες παρενέργειες, ο προγραμματιστής οφείλει να χρησιμοποιήσει δομές ελέγχου, όπως οι σημαφόροι (semaphores) και οι μεταβλητές συνθήκης (condition variables) για να επιτευχθεί συγχρονισμός. Στο σχήμα 5.1 παρουσιάζεται η δομή του πεδίου διευθύνσεων μιας διεργασίας με ένα νήμα ελέγχου, ενώ στο σχήμα 5.2 παρουσιάζεται η αντίστοιχη εικόνα για μια διεργασία με πολλαπλά νήματα ελέγχου.



Σχήμα 5.1: Πρόγραμμα Εφαρμογής με ένα Νήμα Ελέγχου.

Το πρόγραμμα εκτελείται από μια διεργασία της οποίας το πεδίο διευθύνσεων χωρίζεται σε περιοχές για να διατηρείται ο κώδικας του προγράμματος, οι μεταβλητές καθολικής εμβέλειας, και η στοίβα εκτέλεσης. Υπάρχει επίσης μια περιοχή που η διεργασία μπορεί να διαχειριστεί δυναμικά. Μέρος της τρέχουσας κατάστασης της διεργασίας αποτελούν και οι τιμές των καταχωρητών, όπως έχουν διαμορφωθεί από την ως τώρα εκτέλεση του προγράμματος.

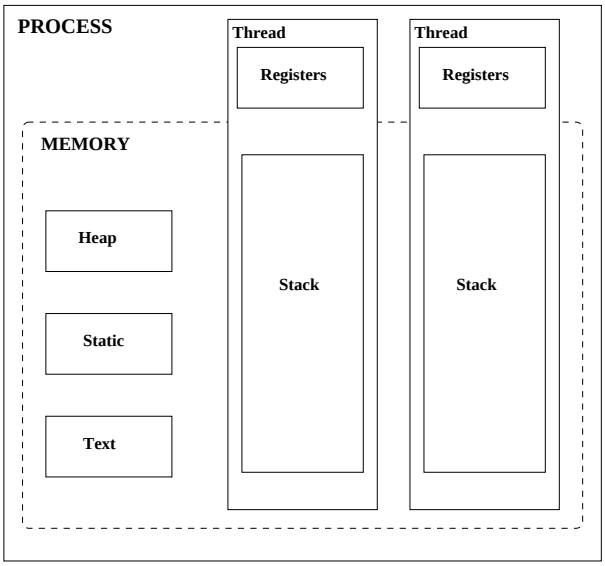
Το πεδίο διευθύνσεων μιας διεργασίας, που εκτελεί ένα προγραμμα σε γλώσσα C στο λειτουργικό σύστημα UNIX, μπορεί να χωριστεί σε τέσσερεις περιοχές:

- Text. Εδώ αποθηκεύεται ο κώδικας του προγράμματος.
- Static. Εδώ κρατούνται οι μεταβλητές καθολικής εμβέλειας του προγράμματος.
- Run−time Stack (στοίβα εκτέλεσης). Χρησιμοποιείται ως χώρος προσωρινής αποθήκευσης κατά την κλήση και εκτέλεση ρουτινών. Στην περιοχή αυτή μεταξύ

άλλων κρατούνται οι μεταβλητές τοπικής εμβέλειας μιας ρουτίνας, καθώς και μια διεύθυνση επιστροφής ώστε με την ολοκλήρωση της εκτέλεσης μιας ρουτίνας να είναι δυνατή η συνέχιση της εκτέλεσης από το σημείο στο οποίο κλήθηκε η ρουτίνα αυτή.

- Heap. Την περιοχή αυτή διαχειρίζεται δυναμικά η διεργασία, όταν ζητάει να δεσμεύσει, κατά την εκτέλεσή της, χώρο μνήμης για κάποια χρήση. Οταν ο χώρος αυτός δεν χρειάζεται πλέον, η διεργασία μπορεί να τον αποδεσμεύσει επιστρέφοντάς τον στην λίστα διαθεσίμων (free list) που διατηρεί το λειτουργικό σύστημα για τον έλεγχο της περιοχής αυτής.

Η κατάσταση αυτών των περιοχών της μνήμης αποτελεί μέρος της κατάστασης της διεργασίας (process state). Την κατάσταση αυτή συμπληρώνουν οι τιμές των καταχωρητών, ελέγχου και γενικής χρήσης, όπως αυτές διαμορφώνονται από την μέχρι στιγμής εκτέλεση του προγράμματος. Οταν η διεργασία περιλαμβάνει πολλαπλά νήματα ελέγχου, κάθε νήμα ελέγχου έχει ιδιωτική στοίβα εκτέλεσης, καθώς και ιδιωτικό αντίγραφο των τιμών των καταχωρητών, ώστε να είναι σε θέση να εκτελέσει κώδικα ανεξάρτητα από τα άλλα, όμως όλα τα νήματα ελέγχου μοιράζονται τις περιοχές Text, Static, και Heap.



Σχήμα 5.2: Πρόγραμμα Εφαρμογής με Πολλαπλά Νήματα Ελέγχου.

Για κάθε νήμα ελέγχου το σύστημα διατηρεί ξεχωριστή στοίβα εκτέλεσης, καθώς και τις τιμές των καταχωρητών (ελέγχου και γενικής χρήσης) όπως έχουν διαμορφωθεί από την ως τώρα εκτέλεσή του. Οι περιοχές της μνήμης όπου φυλάσσεται ο κώδικας του προγράμματος και οι καθολικής εμβέλειας μεταβλητές, καθώς και η περιοχή που το πρόγραμμα διαχειρίζεται δυναμικά είναι κοινές για όλα τα νήματα ελέγχου.

5.1.3 Μέθοδος Προσομοίωσης

Για την προσομοίωση του συστήματος υιοθετείται το μοντέλο προσομοίωσης διακριτών συμβάντων (discrete event simulation model). Το μοντέλο αυτό υποθέτει ότι

στο υπό μελέτη σύστημα σημειώνονται μεταβολές κατάστασης σε διακριτές χρονικές στιγμές στον άξονα του χρόνου προσομοίωσης [Fuj90]. Το μοντέλο συστήματος μεταβαίνει από μια κατάσταση σε μια άλλη όταν εμφανίζονται συγκεκριμένα συμβάντα (events). Ενα παράδειγμα είναι το μοντέλο ενός δικτύου επικοινωνίας, το οποίο περιλαμβάνει κάποιες μεταβλητές κατάστασης που δείχνουν το μέγεθος κάθε ουράς μηνυμάτων στο σύστημα, καθώς και την τρέχουσα κατάσταση κάθε συνδέσμου επικοινωνίας (busy/idle). Στο μοντέλο αυτό, παραδείγματα συμβάντων που οδηγούν σε μεταβολές κατάστασης, που αντιστοιχούν σε αλλαγές τιμών μεταβλητών του μοντέλου, είναι η άφιξη μηνυμάτων σε κάποιο κόμβο του συστήματος, η μεταβίβαση ενός μηνύματος από ένα κόμβο σε κάποιον άλλο βάσει του αλγορίθμου δρομολόγησης που εφαρμόζεται στο σύστημα, καθώς και μια βλάβη σε ένα σύνδεσμο επικοινωνίας.

Η μέθοδος προσομοίωσης διακριτών συμβάντων είναι ιδιαίτερη χρήσιμη για την προσομοίωση ασύγχρονων συστημάτων, στα οποία η εμφάνιση συμβάντων δεν είναι συγχρονισμένη με βάση κάποιο καθολικό ρολόι συστήματος (global clock). Σε τέτοια συστήματα, οι εμφανίσεις συμβάντων σημειώνονται κατά ακανόνιστα χρονικά διαστήματα, που καθορίζονται από την τρέχουσα κατάσταση και τον τρόπο λειτουργίας του συστήματος. Ο προσομοιωτής TPsim είναι ένα παράδειγμα ασύγχρονου συστήματος.

Η υλοποίηση ενός προσομοιωτή για ένα μοντέλο διακριτών συμβάντων στηρίζεται σε τρεις βασικές δομές δεδομένων:

α. Μεταβλητές Κατάστασης (state variables). Περιγράφουν την τρέχουσα κατάσταση του προσομοιούμενου συστήματος, όπως έχει διαμορφωθεί από τα μέχρι στιγμής συμβάντα.

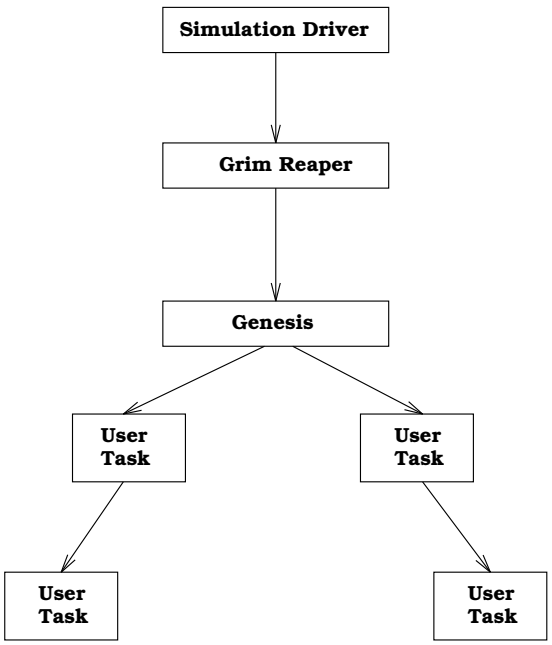
β. Κατάλογος Συμβάντων (event list). Καταγράφει τα συμβάντα τα οποία έχουν δρομολογηθεί αλλά ο προσομοιωτής δεν έχει ακόμα λάβει υπ'όψιν για να επιφέρουν αλλαγές στην κατάσταση του συστήματος. Κάθε συμβάν παριστάνεται με μια δομή που περιλαμβάνει τον τύπο του συμβάντος, που ορίζει το είδος της μεταβολής που η εμφάνιση του συμβάντος επιφέρει στην κατάσταση του συστήματος, και μια χρονοσφραγίδα, που ορίζει πότε εμφανίζεται το συμβάν, με την επακόλουθη μεταβολή κατάστασης.

γ. Ρολόι Προσομοίωσης (simulation clock). Υλοποιεί την έννοια του χρόνου προσομοίωσης, σημειώνοντας πόσο έχει προχωρήσει (στον χρόνο) η προσομοίωση του συστήματος. Ενημερώνεται βάσει των χρονοσφραγίδων των συμβάντων που ο προσομοιωτής ανακτά από τον κατάλογο συμβάντων.

Ο προσομοιωτής χειρίζεται τα συμβάντα που ανακτά από τον κατάλογο συμβάντων, τροποποιώντας τις μεταβλητές κατάστασης του συστήματος, και ενδεχομένως δρομολογώντας νέα συμβάντα σε μελλοντικό χρόνο. Κάθε φορά πρέπει να ανακτάται από τον κατάλογο το συμβάν με την *μικρότερη* χρονοσφραγίδα. Με τον τρόπο αυτό εξασφαλίζεται ότι δεν παραβιάζεται η αιτιότητα (causality): εάν το συμβάν E_a έχει την μικρότερη χρονοσφραγίδα αλλά ο προσομοιωτής επέλεγε να χειριστεί το συμβάν E_b, με χρονοσφραγίδα μεγαλύτερη από αυτή του E_a, τότε θα μπορούσε να μεταβάλλει μεταβλητές κατάστασης που επηρεάζουν τον χειρισμό του E_a, με συνέπεια να δημιουργηθεί εξάρτηση ανάμεσα στο συμβάν αυτό και μια *μελλοντική* κατάσταση του συστήματος, στην οποία το συμβάν δεν έχει ακόμα εμφανιστεί. Τέτοιες εξαρτήσεις παραβιάζουν την αρχή της αιτιότητας, που ορίζει ότι το αίτιο προηγείται του αποτελέσματος.

5.1.4 Οργάνωση του Περιβάλλοντος Εκτέλεσης

Η βιβλιοθήκη υποστήριξης προσομοίωσης PARASOL επιτρέπει την κατασκευή προσομοιωτών διακριτών συμβάντων, παρέχοντας επιπλέον την δυνατότητα να χρησιμοποιηθούν νήματα ελέγχου για την εκτέλεση κώδικα που μοντελοποιεί τμήματα του προσομοιούμενου συστήματος. Η δυνατότητα αυτή επιτρέπει την περιγραφή του μοντέλου προσομοίωσης σε υψηλότερο αφαιρετικό επίπεδο απ'ότι η περιγραφή βάσει μόνο των συμβάντων, αφού η εκτέλεση ενός νήματος ελέγχου περικλείει την εμφάνιση ενός (ενδεχομένως μεγάλου) αριθμού από συμβάντα.



Σχήμα 5.3: Ιεραρχία Νημάτων Ελέγχου.

Ο οδηγός προσομοίωσης (simulation driver) της βιβλιοθήκης υποστήριξης προσομοίωσης PARASOL ξεκινά το νήμα ελέγχου grim reaper, για να επιτρέπει τον τερματισμό νημάτων ελέγχου που ορίζει ο χρήστης χωρίς απώλεια μνήμης. Επίσης, ξεκινά το νήμα ελέγχου genesis, που έχει την ευθύνη της σύνθεσης του περιβάλλοντος εκτέλεσης. Ολα τα νήματα ελέγχου που ξεκινά ο χρήστης είναι απόγονοι του νήματος αυτού, που αποτελεί την γέφυρα ανάμεσα στον οδηγό προσομοίωσης και τον κώδικα του χρήστη.

Στο σχήμα 5.3 απεικονίζεται η δομή του περιβάλλοντος εκτέλεσης, ως ιεραρχία από νήματα ελέγχου, καθώς για κάθε νήμα ελέγχου ορίζεται ένα άλλο ως γεννήτορας. Κάθε νήμα ελέγχου συνδέεται με ένα κόμβο επεξεργασίας, του οποίου οι "πόροι" χρησιμοποιούνται για την εκτέλεση του κώδικα που αντιστοιχεί στο νήμα ελέγχου. Οι πόροι αυτοί καταναλώνονται με την κλήση ορισμένων ρουτινών της βιβλιοθήκης, με τις οποίες ουσιαστικά προχωρά το ρολόι της προσομοίωσης. Χωρίς τις κλήσεις αυτές, ένα νήμα ελέγχου εκτελεί τον κώδικά του *χωρίς* να προχωρά το ρολόι προσομοίωσης. Με τις κλήσεις αυτές, καθώς και με τις κλήσεις ρουτινών συγχρονισμού και επικοινωνίας, ο έλεγχος επιστρέφει στον οδηγό προσομοίωσης, κατά τρόπο αντίστοιχο με την εκτέλεση μιας κλήσης συστήματος (system call) στο λειτουργικό σύστημα UNIX. Ο οδηγός προσομοίωσης της PARASOL ξεκινά το νήμα ελέγχου grim reaper κατά την εκκίνηση του προσομοιωτή. Αυτό με την σειρά του ξεκινά το νήμα ελέγχου genesis,

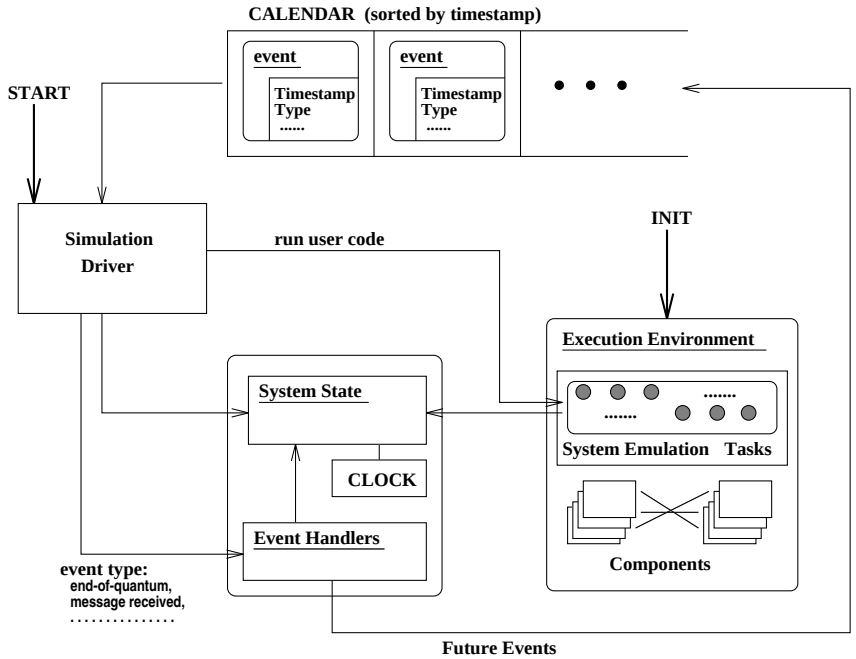
που είναι το πρώτο νήμα ελέγχου που εκτελεί κώδικα που ορίζεται από τον χρήστη, για την εξομοίωση του υπό μελέτη συστήματος. Καθώς το νήμα αυτό είναι το μοναδικό που είναι γνωστό στον οδηγό προσομοίωσης κατά την εκκίνηση του συστήματος, έχει την ευθύνη να ετοιμάσει το περιβάλλον εκτέλεσης για την προσομοίωση του υπό μελέτη συστήματος, κατασκευάζοντας κόμβους επεξεργασίας για να μοντελοποιήσει τα δομικά του στοιχεία, και συνδέοντας με τους κόμβους αυτούς νήματα ελέγχου για την εξομοίωση λειτουργιών του συστήματος. Μετά την εκκίνηση του νήματος genesis, το νήμα grim reaper αρχίζει ένα ατέρμονα βρόχο εξυπηρέτησης, στον οποίο λαμβάνει μηνύματα από νήματα ελέγχου από το δένδρο με ρίζα το νήμα genesis όταν αυτά τερματίζουν την εκτέλεσή τους, για να αποδεσμεύσει την μνήμη που δόθηκε σε αυτά δυναμικά για να χρησιμοποιηθεί ως στοίβα εκτέλεσης. Το νήμα αυτό προστατεύει λοιπόν το σύστημα από το ενδεχόμενο απώλειας μνήμης (memory leak).

5.1.5 Ο Μηχανισμός Ελέγχου της Προσομοίωσης

Ο μηχανισμός ελέγχου της προσομοίωσης (σχήμα 5.4) έχει την ευθύνη να χειριστεί τα συμβάντα που χαρακτηρίζουν την λειτουργία του συστήματος, και να συντονίσει την εκτέλεση των διαφόρων νημάτων ελέγχου. Η βασική δομή δεδομένων που διαχειρίζεται ο οδηγός προσομοίωσης είναι ο κατάλογος συμβάντων, που ονομάζεται calendar, όπου αποθηκεύονται δομές που παριστάνουν συμβάντα. Με κάθε συμβάν συνδέεται μια χρονοσφραγίδα, που καθορίζει την σχετική σειρά των συμβάντων στον κατάλογο. Ο οδηγός προσομοίωσης διατηρεί τον κατάλογο ταξινομημένο κατά αύξουσα τάξη ως προς τις χρονοσφραγίδες. Κάθε φορά που ο οδηγός προσομοίωσης παίρνει τον έλεγχο, ελέγχει την ουρά των νημάτων ελέγχου που είναι σε θέση να προχωρήσουν με την εκτέλεσή τους. Η ουρά αυτή είναι ξεχωριστή για κάθε κόμβο προσομοίωσης. Εάν δεν υπάρχει κάποιο νήμα ελέγχου έτοιμο να συνεχίσει την εκτέλεσή του, ο οδηγός προσομοίωσης αφαιρεί από τον κατάλογο συμβάντων το συμβάν με την μικρότερη χρονοσφραγίδα, και ανάλογα με τον τύπο του συμβάντος, καλείται η ρουτίνα που θα το χειριστεί. Η ρουτίνα αυτή (event handler) ενημερώνει κατάλληλα τις δομές δεδομένων που περιγράφουν την κατάσταση του περιβάλλοντος εκτέλεσης, ενδεχομένως δρομολογώντας ένα ή περισσότερα συμβάντα για επόμενες χρονικές στιγμές. Η δρομολόγηση συμβάντων γίνεται μέσω κλήσεων ρουτινών της βιβλιοθήκης, όπως οι ρουτίνες για αποστολή/λήψη μηνύματος, αλλά μπορεί να γίνει και άμεσα από τον κώδικα του χρήστη, εάν αυτός το επιθυμεί. Η δυνατότητα αυτή είναι πολύ χρήσιμη για να μοντελοποιηθούν κάποια φαινόμενα κατά την λειτουργία του συστήματος χωρίς να υπάρχει κάποιο ειδικό νήμα ελέγχου. Η μεταβολή στην κατάσταση του περιβάλλοντος εκτέλεσης μπορεί να οδηγήσει στην επιστροφή του ελέγχου σε κώδικα που προσομοιώνει την λειτουργία του συστήματος.

5.1.6 Σύνδεση με τον Μεταφραστή της Γλώσσας Προδιαγραφής

Ο μεταφραστής της γλώσσας προδιαγραφής έχει την ευθύνη (βλ. σχήμα 4.1) να προσαρμόσει το περιβάλλον εκτέλεσης ώστε να ξεκινήσει η προσομοίωση, ως εκτέλεση μιας διεργασίας με πολλαπλά νήματα ελέγχου. Μόλις επεξεργαστεί το αρχείο εισόδου με την περιγραφή του συστήματος και του φόρτου εξυπηρέτησης, ο μεταφραστής προχωρά στην εκκίνηση της προσομοίωσης (βλ. σήμα START στα σχήματα 4.1 και 5.4), ξεκινώντας το νήμα ελέγχου genesis, που για την βιβλιοθήκη υποστήριξης προσομοίωσης PARASOL είναι το πρώτο νήμα ελέγχου έξω από το επίπεδο του πυ-



Σχήμα 5.4: Λειτουργία του Μηχανισμού Προσομοίωσης.

Οταν ο οδηγός προσομοίωσης δεν βρίσκει κάποιο νήμα ελέγχου να εκτελέσει, παίρνει από τον κατάλογο των συμβάντων (calendar) το επόμενο (χρονικά) συμβάν και ενημερώνει το λογικό ρολόι της προσομοίωσης. Βάσει του τύπου του συμβάντος προσδιορίζει την ρουτίνα (event handler) που θα κληθεί για να το χειριστεί. Η ρουτίνα ενημερώνει τις δομές δεδομένων που περιγράφουν την κατάσταση του συστήματος και ενδεχομένως δρομολογεί ένα ή περισσότερα συμβάντα. Η μεταβολή στην κατάσταση του συστήματος πιθανώς να επιτρέψει στον οδηγό προσομοίωσης να δώσει τον έλεγχο σε κάποιο νήμα ελέγχου που έχει δοθεί από τον χρήστη. Η εκκίνηση του συστήματος (σήματα START, INIT) γίνεται από τον μεταφραστή της γλώσσας προδιαγραφής (σχήμα 4.1).

ρήνα της βιβλιοθήκης. Ο μεταφραστής δίνει τον έλεγχο στον οδηγό προσομοίωσης, ο οποίος, μετά την προετοιμασία ορισμένων βασικών δομών δεδομένων, όπως ο κατάλογος συμβάντων και ο πίνακας κατάστασης των νημάτων ελέγχου, δίνει τον έλεγχο στο μοναδικό ως εκείνο το σημείο νήμα ελέγχου σε επίπεδο χρήστη, το νήμα ελέγχου genesis. Αυτό το νήμα ελέγχου συνθέτει το περιβάλλον εκτέλεσης (βλ. σήμα INIT στα σχήματα 4.1 και 5.4) με αντικείμενα που υποστηρίζει η βιβλιοθήκη υποστήριξης (ουσιαστικά μοντέλα πόρων), και ξεκινά τα νήματα ελέγχου που εξομοιώνουν την λειτουργία των υποσυστημάτων του προσομοιούμενου συστήματος.

Ο πίνακας συμβόλων (symbol table) που κατασκευάζει ο μεταφραστής κατά την επεξεργασία του αρχείου εισόδου που παρέχει ο χρήστης χρησιμοποιείται στην φάση της σύνθεσης του μοντέλου προσομοίωσης ως μια αφαιρετική περιγραφή του μοντέλου. Ο πίνακας συμβόλων είναι προσπελάσιμος από όλα τα νήματα ελέγχου, καθώς αυτά μοιράζονται το πεδίο διευθύνσεων μιας διεργασίας του λειτουργικού συστήματος. Κατά την διάρκεια της προσομοίωσης, αυτή η δομή δεδομένων χρησιμοποιείται από τους κόμβους επεξεργασίας του προσομοιούμενου συστήματος ως κατάλογος συστήματος (system catalog) για τον εντοπισμό αρχείων δεδομένων και την εύρεση διευθύνσεων θυρών επικοινωνίας κατά την προσομοίωση της (κατανεμημένης) εκτέλεσης δοσοληψιών. Επίσης, η δομή αυτή χρησιμοποιείται για την καταγραφή

μετρήσεων που ο προσομοιωτής συλλέγει κατά την λειτουργία του. Αρκετές από τις μετρήσεις αυτές χρησιμοποιούνται από τις πολιτικές διαχείρισης πόρων που υλοποιούνται από τα υποσυστήματα του προσομοιωτή. Μόλις *όλα* τα νήματα ελέγχου που εξομοιώνουν υποσυστήματα του υπό μελέτη συστήματος έχουν ξεκινήσει, και έχουν προετοιμάσει τις όποιες δομές δεδομένων και μεταβλητές κατάστασης που χρειάζονται για την λειτουργία τους, το νήμα ελέγχου genesis προχωρά στην εκκίνηση του μηχανισμού που προσομοιώνει τις αφίξεις αιτήσεων εξυπηρέτησης μονάδων φόρτου, βάσει της προδιαγραφής του φόρτου που διατυπώθηκε στο αρχείο εισόδου, και στην συνέχεια αποχωρεί από την ουρά των νημάτων ελέγχου που περιμένουν να εκτελεστούν, εκτελώντας μια κλήση `suspend`. Αυτό είναι απαραίτητο ώστε τα άλλα νήματα ελέγχου, που εξομοιώνουν την λειτουργία του συστήματος, να έχουν την ευκαιρία να εκτελέσουν κώδικα. Ο συγχρονισμός για την εκκίνηση του μηχανισμού που προσομοιώνει τις αφίξεις αιτήσεων εξυπηρέτησης υλοποιείται με χρήση δύο σημαφόρων, που υλοποιούν μια δομή συγχρονισμού τύπου barrier.

5.2 Προσομοίωση Υλικού Συστήματος

Για την προσομοίωση του υλικού (hardware) συστήματος, κατά την εκκίνηση του προσομοιωτή δημιουργούνται στιγμιότυπα (instances) του τύπου κόμβων επεξεργασίας που παρέχει η βιβλιοθήκη υποστήριξης. Δημιουργείται ένα στιγμιότυπο για κάθε κόμβο επεξεργασίας στο μοντέλο συστήματος και για κάθε συσκευή αποθήκευσης δεδομένων. Με τον τρόπο αυτό μοντελοποιείται και το δίκτυο διασύνδεσης μεταξύ κάθε ομάδας κόμβων που έχει οριστεί στο μοντέλο συστήματος.

Για τους κόμβους που αντιστοιχούν σε κόμβους επεξεργασίας του μοντέλου συστήματος, η πολιτική χρονοπρογραμματισμού ορίζεται να είναι η preemptive priority με καθοριζόμενο από τον χρήστη του προσομοιωτή quantum χρονοπρογραμματισμού. Για κάθε κόμβο επεξεργασίας στο προσομοιούμενο σύστημα, τα νήματα ελέγχου που εξομοιώνουν τμήματα του συστήματος επεξεργασίας δοσοληψιών, αλλά και τα νήματα ελέγχου που εκτελούν τις μονάδες φόρτου, χρησιμοποιούν τους "πόρους" που αναλογούν στον αντίστοιχο κόμβο επεξεργασίας της βιβλιοθήκης προσομοίωσης κατά τη διάρκεια της προσομοίωσης. Η "κατανάλωση" πόρων σημειώνεται με την κλήση ρουτινών της βιβλιοθήκης υποστήριξης προσομοίωσης. Καθώς κάθε νήμα ελέγχου συνδέεται με ένα κόμβο επεξεργασίας, η κλήση ρουτινών κατανάλωσης πόρων από ένα νήμα ελέγχου που εξομοιώνει την εκτέλεση μιας μονάδας φόρτου ή την λειτουργία ενός υποσυστήματος που αποτελεί τμήμα του συστήματος επεξεργασίας δοσοληψιών έχει το ανάλογο αντίκτυπο στον χρόνο απόκρισης των μονάδων φόρτου που εξυπηρετεί το σύστημα, και λαμβάνεται υπ'όψιν στον υπολογισμό του βαθμού χρησιμοποίησης (utilization) για τον αντίστοιχο πόρο του προσομοιούμενου συστήματος.

Για τους κόμβους που αντιστοιχούν σε συσκευές περιφερειακής μνήμης, η πολιτική χρονοπρογραμματισμού ορίζεται να είναι η FCFS. Οπως εξηγείται αναλυτικότερα στην ενότητα 5.3.6, η λειτουργία μιας συσκευής περιφερειακής μνήμης εξομοιώνεται από ένα νήμα ελέγχου που καταναλώνει τους "πόρους" που αντιστοιχούν στον κόμβο που μοντελοποιεί την συσκευή. Ανάλογα, το δίκτυο διασύνδεσης για κάθε ομάδα κόμβων μοντελοποιείται ως ένας κόμβος επεξεργασίας, με την FCFS πολιτική χρονοπρογραμματισμού. Ενα νήμα ελέγχου που συνδέεται με τον κόμβο εξομοιώνει την λογική της μετάδοσης μηνυμάτων. Αυτό το νήμα ελέγχου εκτελεί έναν ατέρμονα βρόχο εξυπηρέτησης, στον οποίο λαμβάνει μηνύματα από το υποσύστημα επικοινω-

νίας κάποιου κόμβου, ελέγχει εάν ο κόμβος αυτός ανήκει στην ομάδα κόμβων που συνδέονται με το δίκτυο αυτό, και κατόπιν είτε διαβιβάζει το μήνυμα στο υποσύστημα επικοινωνίας του κόμβου προορισμού, εάν ο κόμβος ανήκει στην τοπική ομάδα κόμβων, είτε δρομολογεί το μήνυμα στο backbone δίκτυο ώστε να φτάσει τελικά στο τοπικό δίκτυο όπου ανήκει ο κόμβος προορισμού. Η ''κατανάλωση πόρων'' στον κόμβο επεξεργασίας που παριστάνει ένα τοπικό δίκτυο αντιστοιχεί στο κόστος μετάδοσης του μηνύματος.

5.3 Προσομοίωση Υποσυστημάτων Λογισμικού

Ο προσομοιωτής TPsim περιλαμβάνει τα υποσυστήματα λογισμικού που σημειώνονται στο σχήμα 4.2 (ενότητα 4.2).

5.3.1 Κατανεμημένη Εκτέλεση Δοσοληψιών

Η εκτέλεση μιας δοσοληψίας μοντελοποιείται ως μια ακολουθία από προσπελάσεις σε δεδομένα, ακολουθούμενες από κάποια ''επεξεργασία'', που προσομοιώνεται ως κατανάλωση χρόνου του επεξεργαστή. Δοσοληψίες που χρειάζονται να προσπελάσουν δεδομένα που βρίσκονται διεσπαρμένα σε περισσότερους από ένα κόμβους εκτελούνται με τρόπο παρόμοιο με αυτόν του επόπτη επεξεργασίας δοσοληψιών CICS της IBM [Kag89], βάσει ενός μηχανισμού μεταβίβασης αίτησης προσπέλασης (function request shipping). Το μοντέλο εκτέλεσης περιγράφηκε στην ενότητα 4.2.

5.3.2 Διαχείριση Δοσοληψιών

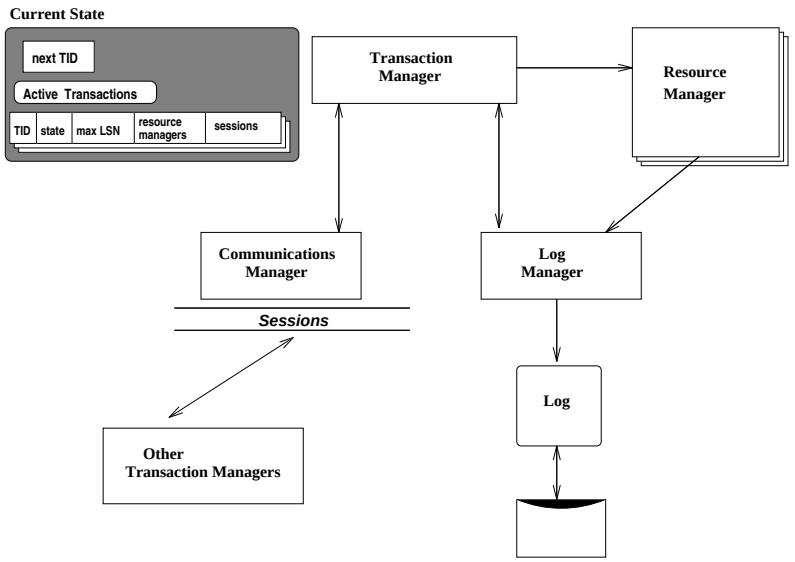
Ο διαχειριστής δοσοληψιών διατηρεί πληροφορίες για κάθε δοσοληψία που βρίσκεται σε εξέλιξη σε ένα κόμβο, όπως φαίνεται στο σχήμα 5.5. Υλοποιεί τις ιδιότητες της ατομικότητας και της μονιμότητας, συντονίζοντας τις ενέργειες των διαχειριστών πόρων του συστήματος και του μηχανισμού καταγραφής μεταβολών κατά τον τερματισμό μιας δοσοληψίας.

Η ιδιότητα της ατομικότητας εξασφαλίζεται μέσω του πρωτοκόλλου δέσμευσης two−phase commit [Gra78].

5.3.3 'Ελεγχος Ταυτόχρονης Πρόσβασης

Ο προσομοιωτής υλοποιεί το πρωτόκολλο κλειδώματος δύο φάσεων (two−phase locking) για τον έλεγχο ταυτόχρονης προσπέλασης [GLPT76, Gra78, BG81]. Σε κάθε κόμβο του προσομοιούμενου συστήματος υπάρχει ένα νήμα ελέγχου που χρησιμοποιεί τις δομές δεδομένων που απεικονίζονται στο σχήμα 5.6 (με ένα παράδειγματιγμιότυπο) για να υλοποιήσει το πρωτόκολλο ελέγχου. Για κάθε προσπέλαση, η δοσοληψία χρειάζεται να κλειδώσει τα δεδομένα που θα προσπελάσει. Ο τύπος του κλειδώματος καθορίζει εάν η δοσοληψία ζητά αποκλειστική προσπέλαση στα δεδομένα, ώστε να είναι σε θέση να τα μεταβάλλει χωρίς να παραβιαστεί η ιδιότητα της απομόνωσης, ή μη αποκλειστική προσπέλαση, ώστε να επιτρέπεται η από κοινού προσπέλαση για ανάγνωση από πολλαπλές δοσοληψίες. Για την παρούσα υλοποίηση, δεδομένα κλειδώνονται σε επίπεδο σελίδων (page granularity).

Υπάρχει το ενδεχόμενο δύο ή περισσότερες δοσοληψίες να εμπλακούν σε αδιέξοδο (deadlock). Για να αντιμετωπιστεί αυτό το ενδεχόμενο ο προσομοιωτής περιλαμβάνει



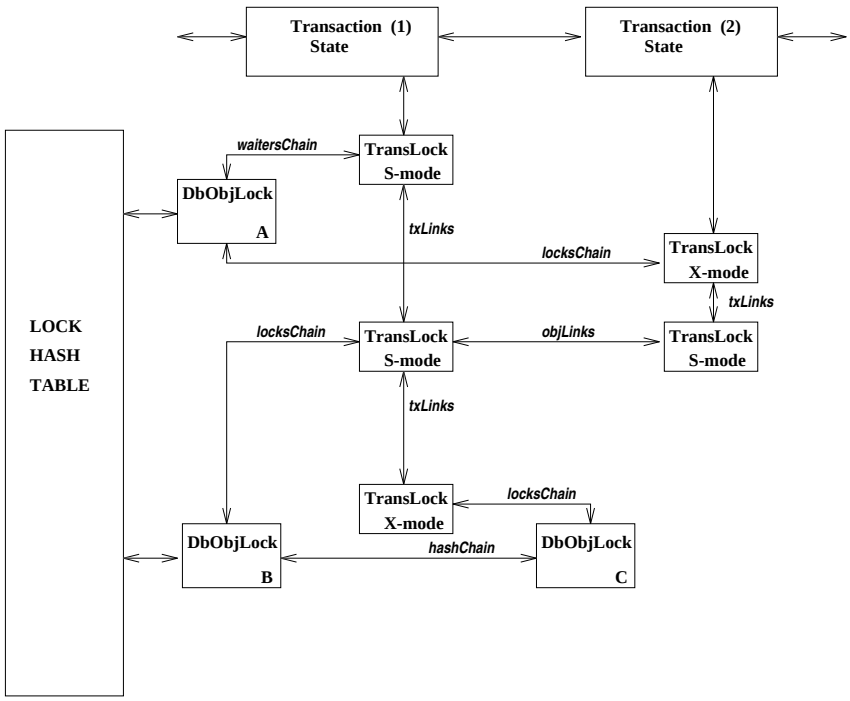
Σχήμα 5.5: Η τρέχουσα κατάσταση του διαχειριστή δοσοληψιών περιλαμβάνει την κατάσταση όλων των δοσοληψιών που βρίσκονται σε εξέλιξη στον κόμβο.

ένα υποσύστημα για την ανίχνευση και επίλυση τέτοιων βρόχων αναμονής. Ενα ειδικό νήμα ελέγχου ενεργοποιείται περιοδικά σε κάθε κόμβο για να ελέγξει εαν υπάρχει deadlock. Για τον σκοπό αυτό εξετάζει τον πίνακα lock table (βλ. σχήμα 5.6), που μπορεί να θεωρηθεί ότι παριστάνει την σχέση wait−for μεταξύ δοσοληψιών [Gra78]. Εάν υπάρχει κύκλος στον γράφο αυτής της σχέσης, τότε υπάρχει deadlock. Στην περίπτωση αυτή, το σύστημα τερματίζει την εκτέλεση μιας από τις δοσοληψίες που εμπλέκονται στον κύκλο, ώστε να αρθεί το αδιέξοδο.

5.3.4 Διαχείριση Ενταμιευτών

Ενας βασικός διαχειριστής πόρων σε ένα σύστημα διαχείρισης βάσεων δεδομένων είναι ο διαχειριστής ενταμιευτών (buffer manager) [EH84, CD85]. Τα δεδομένα αποθηκεύονται εν γένει σε μαγνητικούς δίσκους, καθώς χαρακτηρίζονται από χαμηλό κόστος αποθήκευσης ανά αποθηκευμένο byte δεδομένων, και, όντας σταθερά (non−volatile) μέσα αποθήκευσης, εξασφαλίζουν την μονιμότητα για τα δεδομένα που αποθηκεύουν. Ομως, στα σημερινά συστήματα υπολογιστών η πληροφορία που αποθηκεύεται στο δίσκο μπορεί να χρησιμοποιηθεί και να μεταβληθεί μόνο στην μνήμη. Συνεπώς, όταν μια δοσοληψία επενεργεί στα δεδομένα κάποιας βάσης αποθηκευμένης σε μαγνητικό δίσκο, τα σχετικά με την προσπέλαση δεδομένα πρέπει να μεταφερθούν στην κύρια μνήμη, και, μετά την επιτυχή περάτωση της δοσοληψίας, να γραφούν πάλι στο δίσκο. Τα συστήματα διαχείρισης βάσεων δεδομένων περιλαμβάνουν ένα σύστημα ενταμιευτών στη μνήμη του υπολογιστή, στους οποίους αποθηκεύονται προσωρινά τα δεδομένα της βάσης που μεταφέρονται από το δίσκο στη μνήμη.

Η μεταφορά δεδομένων από και προς το δίσκο γίνεται σε βασικές μονάδες που ονομάζονται σελίδες δίσκου (disk pages). Τα περιεχόμενα της βάσης δεδομένων οργανώνονται σε ομάδες ίσες με τις σελίδες δίσκου που ονομάζονται επίσης σελίδες



Σχήμα 5.6: Δομές Δεδομένων για την Υλοποίηση Ελέγχου Ταυτόχρονης Πρόσβασης.

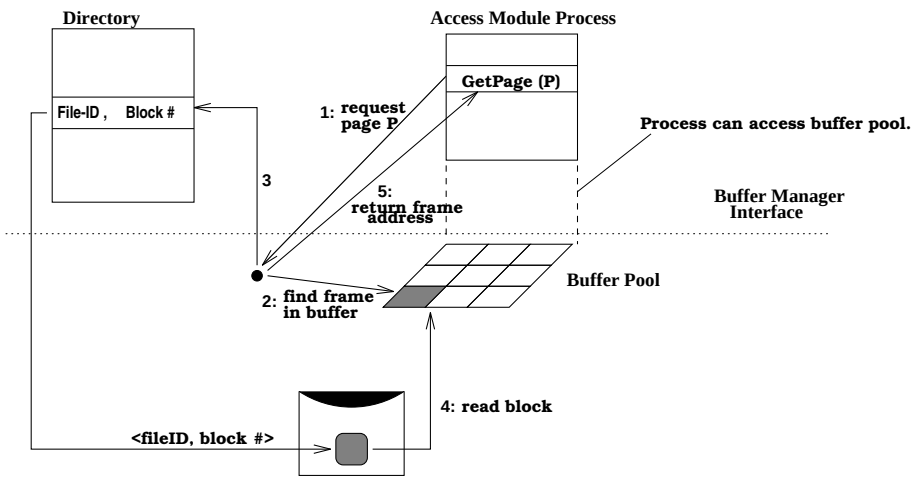
Ο προσομοιωτής διατηρεί πίνακα κατακερματισμού για τα αντικείμενα της βάσης δεδομένων για τα οποία οι δοσοληψίες που βρίσκονται σε εξέλιξη στο σύστημα έχουν ζητήσει δικαιώματα προσπέλασης. Τα αντικείμενα της βάσης δεδομένων προσδιορίζονται από το αναγνωριστικό του αρχείου δεδομένων και την διεύθυνση της αντίστοιχης σελίδας. Στο σχήμα, η δοσοληψία−1 έχει δικαίωμα ανάγνωσης για το αντικείμενο B, δικαίωμα εγγραφής για το αντικείμενο C, και περιμένει να αποκτήσει δικαίωμα εγγραφής για το αντικείμενο A. Η δοσοληψία−2 έχει δικαίωμα ανάγνωσης για το αντικείμενο B (από κοινού με την δοσοληψία−1) και δικαίωμα εγγραφής στο αντικείμενο A.

και μπορούν να μεταφερθούν από το δίσκο στη μνήμη με μια μονάχα προσπέλαση στο δίσκο. Οι ενταμιευτές των συστημάτων διαχείρισης βάσεων δεδομένων χωρίζονται σε τμήματα, κάθε ένα από τα οποία έχει μέγεθος ίσο με το μέγεθος των σελίδων δίσκου.

Ο διαχειριστής ενταμιευτών προσφέρει στους χρήστες του (που είναι κυρίως τα υποσυστήματα του συστήματος διαχείρισης δεδομένων που έχουν την ευθύνη για την προσπέλαση σε δεδομένα – access modules), μια επαφή χρήσης (interface) μέσω της οποίας πραγματοποιούνται όλες οι προσπελάσεις σε δεδομένα που βρίσκονται στους ενταμιευτές από τις διάφορες εφαρμογές του συστήματος διαχείρισης βάσεων δεδομένων. Κάθε πράξη ανάκλησης δεδομένων της βάσης ονομάζεται λογική αναφορά (logical reference). Μόλις ζητηθεί μια λογική αναφορά, ο διαχειριστής ενταμιευτών εφαρμόζει ένα αλγόριθμο αναζήτησης προκειμένου να διαπιστώσει αν η ζητούμενη σελίδα βρίσκεται σε κάποιο ενταμιευτή. Ο προσομοιωτής TPsim υλοποιεί περιοχές ενταμιευτών με χρήση καταλόγου κατακερματισμού (hash table). Το κλειδί αναζήτησης για τον κατάλογο κατακερματισμού είναι η λογική διεύθυνση της ζητούμενης σελίδας. Κατά την εκκίνηση του συστήματος καθορίζεται ως παράμετρος ο μέγιστος αριθμός σελίδων ανά περιοχή ενταμιευτών.

Το σχήμα 5.7 απεικονίζει την λειτουργία του διαχειριστή ενταμιευτών που υλο−

ποιήθηκε ως τμήμα του προσομοιωτή TPsim. Αν κατά την επεξεργασία μιας λογικής αναφοράς διαπιστωθεί ότι η ζητούμενη σελίδα δε βρίσκεται στον κατάλληλο ενταμιευτή, τότε η σελίδα αυτή πρέπει να διαβαστεί από τον δίσκο δεδομένων που κρατά την σελίδα. Ο δίσκος, και η φυσική διεύθυνση όπου αποθηκεύεται μια σελίδα εντοπίζεται μέσω του καταλόγου αρχείων (file directory). Κάθε προσπέλαση στο δίσκο ονομάζεται φυσική αναφορά (physical reference) και αποτελεί μια από τις πιο ακριβές πράξεις στα συστήματα διαχείρισης βάσεων δεδομένων. Οι φυσικές αναφορές είναι δύο ειδών: αναφορές ανάγνωσης και αναφορές εγγραφής. Οι αναφορές ανάγνωσης συμβαίνουν οποτεδήποτε η σελίδα στην οποία αντιστοιχεί μια λογική αναφορά δεν βρίσκεται στον ενταμιευτή. Οι αναφορές εγγραφής συμβαίνουν όταν έχει ολοκληρωθεί μια δοσοληψία και οι σελίδες που έχουν μεταβληθεί κατά τη διάρκειά της πρέπει να αποθηκευτούν σε σταθερή μνήμη (δηλαδή στο δίσκο), αλλά και όταν πρέπει να πραγματοποιηθεί μια αναφορά ανάγνωσης και όλα τα τμήματα του ενταμιευτή περιέχουν σελίδες που έχουν υποστεί αλλαγές. Τότε πρέπει κάποια σελίδα να γραφτεί στο δίσκο παραχωρώντας τη θέση της στον ενταμιευτή στη σελίδα που πρέπει να διαβαστεί. Ο καθορισμός της σελίδας η οποία παραχωρεί τη θέση της, όταν συμβαίνει κάποια αναφορά ανάγνωσης και όλες οι θέσεις του ενταμιευτή είναι πλήρεις, γίνεται από τον αλγόριθμο αντικατάστασης (replacement algorithm).



Σχήμα 5.7: Λειτουργία του Διαχειριστή Ενταμιευτών.

Ο προσομοιωτής TPsim υλοποιεί τον αλγόριθμο LRU (Least Recently Used), καθώς αυτός είναι ο πιο διαδεδομένος αλγόριθμος αντικατάστασης σε συστήματα διαχείρισης βάσεων δεδομένων (αν και όχι πάντα βέλτιστος [CD85]). Ο αλγόριθμος αυτός υλοποιείται διατηρώντας μια δομή διπλά συνδεδεμένης λίστας για τις σελίδες που βρίσκονται στην περιοχή ενταμιευτών, όπου η κεφαλή της λίστας αντιστοιχεί στη σελίδα που προσπελάστηκε λιγότερο πρόσφατα, και συνεπώς είναι η πρώτη υποψήφια για αντικατάσταση.

Σε κάθε κόμβο του προσομοιούμενου συστήματος, ο προσομοιωτής TPsim έχει ένα νήμα ελέγχου που λειτουργεί ως διαχειριστής ενταμιευτών, ώστε οι δοσοληψίες που εκτελούνται στο σύστημα να μπορούν να προσπελάσουν σελίδες. Ο διαχειριστής ενταμιευτών που υλοποιήθηκε μπορεί να χειριστεί πολλαπλές περιοχές ενταμιευ–

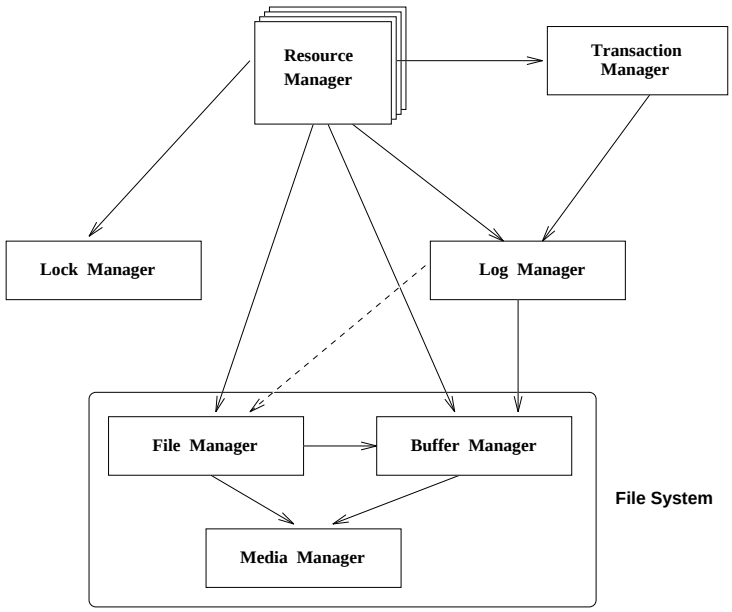
τών, γεγονός που επιτρέπει στον χρήστη του προσομοιωτή να μελετήσει τεχνικές διαχείρισης για πολλαπλές περιοχές ενταμιευτών, όπως αυτές που προτείνονται στις αναφορές [CFW+95, BCL93b].

Σύμφωνα με την ταξινόμηση της αναφοράς [EH84], η λειτουργία του διαχειριστή δοσοληψιών που υλοποιήθηκε ως τμήμα του προσομοιωτή TPsim ακολουθεί τους κανόνες STEAL και NO−FORCE. Σύμφωνα με τον κανόνα STEAL, μια σελίδα μπορεί να μεταφερθεί στον δίσκο, όταν ο αλγόριθμος αντικατάστασης το αποφασίσει, ακόμα και αν η δοσοληψία που εκτέλεσε προσπέλαση εγγραφής στην σελίδα είναι ακόμα σε εξέλιξη. Ο κανόνας αυτός συνεπάγεται ότι για την ακύρωση των αλλαγών που επέφερε μια δοσοληψία χρειάζεται ο μηχανισμός καταγραφής μεταβολών να κρατά αρκετή πληροφορία για να είναι δυνατή η ακύρωση των αλλαγών σε κάθε σελίδα που προσπελάστηκε, και ενδεχομένως να χρειάζονται προσπελάσεις στην περιφερειακή μνήμη. Ο εναλλακτικός κανόνας NO−STEAL ορίζει ότι οι σελίδες που έχουν υποστεί αλλαγές δεν επιτρέπεται να επιλεγούν για αντικατάσταση. Ο κανόνας NO−FORCE ορίζει ότι μια σελίδα μένει στην περιοχή ενταμιευτών μέχρι να αντικατασταθεί, σε αντιδιαστολή με τον FORCE που επιβάλλει κατά τον επιτυχή τερματισμό μιας δοσοληψίας να μεταφέρονται όλες οι σελίδες που αυτή τροποποίησε στην περιφερειακή μνήμη.

5.3.5 Μηχανισμός Καταγραφής Μεταβολών

Ο προσομοιωτής TPSim υλοποιεί επίσης ένα μηχανισμό καταγραφής μεταβολών. Ακολουθώντας το γενικό μοντέλο συστημάτων επεξεργασίας δοσοληψιών που παρουσιάστηκε στο κεφάλαιο 2, το μοντέλο προσομοίωσης θεωρεί ότι υπάρχει ένα υποσύστημα καταγραφής μεταβολών σε κάθε κόμβο του προσομοιούμενου συστήματος, το οποίο μπορούν να χρησιμοποιήσουν ο διαχειριστής δοσοληψιών και οι διάφοροι διαχειριστές πόρων του συστήματος, ιδίως το σύστημα διαχείρισης βάσεων δεδομένων. Οι αλληλεπιδράσεις των διαχειριστών πόρων από την σκοπιά του μηχανισμού καταγραφής μεταβολών απεικονίζονται στο σχήμα 5.8. Η επίδοση αυτού του υποσυστήματος είναι κρίσιμη για την επίδοση ολόκληρου του συστήματος.

Ο προσομοιωτής υλοποιεί το πρωτόκολλο write−ahead logging [Gra78, RH83], σύμφωνα με το οποίο κάθε φορά που μια δοσοληψία επιθυμεί να μεταβάλει ένα αντικείμενο πρέπει προηγουμένως να εισάγει στο αρχείο καταγραφής μεταβολών (log) μια εγγραφή με αρκετή πληροφορία ώστε να είναι δυνατή η ακύρωση της μεταβολής εάν τελικά η δοσοληψία δεν τερματίσει επιτυχώς, αλλά και η επανάληψη της μεταβολής εάν, λόγω βλάβης στο σύστημα, η μεταβολή, αν και η δοσοληψία έχει τερματίσει επιτυχώς, δεν έχει καταστεί μόνιμη (durable). Κάθε εγγραφή στο log περιλαμβάνει ένα μοναδικό αναγνωριστικό, που ονομάζεται LSN (log sequence number). Κάθε αντικείμενο από την βάση δεδομένων (κάθε φυσική σελίδα για τον προσομοιωτή TPsim) περιέχει το LSN για την πλέον πρόσφατη εγγραφή του log που αφορά το αντικείμενο. Το αναγνωριστικό αυτό είναι ένα ζεύγος της μορφής $< fileID, offset >$, ώστε να δίδει την σχετική θέση της αρχής της εγγραφής μέσα στο φυσικό αρχείο που υλοποιεί το log, επιτρέποντας έτσι τον εντοπισμό μιας εγγραφής για την οποία είναι γνωστό το αναγνωριστικό. Εκ κατασκευής, εάν LSN(A) είναι το αναγνωριστικό μιας εγγραφής στο log που αφορά ένα αντικείμενο από την βάση δεδομένων, και LSN(B) είναι το αναγνωριστικό για μια άλλη εγγραφή που αφορά το ίδιο αντικείμενο και έπεται χρονικά της πρώτης, τότε ισχύει ότι LSN(A) < LSN(B). Η



Σχήμα 5.8: Αλληλεπιδράσεις μεταξύ Διαχειριστών Πόρων από την σκοπιά του Διαχειριστή του Μηχανισμού Καταγραφής Μεταβολών.

Οι διαχειριστές πόρων σε σύστηματα επεξεργασίας δοσοληψιών (όπως συστήματα διαχείρισης βάσεων δεδομένων) χρησιμοποιούν, μαζί με το υποσύστημα διαχείρισης δοσοληψιών, τον μηχανισμό καταγραφής μεταβολών.

ιδιότητα αυτή [1] (“μονότονη αύξηση”) είναι ιδιαίτερα χρήσιμη για την υλοποίηση του πρωτοκόλλου write−ahead logging. Επίσης, είναι χρήσιμη κατά την επανεκκίνηση του συστήματος μετά από βλάβη, όταν πρέπει να επαναληφθούν οι αλλαγές που επέφεραν δοσοληψίες που τερμάτισαν επιτυχώς για την ανάκτηση της κατάστασης πριν την βλάβη. Στην φάση αυτή χρειάζεται, για κάθε σελίδα, να ληφθούν υπ’όψιν μόνο οι εγγραφές του log που έχουν LSN μεγαλύτερο από αυτό που σημειώνεται στην σελίδα.

Το πρωτόκολλο write−ahead logging απαιτεί από τον διαχειριστή ενταμιευτών, όταν αυτός αποφασίσει, βάσει του αλγόριθμου αντικατάστασης, να μεταφέρει μια σελίδα στην περιφερειακή μνήμη, να ζητήσει από τον διαχειριστή του log να επιβεβαιώσει ότι οι εγγραφές που αφορούν την σελίδα βρίσκονται στην περιφερειακή (μόνιμη) μνήμη. Για να γίνει αυτό, ο διαχειριστής του log φροντίζει να μεταφερθούν στην μόνιμη μνήμη όλες οι εγγραφές που έχουν αναγνωριστικό μικρότερο ή ίσο με το LSN για την σελίδα. Επίσης, το πρωτόκολλο απαιτεί κατά τον επιτυχή τερματισμό μιας δοσοληψίας ο διαχειριστής του log να επιβεβαιώσει ότι όλες οι εγγραφές με αναγνωριστικό μικρότερο ή ίσο με το LSN της εγγραφής που αντιστοιχεί στην μετάβαση της δοσοληψίας στην κατάσταση COMMITTED (βλ. σχήμα 1.2) έχουν μεταφερθεί στην μόνιμη μνήμη.

Για τον τερματισμό μιας δοσοληψίας είναι απαραίτητο οι εγγραφές που αυτή πραγματοποίησε στο αρχείο καταγραφής μεταβολών να γραφούν στην περιφερειακή μνήμη, ώστε να εξασφαλιστεί η ιδιότητα της μονιμότητας. Συνεπώς, ο ρυθμός μετάδοσης δε−

[1] Ενα λεπτό σημείο για την υλοποίηση είναι ότι, στην περίπτωση που μια δοσοληψία τερματίσει ανεπιτυχώς, κατά την ακύρωση (UNDO) των αλλαγών που αυτή επέφερε, πρέπει να γράφονται στο log εγγραφές, που ονομάζονται compensating log records. Για περισσότερες λεπτομέρειες βλ. αναφορά [MHPS92].

δομένων που μπορεί να υποστηρίξει ένας δίσκος θέτει το άνω φράγμα στον ρυθμό με τον οποίο τερματίζουν οι δοσοληψίες (throughput). Οι περισσότερες δοσοληψίες γράφουν μόνο ένα μικρό αριθμό εγγραφών, εν γένει μικρού μεγέθους, στο αρχείο καταγραφής μεταβολών, με αποτέλεσμα η τελευταία σελίδα του αρχείου να γράφεται στο δίσκο από *κάθε* δοσοληψία που γράφει στο αρχείο. Εάν κάθε δοσοληψία πρέπει να γράψει στον ίδιο δίσκο για να τερματίσει, τότε ο μέγιστος ρυθμός εξυπηρέτησης δοσοληψιών είναι ίσος με τον μέγιστο ρυθμό εξυπηρέτησης πράξεων I/O, που για την τρέχουσα γενιά είναι περίπου 30 πράξεις I/O ανά δευτερόλεπτο. Τα περισσότερα συστήματα (ξεκινώντας από το IMS [GK85]) υποστηρίζουν μια βελτιστοποίηση της διαδικασίας που επιτρέπει να γραφούν στον δίσκο οι εγγραφές του αρχείου καταγραφής μεταβολών για μια ομάδα από δοσοληψίες με *μια* πράξη I/O, ώστε το κόστος της πράξης I/O να χρεώνεται ομοιόμορφα σε μια ομάδα δοσοληψιών, αντί σε κάθε δοσοληψία ξεχωριστά. Η βελτιστοποίηση αυτή ονομάζεται group commit, και είναι απαραίτητη για να επιτευχθεί υψηλός ρυθμός εξυπηρέτησης δοσοληψιών. Γενικά, εάν G είναι το πλήθος των δοσοληψιών σε μια ομάδα, και I είναι ο ρυθμός εξυπηρέτησης πράξεων I/O για τον δίσκο καταγραφής μεταβολών, τότε ο μέγιστος ρυθμός εξυπηρέτησης δοσοληψιών είναι $I \times G$ δοσοληψίες ανά δευτερόλεπτο. Συνεπώς, η επιλογή του μεγέθους της ομάδας είναι κρίσιμη για την απόδοση του συστήματος [SJR91]. Από την μια πλευρά, το μέγεθος της ομάδας πρέπει να είναι αρκετά μεγάλο ώστε να υπάρχει σημαντικό κέρδος από την ομοιόμορφη κατανομή του κόστους της πράξης I/O στις δοσοληψίες–μέλη της ομάδας, από την άλλη όμως δεν πρέπει οι δοσοληψίες να υποχρεώνονται να περιμένουν μέχρι να σχηματιστεί μια μεγάλη ομάδα διότι αυτό επιδρά αρνητικά στην απόδοση του συστήματος.

Είναι σαφές ότι η τεχνική αυτή αποτελεί βελτιστοποίηση που αποφέρει βελτίωση της απόδοσης όταν μεγάλος αριθμός από δοσοληψίες είναι ταυτόχρονα σε εξέλιξη, ενώ όταν ο φόρτος του συστήματος είναι χαμηλός δεν μπορεί να επιφέρει κανένα κέρδος. Ο μέγιστος ρυθμός εξυπηρέτησης πράξεων I/O για τον δίσκο καταγραφής μεταβολών καθορίζει το σημείο καμπής (crossover point) για την τεχνική group commit: όταν ο ρυθμός με τον οποίο οι δοσοληψίες ζητούν να τερματίσουν είναι υψηλότερος από τον μέγιστο ρυθμό εξυπηρέτησης πράξεων I/O για τον δίσκο, η εφαρμογή της τεχνικής αυτής αυξάνει τον ρυθμό εξυπηρέτησης δοσοληψιών.

Το υποσύστημα καταγραφής μεταβολών του προσομοιωτή TPsim υποστηρίζει την τεχνική group commit. Η υλοποίηση ακολουθεί την αναφορά [D. 84]. Οι δοσοληψίες που έχουν περάσει στην φάση του τερματισμού εισάγουν εγγραφές στο τμήμα του αρχείου καταγραφών που διατηρείται στην μνήμη, και κατόπιν περιμένουν να ειδοποιηθούν ότι οι εγγραφές αυτές έχουν μεταφερθεί στον δίσκο. Μια επόμενη δοσοληψία γράφει το τμήμα αυτό στον δίσκο και η ενέργεια αυτή επιτρέπει και στις δοσοληψίες που είχαν προηγουμένως εμπλακεί στην διαδικασία να προχωρήσουν προς την περάτωσή τους. Η ακριβής χρονική στιγμή κατά την οποία θα πραγματοποιηθεί η πράξη I/O καθορίζεται από τρεις παραμέτρους:

- *group threshold*: ελάχιστος αριθμός δοσοληψιών που πρέπει να είναι σε εξέλιξη στο σύστημα για να αρχίσει να εφαρμόζεται η τεχνική group commit.
- *wait threshold*: ποσοστό των δοσοληψιών που βρίσκονται σε εξέλιξη και περιμένουν να τερματίσουν.
- *logdelay threshold*: αριθμός εγγραφών στο τμήμα του αρχείου καταγραφής μεταβολών που βρίσκεται στην μνήμη που πρέπει να συγκεντρωθούν πριν γίνει μια πράξη I/O για να μεταφερθούν στο δίσκο.

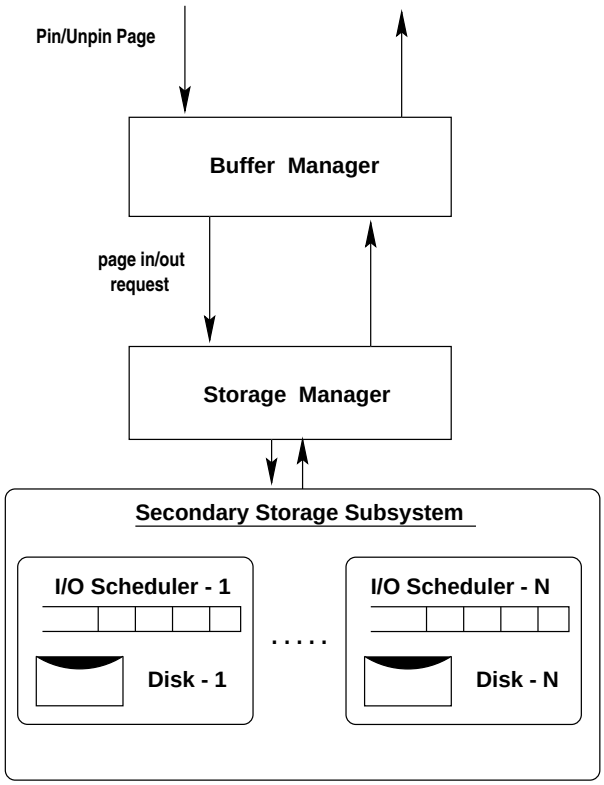
Οι παράμετροι *wait threshold* και *logdelay threshold* εκφράζουν τις συγκρουόμενες απαιτήσεις για τον σχηματισμό ομάδων δοσοληψιών με μέγεθος αρκετά μεγάλο ώστε το κέρδος από την ομοιόμορφη κατανομή του κόστους του I/O να είναι σημαντικό, αλλά αρκετά μικρό για να αποφευχθεί μείωση της απόδοσης του συστήματος. Η παράμετρος *group threshold* εκφράζει το σημείο καμπής για το μέγεθος ομάδας πέρα από το οποίο η εφαρμογή της τεχνικής group commit αυξάνει την απόδοση του συστήματος.

5.3.6 Διαχείριση Περιφερειακής Μνήμης

Για την εξομοίωση της λειτουργίας του συστήματος περιφερειακής μνήμης, ο προσομοιωτής TPsim χρησιμοποιεί μια σειρά από νήματα ελέγχου, τα οποία, μαζί με τον διαχειριστή ενταμιευτών, μοντελοποιούν την ιεραρχία μνήμης του προσομοιούμενου συστήματος. Σε κάθε κόμβο υπάρχει ένα νήμα ελέγχου (storage manager) που διαβιβάζει τις αιτήσεις για πράξεις I/O από τον διαχειριστή ενταμιευτών στο νήμα ελέγχου που μοντελοποιεί τον ελεγκτή της αντίστοιχης συσκευής περιφερειακής μνήμης. Στην παρούσα υλοποίηση, όλες οι συσκευές περιφερειακής μνήμης θεωρούνται ότι είναι μαγνητικοί δίσκοι. Το νήμα ελέγχου storage manager εντοπίζει την συσκευή περιφερειακής μνήμης που πρέπει να προσπελαστεί βάσει του σχήματος ανάθεσης σελίδων σε συσκευές περιφερειακής μνήμης που εφαρμόζεται στο προσομοιούμενο σύστημα.

Η καθυστέρηση λόγω προσπελάσεων στην περιφερειακή μνήμη αποτελεί ένα πολύ σημαντικό ποσοστό του χρόνου απόκρισης μιας δοσοληψίας. Από τους παράγοντες που επηρεάζουν την καθυστέρηση μιας προσπέλασης σε ένα δίσκο (καθυστέρηση αναζήτησης, καθυστέρηση λόγω περιστροφής, καθυστέρηση μεταφοράς δεδομένων), η καθυστέρηση αναζήτησης (seek delay) εξαρτάται άμεσα από τις προηγούμενες προσπελάσεις, που καθορίζουν την τρέχουσα θέση της κεφαλής ανάγνωσης/εγγραφής, συνεπώς μπορεί να ελεγχθεί ως ένα βαθμό από το σύστημα με τον χρονοπρογραμματισμό (scheduling) των αιτήσεων προσπέλασης που περιμένουν να εξυπηρετηθούν. Ο χρονοπρογραμματισμός ουσιαστικά καθορίζει την σειρά εξυπηρέτησης των αιτήσεων προσπέλασης. Στον προσομοιωτή TPsim υλοποιείται ο αλγόριθμος SCAN [AGM90], βάσει του οποίου οι αιτήσεις προσπέλασης διατάσσονται, βάσει της φυσικής διεύθυνσης του ζητούμενου τμήματος, κατά αύξουσα ή φθίνουσα τάξη ανάλογα με την φορά κίνησης της κεφαλής ανάγνωσης/εγγραφής. Για δοθείσα ακολουθία προσπελάσεων, ο αλγόριθμος αυτός ελαχιστοποιεί την απόσταση που διανύει συνολικά η κεφαλή ανάγνωσης/εγγραφής.

Η λειτουργία κάθε δίσκου μοντελοποιείται από ένα νήμα ελέγχου, που χειρίζεται αιτήσεις προερχόμενες από το νήμα ελέγχου που μοντελοποιεί τον ελεγκτή του δίσκου. Το νήμα αυτό εξομοιώνει την καθυστέρηση προσπέλασης βάσει της περιγραφής του τύπου του δίσκου που δόθηκε στην εκκίνηση του προσομοιωτή. Μετά την καθυστέρηση που αντιστοιχεί στην ζητούμενη προσπέλαση (βλ. ενότητα 4.3.1), ειδοποιείται ο ελεγκτής της συσκευής, ώστε να διαβιβάσει την επόμενη (αν υπάρχει) αίτηση προσπέλασης. Ο ελεγκτής επίσης, στην περίπτωση ανάγνωσης δεδομένων, ενημερώνει τον διαχειριστή ενταμιευτών. Το γεγονός ότι η εξυπηρέτηση των προσπελάσεων σε φυσικό επίπεδο γίνεται από διαφορετικό νήμα ελέγχου, αντί από το νήμα ελέγχου που μοντελοποιεί τον ελεγκτή της συσκευής, επιτρέπει στον προσομοιωτή να εξομοιώσει ρεαλιστικά την *ασύγχρονη* αλληλεπίδραση μεταξύ ελεγκτή και συσκευής.



Σχήμα 5.9: Διαχείριση Περιφερειακής Μνήμης.

Το υποσύστημα περιφερειακής μνήμης αποτελείται από ένα αριθμό από δίσκους δεδομένων. Στον προσομοιωτή TPsim υπάρχει ένα νήμα ελέγχου (storage manager) που διαβιβάζει τις αιτήσεις προσπέλασης στον κατάλληλο κάθε φορά δίσκο. Για κάθε δίσκο υπάρχει ένα νήμα ελέγχου που εξομοιώνει την λειτουργία του δίσκου στο φυσικό επίπεδο, και ένα νήμα ελέγχου που εξομοιώνει την λειτουργία του ελεγκτή του δίσκου, και τον χρονοπρογραμματισμό των προσπελάσεων.

5.3.7 Διαχείριση Ουρών

Ο προσομοιωτής TPsim περιλαμβάνει ένα υποσύστημα διαχείρισης ουρών για την ανταλλαγή δεδομένων μεταξύ δοσοληψιών (queueing system). Το υποσύστημα αυτό παρέχει μια σειρά από βασικές υπηρεσίες επικοινωνίας. Σε κάθε κόμβο του προσομοιούμενου συστήματος υπάρχει ένα νήμα ελέγχου που λειτουργεί ως διαχειριστής ουρών. Οι δοσοληψίες μπορούν να ζητήσουν από το διαχειριστή ουρών να δημιουργήσει μια νέα ουρά, να εισάγει μια νέα εγγραφή σε μια υπάρχουσα ουρά, ή να επιστρέψει μια εγγραφή αφού την αφαιρέσει από μια ουρά. Για τον διαχειριστή ουρών μια εγγραφή είναι απλά μια δομή αποτελούμενη από ένα πεδίο δεδομένων, που είναι ουσιαστικά μια σειρά από bytes, και μερικά πεδία με πληροφορίες ελέγχου. Το πεδίο δεδομένων έχει νόημα μόνο για την δοσοληψία που εισάγει την εγγραφή σε μια ουρά, και για την δοσοληψία που τελικά θα το παραλάβει. Τα πεδία ελέγχου περιλαμβάνουν ένα αναγνωριστικό για την εγγραφή που χρησιμοποιείται μόνο από τον διαχειριστή ουρών, ένα αναγνωριστικό για την εγγραφή που καθορίζεται από κάποιο πρόγραμμα εφαρμογής, και ένα δείκτη προτεραιότητας.

5.3.8 Δρομολόγηση και Χρονοπρογραμματισμός Δοσοληψιών

Ο χρόνος απόκρισης μιας δοσοληψίας, ή γενικότερα μιας μονάδας φόρτου, εξαρτάται σε μεγάλο βαθμό από την επιλογή του κόμβου που θα εκτελέσει την δοσοληψία, και τον καθορισμό της προτεραιότητας της διεργασίας ή του νήματος ελέγχου που θα εκτελέσει τον κώδικα του αντίστοιχου προγράμματος εφαρμογής. Ο χρήστης του προσομοιωτή TPsim μπορεί να καθορίσει ως παράμετρο τον αλγόριθμο δρομολόγησης δοσοληψιών, καθώς και τον αλγόριθμο χρονοπρογραμματισμού δοσοληψιών.

Ο αλγόριθμος δρομολόγησης δοσοληψιών εκτελείται, για κάθε αίτηση εξυπηρέτησης που φτάνει σε ένα κόμβο, από ένα νήμα ελέγχου που ονομάζεται "διαχειριστής φόρτου" (load manager). Αυτό το νήμα ελέγχου διατηρεί πληροφορίες για την τρέχουσα κατάσταση του συστήματος, από απόψεως φόρτου, και, με δεδομένη την κλάση φόρτου στην οποία ανήκει η δοσοληψία, επιλέγει τον κόμβο στον οποίο θα εκτελεστεί ο κώδικας που την υλοποιεί. Η παρούσα υλοποίηση του προσομοιωτή υποστηρίζει τους παρακάτω αλγορίθμους: RANDOM, Join−the−Shortest−Queue (JSQ), Minimum−Response−Time (MRT) [YBL88, YL91], καθώς και τους WFW, WFWC, SGOR [FNGD92, FNGD93]. Η επιλογή του κόμβου εκτέλεσης μιας δοσοληψίας είναι κρίσιμη για την επίδοση ενός κατανεμημένου συστήματος επεξεργασίας δοσοληψιών, καθώς καθορίζει την κατανομή του φόρτου εξυπηρέτησης στους κόμβους του συστήματος. Μια επισκόπηση θεμάτων σχετικών με την δρομολόγηση δοσοληψιών παρουσιάζεται στην αναφορά [GHK+95], καθώς και στην αναφορά [Rah92].

Ο αλγόριθμος χρονοπρογραμματισμού δοσοληψιών καθορίζει την προτεραιότητα του νήματος ελέγχου που εκτελεί τον κώδικα της δοσοληψίας. Οι αλγόριθμοι που υποστηρίζει η παρούσα υλοποίηση παρουσιάζονται λεπτομερώς στο κεφάλαιο 7.

5.4 Προσομοίωση Φόρτου Εξυπηρέτησης

Ο μηχανισμός προσομοίωσης φόρτου εξυπηρέτησης τροφοδοτεί τον προσομοιωτή με συμβάντα που αντιστοιχούν σε αιτήσεις εξυπηρέτησης. Τα υποσυστήματα του προσομοιωτή συνεργάζονται ώστε να εξομοιωθεί η παροχή υπηρεσιών σε κάθε μονάδα φόρτου. Το μοντέλο φόρτου μπορεί να είναι είτε κλειστό είτε ανοικτό, και περιλαμβάνει κλάσεις φόρτου για τις οποίες η μονάδα φόρτου είναι είτε μια δοσοληψία είτε μια σειρά από δοσοληψίες.

5.4.1 Απλές Μονάδες Φόρτου

Η προσομοίωση της εξυπηρέτησης μονάδων φόρτου ξεκινά με την λήψη μηνύματος − αίτησης εξυπηρέτησης από το υποσύστημα επικοινωνίας ενός κόμβου. Ο τρόπος με τον οποίο γεννάται αυτό το μήνυμα εξαρτάται από τον τύπο του μοντέλου φόρτου (κλειστό/ανοικτό), αλλά δεν επηρεάζει τον τρόπο εξυπηρέτησης της μονάδας φόρτου. Μια απλή μονάδα φόρτου ουσιαστικά είναι η εκτέλεση ενός προγράμματος εφαρμογής ως μια δοσοληψία. Η αίτηση εξυπηρέτησης, που περιέχει μεταξύ άλλων το αναγνωριστικό της κλάσης της δοσοληψίας, διαβιβάζεται στο υποσύστημα δρομολόγησης δοσοληψιών του κόμβου, ώστε να προσδιοριστεί ο κόμβος στον οποίο θα εκτελεστεί ο κώδικας του σχετικού προγράμματος εφαρμογής, και να καθοριστεί η προτεραιότητα της δοσοληψίας. Κατόπιν αποστέλλεται μήνυμα για την έναρξη της εκτέλεσης της δοσοληψίας στον κόμβο που επιλέχθηκε, και το μήνυμα αυτό διαβιβάζεται στο νήμα

ελέγχου dispatcher που έχει την ευθύνη να ξεκινήσει ένα νήμα ελέγχου που θα εκτελέσει τον κώδικα που υλοποιεί την δοσοληψία. Αυτό το νήμα ελέγχου θα εκτελεστεί με την καθορισμένη προτεραιότητα.

Ο dispatcher ελέγχει τον βαθμό πολυπρογραμματισμού, συνδέοντας νήματα ελέγχου με δοσοληψίας μόνο εφόσον ο αριθμός των νημάτων ελέγχου που εξυπηρετούν δοσοληψίες σε εξέλιξη στον κόμβο δεν υπερβαίνει το όριο που καθορίστηκε κατά την εκκίνηση του συστήματος, βάσει της περιγραφής του συστήματος στην γλώσσα προδιαγραφής. Με τον τρόπο αυτό υλοποιείται μιας μορφής ελέγχου φόρτου (load control). Εάν δεν είναι δυνατή η άμεση εκκίνηση νήματος ελέγχου για την εκτέλεση του προγράμματος εφαρμογής που υλοποιεί την δοσοληψία, η σχετική αίτηση εξυπηρέτησης φυλάσσεται σε μια ουρά που διαχειρίζεται το υποσύστημα διαχείρισης ουρών του κόμβου. Η ουρά αυτή διατηρείται διατεταγμένη σύμφωνα με την προτεραιότητα της δοσοληψίας που περιμένει να εκτελεστεί. Ετσι, όταν μια δοσοληψία ολοκληρώσει την εκτέλεσή της, επιτρέποντας την εκκίνηση μιας άλλης χωρίς να σημειωθεί υπέρβαση του ορίου στον βαθμό πολυπρογραμματισμού για τον κόμβο, ο dispatcher επιλέγει την δοσοληψία με την υψηλότερη προτεραιότητα.

Ο προσομοιωτής ξεκινά την εκτέλεση μιας δοσοληψίας εντοπίζοντας, σε μια δομή δεδομένων που ονομάζεται PCT (program control table), το πρόγραμμα εφαρμογής που υλοποιεί δοσοληψίες αυτής της κλάσης, ως ακολουθία προσπελάσεων σε δεδομένα. Η εκτέλεση δοσοληψιών ακολουθεί το μοντέλο που περιγράφηκε στην ενότητα 5.3.1. Ο πίνακας PCT δίδει, για την κλάση της δοσοληψίας, ένα δείκτη σε μια ρουτίνα σε γλώσσα C που υλοποιεί την δοσοληψία, με την κλήση ρουτινών που εξομοιώνουν τα βήματά της. Ο δείκτης αυτός δίδεται από τον dispatcher ως όρισμα στο νήμα ελέγχου που αναλαμβάνει να εξυπηρετήσει την δοσοληψία. Αυτός ο τρόπος εκκίνησης βασίζεται στον αντίστοιχο μηχανισμό του επόπτη επεξεργασίας δοσοληψιών CICS της IBM [Kag89].

5.4.2 Σύνθετες Μονάδες Φόρτου

Με δεδομένο τον μηχανισμό για την προσομοίωση απλών μονάδων φόρτου, η προσομοίωση σύνθετων μονάδων φόρτου διευκολύνεται σημαντικά. Για κάθε σύνθετη μονάδα φόρτου, ο μεταφραστής της γλώσσας προδιαγραφής ορίζει μια ειδική κλάση δοσοληψιών. Εάν το όνομα της κλάσης σύνθετων μονάδων φόρτου είναι `Workflow`, ο μεταφραστής ονομάζει αυτή την ειδική κλάση δοσοληψιών `Workflow:anchor`. Το πρόγραμμα εφαρμογής που σχετίζεται με την κλάση αυτή έχει την ευθύνη να υλοποιήσει την ροή δεδομένων και ελέγχου που απαιτείται για την εκτέλεση των σύνθετων μονάδων φόρτου. Με τον τρόπο αυτό διευκολύνεται επίσης η προσομοίωση της άφιξης αιτήσεων εξυπηρέτησης της κλάσης `Workflow`, καθώς αυτή ανάγεται στην προσομοίωση της άφιξης αιτήσεων εξυπηρέτησης της απλής κλάσης `Workflow:anchor` (βλ. ενότητες 5.4.3 και 5.4.4).

Οταν το υποσύστημα επικοινωνίας ενός κόμβου λάβει μια αίτηση εξυπηρέτησης για μια σύνθετη μονάδα φόρτου, ξεκινά *τοπικά*, μέσω του νήματος ελέγχου dispatcher, η εκτέλεση του αντίστοιχου προγράμματος εφαρμογής, ακριβώς όπως ξεκινά η εκτέλεση του προγράμματος εφαρμογής που υλοποιεί μια δοσοληψία (βλ. 5.4.1). Το πρόγραμμα αυτό έχει την ευθύνη να υλοποιήσει την ροή ελέγχου για την σύνθετη μονάδα φόρτου, στέλνοντας αιτήσεις εξυπηρέτησης δοσοληψίας για κάθε δοσοληψία που συγκροτεί την σύνθετη μονάδα φόρτου, καθώς και την ροή δεδομένων, ανακτώντας τα δεδομένα που αφήνει κάθε τέτοια δοσοληψία σε ουρές που διαχειρίζεται το

υποσύστημα διαχείρισης ουρών. Το σύστημα εξυπηρετεί τις αιτήσεις για εκτέλεση δοσοληψιών στα πλαίσια μιας σύνθετης μονάδας φόρτου, κάθε μια από τις οποίες δρομολογείται για εκτέλεση από το υποσύστημα δρομολόγησης δοσοληψιών του κόμβου που εκτελεί το πρόγραμμα που συντονίζει την σύνθετη μονάδα φόρτου. Οταν το τοπικό υποσύστημα επικοινωνίας λάβει μήνυμα ότι τερματίστηκε η εκτέλεση μιας δοσοληψίας που είναι ενδιάμεσο βήμα μιας σύνθετης μονάδας φόρτου, ενημερώνεται το πρόγραμμα εφαρμογής που συντονίζει την εκτέλεσή της. Το πρόγραμμα εφαρμογής με την σειρά του είτε στέλνει αίτηση εξυπηρέτησης για το επόμενο βήμα, είτε, εάν ολοκληρώθηκε η σειρά των δοσοληψιών που υλοποιεί την μονάδα φόρτου, τερματίζει. Ο προσομοιωτής μπορεί κατόπιν να υπολογίσει τον χρόνο απόκρισης για την σύνθετη μονάδα φόρτου.

5.4.3 Κλειστό Μοντέλο Φόρτου Εξυπηρέτησης

Στο κλειστό μοντέλο φόρτου εξυπηρέτησης θεωρείται ότι υπάρχει ένας αριθμός από χρήστες (τερματικά) που συνεχώς στέλνουν αιτήσεις εξυπηρέτησης, περιμένουν να λάβουν μήνυμα απάντησης, και προχωρούν στην αποστολή της επόμενης αίτησης εξυπηρέτησης, μετά από τυχαίας διάρκειας αναμονή (βλ. ενότητα 4.3.4). Μια αίτηση εξυπηρέτησης είναι ένα μήνυμα με το οποίο ένας χρήστης ζητά να εκτελεστεί ένα πρόγραμμα εφαρμογής, που με την σειρά του εκτελεί μια ή περισσότερες δοσοληψίες, ανάλογα με τον τύπο της κλάσης φόρτου στην οποία κατατάσσεται το πρόγραμμα εφαρμογής.

Μια προφανής υλοποίηση αυτού του μοντέλου φόρτου θα ήταν να αντιστοιχεί ένα νήμα ελέγχου σε κάθε χρήστη, το οποίο να εκτελεί τον κύκλο που χαρακτηρίζει τους χρήστες. Η λύση αυτή έχει όμως απαγορευτικές απαιτήσεις σε μνήμη για συστήματα με ρεαλιστικό πλήθος χρηστών, ενώ επιπλέον επιβαρύνει πολύ τον χρόνο εκτέλεσης του προσομοιωτή αυξάνοντας το ποσοστό του χρόνου που ξοδεύεται στην μεταφορά του ελέγχου από ένα νήμα ελέγχου σε άλλο (context switch overhead). Για την αποφυγή αυτού του προβλήματος ο προσομοιωτής υλοποιεί το κλειστό μοντέλο φόρτου κάνοντας χρήση της δυνατότητας που παρέχει η βιβλιοθήκη υποστήριξης προσομοίωσης για εισαγωγή συμβάντων στον κατάλογο συμβάντων που διαχειρίζεται ο οδηγός προσομοίωσης. Για την εισαγωγή είναι απαραίτητο να καθοριστεί η χρονοσφραγίδα του συμβάντος. Κατά την εκκίνηση του προσομοιωτή, το νήμα ελέγχου genesis φροντίζει να εισάγει ένα συμβάν τύπου REQUEST_ARRIVAL στον κατάλογο συμβάντων, για κάθε χρήστη στο προσομοιούμενο σύστημα. Ο προσομοιωτής περιλαμβάνει μια ρουτίνα για τον χειρισμό τέτοιου τύπου συμβάντων, η οποία καλείται από τον οδηγό προσομοίωσης όταν αυτός εξάγει το συμβάν από τον κατάλογο (βλ. 5.1.5). Η ρουτίνα αυτή αναλαμβάνει να στείλει το κατάλληλο μήνυμα – αίτηση εξυπηρέτησης – στο υποσύστημα διαχείρισης επικοινωνίας του κόμβου στον οποίο συνδέεται το τερματικό του χρήστη κατά το μοντέλο προσομοίωσης. Μετά την ολοκλήρωση της εξυπηρέτησης, το υποσύστημα αυτό αναλαμβάνει να δρομολογήσει την επόμενη άφιξη μηνύματος τύπου REQUEST_ARRIVAL, λαμβάνοντας υπ'όψιν τον μέσο χρόνο μεταξύ διαδοχικών αιτήσεων εξυπηρέτησης που έχει οριστεί ως παράμετρος για την κλάση στην οποία ανήκει ο χρήστης.

5.4.4 Ανοικτό Μοντέλο Φόρτου Εξυπηρέτησης

Στο ανοικτό μοντέλο φόρτου θεωρείται ότι υπάρχει, σε κάθε κόμβο του προσομοιούμενου συστήματος, για κάθε κλάση χρηστών που είναι ενεργή στον κόμβο, μια πηγή φόρτου που στέλνει αιτήσεις εξυπηρέτησης στο υποσύστημα διαχείρισης επικοινωνίας του κόμβου. Στο μοντέλο αυτό, η πηγή φόρτου στέλνει την επόμενη αίτηση μετά από καθυστέρηση της οποίας η μέση διάρκεια ορίζεται ως παράμετρος της κλάσης χρηστών, *χωρίς* να περιμένει να ολοκληρωθεί η εξυπηρέτηση της προηγούμενης. Το μοντέλο αυτό επιτρέπει λοιπόν στον χρήστη του προσομοιωτή να ''κατακλύσει'' το σύστημα με μονάδες φόρτου, και να μελετήσει πώς αντιμετωπίζει την κατάσταση αυτή. Για την υλοποίηση του μοντέλου χρησιμοποιείται ένα νήμα ελέγχου σε κάθε κόμβο για κάθε κλάση χρηστών που είναι ενεργή στον κόμβο ως μοντέλο της πηγής φόρτου, καθώς συνολικά ο αριθμός των πηγών φόρτου σε τέτοιου τύπου μοντέλα είναι κατά πολύ μικρότερος από τον αριθμό χρηστών σε ένα κλειστό μοντέλο.

5.4.5 Συλλογή Μετρήσεων

Στόχος κάθε μελέτης προσομοίωσης είναι η συλλογή μετρήσεων για ορισμένες μεταβλητές του μοντέλου συστήματος ώστε να σχηματιστεί μια εικόνα της λειτουργίας και επίδοσης του συστήματος υπό δοθείσες συνθήκες, ή ακόμα για να αξιολογηθούν εναλλακτικές σχεδιαστικές επιλογές για τμήματα του συστήματος. Η βιβλιοθήκη υποστήριξης προσομοίωσης παρέχει την δυνατότητα να κρατηθούν τιμές ορισμένων μεταβλητών κατά την διάρκεια της προσομοίωσης, σε μια δομή δεδομένων η οποία ενημερώνεται κάθε φορά που μεταβάλλεται η τιμή μιας μεταβλητής που ενδιαφέρει τον χρήστη του προσομοιωτή. Η ενημέρωση είναι ευθύνη του κώδικα που εξομοιώνει την λειτουργία του συστήματος. Ο προσομοιωτής TPsim αξιοποιεί αυτή τη δυνατότητα για να εκτιμήσει μεταβλητές όπως ο μέσος χρόνος απόκρισης ανά κλάση μονάδων φόρτου. Επιπλέον, η βιβλιοθήκη υποστήριξης προσομοίωσης επιτρέπει να εκτιμηθεί για κάθε κόμβο επεξεργασίας, που αντιστοιχεί σε ένα πόρο του προσομοιούμενου συστήματος, ο βαθμός χρησιμοποίησης (utilization). Ο προσομοιωτής παρέχει τέλος την δυνατότητα υπολογισμού ιστογραμμάτων για την κατανομή των χρόνων απόκρισης ανά κλάση μονάδων φόρτου.

Κεφάλαιο 6

Περιβάλλον Υποστήριξης Πειραμάτων

Στο κεφάλαιο αυτό παρουσιάζεται η μέθοδος που αναπτύχθηκε για την πραγματοποίηση πειραμάτων προσομοίωσης και την συλλογή μετρήσεων. Κάθε πείραμα αποτελείται από μια σειρά από βήματα, καθένα από τα οποία μπορεί να παράγει υψηλό όγκο δεδομένων και απαιτεί σημαντικό χρόνο για την ολοκλήρωσή του. Κάθε βήμα σε ένα πείραμα προσομοίωσης αποτελείται από μια σειρά από εκτελέσεις του προγράμματος προσομοίωσης. Ο συντονισμός των βημάτων για την διεξαγωγή ενός πειράματος και η συλλογή των σχετικών μετρήσεων είναι μια επίπονη και χρονοβόρα εργασία, συνεπώς είναι επιθυμητό να απαλλαγεί κατά το δυνατόν ο χρήστης από αυτή. Για τον σκοπό αυτό σχεδιάστηκε και υλοποιήθηκε ένα περιβάλλον υποστήριξης αυτής της πειραματικής διαδικασίας. Παρακάτω περιγράφεται λεπτομερώς η διαδικασία διεξαγωγής πειραμάτων προσομοίωσης, μέσα στα πλαίσια αυτής της εργασίας, και, ουσιαστικά, με τον τρόπο αυτό διατυπώνονται οι προδιαγραφές σχεδίασης για το περιβάλλον υποστήριξης που υλοποιήθηκε. Επίσης, δίδονται στοιχεία για την υλοποίηση του περιβάλλοντος, και περιγράφεται η διαδικασία συλλογής μετρήσεων από τον προσομοιωτή TPsim μέσα σε αυτό το περιβάλλον. Μια αναφορά σε περιβάλλοντα υποστήριξης πειραμάτων που έχουν παρουσιαστεί στην διεθνή βιβλιογραφία συμπληρώνει το κεφάλαιο.

6.1 Η Διαδικασία Διεξαγωγής Πειραμάτων Προσομοίωσης

Ενα μοντέλο προσομοίωσης για ένα σύστημα χαρακτηρίζεται γενικά από ένα αριθμό από μεταβλητές των οποίων οι τιμές καθορίζουν την στατική διαμόρφωση του συστήματος, δηλαδή τα δομικά του στοιχεία και τις μεταξύ τους σχέσεις, καθώς και την δυναμική του συμπεριφορά και επίδοση [SC81]. Κάθε πείραμα προσομοίωσης έχει ως στόχο την εκτίμηση των τιμών ορισμένων μεταβλητών για το προσομοιούμενο σύστημα, για δοθείσες συνθήκες λειτουργίας, με άλλα λόγια για δοθείσες τιμές για ορισμένες από τις μεταβλητές που χαρακτηρίζουν το σύστημα. Με αυτό το σκεπτικό, οι μεταβλητές σε ένα πείραμα μπορούν να διακριθούν σε *εξαρτημένες* και *ανεξάρτητες*, όπου ανεξάρτητες ονομάζονται οι μεταβλητές των οποίων οι τιμές καθορίζονται ως συνθήκες του πειράματος, και εξαρτημένες ονομάζονται οι μεταβλητές των οποίων οι τιμές πρέπει να εκτιμηθούν με το πείραμα. Γενικά, ένα πείραμα έχει ως στόχο να εκτιμήσει τις τιμές των εξαρτημένων μεταβλητών για ποικίλες τιμές των ανεξάρτητων μεταβλητών. Η συνήθης περίπτωση είναι η εκτίμηση των εξαρτημένων μεταβλητών για μια σειρά τιμών για *μία* από τις ανεξάρτητες μεταβλητές, διατηρώντας σταθερές

κάποιες αρχικές τιμές για τις υπόλοιπες. Με τον τρόπο αυτό είναι δυνατόν να αξιολογηθεί ποσοτικά η επίδραση του παράγοντα που παριστάνει αυτή η ανεξάρτητη μεταβλητή στην δυναμική συμπεριφορά του συστήματος και την επίδοσή του, σε σχέση με τα θεωρούμενα μέτρα επίδοσης, υπό τις συνθήκες που ορίζουν οι τιμές για τις υπόλοιπες ανεξάρτητες μεταβλητές του μοντέλου προσομοίωσης.

Ενα πείραμα προσομοίωσης μπορεί να περιγραφεί πλήρως εάν διαχωριστούν οι μεταβλητές που συνιστούν το μοντέλο του προσομοιούμενου συστήματος σε εξαρτημένες και ανεξάρτητες, και κατόπιν καθοριστούν οι τιμές, ή η σειρά τιμών, για τις ανεξάρτητες μεταβλητές. Είναι σημαντικό να επισημανθεί ότι η περιγραφή αυτή μπορεί να διατυπωθεί σε μια τυπική γλώσσα, γεγονός που αποτελεί το πρώτο βήμα για την αυτοματοποίηση της πειραματικής διαδικασίας.

Μια ενδιαφέρουσα παρατήρηση είναι ότι η διεξαγωγή ενός πειράματος προσομοίωσης είναι μια περίπτωση workflow (βλ. κεφάλαιο 7). Στα πλαίσια ενός πειράματος, μπορούν να διακριθούν *καταστάσεις* που αντιστοιχούν στα στάδια του πειράματος, και *μεταβάσεις* μεταξύ σταδίων. Οι μεταβάσεις αυτές αντιπροσωπεύουν ροή δεδομένων και ελέγχου μεταξύ των βημάτων του πειράματος. Οι καταστάσεις και οι μεταβάσεις αυτές καθορίζονται από κάποιο πειραματικό πρωτόκολλο–προδιαγραφή για το πείραμα. Σε κάθε μετάβαση, παράγονται δεδομένα σαν αποτέλεσμα της ολοκλήρωσης ενός πειραματικού βήματος.

6.2 Προδιαγραφές για το Περιβάλλον Υποστήριξης

Στην σχεδίαση του περιβάλλοντος υποστήριξης πρέπει να ληφθούν υπ'όψιν σαν προδιαγραφές λειτουργίας τα χαρακτηριστικά της διαδικασίας διεξαγωγής πειραμάτων προσομοίωσης. Αλλες απαιτήσεις προκύπτουν από την ανάγκη για καθορισμό και συντονισμό της ροής εκτέλεσης των βημάτων ενός πειράματος, το υψηλό κόστος εκτέλεσης των επιμέρους βημάτων ενός πειράματος, και την ανάγκη για συλλογή και συσχέτιση πειραματικών δεδομένων από πολλαπλά πειράματα, καθένα από τα οποία μπορεί να παράγει μεγάλο όγκο δεδομένων–μετρήσεων.

Ενα πείραμα αποσκοπεί στην παραγωγή μιας συλλογής μετρήσεων, όπου κάθε μέτρηση αποσκοπεί στην εκτίμηση των τιμών των εξαρτημένων μεταβλητών του πειράματος. Κάθε μέτρηση ουσιαστικά προσδιορίζει ένα σημείο ("data point") στον χώρο που ορίζεται από τις ανεξάρτητες και τις εξαρτημένες μεταβλητές του πειράματος. Οι μετρήσεις στα πλαίσια ενός πειράματος διαφέρουν ως προς τις τιμές των ανεξάρτητων μεταβλητών. Για κάθε μέτρηση είναι απαραίτητο, λόγω της πιθανοκρατικής φύσης του μοντέλου προσομοίωσης, να πραγματοποιηθεί ένας αριθμός από ανεξάρτητες επαναλήψεις ώστε να εξασφαλιστεί η στατιστική αξιοπιστία των μετρήσεων. Συνεπώς, το περιβάλλον υποστήριξης πρέπει να παρέχει στον χρήστη ένα μηχανισμό για την λεπτομερή περιγραφή του πειράματος. Είναι επιθυμητό η περιγραφή αυτή να δίδεται ως ένα "σενάριο" σε μια τυπική γλώσσα. Με τον τρόπο αυτό είναι δυνατή η αυτοματοποίηση της διαδικασίας της εκτέλεσης ενός πειράματος με χρήση ενός διερμηνέα εντολών (interpreter) (όπως για παράδειγμα το "κέλυφος εντολών" – command shell – στο περιβάλλον του λειτουργικού συστήματος UNIX) για την γλώσσα. Ο χρήστης θα μπορούσε να δώσει ένα αρχείο με εντολές ("script") για την εκτέλεση των βημάτων του πειράματος, και των αναγκαίων επαναλήψεών τους. Η λύση αυτή έχει το μειονέκτημα ότι πολλές παράμετροι του πειράματος πρέπει να κωδικοποιηθούν σαν ορίσματα στο πρόγραμμα προσομοίωσης. Ετσι κρίνεται σκόπιμο

η γλώσσα περιγραφής να επιτρέπει την περιγραφή του πειράματος χωρίς να είναι απαραίτητο να δοθούν ακριβώς οι εντολές που θα πρέπει να εκτελεστούν, αφήνοντας την ευθύνη για την υλοποίηση της απαραίτητης για τον συντονισμό του πειράματος ροής ελέγχου. Ο διερμηνέας οφείλει να είναι σε θέση να συντονίσει με τρόπο διαφανή για τον χρήστη την διεξαγωγή του πειράματος, εάν αυτός δώσει τις τιμές των ανεξάρτητων μεταβλητών για τις οποίες επιθυμεί να συλλέξει μετρήσεις για τις τιμές των εξαρτημένων μεταβλητών.

Μια ιδιαίτερα σημαντική απαίτηση προκύπτει από το γεγονός ότι ο χρήστης βρίσκεται συχνά στην θέση να πρέπει να τροποποιήσει τις προδιαγραφές του πειράματος (experimental protocol). Η απαίτηση αυτή καλύπτεται με την χρήση διερμηνέα για τον συντονισμό του πειράματος, μια και ο χρήστης έχει μόνο να τροποιήσει την περιγραφή του πειράματος που διατυπώνεται στην τυπική γλώσσα που δέχεται ο διερμηνέας. Στην περίπτωση αυτή, όσο απλούστερη είναι η γλώσσα περιγραφής του πειράματος, τόσο πιο εύκολα θα γίνουν οι απαραίτητες αλλαγές. Ενας επιπλέον λόγος για την χρήση τυπικής γλώσσας για την περιγραφή ενός πειράματος είναι το γεγονός ότι η περιγραφή αυτή μπορεί να χρησιμεύσει και ως τεκμηρίωση (documentation) για το πείραμα, και να αποθηκευθεί μαζί με τα παραγόμενα δεδομένα για ενδεχόμενη μελλοντική χρήση.

Επίσης, για λόγους επίδοσης είναι επιθυμητό να εκτελούνται διάφορα βήματα ενός πειράματος παράλληλα σε διαφορετικούς υπολογιστικούς κόμβους ενός δικτύου. Είναι συνεπώς επιθυμητό το περιβάλλον υποστήριξης να επιτρέπει την εκτέλεση βημάτων παράλληλα, με τρόπο κατά το δυνατόν διαφανή στον χρήστη. Επιπλέον, είναι σημαντικό το περιβάλλον υποστήριξης να επιτρέπει σε περισσότερα από ένα πειράματα να είναι ταυτόχρονα σε εξέλιξη. Η απαίτηση αυτή, που προκύπτει από την φύση της πειραματικής διαδικασίας, καθώς εν γένει ο χρήστης ενδιαφέρεται για την δυναμική συμπεριφορά και επίδοση του προσομοιούμενου συστήματος για ένα ευρύ φάσμα τιμών για τις διάφορες παραμέτρους του μοντέλου, επιβάλλει στο περιβάλλον υποστήριξης να μπορεί να διακρίνει μεταξύ των δεδομένων που παράγονται στα πλαίσια διαφορετικών πειραμάτων.

Τα πειράματα προσομοίωσης εν γένει παράγουν υψηλό όγκο δεδομένων–μετρήσεων. Τα δεδομένα αυτά αναλύονται από τους χρήστες για την εξαγωγή συμπερασμάτων σχετικά με το υπό μελέτη σύστημα, και στη συνέχεια φυλάσσονται ώστε να είναι δυνατή η αντιπαραβολή και συσχέτισή τους με δεδομένα από άλλα συναφή πειράματα. Με δεδομένο ότι κάθε πείραμα απαιτεί την συλλογή ενός αριθμού από μετρήσεις, και κάθε μέτρηση προκύπτει με την εκτέλεση ενός πλήθους από ανεξάρτητες επαναλήψεις, δεν είναι αποδεκτό να απαιτείται από τον χρήστη να συλλέξει αυτός τα δεδομένα από κάθε επανάληψη για κάθε μέτρηση. Αν και είναι δυνατόν να αναπτυχθούν βοηθήματα, κυρίως με χρήση εργαλείων για την επεξεργασία αρχείων κειμένου και συμβολοσειρών, όπως για παράδειγμα τα φίλτρα sed και awk [KP84], στο περιβάλλον του λειτουργικού συστήματος UNIX, η λύση αυτή έχει το μειονέκτημα ότι απαιτείται αρκετή προσπάθεια από μέρους του χρήστη για την ανάπτυξη των βοηθητικών προγραμμάτων. Είναι επιθυμητό η συλλογή των δεδομένων, καθώς και ο υπολογισμός στατιστικών παραμέτρων που σχετίζονται με αυτά, να γίνεται με τρόπο διαφανή στον χρήστη. Εάν το περιβάλλον υποστήριξης είναι σε θέση να επιτύχει αυτή την διαφάνεια, ο χρήστης χρειάζεται μόνο να περιγράψει το πείραμα βάσει της γλώσσας περιγραφής πειραμάτων. Το περιβάλλον υποστήριξης θα πρέπει να είναι σε θέση να διαχωρίσει τα δεδομένα που προέρχονται από διαφορετικά πειράματα, και να τα αποθηκεύει ξεχωριστά. Επιπλέον, η συγκέντρωση των δεδομένων από τα πειράματα σε ένα ‘‘κεντρικό’’ σημείο διευκολύνει την διαχείρισή τους.

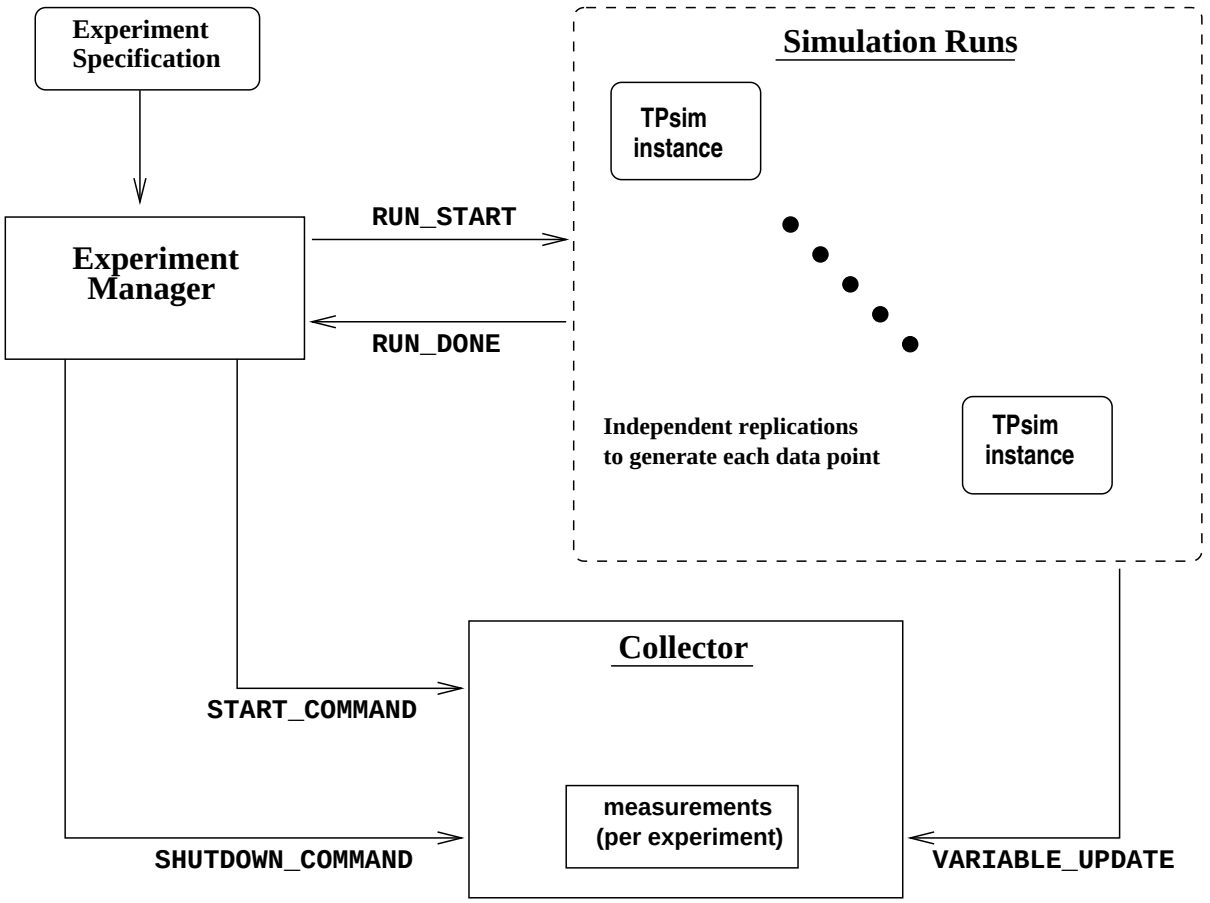
6.3 Υλοποίηση του Περιβάλλοντος Υποστήριξης

Για την εργασία αυτή οι προς εκτίμηση μεταβλητές είναι μια σειρά από μέτρα επίδοσης για το προσομοιούμενο σύστημα επεξεργασίας δοσοληψιών, για δοθείσα διαμόρφωση συστήματος και φόρτο εξυπηρέτησης. Παραδείγματα μέτρων επίδοσης που εκτιμούνται στα πειράματα στα πλαίσια αυτής της εργασίας είναι ο δείκτης επίδοσης και ο μέσος χρόνος απόκρισης για κάθε κλάση μονάδων φορτίου. Σε καθένα από τα πειράματα που παρουσιάζονται στην εργασία αυτή, εκτιμούνται οι τιμές αυτών των μέτρων για μια σειρά από τιμές για *μία* ανεξάρτητη μεταβλητή, διατηρώντας σταθερές κάποιες αρχικές τιμές για τις υπόλοιπες μεταβλητές του μοντέλου προσομοίωσης. Πάντως, η υλοποίηση του περιβάλλοντος επιτρέπει να διεξαχθούν πειράματα στα οποία σημειώνονται ταυτόχρονα μεταβολές σε περισσότερες από μία ανεξάρτητες μεταβλητές.

Το περιβάλλον υποστήριξης υλοποιείται ως μια συλλογή από διεργασίες, που επικοινωνούν μεταξύ τους κατά το πρότυπο client−server. Μια διεργασία συλλέγει τις μετρήσεις που παράγονται από διεργασίες που εκτελούν το πρόγραμμα προσομοίωσης. Για κάθε πείραμα, μια ειδική διεργασία−πελάτης λειτουργεί ως διερμηνέας για την περιγραφή του πειράματος και συντονίζει την εκτέλεση του πειράματος, ξεκινώντας διεργασίες που εκτελούν το πρόγραμμα προσομοίωσης, για τις τιμές των ανεξαρτήτων μεταβλητών που καθορίζονται από τις προδιαγραφές του πειράματος, σε διάφορους υπολογιστικούς κόμβους. Η οργάνωση του περιβάλλοντος παρουσιάζεται συνοπτικά στο σχήμα 6.1.

Ο χρήστης περιγράφει το πείραμα προσομοίωσης σύμφωνα με την γλώσσα περιγραφής πειραμάτων που δέχεται ένας διερμηνέας εντολών που αποτελεί μέρος της διεργασίας που λειτουργεί σαν διαχειριστής πειραμάτων (Experiment Manager). Βάσει της γλώσσας αυτής, καθορίζεται το μοντέλο του συστήματος που θα προσομοιωθεί, δίδονται τιμές στις μεταβλητές που απαρτίζουν το μοντέλο, και δηλώνεται ποιές είναι οι ανεξάρτητες μεταβλητές του πειράματος για τις οποίες θα υπάρξει μεταβολή τιμών κατά την διάρκεια του πειράματος, και ποιά θα είναι η σειρά τιμών για τις μεταβλητές αυτές για κάθε μέτρηση που πρέπει να συλλεχθεί στα πλαίσια του πειράματος. Επίσης, καθορίζεται το πλήθος των ανεξάρτητων επαναλήψεων για κάθε μέτρηση, καθώς και η διάρκεια προσομοίωσης (σε μονάδες χρόνου προσομοίωσης) για κάθε εκτέλεση του προγράμματος προσομοίωσης. Ο χρήστης επιπλέον δίνει και ένα αναγνωριστικό όνομα ως τίτλο για το πείραμα. Ο τίτλος αυτός είναι απαραίτητος για να διακρίνονται μεταξύ τους πειράματα που ταυτόχρονα είναι σε εξέλιξη.

Η συλλογή των μετρήσεων είναι ευθύνη μιας ξεχωριστής διεργασίας (Collector). Ενώ για κάθε πείραμα σε εξέλιξη υπάρχει ένας διαχειριστής πειράματος, υπάρχει μόνο μια διεργασία συλλογής μετρήσεων. Αυτό είναι επιθυμητό καθώς τα βήματα των διαφόρων πειραμάτων μπορεί να εκτελούνται σε περισσότερους από έναν υπολογιστικούς κόμβους, αλλά τελικά οι μετρήσεις που προκύπτουν πρέπει να συγκεντρωθούν σε ένα ''κεντρικό'' σημείο για αποθήκευση και ανάλυση. Η διεργασία συλλογής μετρήσεων είναι ενεργή σε ένα προκαθορισμένο κόμβο του δικτύου κόμβων όπου διεξάγωνται τα πειράματα, και επικοινωνεί με τις διεργασίες διαχείρισης για καθένα από τα πειράματα σε εξέλιξη μέσω sockets, που είναι ένας μηχανισμός για επικοινωνία μεταξύ διεργασιών (IPC − interprocess communication). Οι διεργασίες διαχείρισης πειραμάτων στέλνουν μηνύματα της μορφής $<$ τύπος υπηρεσίας, παράμετροι $>$ στην διεργασία συλλογής μετρήσεων. Η διεργασία συλλογής μετρήσεων αναγνωρίζει τα παρακάτω είδη μηνυμάτων για παροχή υπηρεσιών σε διεργασίες διαχείρισης πειραμάτων:



Σχήμα 6.1: Οργάνωση του Περιβάλλοντος Υποστήριξης Πειραμάτων.

Η διεργασία Experiment Manager συντονίζει την εκτέλεση του πειράματος. Η διεργασία Collector συλλέγει τις μετρήσεις που παράγονται από διαδοχικές εκτελέσεις του προγράμματος προσομοίωσης TPsim. Βήματα από περισσότερα από ένα πειράματα μπορούν να είναι ταυτόχρονα σε εξέλιξη σε διάφορους κόμβους.

α. Εναρξη πειράματος (τύπος υπηρεσίας: START_COMMAND). Με αυτό το μήνυμα μια διεργασία διαχείρισης δηλώνει στην διεργασία συλλογής μετρήσεων ότι έχει ξεκινήσει η εκτέλεση ενός νέου πειράματος, του οποίου ο τίτλος δίδεται μέσα στο μήνυμα, και ζητά να δεσμευτεί χώρος στις δομές δεδομένων που διαχειρίζεται η διεργασία συλλογής μετρήσεων για τις μετρήσεις που θα προκύψουν με την εκτέλεση των βημάτων του πειράματος. Μετά από ένα τέτοιο μήνυμα, η διεργασία συλλογής μετρήσεων είναι σε θέση να δεχθεί μηνύματα με τιμές για τις εξαρτημένες μεταβλητές του μοντέλου από τις διεργασίες που εκτελούν τα βήματα του πειράματος προσομοίωσης.

β. Τερματισμός πειράματος (τύπος υπηρεσίας: SHUTDOWN_COMMAND). Με αυτό το μήνυμα μια διεργασία διαχείρισης δηλώνει στην διεργασία συλλογής μετρήσεων ότι το πείραμα του οποίου ο τίτλος δίδεται μέσα στο μήνυμα έχει ολοκληρωθεί. Η διεργασία συλλογής μετρήσεων είναι τότε σε θέση να παράγει ένα αρχείο όπου συνοψίζονται οι μετρήσεις από το πείραμα. Επίσης, στο αρχείο

αυτό γράφονται και κάποια στατιστικά στοιχεία για τις μετρήσεις, καθώς αυτές προκύπτουν από ένα αριθμό από ανεξάρτητες επαναλήψεις. Τα στοιχεία αυτά περιλαμβάνουν την ελάχιστη, μέγιστη και μέση τιμή από τις τιμές που προέκυψαν από τις ανεξάρτητες επαναλήψεις, την τυπική απόκλιση, και τα όρια του $\alpha\%$ διαστημάτος εμπιστοσύνης, για $\alpha = 90\%, 95\%, 99\%$ [Wes76]. Οι τιμές αυτών των στατιστικών παραμέτρων είναι απαραίτητες για να εκτιμηθεί η "στατιστική σημασία" (statistical significance) των μετρήσεων. Λόγω της πιθανοκρατικής φύσης της διαδικασίας προσομοίωσης, είναι απαραίτητο να λαμβάνονται υπ'όψιν τέτοιου είδους κριτήρια ποιότητας για τις λαμβανόμενες μετρήσεις πριν αυτές χρησιμοποιηθούν για την εξαγωγή συμπερασμάτων για την δυναμική συμπεριφορά και επίδοση του συστήματος.

γ. Ενημέρωση τιμής μεταβλητής (τύπος υπηρεσίας: VARIABLE_UPDATE). Μηνύματα αυτού του τύπου στέλνονται από τις διεργασίες που ενεργοποιούνται από μια διεργασία διαχείρισης για ένα πείραμα με σκοπό την εκτέλεση των (επαναληπτικών) μετρήσεων για κάθε βήμα του πειράματος.

Η διεργασία διαχείρισης πειράματος, μετά την αποστολή μηνύματος στην διεργασία συλλογής μετρήσεων για την γνωστοποίηση της έναρξης του πειράματος (μήνυμα τύπου START_COMMAND), ξεκινά την εκτέλεση των βημάτων του πειράματος. Για τις τιμές των ανεξαρτήτων μεταβλητών του μοντέλου προσομοίωσης που αντιστοιχούν στο κάθε βήμα, η διεργασία διαχείρισης εκτελεί τον καθορισμένο αριθμό επαναλήψεων, καλώντας το πρόγραμμα προσομοίωσης, παρέχοντας τα απαραίτητα ορίσματα με τρόπο εντελώς διαφανή για τον χρήστη. Ειδικά για τον προσομοιωτή TPsim, το μοντέλο του προσομοιούμενου συστήματος καθορίζεται από ένα αρχείο με την περιγραφή του μοντέλου στην γλώσσα προδιαγραφής που υποστηρίζει ο προσομοιωτής. Η διεργασία διαχείρισης ενεργοποιεί τον προσομοιωτή TPsim δίδοντας ως όρισμα το αρχείο αυτό, αλλά και μια σειρά από ζεύγη της μορφής < μεταβλητή, τιμή > , για να καθορίσει τις τιμές των ανεξάρτητων μεταβλητών του μοντέλου. Με την ολοκλήρωση κάθε μέτρησης, η αντίστοιχη διεργασία που εκτέλεσε το πρόγραμμα προσομοίωσης για τις τιμές των ανεξάρτητων μεταβλητών που αντιστοιχούν στο σημείο της μέτρησης στέλνει στην διεργασία συλλογής μετρήσεων μηνύματα (τύπου VARIABLE_UPDATE) με τις τιμές που υπολόγισε για τις εξαρτημένες μεταβλητές του μοντέλου προσομοίωσης. Στα μηνύματα αυτά σημειώνεται και ο τίτλος του αντίστοιχου πειράματος, ώστε η διεργασία συλλογής μετρήσεων να είναι σε θέση να διαχωρίσει τις μετρήσεις από διαφορετικά πειράματα. Οι τιμές από επαναλήψεις μιας μέτρησης φυλάσσονται στις δομές δεδομένων που κρατούνται για το αντίστοιχο πείραμα, ώστε, όταν το πείραμα ολοκληρωθεί, γεγονός που σημειώνεται με την λήψη μηνύματος τερματισμού πειράματος από την διεργασία διαχείρισης που έχει την ευθύνη του συντονισμού του πειράματος, να είναι σε θέση να υπολογίσει διάφορες στατιστικές παραμέτρους για τα δεδομένα αυτά. Μετά την αποστολή των μηνυμάτων τύπου VARIABLE_UPDATE, η διεργασία που εκτέλεσε το πρόγραμμα προσομοίωσης μπορεί να τερματίσει. Η διεργασία διαχείρισης για το αντίστοιχο πείραμα ενημερώνεται για το γεγονός αυτό, ώστε να διατηρείται ενήμερη για την εξέλιξη του πειράματος. 'Οταν όλες οι διεργασίες που ενεργοποιήθηκαν από τη διεργασία διαχείρισης ενός πειράματος έχουν ολοκληρώσει την αποστολή μετρήσεων στην διεργασία συλλογής, η διεργασία διαχείρισης στέλνει μήνυμα (τύπου SHUTDOWN_COMMAND) στην διεργασία συλλογής για να δηλώσει το τέλος του πειράματος.

Αξίζει να τονισθεί στο σημείο αυτό ότι η σχεδίαση του περιβάλλοντος υποστήριξης

πειραμάτων που παρουσιάστηκε παραπάνω μπορεί να χειριστεί γενικά πειράματα προσομοίωσης, πέρα από το συγκεκριμένο πεδίο εφαρμογής του προσομοιωτή TPsim. Η διεργασία συλλογής μετρήσεων υποστηρίζει ένα γενικής εφαρμογής σύνολο υπηρεσιών (εκκίνηση πειράματος, ενημέρωση τιμής εξαρτημένης μεταβλητής, τερματισμός πειράματος) που μπορεί να χρησιμοποιηθεί για κάθε πεδίο εφαρμογών όπου απαιτείται η συλλογή μετρήσεων από προσομοίωσεις ενός συστήματος. Παρόμοια, η διεργασία διαχείρισης πειράματος, που έχει την ευθύνη του συντονισμού της διεξαγωγής του πειράματος, δεν έχει γνώση του πεδίου εφαρμογής, παρά μόνο της ροής ελέγχου που πρέπει να επιβάλλει για την παραγωγή των μετρήσεων που απαιτείται από τις προδιαγραφές του πειράματος. Μόνο το πρόγραμμα προσομοίωσης εξαρτάται (αναγκαστικά) από το πεδίο εφαρμογής. Συνεπώς, το προτεινόμενο περιβάλλον υποστήριξης πειραμάτων θα μπορούσε να χρησιμοποιηθεί για την διαχείριση πειραμάτων προσομοίωσης και σε άλλα πεδία εφαρμογής, πέρα από αυτό του προσομοιωτή TPsim που χρησιμοποιήθηκε στα πλαίσια αυτής της εργασίας, αρκεί οι προσομοιωτές αυτοί να είναι σε θέση να επικοινωνήσουν με την διεργασία συλλογής μετρήσεων, ώστε να μεταδώσουν τις μετρήσεις που συλλέγουν για τις εξαρτημένες μεταβλητές του μοντέλου.

6.4 Επισκόπηση της Σχετικής Βιβλιογραφίας

Πρέπει να τονιστεί ότι η διεκπεραίωση μιας μελέτης προσομοίωσης περιλαμβάνει πολύ περισσότερη δουλειά πέρα από την σχεδίαση και υλοποίηση του προγράμματος προσομοίωσης. Το περιβάλλον που υλοποιήθηκε για την διαχείριση των πειραμάτων που αναφέρονται στην εργασία αυτή παρέχει μηχανισμούς για την υποστήριξη των κυριοτέρων βημάτων της πειραματικής διαδικασίας που εφαρμόστηκε, αν και απέχει από το να μπορεί να χαρακτηριστεί πλήρης λύση για το πρόβλημα της διαχείρισης πειραμάτων. Είναι σημαντικό στο σημείο αυτό να δοθούν στοιχεία για σχετικές προσπάθειες που αναφέρονται στην βιβλιογραφία. Αξίζει να σημειωθεί ότι αν και το πρόβλημα της διαχείρισης πειραμάτων προσομοίωσης ανακύπτει πολύ συχνά λόγω της ευρείας χρήσης της προσομοίωσης ως μεθόδου για την μελέτη σύνθετων συστημάτων, μέχρι στιγμής μόνο αποσπασματικές λύσεις έχουν προταθεί. Η επέκταση του βασικού περιβάλλοντος υποστήριξης που υλοποιήθηκε στα πλαίσια αυτής της εργασίας κρίνεται ως μια σημαντική κατεύθυνση για περαιτέρω δουλειά.

Σχετικό με την παρούσα εργασία είναι το σύστημα *OpenSim* [MM93], το οποίο αποτελείται από μια συλλογή από εργαλεία που επιτρέπουν την διαχείριση πειραμάτων προσομοίωσης μέσω μιας γραφικής επαφής χρήσης (graphical user interface). Το σύστημα υποστηρίζει τον καθορισμό και την διεξαγωγή πειραμάτων προσομοίωσης και παρέχει μηχανισμούς για την συλλογή μετρήσεων και την γραφική τους απεικόνιση. Η γραφική επαφή χρήσης του συστήματος επιτρέπει τον ορισμό των μεταβλητών του μοντέλου προσομοίωσης, οι οποίες διακρίνονται σε μεταβλητές εισόδου (ανεξάρτητες) και μεταβλητές εξόδου (εξαρτημένες). Καθορίζοντας σειρά τιμών για μία ή περισσότερες μεταβλητές εισόδου, ο χρήστης είναι σε θέση να ορίσει τα βήματα του πειράματος. Η επαφή χρήσης διευκολύνει την κατασκευή των αρχείων εισόδου που απαιτούνται για το πρόγραμμα προσομοίωσης, και την διαχείριση των παραγόμενων αρχείων εξόδου. Ο χρήστης μπορεί να κατασκευάσει γραφήματα επιλέγοντας τις μεταβλητές οι οποίες θεωρούνται σταθερές για το πείραμα και την μεταβλητή της οποίας η τιμή μεταβάλλεται (για τον άξονα x του γραφήματος). Το σύστημα αποσπά από τα αρχεία εξόδου που προκύπτουν με την εκτέλεση των βημάτων του πειράματος

τις τιμές που απαιτούνται για το ζητούμενο γράφημα.

Ενα ιδιαίτερα σημαντικό χαρακτηριστικό του συστήματος είναι ότι επιτρέπει την κατανεμημένη εκτέλεση των βημάτων ενός πειράματος, κάνοντας χρήση του συστήματος Condor [LLM88]. Το σύστημα Condor έχει ως στόχο να ελαχιστοποιήσει τους αναξιοποίητους (idle) κύκλους CPU σε ένα δίκτυο από σταθμούς εργασίας. Η λειτουργία του συστήματος στηρίζεται σε έναν μηχανισμό παρακολούθησης του φορτίου σε κάθε κόμβο που αυτό διαχειρίζεται, και σε ένα μηχανισμό για την "μετανάστευση" (migration) διεργασιών από ένα κόμβο σε κάποιον άλλο. Το σύστημα δέχεται αιτήσεις για εκτέλεση προγραμμάτων, και για την εξυπηρέτησή τους εντοπίζει κόμβους με χαμηλό φορτίο, στους οποίους, επιπλέον, δεν έχει παρουσιαστεί δραστηριότητα από πλευράς του χρήστη–ιδιοκτήτη του κόμβου. Η εκτέλεση ενός προγράμματος σε ένα κόμβο διακόπτεται όταν γίνει αντιληπτό ότι ο χρήστης–ιδιοκτήτης του κόμβου αρχίζει να παρουσιάζει δραστηριότητα. Τότε το σύστημα λαμβάνει μια εικόνα της τρέχουσας κατάστασης της διεργασίας που εκτελεί το πρόγραμμα ("checkpoint") και αναλαμβάνει να εντοπίσει κάποιον άλλο κόμβο για να συνεχίσει την εκτέλεση του προγράμματος. Ο κόμβος θα ανακατασκευάσει την κατάσταση της διεργασίας την στιγμή της διακοπής, και θα συνεχίσει την εκτέλεση του προγράμματος. Η διαδικασία αυτή είναι διαφανής στους χρήστες. Αξίζει να σημειωθεί ότι το σύστημα OpenSim παρέχει στον χρήστη και έναν μηχανισμό για την καθορισμό προθεσμιών για την περάτωση των πειραμάτων, με χρήση μιας μεθόδου [Mut92] για την πρόβλεψη της διαθεσιμότητας κύκλων CPU βάσει μετρήσεων από το πρόσφατο παρελθόν. Με τον τρόπο αυτό, το σύστημα είναι σε θέση να δώσει τον χρήστη στοιχεία για τον αναμενόμενο χρόνο περάτωσης του πειράματος.

Η διαχείριση πειραμάτων είναι ένα πολύ σημαντικό πρόβλημα σε κάθε επιστημονικό πεδίο που στηρίζεται στο πείραμα. Στην βιβλιογραφία αναφέρονται εργασίες που αφορούν στην σχεδίαση και υλοποίηση μηχανισμών για την υποστήριξη της πειραματικής έρευνας στο πεδίο της Βιολογίας. Οι αναφορές [RS94], [GRS94] περιγράφουν τα συστήματα MapBase και LabBase για την υποστήριξη πειραμάτων στο πεδίο της Βιολογίας, στα πλαίσια του προγράμματος GENOME για την χαρτογράφηση του DNA διαφόρων οργανισμών. Ο βασικός στόχος στις προσπάθειες αυτές είναι να κατασκευαστεί ένα σύστημα που να διαχειρίζεται ολόκληρη την πειραματική διαδικασία, καθώς και τα παραγόμενα δεδομένα, ο όγκος των οποίων είναι πολύ μεγάλος. Μια επιπλέον απαίτηση είναι το σύστημα να είναι σε θέση να παρέχει δεδομένα στους χρήστες βάσει αιτήσεων που αυτοί διατυπώνουν με χρήση μιας γλώσσας επερωτήσεων. Ο πυρήνας των συστημάτων MapBase και LabBase είναι ένα οντοκεντρικό σύστημα διαχείρισης βάσεων δεδομένων.

Ενα πολύ ενδιαφέρον χαρακτηριστικό του συστήματος LabBase είναι ότι παρέχει έναν μηχανισμό για τον ορισμό ενός διαγράμματος καταστάσεων–μεταβάσεων (state transition diagram) που περιγράφει τα βήματα και τις διαδικασίες του πειράματος. Το σύστημα στη συνέχεια υλοποιεί την ροή ελέγχου που προδιαγράφεται με την μέθοδο αυτή. Το σχήμα της βάσης δεδομένων *δεν* περιλαμβάνει την περιγραφή των πειραμάτων. Η περιγραφή των πειραμάτων γίνεται σε ξεχωριστά αρχεία (scripts) και αυτό διευκολύνει την τροποποίηση του πειραματικού πρωτοκόλλου. Αυτή η σχεδιαστική επιλογή έχει όμως ως συνέπεια να μην καταγράφεται στην βάση δεδομένων η σημασιολογική σύνδεση ανάμεσα σε πειραματικά δεδομένα που έχουν εισαχθεί στην βάση και στην αντίστοιχη μετάβαση στο διάγραμμα καταστάσεων–μεταβάσεων που περιγράφει το πείραμα.

Κεφάλαιο 7

Αλγόριθμοι Χρονοπρογραμματισμού για Σύνθετες Μονάδες Φόρτου

Στο κεφάλαιο αυτό εισάγεται η έννοια της σύνθετης μονάδας φόρτου (complex unit of work) και εξετάζεται η σχέση της με την έννοια της δοσοληψίας. Γίνεται επίσης μια επισκόπιση των μηχανισμών που παρέχουν διαθέσιμα συστήματα επεξεργασίας δοσοληψιών για την υποστήριξη αυτού του τύπου μονάδων φόρτου. Στη συνέχεια παρουσιάζεται μια σειρά από αλγορίθμους χρονοπρογραμματισμού για σύνθετες μονάδες φόρτου σε συστήματα επεξεργασίας δοσοληψιών. Οι αλγόριθμοι αξιολογούνται με κριτήριο την ικανοποίηση στόχων επίδοσης. Η αξιολόγηση πραγματοποιείται με μια σειρά από πειράματα προσομοίωσης που χρησιμοποιούν τον προσομοιωτή TPsim και το περιβάλλον διαχείρισης πειραμάτων προσομοίωσης που αναπτύχθηκαν στα πλαίσια αυτής της εργασίας. Η μελέτη επίδοσης του κεφαλαίου αυτού χρησιμεύει ως μελέτη εφαρμογής του προσομοιωτή TPsim, εστιάζοντας σε ένα συγκεκριμένο πρόβλημα διαχείρισης πόρων.

7.1 Σύνθετες Μονάδες Φόρτου

Ο όρος workflow [Coa] αναφέρεται σε ένα σύνολο από σχετιζόμενες εργασίες που οργανώνονται με σκοπό να διεκπεραιωθεί μια διεργασία (business process) που σχετίζεται με την οργανωτική δομή και λειτουργία μιας επιχείρησης. Η ροή εργασίας καθορίζεται από την δομή της διεργασίας, και περιλαμβάνει προδιαγραφές για την ροή ελέγχου, με τον καθορισμό της σειράς εκτέλεσης των εργασιών και περιορισμών συγχρονισμού μεταξύ αυτών, καθώς και για την ροή δεδομένων και πληροφορίας μεταξύ εργασιών. Γενικά η ροή εργασίας περιλαμβάνει εργασίες που εκτελούνται και από ανθρώπους και από υπολογιστικά συστήματα [GHS94].

Συνεπώς, για την αυτοματοποίηση σύνθετων διεργασιών στο λειτουργικό περιβάλλον ενός οργανισμού προκύπτει η ανάγκη παροχής υποστήριξης για *σύνθετες μονάδες φόρτου*, που αποτελούνται από πολλαπλά βήματα. Τα βήματα αυτά είναι ενδεχομένως μεγάλης διάρκειας, και απαιτούν προσπέλαση σε κοινόχρηστα δεδομένα από μια ή περισσότερες βάσεις δεδομένων. Ενδέχεται να είναι απαραίτητη η αλληλεπίδραση και συνεργασία μεταξύ διαφορετικών οργανωτικών μονάδων του οργανισμού. Μια δραστηριότητα μπορεί να ξεκινήσει με ένα βήμα που στην συνέχεια ενεργοποιεί άλλα βήματα κατά τρόπο ασύγχρονο [DHL93]. Κάθε βήμα μπορεί να ζητήσει υπηρεσίες από προγράμματα εφαρμογών που εκτελούν μια ή περισσότερες δοσοληψίες, που προσπε-

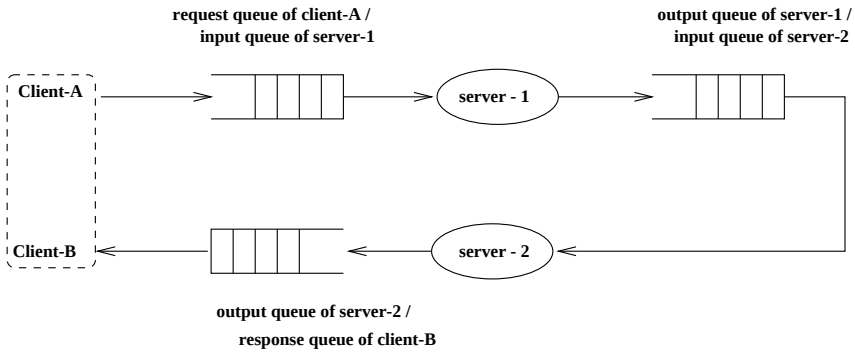
λαύνουν κοινόχρηστα δεδομένα, ενδεχομένως από πολλαπλά συστήματα διαχείρισης βάσεων δεδομένων.

Χρειάζεται συντονισμός της ροής δεδομένων και ελέγχου μεταξύ βημάτων, καθώς και συγχρονισμός όταν γίνεται προσπέλαση σε κοινόχρηστα δεδομένα. Επίσης, είναι απαραίτητος ο χειρισμός σφαλμάτων/εξαιρέσεων κατά την εκτέλεση βημάτων, ώστε να εξασφαλιστούν εγγυήσεις αξιοπιστίας για την εκτέλεση σύνθετων διεργασιών [DGMH+93, GHS94].

7.2 Υποστήριξη Σύνθετων Μονάδων Φόρτου

Ενας επόπτης επεξεργασίας δοσοληψιών παρέχει υποστήριξη για την ανάπτυξη και χρήση εφαρμογών που χειρίζονται υψηλό όγκο δεδομένων, με ταυτόχρονη προσπέλαση από πολλούς χρήστες. Οι υπάρχοντες επόπτες επεξεργασίας δοσοληψιών είναι προσανατολισμένοι στην εξυπηρέτηση "απλών" δοσοληψιών με αποτελεσματικό τρόπο. Ομως πολλές εφαρμογές οργανώνονται ως σύνολο από σχετιζόμενες δοσοληψίες, που χαρακτηρίζονται από σημαντικές αλληλεξαρτήσεις, τόσο στην ροή ελέγχου όσο και στην ροή δεδομένων [DGMH+93]. Αν και υπάρχουν αλληλεξαρτήσεις ανάμεσα στις δοσοληψίες αυτές, η μοντελοποίησή τους γίνεται έξω από το πεδίο εμβέλειας (*scope*) των συστημάτων διαχείρισης βάσεων δεδομένων και δοσοληψιών. Η ροή δεδομένων και ελέγχου είναι διασκορπισμένη στον κώδικα των προγραμμάτων εφαρμογής, και κατά συνέπεια είναι δύσκολη η κατανόηση και διαχείρισή τους, ή ακόμα είναι ευθύνη του χρήστη του συστήματος.

Τα υπάρχοντα συστήματα επεξεργασίας δοσοληψιών παρέχουν λοιπόν μόνο μερική υποστήριξη για σύνθετες μονάδες φόρτου, καθώς οι επόπτες επεξεργασίας δοσοληψιών εν γένει υποστηρίζουν ουρές για πέρασμα δεδομένων μεταξύ δοσοληψιών (βλ. σχήμα 7.1) [BHM90].



Σχήμα 7.1: Χρήση Ουρών για την Υποστήριξη Σύνθετων Μονάδων Φόρτου.

Ξεκινώντας με μια αίτηση από ένα πελάτη, κάθε δοσοληψία παράγει με τον τερματισμό της μια αίτηση εξυπηρέτησης, που αποθηκεύεται σε μια ουρά προσωρινής αποθήκευσης. Ετσι εκτελείται τελικά μια αλυσίδα από δοσοληψίες. Το αν τα αποτελέσματα της τελικής δοσοληψίας πάνε στον αρχικό πελάτη (A) ή σε κάποιο άλλο, εξαρτάται από την εφαρμογή.

Αν και αυτή η κατηγορία σύνθετων μονάδων φόρτου αποτελεί υποσύνολο μόνο της γενικής κατηγορίας μονάδων φόρτου που αναφέρεται με τον όρο workflow, παρουσιάζει ιδιαίτερο ενδιαφέρον καθώς συχνά στην πράξη εργασίες μακράς διάρκειας

διεκπεραιώνονται ως αλυσίδες δοσοληψιών [GMS87], αντί ως μια δοσοληψία μακράς διάρκειας, ώστε να αποφευχθούν αρνητικές επιπτώσεις στην επίδοση του συστήματος [Gra81]. Επιπλέον, η οργάνωση μιας εργασίας μακράς διάρκειας ως αλυσίδα δοσοληψιών είναι συμφέρουσα για την αντιμετώπιση του ενδεχομένου βλαβών, ειδικά σε κατανεμημένο περιβάλλον εκτέλεσης. Καθώς η εργασία αυτή εστιάζει στην μελέτη θεμάτων επίδοσης, αυτή η κλάση σύνθετων μονάδων φόρτου είναι πολύ σημαντική καθώς μπορεί να εμφανιστεί ως δομικό στοιχείο σε μονάδες φόρτου με πιο σύνθετη δομή.

Στην εργασία αυτή, θεωρείται ότι υπάρχουν P κλάσεις σύνθετων μονάδων φόρτου, οι WC_p, $p = 1, \ldots, P$, και κάθε σύνθετη μονάδα φόρτου της κλάσης WC_p αποτελείται από μια αλυσίδα δοσοληψιών ("βήματα") $\mathrm{T}_{p1}, \ldots, T_{p,n_p}$. Κάθε βήμα T_{pt} θεωρείται ότι έχει ταξινομηθεί (βλ. για παράδειγμα [Lab95]) σε μια κλάση δοσοληψιών C_i, $i = 1, \ldots, K$. Για κάθε κλάση σύνθετων μονάδων φόρτου ορίζεται ένας στόχος επίδοσης G_p, ως το επιθυμητό άνω όριο για τον μέσο χρόνο απόκρισης για την κλάση. Οπως τονίστηκε στο κεφάλαιο 3, οι στόχοι αυτοί είναι στατιστικής φύσης, καθώς δεν ορίζουν συγκεκριμένες προθεσμίες για ξεχωριστές μονάδες φόρτου, όπως συμβαίνει στα συστήματα πραγματικού χρόνου [AGM88].

7.3 Αλγόριθμοι Χρονοπρογραμματισμού

Οι αλγόριθμοι χρονοπρογραμματισμού που εξετάζονται ανήκουν στην κατηγορία των priority−driven, preemptive αλγορίθμων. Στα πλαίσια της εργασίας αυτής εξετάζονται εναλλακτικές μέθοδοι για τον υπολογισμό της προτεραιότητας για κάθε δοσοληψία−βήμα μιας σύνθετης μονάδας φόρτου. Θεωρείται ότι ο επεξεργαστής επιλέγει, σε κάθε quantum, για εκτέλεση την δοσοληψία με τον μεγαλύτερο δείκτη προτεραιότητας, και εξυπηρετεί δοσοληψίες που έχουν τον ίδιο δείκτη προτεραιότητας με πολιτική round−robin (κυκλικά). Τροποποιώντας δυναμικά την προτεραιότητα των δοσοληψιών που το σύστημα έχει να εκτελέσει, είναι δυνατή η υλοποίηση ευρέος φάσματος πολιτικών χρονοπρογραμματισμού. Η ιδέα αυτή έχει προταθεί και στην αναφορά [AGMK94] για την εξομοίωση πολιτικών χρονοπρογραμματισμού πραγματικού χρόνου σε standard (όχι real−time) λειτουργικά συστήματα.

Ο υπολογισμός της προτεραιότητας των δοσοληψιών έχει ιδιαίτερα σημαντική επίδραση στον χρόνο απόκρισης της δοσοληψίας, και συνεπώς της σύνθετης μονάδας φόρτου στην οποία αυτή εντάσσεται, καθώς καθορίζει τον τρόπο με τον οποίο κατανέμεται ο χρόνος του επεξεργαστή ανάμεσα στις εξελισσόμενες δοσοληψίες. Στα πλαίσια της εργασία αυτής μελετήθηκαν αλγόριθμοι που λαμβάνουν υπ'όψιν κάποιους ή και όλους τους παρακάτω παράγοντες για τον καθορισμό της προτεραιότητας δοσοληψιών:

- Απαιτήσεις σε πόρους για την εκτέλεση της δοσοληψίας. Ο παράγοντας αυτός αντικατοπτρίζεται σε μια εκτίμηση για τον αναμενόμενο χρόνο εξυπηρέτησης της δοσοληψίας, χωρίς να λαμβάνονται υπ'όψιν καθυστερήσεις λόγω αναμονής (queueing delay) για προσπέλαση σε πόρους.

- Στόχοι Επίδοσης για τις κλάσεις σύνθετων μονάδων φόρτου. Οι στόχοι αυτοί καθορίζονται στατικά, ως απαιτήσεις για τον μέσο χρόνο απόκρισης για την κλάση.

– Τρέχουσα κατάσταση του συστήματος, όσον αφορά την ικανοποίηση των στόχων επίδοσης των κλάσεων σύνθετων μονάδων φόρτου. Ο παράγοντας αυτός εκφράζεται με τις τιμές των δεικτών επίδοσης για τις κλάσεις.

– Πληροφορία ανάδρασης (feedback) σχετικά με τον χρόνο απόκρισης για προηγούμενα βήματα μιας σύνθετης μονάδας φόρτου της οποίας η εκτέλεση είναι σε εξέλιξη. Το σύστημα παρακολουθεί την επίδοσή του για κάθε δοσοληψία, και προσπαθεί να ικανοποιήσει *επιμέρους στόχους* επίδοσης κατά την εξυπηρέτηση μιας σύνθετης μονάδας φόρτου. Με βάση τον καθορισμό επιμέρους στόχων, είναι δυνατή η χρήση μηχανισμών ανάδρασης για να τροποποιηθεί η προτεραιότητα που δίδεται σε επόμενα βήματα μιας μονάδας φόρτου, με απώτερο σκοπό να μειωθεί η πιθανότητα συνολικά η κλάση φόρτου να αποκλίνει από τον προκαθορισμένο στόχο επίδοσης.

7.3.1 Επιμέρους Στόχοι Επίδοσης

Για την ικανοποίηση στόχων επίδοσης για κλάσεις μονάδων φόρτου χρειάζεται ένας μηχανισμός ανάδρασης (feedback) για την ρύθμιση παραμέτρων ελέγχου της ανάθεσης πόρων όταν μια κλάση φόρτου αποκλίνει από τον καθορισμένο γι'αυτήν στόχο επίδοσης (βλ. σχήματα 3.1 και 3.2). Για σύνθετες μονάδες φόρτου, που εκτελούνται ως μια σειρά από δοσοληψίες, η ικανοποίηση στόχων επίδοσης δυσχεραίνεται από τις εξαρτήσεις (ελέγχου και δεδομένων) μεταξύ των βημάτων. Για τον σκοπό αυτό εισάγεται η έννοια του *επιμέρους στόχου επίδοσης* ως βοήθημα για την χρήση μηχανισμών ανάδρασης.

Υποθέτοντας ότι για κάθε κλάση δοσοληψιών είναι διαθέσιμες πληροφορίες για την μέση κατανάλωση πόρων, είναι δυνατός ο *δυναμικός* καθορισμός στόχων επίδοσης για τις δοσοληψίες που συγκροτούν μια σύνθετη μονάδα φόρτου. Στην αναφορά [FNGD93], που εξετάζει τεχνικές για την ικανοποίηση στόχων για κλάσεις δοσοληψιών λαμβάνοντας υπ'όψιν πληροφορία για την μέση κατανάλωση πόρων για κάθε κλάση δοσοληψιών, θεωρείται ότι οι στόχοι επίδοσης ορίζονται στατικά. Ακολουθώντας την αναφορά [FNGD93], θεωρείται ότι για κάθε κλάση δοσοληψιών είναι γνωστός ο μέσος χρόνος χρήσης του επεξεργαστή, η μέση καθυστέρηση για προσπελάσεις σε δεδομένα, η μέση καθυστέρηση για εγγραφές στο log, και το μέσο κόστος εκτέλεσης του πρωτοκόλλου δέσμευσης, για κάθε κλάση δοσοληψιών. Με βάση τα στοιχεία αυτά, υπολογίζεται (ως άθροισμα των παραπάνω συνιστωσών) ο αναμενόμενος χρόνος εξυπηρέτησης $RT_{service}(T_{pt})$ για κάθε βήμα T_{pt}, *χωρίς* να ληφθούν υπ'όψιν καθυστερήσεις λόγω αναμονής για προσπέλαση σε πόρους. Η εκτίμηση αυτή, που αντικατοπτρίζει τις εγγενείς απαιτήσεις σε πόρους για την εκτέλεση ενός βήματος T_{pt}, χρησιμοποιείται για να δοθεί στο βήμα T_{pt} ένας στόχος επίδοσης $g(T_{pt})$, "αναλογικά" σε σχέση με τον συνολικό στόχο επίδοσης για την κλάση στην οποία ανήκει η σύνθετη μονάδα φόρτου στην οποία εντάσσεται το βήμα. Ο επιμέρους στόχος υπολογίζεται βάσει του παρακάτω τύπου:

$$g(T_{pt}) \;=\; G_p \cdot \frac{RT_{service}(T_{pt})}{\sum_{t=1}^{n_p} RT_{service}(T_{pt})}, \tag{7.1}$$

όπου T_{pt}, $t = 1, \ldots, n_p$ είναι τα βήματα που συγκροτούν μια σύνθετη μονάδα φόρτου της κλάσης WC_p. Ο υπολογισμός του επιμέρους στόχου επίδοσης για ένα βήμα είναι δυναμικός, καθώς εξαρτάται από τον συνολικό στόχο επίδοσης, με συνέπεια να είναι ο επιμέρους στόχος για μια δοσοληψία κλάσης C_i να είναι γενικά διαφορετικός ανάλογα

με την κλάση της σύνθετης μονάδας φόρτου στην οποία εντάσσεται η συγκεκριμένη δοσοληψία.

Οι επιμέρους στόχοι για τα βήματα χρησιμοποιούνται από κάποιους από τους αλγορίθμους που προτείνονται στην εργασία αυτή ως σημεία αναφοράς, ώστε το σύστημα να έχει την δυνατότητα να επέμβει διορθωτικά όταν, σε κάποιο σημείο ελέγχου, βρεθεί ότι οι επιμέρους στόχοι δεν ικανοποιούνται, με συνέπεια να είναι πολύ πιθανή η απόκλιση από τους στόχους επίδοσης για τις κλάσεις σύνθετων μονάδων φόρτου που ενδιαφέρουν τους χρήστες του συστήματος.

7.3.2 Περιγραφή των Αλγόριθμων

Στα πλαίσια της εργασίας αυτής μελετήθηκαν οι παρακάτω αλγόριθμοι, ως συνέχεια της εργασίας [MN95]:

- RR: Round−robin εξυπηρέτηση δοσοληψιών (όλες οι δοσοληψίες έχουν ίση προτεραιότητα). Η πολιτική αυτή δεν λαμβάνει υπ'όψιν καμία πληροφορία για το workflow ή για τα επιμέρους βήματά του.

- STATIC: Η προτεραιότητα ενός βήματος T_{pt} ενός workflow της κλάσης WC_p ορίζεται να είναι ίση με τον στόχο επίδοσης G_p για την κλάση. Η πολιτική αυτή δεν κάνει χρήση πληροφορίας για τις απαιτήσεις πόρων των επιμέρους βημάτων ενός workflow, λαμβάνοντας υπ'όψιν μόνο τον προκαθορισμένο για την κλάση στόχο επίδοσης. Η πολιτική αυτή σαφώς ευνοεί μονάδες φόρτου με χαμηλό στόχο επίδοσης, αντίστοιχα με την πολιτική earliest−deadline−first για συστήματα πραγματικού χρόνου [AGM88].

- rSLACK: Η προτεραιότητα ενός βήματος t ενός workflow κλάσης WC_p ορίζεται να είναι ίση με $G_p - \sum_{m=t+1}^{n_p} g(T_{pm})$. Με άλλα λόγια, η προτεραιότητα ορίζεται να είναι ίση με την διαφορά ανάμεσα στον στόχο επίδοσης για την κλάση και το άθροισμα των επιμέρους στόχων επίδοσης για τα εναπομείναντα βήματα του workflow. Συνεπώς, ο αλγόριθμος αυτός δίδει διαδοχικά αυξανόμενη προτεραιότητα σε διαδοχικά βήματα ενός workflow.

- PI: Η προτεραιότητα ενός βήματος ενός workflow της κλάσης WC_p ορίζεται να είναι ίση με την τρέχουσα τιμή του δείκτη επίδοσης PI_p της κλάσης [FNGD93]. Ο δείκτης επίδοσης υπολογίζεται βάσει της παρακάτω σχέσης (βλ. και εξίσωση 3.1) :

$$PI_p \;=\; \frac{R_p}{G_p}, \tag{7.2}$$

όπου R_p είναι η τρέχουσα εκτίμηση του μέσου χρόνου απόκρισης για την κλάση WC_p. Η εκτίμηση αυτή τροποποιείται κάθε φορά που ένα workflow T της κλάσης WC_p περατώνεται ως εξής:

$$R_p \;\leftarrow\; (1-\alpha)\cdot R_p \;+\; \alpha \cdot R(T), \tag{7.3}$$

όπου $RT(T)$ είναι ο χρόνος απόκρισης για το workflow T, και α είναι μια σταθερά ($0 \leq \alpha \leq 1$) που καθορίζει την σχετική βαρύτητα των πρόσφατων μετρήσεων του χρόνου απόκρισης έναντι μετρήσεων του χρόνου απόκρισης για μονάδες φόρτου που περατώθηκαν στο παρελθόν. Η βαρύτητα μιας μέτρησης του χρόνου απόκρισης φθίνει εκθετικά με τον χρόνο στην εκτίμηση του μέσου χρόνου

απόκρισης (κανόνας *exponential decay*). Για τα πειράματα που αναφέρονται στην εργασία αυτή χρησιμοποιήθηκε η τιμή $\alpha = 0.8$. Χρησιμοποιείται λοιπόν μια εκτίμηση του μέσου χρόνου απόκρισης που λαμβάνει υπ'όψιν την πρόσφατη ιστορία του συστήματος, αντί για τον άμεσα μετρούμενο χρόνο απόκρισης, ώστε να αποσβεστεί η επίδραση παροδικών μεταβολών στην ευσταθή κατάσταση (steady−state) του συστήματος. Η πολιτική αυτή συνεπώς είναι δυναμικής φύσης, καθώς λαμβάνει υπ'όψιν την κατάσταση του συστήματος ως προς την ικανοποίηση των στόχων επίδοσης, δείχνοντας εύνοια στις κλάσεις που κατά την χρονική στιγμή της ενεργοποίησης του αλγορίθμου είναι οι πιο πιθανές να αποκλίνουν από τον καθορισμένο στόχο επίδοσης. Η πολιτική αυτή ουσιαστικά στηρίζεται σε ένα μηχανισμό ανάδρασης, αφού ο χρόνος απόκρισης για workflows που έχουν τερματίσει καθορίζει την τρέχουσα τιμή του δείκτη επίδοσης, και συνεπώς την προτεραιότητα για τα επόμενα workflows.

- WPI: Η προτεραιότητα ενός βήματος t ενός workflow κλάσης WC_p ορίζεται να είναι $w_t \cdot PI_p$, όπου w_t είναι ένας συντελεστής που υπολογίζεται σύμφωνα με τον ψευδοκώδικα του σχήματος 7.2, όπου PI_p είναι η τρέχουσα τιμή του δείκτη επίδοσης για την κλάση. Ο συντελεστής βάρους w_t επιτρέπει την αναπροσαρμογή της προτεραιότητας που δίδεται στις δοσοληψίες−βήματα ενός workflow, καθώς η εκτέλεση του workflow προχωρά και μετά από κάθε βήμα το σύστημα έχει όλο και πιο ισχυρές ενδείξεις για το αν ο συνολικός χρόνος απόκρισης θα είναι κάτω από το όριο−στόχο για την κλάση. Ο υπολογισμός της προτεραιότητας λαμβάνει υπ'όψιν την τρέχουσα τιμή του δείκτη επίδοσης ώστε να συμπεριληφθεί στον υπολογισμό η γνώση για την τρέχουσα κατάσταση ως προς την ικανοποίηση του στόχου επίδοσης για την κλάση. Οι συντελεστές w_t εξαρτώνται από τις απαιτήσεις πόρων των βημάτων ενός workflow, καθώς στον υπολογισμό τους λαμβάνονται υπ'όψιν οι επιμέρους στόχοι επίδοσης για τα βήματα. Ο ψευδοκώδικας του σχήματος 7.2 δίδει ουσιαστικά ένα μηχανισμό ανάδρασης για τον κατά το δυνατόν έλεγχο πάνω στον ρυθμό εξυπηρέτησης ενός workflow. Η συνάρτηση $F(.)$ που εμφανίζεται στον ψευδοκώδικα του σχήματος 7.2 καθορίζει την μεταβολή των συντελεστών βάρους όταν ο χρόνος απόκρισης για ένα βήμα του workflow ξεπεράσει τον επιμέρους στόχο επίδοσης για το βήμα. Η διαίσθηση πίσω από τον ορισμό της $F(.)$ που χρησιμοποιήθηκε είναι ότι, όταν για ένα βήμα δεν ικανοποιηθεί ο επιμέρους στόχος επίδοσης, η ικανοποίηση του στόχου επίδοσης για ολόκληρο το workflow γίνεται πιο δύσκολη, επομένως έχει νόημα να "επιταχυνθεί" αναλογικά το επόμενη στη σειρά εκτέλεσης βήμα, ώστε να καλυφθεί κατά το δυνατόν η καθυστέρηση του προηγούμενου.
- WT: Η πολιτική αυτή αποτελεί απλοποίηση της WPI. Η προτεραιότητα ενός βήματος t ενός workflow της κλάσης WC_p ορίζεται να είναι $w_t \cdot g(t)$. Η πολιτική αυτή δεν λαμβάνει υπ'όψιν την τρέχουσα κατάσταση του συστήματος ως προς την ικανοποίηση των στόχων επίδοσης.

7.4 Πειραματική Αξιολόγηση των Αλγορίθμων

Το μοντέλο του συστήματος που προσομοιώθηκε στην γλώσσα προδιαγραφής που δέχεται ο προσομοιωτής TPsim δίδεται στο παράρτημα Β, ενώ στο παράρτημα Γ δίδεται ενδεικτικά η προδιαγραφή ενός από τα πειράματα που διεξήχθησαν για την

t := 1 ;
$w_m := \frac{g(T_{pm})}{\sum_{m=1}^{n_p} g(T_{pm})},\ for\ m = 1, \ldots, n_p$;
while (t $\leq n_p$) {

Initiate−Transaction(T_{pt}, service−node, $prio_{pt}$);
/* Assign priority index $prio_{pt}$ to the transaction initiation request for T_{pt}.
Use $prio_{pt}$ as the priority for task T_{pt}, relay request to service−node */
Wait−for−Transaction−Completion(T_{pt}) ;

/* Wait for service−node to send notification of the completion of T_{pt}.
Service−node also sends the actual response time, r_{pt}, of T_{pt}. */
$w_m := \frac{g(T_{pm})}{\sum_{m=t+1}^{n_p} g(T_{pm})},\ for\ m = t+1, \ldots, n_p$;

if($r_{pt} > g(T_{pt})$) {

/* T_{pt} missed its response time goal */
$w_{t+1} := w_{t+1} + \sum_{m=t+2}^{n_p} F(w_m, r_{pt}, g(T_{pt}))$;
$w_m := w_m - F(w_m, r_{pt}, g(T_{pt}))$,

for $m = t+2, \ldots, n_p$;
}
t := t + 1 ;

}

$$F(w_m, r_{pt}, g(T_{pt})) := w_m \cdot \frac{r_{pt} - g(T_{pt})}{g(T_{pt})}.$$

Σχήμα 7.2: Δυναμική Αναπροσαρμογή των Συντελεστών Βάρους w_t

αξιολόγηση των αλγορίθμων στην γλώσσα που δέχεται ο διερμηνέας της διεργασίας διαχείρισης πειραμάτων.

7.4.1 Παράμετροι Μοντέλου Προσομοίωσης

Το σύστημα επεξεργασίας δοσοληψιών αποτελείται από δύο κόμβους, από τους οποίους ο ένας λειτουργεί ως front−end. Ο άλλος κόμβος λειτουργεί ως back−end, εκτελώντας τις σύνθετες μονάδες φόρτου που υποβάλλονται στον front−end κόμβο για εξυπηρέτηση από ένα αριθμό από τερματικά, καθώς χρησιμοποιείται το κλειστό μοντέλο φόρτου εξυπηρέτησης. Ο back−end κόμβος έχει ένα επεξεργαστή των 50 MIPS, και εφαρμόζει preemptive priority scheduling, με round−robin εξυπηρέτηση μεταξύ μονάδων εκτέλεσης με την ίδια προτεραιότητα. Το quantum του επεξεργαστή έχει διάρκεια 10 msec. Ο μέγιστος βαθμός πολυπρογραμματισμού (MPL) είναι 200. Ο ρυθμός μετάδοσης πακέτων (μεγέθους 1024 bytes) μεταξύ του front−end και του back−end είναι 4 MBit/sec. Η βάση δεδομένων αποτελείται από 50,000 σελίδες, που κατανέμονται με την round−robin μέθοδο σε δύο δίσκους δεδομένων. Το log διατηρείται σε ξεχωριστό δίσκο. Οι δίσκοι είναι πανομοιότυποι, με τις παραμέτρους που δίδονται στο παράρτημα Β, που δίδουν μέσο χρόνο προσπέλασης περίπου ίσο με 10 msec. Ο κόμβος διαθέτει ενταμιευτή δεδομένων μεγέθους 20,000 σελίδων (40% της βάσης δεδομένων). Θεωρείται ότι 80% των προσπελάσεων είναι για εγγραφή. Η κατανομή των αναφορών στις σελίδες θεωρείται ότι ακολουθεί τον κανόνα 80−20:

80% των προσπελάσεων αφορούν 20% των δεδομένων. Το μέγεθος του ενταμιευτή δεδομένων είναι αρκετό ώστε να αξιοποιηθεί στο έπακρο η τοπικότητα των αναφορών που χαρακτηρίζει τις προσπελάσεις.

Με τις παραμέτρους αυτές, το προσομοιούμενο σύστημα δεν εμφανίζει bottleneck. Ετσι τα παρακάτω πειράματα εστιάζουν ακριβώς στην επίδραση που έχει ο αλγόριθμος χρονοπρογραμματισμού δοσοληψιών στην επίδοση του συστήματος. Για τα πειράματα αυτά, ο βαθμός χρησιμοποίησης (utilization) των δίσκων δεδομένων είναι γύρω στο 50%, ενώ ο βαθμός χρησιμοποίησης του επεξεργαστή είναι γύρω στο 90–95%.

Υπάρχουν δύο κλάσεις σύνθετων μονάδων φόρτου, $WC1$ και $WC2$, με ίση συχνότητα άφιξης, και τέσσερεις διαφορετικές κλάσεις δοσοληψιών, A, B, C, D. Οι μονάδες φόρτου της κλάσης $WC1$ είναι αλυσίδες από διαδοχικές δοσοληψίες των κλάσεων A, B, D, ενώ οι μονάδες φόρτου της κλάσης $WC2$ είναι αλυσίδες από διαδοχικές δοσοληψίες των κλάσεων C, A, B, D, C. Στον παρακάτω πίνακα δίδονται οι παράμετροι (profiles) για τις κλάσεις δοσοληψιών A, B, C, D:

Πίνακας 7.1: Profiles για τις κλάσεις δοσοληψιών

transaction class	avg application pathlength	min/max DB calls
TX–CLASS–A	40,000	1 – 4
TX–CLASS–B	60,000	1 – 8
TX–CLASS–C	80,000	1 – 12
TX–CLASS–D	100,000	1 – 16

7.4.2 Προδιαγραφή Πειραμάτων

Ακολουθείται το κλειστό μοντέλο φόρτου (βλ. ενότητα 5.4.3), με μεταβλητό αριθμό τερματικών. Παρουσιάζονται αποτελέσματα για 100, 125, 150, και 175 τερματικά, με think–time 4.0 sec. Οι στόχοι επίδοσης για τις κλάση $WC1$ και $WC2$ ορίζονται να είναι 3.0 sec και 5.0 sec αντίστοιχα.

Για κάθε μέτρηση που δίδεται παρακάτω έγιναν 20 επαναλήψεις, όπου κάθε επανάληψη προσομοιώνει 3600 sec χρόνου λειτουργίας του συστήματος. Στον χρόνο αυτό σημειώνονται τουλάχιστον 35,000 αφίξεις σύνθετων μονάδων φόρτου από κάθε κλάση. Μετρήσεις του χρόνου απόκρισης αρχίζουν να κρατούνται μόλις συμπληρωθούν 10,000 αφίξεις από κάθε κλάση, ώστε το προσομοιούμενο σύστημα να βρεθεί σε ευσταθή κατάσταση. Εκτελώντας 20 ανεξάρτητες επαναλήψεις για κάθε μέτρηση είναι δυνατός ο υπολογισμός των διαστημάτων εμπιστοσύνης (confidence intervals). Με τον τρόπο αυτό τεκμηριώνεται η στατιστική αξιοπιστία των μετρήσεων.

Το κύριο μέτρο επίδοσης που λαμβάνεται υπ'όψιν είναι ο μέγιστος δείκτης επίδοσης μεταξύ των κλάσεων σύνθετων μονάδων φόρτου που εξυπηρετεί το σύστημα. Δίδεται ακόμα ο μέσος χρόνος απόκρισης και η τιμή του δείκτη επίδοσης για κάθε κλάση ξεχωριστά, καθώς και ο λόγος του μέγιστου προς τον ελάχιστο δείκτη επίδοσης μεταξύ των κλάσεων. Επίσης, λαμβάνεται υπ'όψιν ένα μέτρο επίδοσης που ονομάζεται "απόσταση παραβίασης στόχων" (violation distance), και ορίζεται ως εξής:

$$VD := \sum_{WC_p} max\{\frac{R_p - G_p}{G_p}, 0\}, \tag{7.4}$$

όπου R_p είναι ο μέσος χρόνος απόκρισης του συστήματος, *όπως μετράται κατά την λειτουργία του συστήματος*, για την κλάση σύνθετων μονάδων φόρτου WC_p, για την οποία έχει οριστεί στόχος επίδοσης G_p. Το μέτρο αυτό δίδει μια συνολική εικόνα για το βαθμό ικανοποίησης των στόχων επίδοσης. Τέλος δίδεται (βλ. πίνακες 7.2 και 7.3) για κάθε μέτρηση η μέση τιμή, η τυπική απόκλιση, και το σχετικό εύρος δ_α ($\frac{\text{εύρος διαστήματος}}{\text{μέτρηση}}$) του $\alpha\%$ διαστήματος εμπιστοσύνης, για $\alpha = 90\%, 95\%, 99\%$.

7.5 Πειραματικά Αποτελέσματα

Στον πίνακα 7.4 συνοψίζονται οι μετρήσεις από τα πειράματα που πραγματοποιήθηκαν. Σημειώνονται [1] οι τιμές των δεικτών επίδοσης για τις κλάσεις $WC1$ και $WC2$. Στον πίνακα 7.5 σημειώνονται ο λόγος της μέγιστης προς την ελάχιστη τιμή μεταξύ των δεικτών επίδοσης, και η τιμή του μέτρου επίδοσης VD. Στο σχήμα 7.3 απεικονίζονται, με ραβδόγραμμα, οι μετρήσεις του μέγιστου δείκτη επίδοσης για τους αλγορίθμους που μελετήθηκαν ως συνάρτηση του αριθμού των τερματικών. Στο σχήμα 7.4 απεικονίζεται η μεταβολή του μέτρου επίδοσης VD ως συνάρτηση του αριθμού των τερματικών.

Ο αλγόριθμος RR είναι ο απλούστερος από αυτούς που μελετήθηκαν, και παρέχει μια χρήσιμη βάση σύγκρισης. Αντιμετωπίζει "δίκαια" τις κλάσεις φόρτου. Σύμφωνα με τον πίνακα 7.4, ο αλγόριθμος RR, μαζί με τον WPI, δίδει την χαμηλότερη τιμή για τον λόγο του μέγιστου προς τον ελάχιστο δείκτη επίδοσης. Το γράφημα 7.3 δείχνει ότι, αν και γενικά δεν έχει την καλύτερη επίδοση, έχει καλύτερη επίδοση από όλους, εκτός από τον WPI, για υψηλό φόρτο (αριθμός τερματικών = 175).

Οι αλγόριθμοι STATIC και rSLACK δεν αντιμετωπίζουν "δίκαια" τις κλάσεις φόρτου, όπως φαίνεται από τον πίνακα 7.4. Ο αλγόριθμος STATIC, παρόμοια με τον earliest−deadline−first, ευνοεί συστηματικά τις δοσοληψίες που είναι βήματα σε workflows της κλάσης με τον μικρότερο στόχο επίδοσης. Ο δυναμικός αλγόριθμος rSLACK ευνοεί συστηματικά τις δοσοληψίες που είναι βήματα σε workflows της κλάσης με τον μεγαλύτερο στόχο επίδοσης. Το αποτέλεσμα είναι να ευνοείται η μια κλάση σε βάρος της άλλης, χωρίς να λαμβάνεται υπ'όψιν η ικανοποίηση των στόχων επίδοσης. Ο rSLACK τροποποιεί δυναμικά την προτεραιότητα που αναθέτει στα βήματα ενός workflow, χωρίς να λαμβάνει υπ'όψιν την τρέχουσα κατάσταση ως προς την ικανοποίηση στόχων επίδοσης. Στηρίζεται μόνο στις προδιαγραφές των στόχων επίδοσης, και τους επιμέρους στόχους επίδοσης για τα βήματα.

Τα μειονεκτήματα αυτά των προηγούμενων αλγορίθμων υποδεικνύουν ότι είναι απαραίτητο να λαμβάνεται υπ'όψιν η τρέχουσα κατάσταση των κλάσεων ως προς την ικανοποίηση των στόχων επίδοσης. Οι αλγόριθμοι PI και WPI χρησιμοποιούν για τον σκοπό αυτό την τρέχουσα τιμή των δεικτών επίδοσης για τις κλάσεις. Με τον τρόπο αυτό, το σύστημα αποκτά ένα μηχανισμό ανάδρασης, ώστε να ανταποκρίνεται σε μεταβολές στην τρέχουσα κατάσταση των κλάσεων, μεταβάλλοντας ανάλογα την προτεραιότητα των δοσοληψιών−βημάτων. Ο αλγόριθμος WPI, όπως και ο WT, χρησιμοποιεί ένα μηχανισμό ανάδρασης, ώστε σε ενδιάμεσα σημεία ελέγχου κατά την εξυπηρέτηση μιας σύνθετης μονάδας φόρτου (συγκεκριμένα μετά τον τερματισμό κάθε δοσοληψίας−βήματος) να είναι δυνατή η ρύθμιση της προτεραιότητας. Συνεπώς, είναι σε θέση να αντιδράσει αποτελεσματικότερα απ'ότι ο PI, ο οποίος αναθέτει την ίδια προτεραιότητα σε όλες τις δοσοληψίες−βήματα. Ο WPI είναι ο μόνος από τους

[1] Υπενθυμίζεται ότι κάθε μέτρηση που σημειώνεται είναι ο μέσος όρος από 20 ανεξάρτητες επαναλήψεις.

αλγορίθμους που μελετήθηκαν που διατηρεί τον μέγιστο δείκτη επίδοσης μικρότερο από την μονάδα (δηλαδή ικανοποιεί τους στόχους επίδοσης όλων των κλάσεων) για αριθμό τερματικών ίσο με 150.

Ο αλγόριθμος WT χρησιμοποιεί τον ίδιο μηχανισμό ανάδρασης με τον WPI, όμως δεν λαμβάνει υπ'όψιν τον δείκτη επίδοσης για κάθε κλάση. Είναι ενδιαφέρον ότι έχει την καλύτερη επίδοση για χαμηλό σχετικά φόρτο (αριθμός τερματικών = 100), μαζί με τον rSLACK. Ομως, για υψηλότερο φόρτο (150 ή 175 τερματικά) η επίδοσή του δεν είναι καλύτερη από αυτή των RR και STATIC. Η συμπεριφορά αυτή εξηγείται από το γεγονός ότι ο WT βασίζεται στους επιμέρους στόχους επίδοσης, των οποίων ο υπολογισμός αγνοεί τις καθυστερήσεις λόγω αναμονής για πρόσβαση σε πόρους. Οι καθυστερήσεις αυτές είναι μάλλον μικρές στην περίπτωση χαμηλού φόρτου (για τη δοθείσα διαμόρφωση του συστήματος), οπότε οι επιμέρους στόχοι επίδοσης είναι κοντά στον πραγματικό χρόνο εκτέλεσης των βημάτων. Ομως για υψηλότερο φόρτο η διαφορά ανάμεσα στον επιμέρους στόχο επίδοσης και τον πραγματικό χρόνο εκτέλεσης για ένα βήμα γίνεται αρκετά σημαντική. Το ίδιο επιχείρημα ισχύει και για τον αλγόριθμο rSLACK. Ενώ όμως ο rSLACK συνεχίζει "τυφλά" να ευνοεί τις κλάσεις με μεγάλες τιμές ως στόχους επίδοσης, ο WT αξιοποιεί τον μηχανισμό ανάδρασης και έχει τουλάχιστον καλύτερη επίδοση από τον rSLACK.

Η συμπεριφορά των αλγορίθμων PI, rSLACK, WPI, και WT επηρεάζεται άμεσα από την επιλογή των τιμών για τους στόχους επίδοσης, σε αντιδιαστολή με τους RR και STATIC. Ο αλγόριθμος RR δίδει την ίδια προτεραιότητα σε όλες τις δοσοληψίες, ανεξάρτητα στόχων, ενώ ο αλγόριθμος STATIC ευνοεί συστηματικά τις δοσοληψίες που είναι βήματα σε workflows της κλάσης με τον μικρότερο στόχο επίδοσης. Οταν ένας στόχος επίδοσης δεν είναι εφικτός (είναι δηλαδή πολύ "αισιόδοξος"), η επίδοση των δυναμικών αλγορίθμων επηρεάζεται δυσμενώς. Την μεγαλύτερη ευαισθησία αναμένεται να δείξουν οι αλγόριθμοι WT και rSLACK, των οποίων η συμπεριφορά καθορίζεται σημαντικά από την επιλογή των στόχων επίδοσης. Ο WT είναι ο πλέον ευαίσθητος λόγω της εξάρτησής του από τις τιμές των επιμέρους στόχων επίδοσης, που καθορίζονται άμεσα από τις τιμές των στόχων επίδοσης για τις κλάσεις. Σχετικά πειράματα αναφέρονται στην εργασία [MN95]. Πάντως, ο μηχανισμός ανάδρασης στον WT επιτρέπει στο σύστημα να προσαρμοστεί ως ένα βαθμό. Ο PI, και κατ'αναλογία ο WPI, εμφανίζουν πιο σταθερή συμπεριφορά, καθώς παρακολουθούν τις μεταβολές των δεικτών επίδοσης για τις κλάσεις και προσαρμόζονται κατά το δυνατόν. Στον πίνακα 7.8 και στα γραφήματα 7.5 και 7.6 συνοψίζονται τα αποτελέσματα από μια σειρά πειράματα που επαληθεύουν τις παραπάνω παρατηρήσεις. Στα πειράματα δεν εξετάστηκε ο αλγόριθμος RR διότι δεν επηρεάζεται από τις τιμές των στόχων επίδοσης. Επίσης, δεν εξετάστηκε ο αλγόριθμος STATIC διότι συστηματικά ευνοεί την κλάση με τον μικρότερο στόχο επίδοσης, ενώ όταν οι στόχοι επίδοσης είναι ίσοι εκφυλίζεται στον RR. Για τους υπόλοιπους αλγορίθμους (rSLACK, PI, WPI, WT) έγιναν πειράματα για μια σειρά τιμών για τον στόχο επίδοσης της κλάσης $WC1$ ($G_{WC1} = 2.0, 3.0, 4.0, 5.0$), διατηρώντας σταθερό τον στόχο επίδοσης για την κλάση $WC2$ ($G_{WC2} = 5.0$), και τον αριθμό των τερματικών (ίσο με 150).

Στην ενότητα αυτή γίνεται επίσης μια σύγκριση των αλγορίθμων WPI και WT ως προς το ποσοστό των βημάτων σύνθετων μονάδων φόρτου των οποίων η προτεραιότητα τροποποιείται επειδή δεν ικανοποιούνται οι αντίστοιχοι επιμέρους στόχοι επίδοσης (βλ. σχήμα 7.2). Οι δυο αλγόριθμοι στηρίζονται στο ίδιο σχήμα ανάδρασης, υπολογίζοντας την προτεραιότητα ενός βήματος ως γινόμενο δυο παραγόντων, από τους οποίους ο ένας είναι ο συντελεστής w_t. Διαφέρουν μόνο στην επιλογή του δεύτερου

παράγοντα. Ο πίνακας 7.9 δίδει το ποσοστό αυτό για κάθε κλάση ξεχωριστά αλλά και συνολικά για όλες τις κλάσεις, ως συνάρτηση του στόχου επίδοσης G_{WC1}, με $G_{WC2} = 5.0$ και 150 τερματικά. Και για τους δυο αλγορίθμους το ποσοστό αυτό φθίνει όσο αυξάνει η τιμή του στόχου G_{WC1}, καθώς το σύστημα πιέζεται λιγότερο για να ικανοποιήσει τους στόχους επίδοσης. Στο γράφημα 7.7 δίδεται μια γραφική παράσταση του ποσοστού αυτού, συνολικά για όλες τις κλάσεις, και φαίνεται παραστατικά ότι ο αλγόριθμος WPI επεμβαίνει λιγότερο συχνά από τον WT για να τροποποιήσει την προτεραιότητα ενός βήματος. Η διαφορά αυτή είναι σταθερά γύρω στο 4%. Είναι σημαντικό ότι μια διαφορά αυτής της τάξης μεγέθους επηρεάζει σε μεγάλο βαθμό το αν τελικά ικανοποιούνται οι στόχοι επίδοσης.

Συνολικά, προκύπτει το συμπέρασμα ότι πολιτικές όπως οι PI και WPI έχουν καλύτερη επίδοση σε σχέση με πολιτικές που δεν λαμβάνουν υπ'όψιν πληροφορία για την τρέχουσα κατάσταση ως προς την ικανοποίηση στόχων επίδοσης για τις κλάσεις. Η πολιτική PI υλοποιεί ουσιαστικά ένα μηχανισμό ανάδρασης με βάση την τρέχουσα κατάσταση ως προς την ικανοποίηση στόχων, καθώς στηρίζεται στην τρέχουσα τιμή του δείκτη επίδοσης για κάθε κλάση. Η πολιτική WPI επιπλέον λαμβάνει υπ'όψιν την ικανοποίηση επιμέρους στόχων επίδοσης κατά την εκτέλεση σύνθετων μονάδων φόρτου, με συνέπεια το σύστημα να έχει στην διάθεσή του περισσότερα σημεία ελέγχου, καθώς η προτεραιότητα ενός βήματος υπολογίζεται ξεχωριστά μετά τον τερματισμό του προηγούμενου.

7.6 Επισκόπιση της Σχετικής Βιβλιογραφίας

Η σχεδίαση των αλγορίθμων που παρουσιάζονται στο κεφάλαιο αυτό βασίστηκε σε προηγούμενη δουλειά πάνω σε θέματα ικανοποίησης στόχων επίδοσης (βλ. κεφάλαιο 3). Αυτή όμως η προηγούμενη δουλειά δεν περιλαμβάνει μελέτη τεχνικών για σύνθετες μονάδες φόρτου.

Σχετική δουλειά αναφέρεται στην εργασία [KGM93], όπου εξετάζονται δυναμικές τεχνικές χρονοπρογραμματισμού σε ένα κατανεμημένο περιβάλλον soft real−time, όπου ο στόχος είναι να περιοριστεί κατά το δυνατόν το ποσοστό των εργασιών των οποίων η εκτέλεση περατώνεται σε χρόνο μεγαλύτερο από την προθεσμία (deadline) που έχει οριστεί γι'αυτές. Οι εργασίες, που ονομάζονται "global tasks", αποτελούνται από μια αλυσίδα από επιμέρους βήματα, που ονομάζονται "local tasks", το καθένα από τα οποία πρέπει να εκτελεστεί σε συγκεκριμένο κόμβο του συστήματος. Η εργασία αυτή μελετά τρόπους για τον καθορισμό της προθεσμίας για κάθε βήμα, δοθείσης της συνολικής προθεσμίας για μια εργασία. Τα βήματα δεν είναι δοσοληψίες, ενώ δεν μελετούνται συστήματα που έχουν να εξυπηρετήσουν πολλαπλές κλάσεις εργασιών, θεωρώντας σταθερό το πλήθος των βημάτων για κάθε global task.

Ενδιαφέρον παρουσιάζει η εργασία [PLC92], που ασχολείται με real−time συστήματα διαχείρισης βάσεων δεδομένων, και μελετά το πρόβλημα, που αναφέρθηκε και στην παρουσίαση του αλγορίθμου STATIC στην ενότητα 7.3.2, της "προκατάληψης" (bias) που δείχνει ο κλασσικός αλγόριθμος earliest−deadline−first κατά μονάδων φόρτου (δοσοληψιών για την εργασία αυτή) μακράς διάρκειας. Η μελέτη αυτή υποδεικνύει παραστατικά το πρόβλημα, δείχνοντας με προσομοιώσεις ότι δοσοληψίες μακράς διάρκειας αποτυγχάνουν να τηρήσουν την προθεσμία που τους έχει δοθεί συχνότερα από δοσοληψίες μικρότερης διάρκειας. Για την αντιμετώπιση του προβλήματος εισάγεται η έννοια της *εικονικής προθεσμίας* (virtual deadline). Η εικονική προθεσμία

τροποποιείται δυναμικά όσο προχωρά η εκτέλεση της δοσοληψίας, και είναι συνάρτηση του μεγέθους της. Η ιδέα αυτή είναι αρκετά σχετική με την ιδέα των επιμέρους στόχων επίδοσης που προτείνεται στην παρούσα εργασία.

Σχετική δουλειά έχει γίνει και στο πεδίο της θεωρίας ελέγχου (control theory). Είναι σαφές ότι μια πιο τυπική προσέγγιση στο πρόβλημα είναι σημαντική, όμως οι απαραίτητες μαθηματικές τεχνικές είναι ιδιαίτερα επίπονες στην εφαρμογή. Στην αναφορά [BGT91] παρουσιάζεται μια διατύπωση του προβλήματος του χρονοπρογραμματισμού για ένα σύστημα όπου μονάδες φόρτου έχουν να "επισκεφτούν" ένα αριθμό από κόμβους που χρειάζονται τις υπηρεσίες ενός σταθμού εξυπηρέτησης ως πρόβλημα στοχαστικής βελτιστοποίησης. Παράδειγμα συστήματος που ακολουθεί αυτό το αφηρημένο μοντέλο θα μπορούσε να είναι ένα σύστημα με ένα επεξεργαστή και ένα αριθμό από software servers, όπως ένας επόπτης επεξεργασίας δοσοληψιών και ένα σύστημα διαχείρισης βάσεων δεδομένων. Προτείνεται μια μέθοδος για την ελαχιστοποίηση του μέγιστου δείκτη επίδοσης μεταξύ των κλάσεων φόρτου, όμως η μέθοδος έχει το μειονέκτημα ότι είναι περίπλοκη και η υλοποίησή της έχει υψηλό κόστος.

Πίνακας 7.2: Στατιστικές Ιδιότητες των Μετρήσεων για την Κλάση WC1

terminals	**Μετρήσεις Χρόνου Απόκρισης για την Κλάση WC1**					
	Αλγόριθμος	Μέση Τιμή	Απόκλιση	δ_{90}	δ_{95}	δ_{99}
100	RR	1.349	0.002	0.281%	0.335%	0.441%
	STATIC	0.443	0.001	0.246%	0.293%	0.385%
	rSLACK	1.175	0.003	0.406%	0.484%	0.636%
	PI	1.265	0.002	0.258%	0.308%	0.404%
	WPI	0.967	0.006	1.013%	1.207%	1.587%
	WT	0.870	0.002	0.315%	0.376%	0.494%
125	RR	2.045	0.003	0.218%	0.260%	0.342%
	STATIC	0.551	0.001	0.257%	0.306%	0.403%
	rSLACK	2.325	0.005	0.338%	0.403%	0.530%
	PI	2.179	0.006	0.432%	0.515%	0.677%
	WPI	1.715	0.030	2.879%	3.430%	4.508%
	WT	1.320	0.002	0.246%	0.293%	0.385%
150	RR	2.816	0.003	0.202%	0.240%	0.316%
	STATIC	0.686	0.001	0.252%	0.301%	0.395%
	rSLACK	4.021	0.007	0.296%	0.352%	0.463%
	PI	3.303	0.028	1.394%	1.661%	2.183%
	WPI	2.848	0.085	4.911%	5.852%	7.691%
	WT	1.719	0.003	0.325%	0.387%	0.509%
175	RR	3.653	0.003	0.135%	0.161%	0.212%
	STATIC	0.853	0.001	0.274%	0.326%	0.428%
	rSLACK	6.025	0.012	0.337%	0.401%	0.527%
	PI	4.318	0.026	0.997%	1.188%	0.562%
	WPI	3.837	0.127	5.451%	6.495%	8.537%
	WT	2.205	0.007	0.531%	0.632%	0.831%

Πίνακας 7.3: Στατιστικές Ιδιότητες των Μετρήσεων για την Κλάση WC2

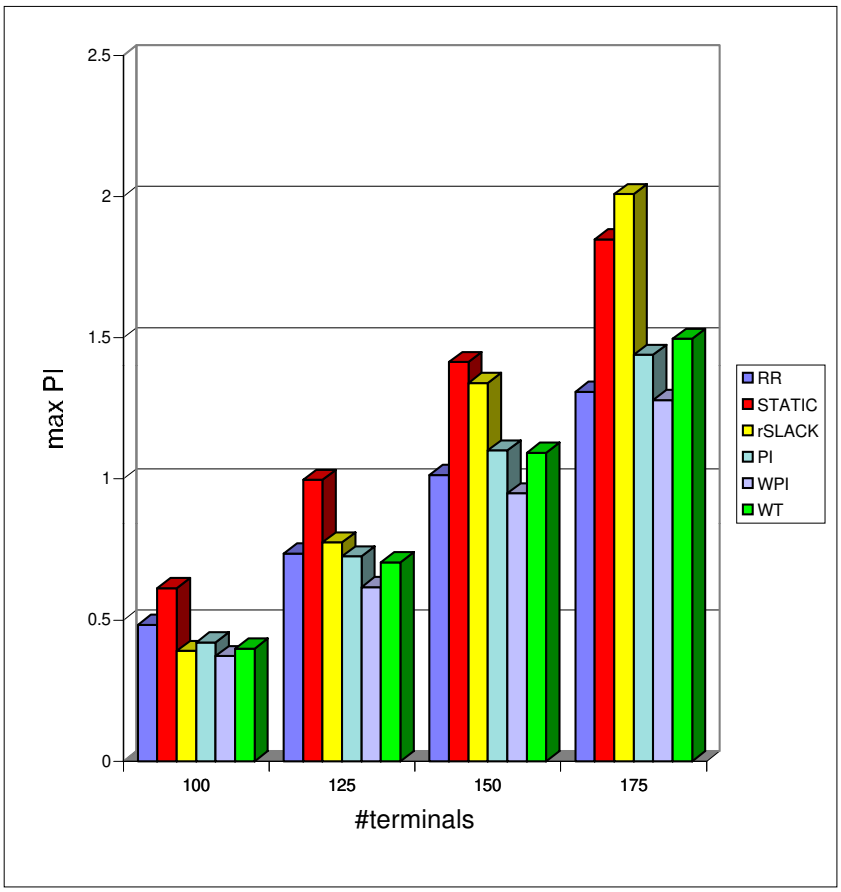
terminals	**Μετρήσεις Χρόνου Απόκρισης για την Κλάση WC2**					
	Αλγόριθμος	Μέση Τιμή	Απόκλιση	δ_{90}	δ_{95}	δ_{99}
100	RR	2.147	0.004	0.256%	0.305%	0.400%
	STATIC	3.068	0.005	0.286%	0.340%	0.447%
	rSLACK	1.838	0.003	0.265%	0.316%	0.416%
	PI	1.847	0.002	0.207%	0.247%	0.325%
	WPI	1.868	0.007	0.600%	0.715%	0.940%
	WT	1.997	0.005	0.378%	0.451%	0.593%
125	RR	3.682	0.006	0.255%	0.304%	0.399%
	STATIC	4.989	0.007	0.215%	0.256%	0.337%
	rSLACK	2.644	0.004	0.270%	0.322%	0.423%
	PI	2.840	0.005	0.266%	0.316%	0.415%
	WPI	3.088	0.035	1.875%	2.234%	2.936%
	WT	3.525	0.006	0.261%	0.311%	0.408%
150	RR	5.073	0.005	0.171%	0.204%	0.268%
	STATIC	7.072	0.005	0.121%	0.144%	0.189%
	rSLACK	3.250	0.004	0.192%	0.229%	0.301%
	PI	3.999	0.018	0.734%	0.875%	1.149%
	WPI	4.392	0.095	3.572%	4.256%	5.593%
	WT	5.468	0.006	0.179%	0.214%	0.281%
175	RR	6.549	0.006	0.142%	0.169%	0.223%
	STATIC	9.240	0.008	0.134%	0.160%	0.210%
	rSLACK	3.657	0.005	0.240%	0.286%	0.376%
	PI	5.456	0.021	0.627%	0.747%	0.982%
	WPI	5.906	0.134	3.738%	4.454%	5.854%
	WT	7.482	0.008	0.173%	0.206%	0.271%

Πίνακας 7.4: Σύνοψη Μετρήσεων

terminals	**Δείκτες Επίδοσης**						
	Class	RR	STATIC	rSLACK	PI	W−PI	W−T
100	WC1	0.449	0.110	0.391	0.421	0.322	0.290
	WC2	0.483	0.613	0.367	0.369	0.373	0.399
125	WC1	0.681	0.183	0.775	0.726	0.571	0.440
	WC2	0.736	0.997	0.528	0.568	0.617	0.705
150	WC1	0.938	0.228	1.340	1.101	0.949	0.573
	WC2	1.014	1.414	0.650	0.799	0.878	1.093
175	WC1	1.217	0.284	2.008	1.439	1.279	0.735
	WC2	1.309	1.848	0.731	1.091	1.181	1.496

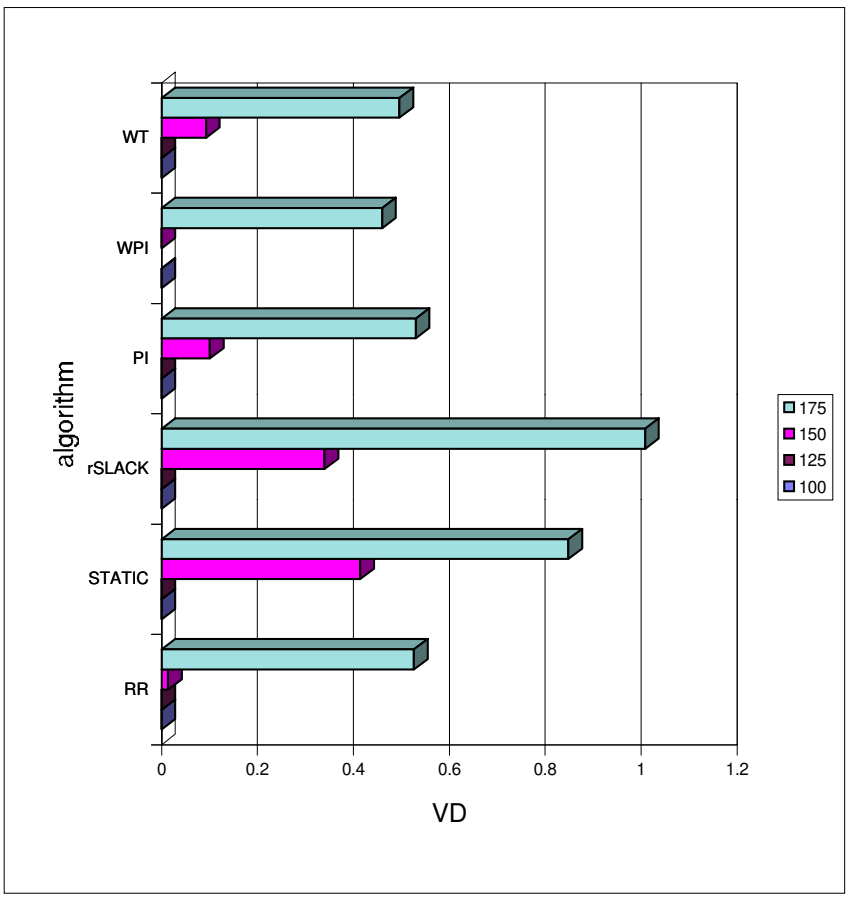
Πίνακας 7.5: Απόσταση Παραβίασης Στόχων και Λόγος min/max Δείκτη Επίδοσης

terminals							
		RR	STATIC	rSLACK	PI	W−PI	W−T
100	VD	0.0	0.0	0.0	0.0	0.0	0.0
	max:min	1.075	5.572	1.065	1.140	1.158	1.375
125	VD	0.0	0.0	0.0	0.0	0.0	0.0
	max:min	1.079	5.548	1.467	1.278	1.080	1.602
150	VD	0.014	0.414	0.340	0.101	0.0	0.093
	max:min	1.081	6.201	2.061	1.377	1.080	1.907
175	VD	0.526	0.848	1.008	0.530	0.460	0.496
	max:min	1.075	6.507	2.746	1.318	1.082	1.998



Σχήμα 7.3: Μεταβολή του Μέγιστου Δείκτη Επίδοσης ως προς τον Αριθμό Τερματικών.

Μόνο ο αλγόριθμος WPI διατηρεί τον μέγιστο δείκτη επίδοσης μικρότερο της μονάδας για 150 τερματικά. Γενικά ο αλγόριθμος αυτός δίδει την μικρότερη τιμή για τον μέγιστο δείκτη επίδοσης για όλους τους αριθμούς τερματικών που προσομοιώθηκαν. Είναι ενδιαφέρον ότι για υψηλό φόρτο (175 τερματικά), όταν δεν είναι εφικτή η ικανοποίηση των στόχων επίδοσης, όλοι οι αλγόριθμοι εκτός του WPI δίδουν χειρότερα αποτελέσματα από τον RR.



Σχήμα 7.4: Μεταβολή του Μέτρου VD ως προς τον Αριθμό Τερματικών.

Ο αλγόριθμος WPI δίδει την χαμηλότερη τιμή για το μέτρο VD για όλους τους αριθμούς τερματικών που προσομοιώθηκαν. Το μέτρο VD γίνεται μεγαλύτερο του μηδενός για τον WPI *μόνο* για 175 τερματικά.

Πίνακας 7.6: Μελέτη Ευαισθησίας ως προς τον Καθορισμό Στόχων: Στατιστικές Ιδιότητες των Μετρήσεων για την Κλάση WC1. Ο στόχος για την κλάση WC2 είναι 5.0 sec, και ο αριθμός των τερματικών ίσος με 150.

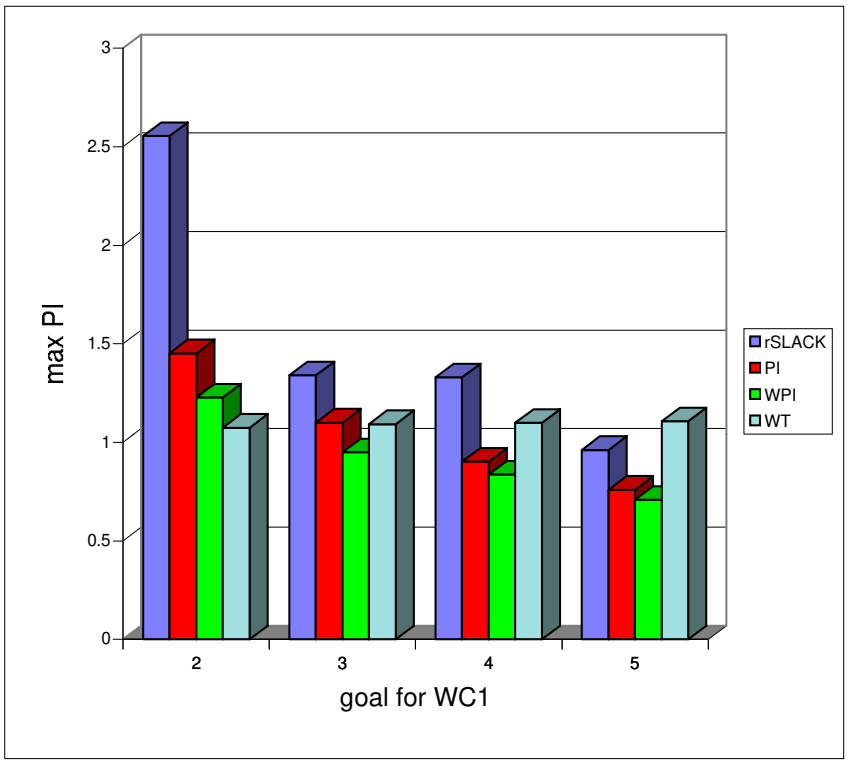
G_{WC1}	**Μετρήσεις Χρόνου Απόκρισης για την Κλάση WC1**					
	Αλγόριθμος	Μέση Τιμή	Απόκλιση	δ_{90}	δ_{95}	δ_{99}
2.0	rSLACK	5.108	0.010	0.327%	0.389%	0.512%
	PI	2.903	0.009	0.494%	0.588%	0.773%
	WPI	2.455	0.007	0.491%	0.585%	0.769%
	WT	1.845	0.004	0.362%	0.431%	0.567%
3.0	rSLACK	4.021	0.007	0.296%	0.352%	0.463%
	PI	3.303	0.028	1.394%	1.661%	2.183%
	WPI	2.848	0.085	4.911%	5.852%	7.691%
	WT	1.719	0.003	0.325%	0.387%	0.509%
4.0	rSLACK	5.320	0.008	0.250%	0.298%	0.392%
	PI	3.611	0.007	0.304%	0.362%	0.475%
	WPI	3.351	0.008	0.394%	0.469%	0.616%
	WT	1.696	0.002	0.238%	0.284%	0.373%
5.0	rSLACK	4.814	0.007	0.251%	0.299%	0.393%
	PI	3.797	0.008	0.329%	0.392%	0.515%
	WPI	3.639	0.008	0.351%	0.418%	0.550%
	WT	1.673	0.003	0.283%	0.337%	0.443%

Πίνακας 7.7: Μελέτη Ευαισθησίας ως προς τον Καθορισμό Στόχων : Στατιστικές Ιδιότητες των Μετρήσεων για την Κλάση WC2. Ο στόχος για την κλάση WC2 είναι 5.0 sec, και ο αριθμός των τερματικών ίσος με 150

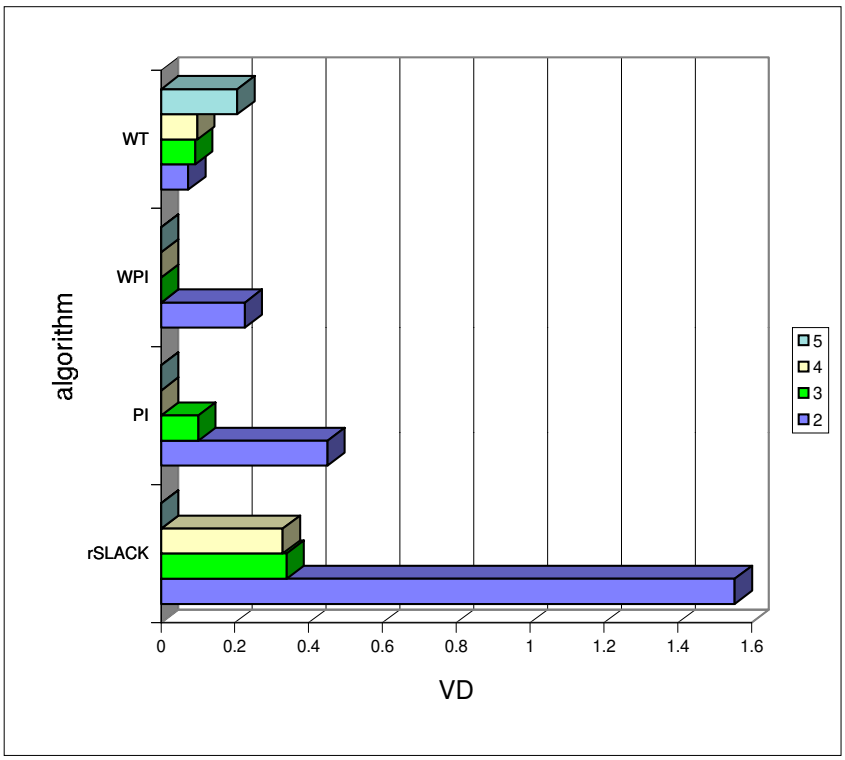
G_{WC1}	**Μετρήσεις Χρόνου Απόκρισης για την Κλάση WC2**					
	Αλγόριθμος	Μέση Τιμή	Απόκλιση	δ_{90}	δ_{95}	δ_{99}
2.0	rSLACK	2.121	0.003	0.327%	0.389%	0.512%
	PI	4.485	0.007	0.267%	0.318%	0.418%
	WPI	4.742	0.007	0.234%	0.279%	0.367%
	WT	5.370	0.006	0.169%	0.201%	0.264%
3.0	rSLACK	3.250	0.004	0.192%	0.229%	0.301%
	PI	3.999	0.018	0.734%	0.875%	1.149%
	WPI	4.392	0.095	3.572%	4.256%	5.593%
	WT	5.468	0.006	0.179%	0.214%	0.281%
4.0	rSLACK	2.342	0.004	0.277%	0.330%	0.434%
	PI	3.705	0.005	0.237%	0.283%	0.372%
	WPI	3.825	0.008	0.340%	0.405%	0.532%
	WT	5.498	0.005	0.157%	0.187%	0.245%
5.0	rSLACK	2.557	0.004	0.253%	0.301%	0.396%
	PI	3.529	0.008	0.352%	0.419%	0.551%
	WPI	3.554	0.007	0.332%	0.396%	0.520%
	WT	5.614	0.008	0.223%	0.266%	0.350%

Πίνακας 7.8: Μεταβολή Δεικτών Επίδοσης και Μέτρου VD ως προς τον στόχο G_{WC1}. Ο στόχος G_{WC2} είναι 5.0 sec, και ο αριθμός των τερματικών ίσος με 150.

G_{WC1}	Δείκτες Επίδοσης και VD				
		rSLACK	PI	W−PI	W−T
2.0	WC1	2.554	1.451	1.227	0.922
	WC2	0.424	0.885	0.948	1.074
	VD	1.554	0.451	0.227	0.074
3.0	WC1	1.340	1.101	0.949	0.573
	WC2	0.650	0.799	0.878	1.093
	VD	0.340	0.101	0.0	0.093
4.0	WC1	1.330	0.902	0.837	0.424
	WC2	0.468	0.741	0.765	1.099
	VD	0.330	0.0	0.0	0.099
5.0	WC1	0.962	0.759	0.727	0.334
	WC2	0.511	0.705	0.708	1.107
	VD	0.0	0.0	0.0	0.107



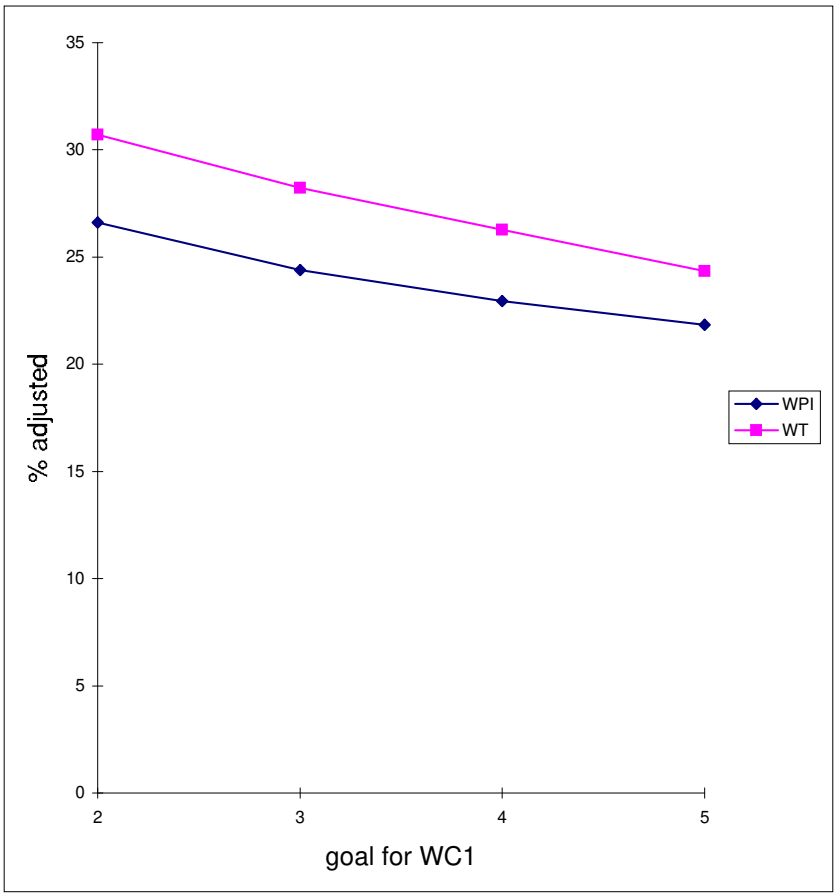
Σχήμα 7.5: Μεταβολή του Μέγιστου Δείκτη Επίδοσης ως προς τον Στόχο G_{WC1}. Ο στόχος G_{WC2} είναι 5.0 sec, και ο αριθμός των τερματικών ίσος με 150.



Σχήμα 7.6: Μεταβολή του Μέτρου VD ως προς τον Στόχο G_{WC1}. Ο στόχος G_{WC2} είναι 5.0 sec, και ο αριθμός των τερματικών ίσος με 150.

Πίνακας 7.9: Δυναμική Συμπεριφορά των Αλγορίθμων WPI και WT: Μεταβολή του ποσοστού των βημάτων των οποίων η προτεραιότητα τροποποιείται ως προς τον στόχο G_{WC1}. Ο στόχος G_{WC2} είναι 5.0 sec, και ο αριθμός των τερματικών ίσος με 150. Σημειώνεται το ποσοστό των βημάτων των οποίων η προτεραιότητα τροποποιείται ξεχωριστά για κάθε κλάση, και συνολικά για όλες τις κλάσεις. Υπενθυμίζεται ότι οι μονάδες φόρτου της κλάσης WC1 αποτελούνται από 3 βήματα, ενώ αυτές της κλάσης WC2 από 5 βήματα.

G_{WC1}	WPI			WT		
	WC1	WC2	overall	WC1	WC2	overall
2.0	30.307%	24.400%	26.622%	37.535%	26.563%	30.706%
3.0	26.152%	23.340%	24.393%	30.482%	26.827%	28.126%
4.0	23.550%	22.582%	22.946%	24.999%	27.047%	26.280%
5.0	21.549%	22.016%	21.841%	19.485%	27.235%	24.334%



Σχήμα 7.7: Μεταβολή του ποσοστού των βημάτων των οποίων τροποποιείται η προτεραιότητα ως προς τον στόχο G_{WC1}. Ο στόχος G_{WC2} είναι 5.0 sec, και ο αριθμός των τερματικών ίσος με 150. Ο αλγόριθμος WPI λειτουργεί πιο "επιλεκτικά' από τον WT, καθώς τροποποιεί την προτεραιότητα μικρότερου ποσοστού βημάτων από σύνθετες μονάδες φόρτου. Το ποσοστό γενικά μειώνεται όσο ο στόχος G_{WC1} γίνεται μεγαλύτερος, καθώς το σύστημα πιέζεται λιγότερο για να ικανοποιήσει τους στόχους επίδοσης.

Κεφάλαιο 8

Συμπεράσματα και Ερευνητικές Κατευθύνσεις

Στην ενότητα αυτή παρουσιάζονται συνοπτικά τα αποτελέσματα της εργασίας, οργανωμένα κατά γνωστική περιοχή. Οι κύριες γνωστικές περιοχές που άπτονται της εργασίας αυτής είναι η προσομοίωση συστημάτων, η επεξεργασία δοσοληψιών, και η διαχείριση πόρων. Η σύνοψη έχει τον χαρακτήρα κατακλείδας για την αναφορά αυτή, και θέτει τις βάσεις για μια σειρά από προτάσεις για επεκτάσεις και παραπέρα έρευνα.

8.1 Σύνοψη Αποτελεσμάτων

Στα πλαίσια της εργασίας αυτής σχεδιάστηκε και αναπτύχθηκε ένας προσομοιωτής για συστήματα επεξεργασίας δοσοληψιών, που μοντελοποιεί όλες τις κύριες λειτουργίες ενός επόπτη εκτέλεσης/επεξεργασίας δοσοληψιών, ενός συστήματος διαχείρισης βάσεων δεδομένων, και ενός υποσυστήματος επικοινωνίας. Ο προσομοιωτής παρέχει στον χρήστη μια γλώσσα προδιαγραφής για την περιγραφή, σε υψηλό επίπεδο, της διαμόρφωσης του υπό μελέτη συστήματος και του φόρτου εξυπηρέτησης. Η σχεδίαση του προσομοιωτή επιτρέπει να διερευνηθούν εναλλακτικές σχεδιαστικές επιλογές για καίρια τμήματα του συστήματος, με έμφαση στην μελέτη εναλλακτικών πολιτικών ανάθεσης πόρων. Ο προσομοιωτής εντάσσεται σε ένα ολοκληρωμένο περιβάλλον υποστήριξης πειραμάτων, που επιτρέπει τον καθορισμό πειραμάτων, και την εκτέλεση των πειραματικών βημάτων και συλλογή μετρήσεων, χωρίς την ανάγκη επίβλεψης από τον χρήστη. Το περιβάλλον αυτό διευκόλυνε σημαντικά την διεξαγωγή μιας εκτενούς σειράς πειραμάτων για την αξιολόγηση τεχνικών διαχείρισης πόρων.

Ο προσομοιωτής TPsim εξομοιώνει σε σημαντικό βαθμό λεπτομέρειας την λειτουργία ενός τυπικού επόπτη επεξεργασίας δοσοληψιών και βασικών διαχειριστών πόρων όπως ένα σύστημα διαχείρισης βάσεων δεδομένων και ένα σύστημα διαχείρισης ουρών. Η σχεδίαση του προσομοιωτή παρέχει σημαντική ευελιξία στον καθορισμό του μοντέλου συστήματος και του μοντέλου φόρτου, επιτρέποντας την μελέτη εναλλακτικών σχεδιαστικών επιλογών κάτω από ποικίλες συνθήκες λειτουργίας.

Υλοποιήθηκαν τεχνικές διαχείρισης πόρων για σημαντικά τμήματα ενός συστήματος επεξεργασίας δοσοληψιών, με έμφαση σε τεχνικές για την ικανοποίηση στόχων επίδοσης που ορίζονται για κλάσεις φόρτου. Προτάθηκαν τεχνικές χρονοπρογραμματισμού για σύνθετες μονάδες φόρτου, και αξιολογήθηκαν με σειρά πειραμάτων προσομοίωσης. Εισήχθηκε η έννοια του καθορισμού επιμέρους στόχων επίδοσης για

τα επιμέρους βήματα σύνθετων μονάδων φόρτου, και αναπτύχθηκαν απλές τεχνικές που βασίζονται σε ένα βρόχο ανάδρασης για τον χρονοπρογραμματισμό των δοσοληψιών που αποτελούν τα επιμέρους βήματα. Βρέθηκε ότι απλές τεχνικές που βασίζονται σε αυτόν τον βρόγχο ανάδρασης μπορούν να ικανοποιήσουν τους στόχους επίδοσης σύνθετων μονάδων φόρτου. Ελέγθηκε τέλος η ευαισθησία των τεχνικών αυτών στον καθορισμό των στόχων επίδοσης.

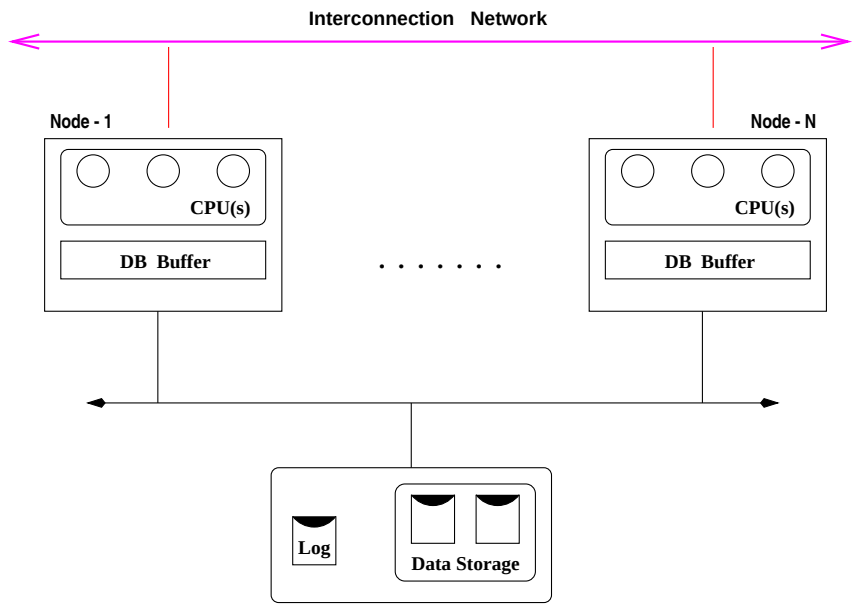
8.2 Επεκτάσεις και Ερευνητικές Κατευθύνσεις

Στην ενότητα αυτή προτείνονται ορισμένες επεκτάσεις των αποτελεσμάτων που προέκυψαν στα πλαίσια της παρούσας εργασίας. Οι προτάσεις αυτές μπορούν να ταξινομηθούν σε δύο κατηγορίες. Η πρώτη κατηγορία έχει περισσότερο εφαρμοσμένο χαρακτήρα και περιλαμβάνει προτάσεις για επεκτάσεις του προσομοιωτή TPsim με σκοπό να διευρυνθεί το πεδίο εφαρμογής του. Η δεύτερη κατηγορία προτάσεων έχει καθαρά ερευνητικό χαρακτήρα και περιλαμβάνει υποδείξεις για παραπέρα δουλειά σε θέματα διαχείρισης πόρων και σύνθετων μονάδων φόρτου τα οποία μελετήθηκαν στα πλαίσια αυτής της εργασίας.

8.2.1 Επεκτάσεις του Προσομοιωτή TPsim

Η πειραματική αξιολόγηση των αλγορίθμων που προτάθηκαν στην εργασία αυτή στηρίχθηκε σε συνθετικά μοντέλα φόρτου. Είναι σημαντικό να επεκταθεί το μοντέλο φόρτου που υποστηρίζει ο προσομοιωτής ώστε να είναι δυνατή η χρήση ιχνών (traces) από πραγματικά συστήματα. Η επέκταση αυτή θα επιτρέψει την αξιολόγηση τεχνικών διαχείρισης πόρων υπό ρεαλιστικές συνθήκες φόρτου. Στο ίδιο πνεύμα, είναι δυνατή η επέκταση του προσομοιωτή ώστε να μπορεί να δεχθεί ως είσοδο το σχήμα εκτέλεσης (execution plan) που παράγει ένας πραγματικός βελτιστοποιητής επερωτήσεων (query optimizer) για σύνθετες επερωτήσεις. Καθώς η παρούσα υλοποίηση του προσομοιωτή μοντελοποιεί προσπελάσεις σε επίπεδο σελίδων, η επέκταση αυτή απαιτεί να υλοποιηθεί μια βιβλιοθήκη από ρουτίνες που εξομοιώνουν την εκτέλεση τελεστών που εμφανίζονται σε ένα σχήμα εκτέλεσης [Sel79], όπως πράξεις σάρωσης (scan) και σύνδεσης (join). Η βιβλιοθήκη θα πρέπει να υποστηρίζει εναλλακτικούς αλγόριθμους για κάθε πράξη. Για παράδειγμα, μια πράξη σάρωσης μπορεί να υλοποιείται με σειριακή σάρωση ενός αρχείου δεδομένων, ή μέσω μιας δομής δεικτοδότησης (index), ενώ και μια πράξη σύνδεσης μπορεί να υλοποιείται με ποικιλία αλγορίθμων (όπως οι merge−join και hash−join). Η βιβλιοθήκη αυτή θα επιτρέψει την προσομοίωση συστημάτων με φόρτο αποτελούμενο από πιο πολύπλοκες μονάδες φόρτου σε σχέση με το συνθετικό μοντέλο φόρτου, όπως οι δοσοληψίες που καθορίζονται στις προδιαγραφές του benchmark TPC−C [Cou93], και οι επερωτήσεις του benchmark TPC−D [Cou95]. Είναι ενδιαφέρον να μελετηθεί η επίδοση συστημάτων για φόρτο εξυπηρέτησης που περιλαμβάνει και δοσοληψίες και επερωτήσεις [BCD+92]. Σε τέτοια συστήματα η δυνατότητα διατύπωσης στόχων επίδοσης για τις κλάσεις φόρτου είναι ιδιαίτερα χρήσιμη.

Η παρούσα υλοποίηση του προσομοιωτή TPsim υποστηρίζει την προσομοίωση συστημάτων που ακολουθούν την Shared−Nothing αρχιτεκτονική. Μια σημαντική επέκταση είναι η υποστήριξη και της αρχιτεκτονικής Shared−Disk [YCDI86, YDR+87] (βλ. σχήμα 8.1).



Σχήμα 8.1: Αρχιτεκτονική Data Sharing για κατανεμημένα συστήματα επεξεργασίας δοσοληψιών.

Κάθε κόμβος περιλαμβάνει ένα ή περισσότερους επεξεργαστές και διαθέτει ενταμιευτές δεδομένων. Ολοι οι κόμβοι μπορούν να προσπελάσουν τα δεδομένα στους κοινόχρηστους δίσκους του συστήματος.

Καθώς κάθε κόμβος έχει ιδιωτικούς ενταμιευτές δεδομένων, είναι δυνατόν να υπάρχουν αντίγραφα της ίδιας σελίδας σε πολλαπλούς κόμβους, οπότε τίθεται θέμα συνέπειας (consistency/coherence). Συνεπώς, ο προσομοιωτής θα πρέπει να επεκταθεί ώστε να υποστηρίζει πρωτόκολλα για την εξασφάλιση της συνέπειας. Για λόγους επίδοσης, τα πρωτόκολλα αυτά γενικά ενσωματώνονται στο πρωτόκολλο ελέγχου ταυτόχρονης προσπέλασης.

Μια παραλλαγή αυτής της αρχιτεκτονικής προσθέτει ένα επιπλέον επίπεδο στην ιεραρχία μνήμης, μεταξύ της μνήμης ενταμιευτών και δίσκων. Η αρχιτεκτονική αυτή αναφέρεται με το όνομα Shared−Intermediate−Memory [YD91b, YD91a], και υλοποιείται στο σύστημα SYSPLEX της IBM [IBM94], με την προσθήκη μιας συσκευής που ονομάζεται coupling facility. Ο προσομοιωτής θα μπορούσε να υποστηρίξει και τέτοια συστήματα, με βάση την επέκταση για την υποστήριξη της αρχιτεκτονικής Shared−Disk.

8.2.2 Ικανοποίηση Στόχων Επίδοσης

Είναι σαφές ότι μια πλήρης λύση στο πρόβλημα της ικανοποίησης στόχων επί−δοσης για πολλαπλές κλάσεις μονάδων φόρτου οφείλει να λάβει υπ'όψιν όλους τους διαθέσιμους πόρους του συστήματος, καθώς και πληθώρα παραμέτρων ελέγχου για κάθε πόρο. Μια τέτοια λύση θα επέτρεπε στο σύστημα, για δοθέντες στόχους επίδοσης για κάθε κλάση μονάδων φόρτου, να ελέγχει δυναμικά την επίδοσή του προσαρμόζο−ντας κατάλληλα τις παραμέτρους ελέγχου των διαφόρων υποσυστημάτων του. Είναι συνεπώς απαραίτητο να αναπτυχθούν τεχνικές που να είναι σε θέση να διαχειριστούν πολλαπλούς πόρους και να λάβουν υπ'όψιν τις αλληλεπιδράσεις που μπορούν να έχουν

οι μεταβολές παραμέτρων ελέγχου.

Η συνύπαρξη πολλαπλών διαχειριστών πόρων θέτει επίσης θέματα ευστάθειας του συστήματος, καθώς οι ενέργειές τους μπορεί να συγκρούονται. Καθώς το σύστημα παρέχει πολλαπλά επίπεδα ελέγχου, υπάρχουν γενικά αρκετοί τρόποι για να καθοδηγηθεί το σύστημα στην ικανοποίηση των στόχων επίδοσης για μια κλάση. Οι τρόποι αυτοί όμως δεν είναι ισοδύναμοι ως προς την επίδραση που έχουν στην ικανοποίηση των στόχων επίδοσης των υπολοίπων κλάσεων. Ο καθορισμός του πόρου (ή των πόρων) που επηρεάζουν πιο πιεστικά την επίδοση του συστήματος για κάθε κλάση μονάδων φόρτου αποτελεί ένα σημαντικό πρόβλημα. Για την λύση του είναι απαραίτητη η παρακολούθηση της συμπεριφοράς των μονάδων φόρτου που εξυπηρετεί το σύστημα από πλευράς κατανάλωσης πόρων και από πλευράς χρόνου απόκρισης. Είναι απαραίτητο να καθοριστεί πώς αναλύεται ο χρόνος παραμονής μιας μονάδας φόρτου στο σύστημα σε χρόνους αναμονής για προσπέλαση σε πόρους συστήματος και σε χρόνους χρήσης των πόρων, ώστε να εντοπιστεί ποιό είναι το κρίσιμο σημείο που περιορίζει την επίδοση του συστήματος για την δοθείσα κλάση φόρτου. Με τον τρόπο αυτό μπορεί κανείς να καθορίσει μια διάταξη των πόρων, για κάθε κλάση φόρτου, βάσει της οποίας να ρυθμίζεται η πολιτική ανάθεσης των πολλαπλών ειδών πόρων που χρειάζονται για την εξυπηρέτηση των αντίστοιχων μονάδων φόρτου.

Ενα ακόμα σημαντικό πρόβλημα προκύπτει από το γεγονός ότι ο φόρτος του συστήματος δεν διατηρείται σταθερός με τον χρόνο, αλλά χαρακτηρίζεται από μεταβολές. Οι αλγόριθμοι ανάθεσης πόρων οφείλουν να λάβουν υπ'όψιν την επίδραση αυτών των μεταβολών, ειδικά εάν στηρίζονται σε μια στατιστική περιγραφή των απαιτήσεων πόρων από τις διάφορες κλάσεις μονάδων φόρτου.

Μια άλλη κατεύθυνση έρευνας είναι η ανάπτυξη τεχνικών για την ικανοποίηση εναλλακτικών τύπων στόχων επίδοσης (βλ. ενότητα 3.3). Ιδιαίτερα ενδιαφέρον είναι το πρόβλημα της ταυτόχρονης υποστήριξης στόχων σχετικών με προθεσμίες για μεμονωμένες μονάδες φόρτου και στόχων σχετικών με μέσο χρόνο απόκρισης για κλάσεις μονάδων φόρτου. Η διατύπωση στόχων με την μορφή προθεσμιών είναι απαραίτητη για την διατύπωση εγγυήσεων ποιότητας εξυπηρέτησης (quality of service). Επίσης, είναι σημαντικό να ληφθεί υπ'όψιν στον ορισμό της ικανοποίησης στόχων επίδοσης το ανεκτό επίπεδο διακύμανσης στην ποιότητα εξυπηρέτησης. Η απαίτηση αυτή θα μπορούσε να εκφραστεί με την διατύπωση επιθυμητών ορίων για την διασπορά του χρόνου απόκρισης των κλάσεων φόρτου. Ο καθορισμός της σπουδαιότητας της κάθε κλάσης φόρτου είναι απαραίτητος, ώστε, όταν το σύστημα αδυνατεί να ικανοποιήσει τους στόχους επίδοσης όλων των κλάσεων (degraded mode), οι διαθέσιμοι πόροι να διατίθενται για να διατηρηθεί η επίδοση του συστήματος για τις ''κρίσιμες'' κλάσεις σε ανεκτά επίπεδα.

Στην παρουσίαση του προβλήματος της ικανοποίησης στόχων επίδοσης για κλάσεις φόρτου θεωρήθηκαν δεδομένοι οι στόχοι επίδοσης για κάθε κλάση. Μια παραλλαγή εμφανίστηκε στο κεφάλαιο 7 όπου ορίστηκαν επιμέρους στόχοι επίδοσης για τα βήματα μιας σύνθετης μονάδας φόρτου. Με βάση την ως τώρα διατύπωση του προβλήματος της διαχείρισης πόρων, το ζητούμενο είναι να ικανοποιηθούν *στατικά καθορισμένοι* στόχοι επίδοσης με την εύρεση κατάλληλης πολιτικής ανάθεσης πόρων. Είναι ενδιαφέρον να ερευνηθεί το πρόβλημα του καθορισμού στόχων επίδοσης, με βάση δεδομένα για τις απαιτήσεις πόρων και την διαμόρφωση και την τρέχουσα κατάσταση του συστήματος. Μια πρώτη προσέγγιση θα μπορούσε να είναι η ανάπτυξη μιας μεθόδου με την οποία να τροποποιούνται δυναμικά οι στόχοι επίδοσης για τις κλάσεις φόρτου ανάλογα με το κατά πόσο οι στόχοι ικανοποιούνται. Η ευαισθησία

των τεχνικών χρονοπρογραμματισμού που υλοποιήθηκαν στον καθορισμό των στόχων επίδοσης συνδέεται άμεσα με το πρόβλημα του δυναμικού καθορισμού των στόχων επίδοσης, συνεπώς η μελέτη αυτού του προβλήματος μπορεί να δώσει ως αποτέλεσμα πιο αποτελεσματικούς αλγορίθμους.

8.2.3 Διαχείριση Σύνθετων Μονάδων Φόρτου

Η κατηγορία των σύνθετων μονάδων φόρτου που μελετήθηκε στα πλαίσια αυτής της εργασίας είναι μια απλή περίπτωση workflows. Είναι σημαντικό να μελετηθούν workflows με πιο σύνθετη δομή. Για τον σκοπό αυτό θα ήταν σκόπιμο να υλοποιηθεί μια σειρά από βασικές πράξεις διαχείρισης δοσοληψιών (transaction management primitives), κατά το πρότυπο των αναφορών [BDG+94, GHMea93], ώστε να είναι δυνατή η μοντελοποίηση δομών ροής ελέγχου και δεδομένων πιο σύνθετων από την απλή δομή αλυσίδας που υποστηρίζει η παρούσα υλοποίηση. Με τον τρόπο αυτό ο προσομοιωτής θα μπορούσε να μοντελοποιήσει την εκτέλεση εφαρμογών που οραγνώνονται ως ένα σύνολο δοσοληψιών βάσει των κανόνων ενός εξελιγμένου μοντέλου δοσοληψιών [Elm92]. Η σχεδίαση του μηχανισμού προσομοίωσης μονάδων φόρτου (βλ. ενότητα 5.4, και ειδικά την παράγραφο 5.4.2) διευκολύνει την επέκταση αυτή, αφού, με δεδομένη μια υλοποίηση των transaction management primitives, ο χρήστης αρκεί να γράψει μια ρουτίνα σε γλώσσα C που να υλοποιεί την ροή ελέγχου και δεδομένων με χρήση αυτών των primitives.

Οπως σημειώνεται στην αναφορά [Den93], η συλλογή και εκτίμηση παραμέτρων επίδοσης για σύνθετες μονάδες φόρτου δυσχεραίνεται από την ανάγκη για συσχέτιση (correlation) μεταξύ των βημάτων που συγκροτούν μια σύνθετη μονάδα φόρτου. Συνεπώς, είναι σημαντικό, ειδικά σε κατανεμημένα συστήματα, να αναπτυχθούν μέθοδοι παρακολούθησης και συλλογής στοιχείων σχετικών με την επίδοση, με κατά το δυνατόν χαμηλότερο κόστος.

Βιβλιογραφία

[AGM88] R. Abbott and H. Garcia−Molina. ''Scheduling Real−Time Transactions: A Performance Evaluation''. In *Proceedings of the 14th International Conference on Very Large Data Bases*, 1988.

[AGM90] R. Abbott and H. Garcia−Molina. ''Scheduling I/O Requests with Deadlines: A Performance Evaluation''. In *IEEE Real−Time System Symposium*, 1990.

[AGMK94] B. Adelberg, H. Garcia−Molina, and B. Kao. ''Emulating Soft Real−Time Scheduling Using Traditional Operating System Schedulers''. In *IEEE Real−Time System Symposium*, 1994.

[Ano85] Anon. ''A Measure of Transaction Processing Power''. *Datamation*, 31(7):112−−118, 1985.

[BCD+92] K. P. Brown, M. J. Carey, D. J. DeWitt, M. Mehta, and J. F. Naughton. ''Resource Allocation and Scheduling for Mixed Database Workloads''. Technical Report TR 1095, Computer Sciences Department, University of Wisconsin−Madison, 1992.

[BCL93a] K. P. Brown, K. J. Carey, and M. Livny. ''Towards an Autopilot in the DBMS Performance Cockpit''. In *Proceedings of the 5th International High Performance Transaction Systems Workshop*, 1993.

[BCL93b] K. P. Brown, M. J. Carey, and M. Livny. ''Managing Memory to Meet Multiclass Workload Response Time Goals''. In *Proceedings of the 19th International VLDB Conference*, 1993. Also available as Technical Report No. 1146, University of Wisconsin.

[BDG+94] A. Biliris, S. Dar, N. Gehani, H.V. Jagadish, and K. Ramamritham. ''ASSET: A System for Supporting Extended Transactions''. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 1994.

[Ber90] P. Bernstein. ''Transaction Processing Monitors''. *Communications of the ACM*, 33(11), November 1990.

[BET91] P. Bernstein, W. Emberton, and V. Trehan. ''DECdta: Digital's Distributed Transaction Processing Architecture''. *Digital Technical Journal*, 3(1), 1991.

[BG81] P. Bernstein and N. Goodman. ''Concurrency Control in Distributed Database Systems''. *ACM Computing Surveys*, 13(2):185−−222, 1981.

[BGT91] P. Bhattacharya, L. Georgiadis, and P. Tsoucas. "Optimal Adaptive Scheduling in Multi−Class $M/GI/1$ Queues with Feedback". In *Proc. of the 29th Allerton Conference on Communication, Control and Computing*, September 1991.

[BHM90] P. Bernstein, M. Hsu, and B. Mann. "Implementing Recoverable Requests using Queues". In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 1990.

[BMCL94] K. P. Brown, M. Mehta, K. J. Carey, and M. Livny. "Towards Automated Performance Tuning for Complex Workloads". In *Proceedings of the 20th International VLDB Conference*, pages 578−−587, 1994.

[Bro94] K. P. Brown. *"PRPL: A Database Workload Specification Language"*. Computer Sciences Department, University of Wisconsin−Madison, 1994.

[CD85] H. Chou and D. DeWitt. "An Evaluation of Buffer Management Strategies for Relational Database Systems". In *Proceedings of the 11th International VLDB Conference*, 1985.

[CDY86] D. W. Cornell, D. M. Dias, and P. S. Yu. "On Multisystem Coupling through Function−Request Shipping". *IEEE Transactions on Software Engineering*, 12(10):1006−−1017, October 1986.

[CFW+95] J. Y. Chung, D. Ferguson, G. Wang, C. Nikolaou, and J. Teng. "Goal−Oriented Dynamic Buffer Pool Management for Database Systems". In *Proceedings of the 1st IEEE International Conference on Engineering of Complex Computer Systems*, 1995. Also available as Technical Report TR−125, ICS−FORTH, October 1994.

[Coa] Workflow Management Coalition. *Glossary − A Workflow Management Coalition Specification*.

[Cou90a] Transaction Processing Council. *TPC Benchmark A : Standard Specification*, 1990.

[Cou90b] Transaction Processing Council. *TPC Benchmark B : Standard Specification*, 1990.

[Cou93] Transaction Processing Council. *TPC Benchmark C : Standard Specification*, 1993.

[Cou95] Transaction Processing Council. *TPC Benchmark D : Standard Specification*, 1995.

[CP84] S. Ceri and S. Pelagatti. *Distributed Databases: Principles and Systems*. McGraw−Hill, 1984.

[D. 84] D. DeWitt and R. Katz and F. Olken and L. Shapiro and M. Stonebraker and D. Wood. "Implementation Techniques for Main Memory Database Systems". In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 1984.

[Den93] P. Denning. "The Fifteenth Level". In *Proceedings of the ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, 1993.

[DG92] D. DeWitt and J. Gray. "Parallel Database Systems: The Future of High Performance Database Processing". *Communications of the ACM*, 35(6), June 1992.

[DGMH+93] U. Dayal, H. Garcia–Molina, M. Hsu, B. Kao, and M.C. Shan. "Third Generation TP Monitors: A Database Challenge". In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 1993.

[DHL93] U. Dayal, M. Hsu, and R. Ladin. "Organizing Long–Running Activities with Triggers and Transactions". In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 1993.

[EH84] W. Effelsberg and T. Haerder. "Principles of Database Buffer Management". *ACM Transactions on Database Systems*, 9(4), 1984.

[Elm92] A. K. Elmagarmid, editor. *Transaction Models for Advanced Database Applications*. Morgan Kaufmann, 1992.

[EMS91] J. L. Eppinger, B. Mummert, and A. Z. Spector. *Camelot and Avalon: A Distributed Transaction Facility*. Morgan Kaufmann, 1991.

[FNGD92] D. Ferguson, C. Nikolaou, L. Georgiadis, and K. Davies. "Goal Oriented, Adaptive Transaction Routing for High Performance Transaction Processing Systems", 1992. IBM Research Report RC18139.

[FNGD93] D. Ferguson, C. Nikolaou, L. Georgiadis, and K. Davies. "Satisfying Response Time Goals in Transaction Processing Systems". In *Proceedings of the 2nd International Conference on Parallel and Distributed Information Systems*, 1993.

[Fuj90] R.M. Fujimoto. "Parallel Discrete Event Simulation". *Communications of the ACM*, 33(10), October 1990.

[GHK+95] G. Georgiannakis, C. Houstis, S. Kapidakis, M. Karavassili, C. Nikolaou, A. Labrinidis, M. Marazakis, E. Markatos, M. Mavronicolas, S. Chabridon, E. Gelenbe, E. Born, L. Richter, and R. Riedl. "Description of the Adaptive Resource Management Problem, Cost functions and Performance objectives". Technical Report 130, ICS/FORTH, Heraklion, Crete, Greece, April 1995. Deliverable WP1/T.1.1–T.1.2/D1 of project LYDIA (available online at ftp.ics.forth.gr:/tech–reports/1995/1995.TR130.LYDIA.D1.ps.gz).

[GHMea93] D. Georgakopoulos, M. Hornik, F. Manola, and et al. "An Extended Transaction Environment for Workflows in Distributed Object Computing". *IEEE Bulletin of the Technical Committee on Data Engineering*, June 1993.

[GHS94] D. Georgakopoulos, M. Hornik, and A. Sheth. "An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure". *Distributed and Parallel Databases*, 3(2), September 1994.

[GK85] A. Gawlick and D. Kinkade. Varieties of Concurrency Control in IMS/VS Fast Path. *Database Engineering*, June 1985.

[GLPT76] J. Gray, R. Lorie, F. Putzolu, and I. Traiger. "granularity of locks and degrees of consistency in a large shared data base". In *"Modeling in Data Base Management Systems"*. Elsevier North Holland, 1976.

[GMS87] H. Garcia−Molina and K. Salem. "SAGAS". In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 1987.

[GR93] J. Gray and A. Reuter. *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann, 1993.

[Gra78] J. Gray. "Notes on Database Operating Systems". In *Operating Systems: An Advanced Course*. Springer−Verlag, 1978.

[Gra81] J. Gray. "The Transaction Concept: Virtues and Limitations". In *Proceedings of the 7th International VLDB Conference*, 1981.

[Gra91] J. Gray. *The Benchmark Handbook for Database and Transaction Processing Systems*. Morgan Kaufmann, 1991.

[GRS94] N. Goodman, S. Rozen, and L. Stein. "Building a Laboratory Information System around a C++−Based Object−Oriented DBMS". In *Proceedings of the 20th International VLDB Conference*, 1994.

[IBM93] IBM Corp. *MVS/ESA Version 4.3 Initialization and Tuning of Guide*, GC28−1643 edition, March 1993.

[IBM94] IBM Corp. *Sysplex Overview − Introducing Data Sharing and Parallelism in a Sysplex*, GC28−1208−00 edition, April 1994.

[Kag89] Y. Kageyama. *CICS Handbook*. Intertext Publications, McGraw−Hill, 1989.

[KGM93] B. Kao and H. Garcia−Molina. "Deadline Assignment in a Distributed Soft Real−Time System". In *Proc. 13−th Int. Conf. on Distr. Comp. Syst.*, 1993.

[KP84] B. W. Kernigham and R. Pike. *The UNIX Programming Environment*. Prentice Hall, 1984.

[Lab95] A. Labrinidis. "Methods to cluster transactions into utilization classes with similar workload characteristics (MSc thesis)". Technical Report TR−134, FORTH−ICS Heraklion, Crete, Greece, August 1995.

[Lis88] B. Liskov. "Distributed Programming in Argus". *Communications of the ACM*, 31(3):300−−312, 1988.

[Liv89] M. Livny. *DeNet User's Guide*. Computer Sciences Department, University of Wisconsin−Madison, 1989.

[LLM88] M. J. Litzkow, M. Livny, and M. W. Mutka. "Condor − A Hunter of Idle Workstations". In *Proceedings of the 8th International Conference on Distributed Computing Systems*, 1988.

[MHPS92] C. Mohan, D. Haderle, H. Pirahesh, and P. Schwarz. "ARIES: A Transaction Recovery Method Supporting Fine−Granularity Locking and Partial Rollbacks Using Write−Ahead Logging". *ACM Transactions on Database Systems*, 17(1), 1992.

[MM93] M. W. Mutka and P. K. McKinley. "Supporting a Simulation Environment with OpenSim". *Simulation*, 61(4):223−−235, 1993.

[MN95] M. Marazakis and C. Nikolaou. "Towards Adaptive Scheduling of Tasks in Transactional Workflows". In *Proceedings of the Winter Simulation Conference*, 1995.

[Mos85] J. E. B. Moss. *Nested Transactions: An Approach to Reliable Distributed Computing*. MIT Press, 1985.

[Mut92] M. W. Mutka. Estimating Capacity For Sharing in a Privately Owned Workstation Environment. *IEEE Transactions on Software Engineering*, 18(4):319−−328, April 1992.

[Nei91] J. E. Neilson. "PARASOL: A Simulator for Distributed and/or Parallel Systems", 1991. Technical Report SCS−TR−192,Carleton University, Canada.

[Nei94] J. E. Neilson. *PARASOL User's Guide* . Carleton University, Canada, 1994.

[NFC92] C. Nikolaou, D. Ferguson, and P. Constantopoulos. "Towards Goal Oriented Resource Management", 1992.

[Noo89] J. Noonan. "Automated Service Level Management and its Supporting Technologies". *Mainframe Journal*, October 1989.

[Nov93] "Programming a Distributed Application: The TUXEDO System Approach". Technical report, Novell Corp., January 1993.

[OV91] M. Ozsu and P. Valduriez. *"Principles of Distributed Database Systems"*. Prentice Hall, 1991.

[PLC92] H. Pang, M. Livny, and M. J. Carey. "Transaction Scheduling in Multiclass Real−Time Database Systems". In *Proceedings of the IEEE Real−Time Systems Symposium*, 1992.

[Rah92] E. Rahm. "A Framework for Workload Allocation in Distributed Transaction Processing Systems". *Journal of Systems Software*, 18:171−−190, 1992.

[RH83] A. Reuter and T. Haerder. "Principles of Transaction−Oriented Database Recovery". *ACM Computing Surveys*, 15(4):237−−318, 1983.

[RS94] S. Rozen and L. Stein. "Constructing a Domain−Specific DBMS using a Persistent Object System". In *Proceedings of the 6th International Workshop on Persistent Object Systems*, 1994.

[SC81] C. H. Sauer and K. M. Chandy. *Computer Systems Performance Modeling*. Prentice Hall, 1981.

[Sch94] H. Schwetman. *CSIM User's Guide*. Mesquite Inc., 1994.

[Sel79] P.G. Sellinger. "Access Path Selection in a Relational Database Management System". In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 1979.

[Sel92] M. Seltzer. *File System Performance and Transaction Support*. PhD thesis, University of California at Berkeley, 1992.

[She93] M. Sherman. "Architecture of the Encina Distributed Transaction Processing Family". In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 1993.

[SJR91] P. M. Spiro, A. M. Joshi, and T. K. Rengarajan. "Designing an Optimized Transaction Commit Protocol". *Digital Technical Journal*, 3(1), 1991.

[SO92] M. Seltzer and M. Olson. "LIBTP: Portable, Modular Transactions for UNIX". In *Proceedings of the Winter Usenix*, 1992.

[SS91] T. Speer and M. Storm. "Digital's TP Monitors". *Digital Technical Journal*, 3(1), 1991.

[Wes76] G. O. Wesolowsky. *Multiple Regression and Analysis of Variance*. John Wiley & Sons, 1976.

[WHMZ93a] G. Weikum, C. Hasse, A. Monkeberg, and P. Zabback. "The COMFORT Automatic Tuning Project". Technical report, Department of Computer Science of the Swiss Federal Institute of Technology, Zurich, Switzerland, 1993.

[WHMZ93b] G. Weikum, C. Hasse, A. Monkeberg, and P. Zabback. "The COMFORT Prototype: A Step Towards Automated Database Performance Tuning". In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1993.

[YBL88] P. S. Yu, S. Balsamo, and Y. H. Lee. "Dynamic Transaction Routing in Distributed Database Systems". *IEEE Transactions on Software Engineering*, 14(9):1307−−1318, September 1988.

[YCDI86] P. S. Yu, D. W. Cornell, D. M. Dias, and B. R. Iyer. "On Affinity Based Routing in Multi−System Data Sharing". In *Proceedings of the 12th International VLDB Conference*, Kyoto, Japan, 1986.

[YD91a] P. S. Yu and A. Dan. "Effect of System Dynamics on Coupling Architectures for Transaction Processing", 1991. IBM Research Report RC16606.

[YD91b] P. S. Yu and A. Dan. "Impact of Affinity on the Performance of Coupling Architectures for Transaction Processing", 1991. IBM Research Report RC16431.

[YDR+87] P. S. Yu, D. M. Dias, J. T. Robinson, B. R. Iyer, and D. W. Cornell. "On Coupling Multi−Systems Through Data Sharing". *Proceeding of the IEEE*, 75(5), May 1987.

[YL91] P. S. Yu and A. Leff. ''On Robust Transaction Routing and Load Sharing''. *ACM Trans. Database Syst.*, 16(3):476−−512, September 1991.

[YTJ91] M. W. Young, D. S. Thomson, and E. Jaffe. ''A Modular Architecture for Distributed Transaction Processing''. In *Proceedings of the Winter Usenix*, 1991.

Παράρτημα Α: Η Γραμματική της Γλώσσας Προδιαγραφής

Στο παράρτημα αυτό δίδεται η γραμματική της γλώσσας προδιαγραφής που δέχεται ο προσομοιωτής TPsim. Η γραμματική παρουσιάζεται στην μορφή που αναγνωρίζει η γεννήτρια μεταφραστών YACC. Με κεφαλαία σημειώνονται τα τελικά σύμβολα (tokens) της γλώσσας.

```
SysConfigFile:  SYS_CONFIG_TOKEN  SystemConfiguration
                DATA_DISTRIBUTION_TOKEN  DataDistribution
                WORKLOAD_DESCRIPTION_TOKEN  WorkloadDescription
                PerformanceGoalSpecificationList
              ;

SystemConfiguration:   ComponentDefinition
                   |   SystemConfiguration ComponentDefinition
                   ;

DataDistribution:   DataDistributionDefinition
                |   DataDistribution DataDistributionDefinition
                ;

WorkloadDescription:   WorkloadClassList  ClientClassList  WorkloadLocalization
                   ;

ComponentDefinition:   IoDeviceDefinition
                   |   NodeClassDefinition
                   |   NodeClusterDefinition
                   |   NodeDefinition
                   |   LinkClassDefinition
                   |   LinkDefinition
                   ;

IoDeviceDefinition:   DEFINE  IO_DEVICE  ID  WITH  '{'
                      IO_COPY_DELAY_TOKEN  ':'  REALNUM  ';'
                      IO_LOAD_DELAY_TOKEN  ':'  REALNUM  ';'
                      IO_SEEK_DELAY_TOKEN  ':'  REALNUM  ';'
                      IO_ROT_DELAY_TOKEN  ':'  REALNUM  ';'
                      NUM_HEADS  ':'  INTNUM  ';'
                      NUM_BLOCKS  ':'  INTNUM  ';'  '}'   {

                  |   DEFINE  IO_DEVICE  ID  WITH  '{'
                      IO_DELAY_MIN_TOKEN  ':'  REALNUM  ';'
                      IO_DELAY_MAX_TOKEN  ':'  REALNUM  ';'   '}'
                  ;
```

```
NodeDefinition:   DEFINE  NODE  ID  OF  CLASS  ID  ';'
                | DEFINE  NODE  ID  OF  CLASS  ID
                    '('  FRONT_END_TOKEN  ')'  ';'
                | DEFINE  NODE  ID  OF  CLASS  ID
                    '('  GATEWAY_TOKEN  ')'  ';'
                ;

NodeClusterDefinition:   DEFINE  NODE_CLUSTER  ID  OF  CLASS  ID  WITH  '{'
                           NUM_NODES  ':'  INTNUM  ';'
                           PACKET_SIZE  ':'  INTNUM  ';'
                           TRANSFER_RATE  ':'  INTNUM  ';'
                           BUS_DISCIPLINE  ':'  ID  ';'  '}'
                       | DEFINE  NODE_CLUSTER  ID  WITH  '{'
                           NUM_NODES  ':'  INTNUM  ';'
                           PACKET_SIZE  ':'  INTNUM  ';'
                           TRANSFER_RATE  ':'  INTNUM  ';'
                           BUS_DISCIPLINE  ':'  ID  ';'
                           MEMBER_NODES  ':'  '{'  ID_List  '}'  '}'
                      ;

NodeClassDefinition:   DEFINE  NODE_CLASS_TOKEN  ID  WITH  '{'
                         CPU_CNT  ':'  INTNUM  ';'
                         MPL_TOKEN  ':'  INTNUM  ';'
                         CPU_RATE  ':'  REALNUM  ';'
                         ATTACH_TASK_COST_TOKEN  ':'  REALNUM  ';'
                         DM_INTERFACE_COST_TOKEN  ':'  REALNUM  ';'
                         DM_CALL_COST_TOKEN  ':'  REALNUM  ';'
                         DM_IO_COST_TOKEN  ':'  REALNUM  ';'
                         FUNCTION_SHIP_SEND_COST_TOKEN  ':'  REALNUM  ';'
                         FUNCTION_SHIP_RECV_COST_TOKEN  ':'  REALNUM  ';'
                         PRIMARY_PREPARE_COST_TOKEN  ':'  REALNUM  ';'
                         SEND_PREPARE_COST_TOKEN  ':'  REALNUM  ';'
                         SEND_COMMIT_COST_TOKEN  ':'  REALNUM  ';'
                         PRIMARY_COMMIT_COST_TOKEN  ':'  REALNUM  ';'
                         RECV_PREPARE_COST_TOKEN  ':'  REALNUM  ';'
                         RECV_COMMIT_COST_TOKEN  ':'  REALNUM  ';'
                         LOG_IO_COST_TOKEN  ':'  REALNUM  ';'
                         SECONDARY_PREPARE_COST_TOKEN  ':'  REALNUM  ';'
                         SECONDARY_COMMIT_COST_TOKEN  ':'  REALNUM  ';'
                         DETACH_TASK_COST_TOKEN  ':'  REALNUM  ';'
                         DM_BUFFER_SIZE_TOKEN  ':'  INTNUM  ';'
                         LOG_IO_DEVICE_TOKEN  ':'  ID  ';'
                         DM_IO_DEVICE_TOKEN  ':'  ID  ';'
                         NUM_DISKS  ':'  INTNUM  ';'  '}'
                     ;

LinkClassDefinition:   DEFINE  LINK_CLASS_TOKEN  ID  WITH  '{'
                         PACKET_SIZE  ':'  INTNUM  ';'
                         TRANSFER_RATE  ':'  INTNUM  ';'  '}'
                     ;

LinkDefinition:   DEFINE  LINK  ID  OF  CLASS  ID  WITH  '{'
                  SRC  ':'  ID  ';'
                  DST  ':'  ID  ';'  '}'
                ;

DataDistributionDefinition:   RelationDefinition
                            | AliasDefinition
                            | InstanceDefinition
                            | FragmentDefinition
                            ;
```

```
ID_List:   ID
       |   ID_List  ’,’  ID
       ;

RelationDefinition:   DEFINE  RELATION  ID  WITH  ’{’
                      RelationSchema
                      TUPLES_PER_PAGE  ’:’  REALNUM  ’;’
                      KEY  EQ  ’{’  ID_List  ’}’  ’;’
                      INDEX  ON  ’{’  ID_List  ’}’  ’;’  ’}’
                  ;
                  |   DEFINE  RELATION  ID  WITH  ’{’
                      RelationSchema
                      TUPLES_PER_PAGE  ’:’  REALNUM  ’;’
                      KEY  EQ  ’{’  ID_List  ’}’  ’;’
                      CLUSTERED_INDEX  ON  ’{’  ID_List  ’}’  ’;’
                      INDEX  ON  ’{’  ID_List  ’}’  ’;’ ’}’
                  ;

AttrDefList:    AttrDef
           |    AttrDefList  AttrDef
           ;

AttrDef:   ID  ’:’  NUMERIC  ’;’
       |   ID  ’:’  SYMBOLIC  ’;’
       ;

RelationSchema:  ATTRIBUTES  ’{’  AttrDefList  ’}’
              ;

InstanceDefinition:   DEFINE  INSTANCE
                      OF  RELATION  ID  AT  ALL_SITES_TOKEN
                      WITH  NUM_ITEMS  ’:’  INTNUM  ’;’
                  |   DEFINE  INSTANCE
                      OF  RELATION  ID  AT
                      ’{’  ID_List  ’}’
                      WITH  NUM_ITEMS  ’:’  INTNUM  ’;’
                  ;

FragmentDefinition:   HorizontalFragmentDefinition
                  |   VerticalFragmentDefinition
                  ;

Predicate:   ’(’ SimplePredicate  ’)’
         |   ’(’  Predicate  ’)’
         |   NOT  ’(’  Predicate  ’)’
         |   ’(’  Predicate  AND  Predicate  ’)’
         |   ’(’  Predicate  OR  Predicate  ’)’
         ;

SimplePredicate:   ID  EQ  ID
               |   ID  NEQ  ID
               |   ID  EQ  INTNUM
               |   ID  NEQ  INTNUM
               |   ID  LEQ  INTNUM
               |   ID  GEQ  INTNUM
               |   ID  LT  INTNUM
               |   ID  GT  INTNUM
               |   ID  EQ  REALNUM
               |   ID  NEQ  REALNUM
               |   ID  LEQ  REALNUM
```

```
              |   ID  GEQ  REALNUM
              |   ID  LT  REALNUM
              |   ID  GT  REALNUM
              ;

H_Fragm_List:   ID  WITH  Predicate  ',’  TUPLES_PER_PAGE  ':’  REALNUM  ';’
            |   H_Fragm_List
                ID  WITH  Predicate  ',’  TUPLES_PER_PAGE  ':’  REALNUM  ';’
            ;

HorizontalFragmentDefinition:   DEFINE  HORIZONTAL  FRAGMENTATION  OF  ID
                                AS  '{’  H_Fragm_List  '}’
                           ;

V_Fragm_List:   ID  WITH  COLUMNS  '{’  ID_List  '}’  ',’
                TUPLES_PER_PAGE  ':’   REALNUM  ';’
            |   V_Fragm_List
                ID  WITH  COLUMNS  '{’  ID_List  '}’  ',’
                TUPLES_PER_PAGE  ':’   REALNUM  ';’
            ;

VerticalFragmentDefinition:   DEFINE  VERTICAL  FRAGMENTATION  OF  ID
                              AS  '{’  V_Fragm_List  '}’
                         ;

WorkloadClassList:  WorkloadClassDefinition
                 |  WorkloadClassList  WorkloadClassDefinition
                 ;

ClientClassList:   ClientClassDefinition
               |   ClientClassList  ClientClassDefinition
               ;

TxClassDistributionList:   TxClassDistribution
                       |   TxClassDistributionList
                           TxClassDistribution
                       ;

TxClassDistribution:  SUBMIT  ID  WITH
                      PROBABILITY_TOKEN  REALNUM  ';’
                  ;

ClientClassDefinition:   DEFINE  CLIENT_CLASS_TOKEN  ID  AS  '{’
                         ARRIVAL_RATE  ':’  REALNUM  ';’
                         TxClassDistributionList
                         '}’
                     ;

ReferenceLocalityDefinition:   InstanceReferenceLocality
                           |   ReferenceLocalityDefinition
                               InstanceReferenceLocality
                           ;

InstanceReferenceLocality:   ACCESS_PROBABILITY  OF INSTANCE
                             ID  ':’  REALNUM  ';’
                             WRITE_PROBABILITY  OF INSTANCE
                             ID  ':’  REALNUM  ';’
                         ;

WorkloadClassDefinition:   DEFINE  TRANSACTION_CLASS  ID  AS  '{’
                           APPLICATION_BURST  ':’  REALNUM  ';’
```

```
                        MIN_BLOCKS_ACCESSED  ':'  INTNUM  ';'
                        MAX_BLOCKS_ACCESSED  ':'  INTNUM  ';'
                        ReferenceLocalityDefinition   '}'
                    |   DEFINE  TRANSACTION_CLASS  ID  AS  '{'
                        APPLICATION_BURST  ':'  REALNUM  ';'
                        MIN_BLOCKS_ACCESSED  ':'  INTNUM  ';'
                        MAX_BLOCKS_ACCESSED  ':'  INTNUM  ';'
                        PROBABILITY_IO_TOKEN  ':'  REALNUM  ';'
                        IO_DELAY_MIN_TOKEN  ':'  REALNUM  ';'
                        IO_DELAY_MAX_TOKEN  ':'  REALNUM  ';'
                        ReferenceLocalityDefinition   '}'
                    |   DEFINE  TRANSACTION_CLASS  ID  AS  '{'
                        APPLICATION_BURST  ':'  REALNUM  ';'
                        DMLstmtList  '}'
                    |   DEFINE  WORKFLOW_CLASS  ID  AS  '{'
                        PROGRAM_TOKEN  ':'  ID  ';'
                        CHAIN_TOKEN  ':'  '{'  ID_List  '}'  ';'  '}'
                    ;

VarList:   Var
       |   VarList  ','  Var
       ;

Var:   ID  FROM_TOKEN ID
   ;

DMLstmtList:   DMLstmt
           |   DMLstmtList  DMLstmt
           ;

AccessMethod:   ACCESS_METHOD_TOKEN  OF  ID  ':'
                   SCAN_CLUSTERED_INDEX_TOKEN  ';'
            |   ACCESS_METHOD_TOKEN  OF  ID  ':'
                   SCAN_UNCLUSTERED_INDEX_TOKEN  ';'
            |   ACCESS_METHOD_TOKEN  OF  ID  ':'
                   SCAN_SEQUENTIAL_TOKEN  ';'
            ;

AccessMethodList:   AccessMethod
                |   AccessMethodList  AccessMethod
                ;

Selectivity:   SELECTIVITY_TOKEN  OF ID  ':'  REALNUM  ';'
           ;

SelectivityList:   Selectivity
               |   SelectivityList  Selectivity
               ;

DMLstmt:   SELECT_TOKEN  VarList  WHERE_TOKEN  Predicate   WITH
           '{'  MERGE_COST_TOKEN  ':'  REALNUM  ';'
                AccessMethodList  SelectivityList  '}'  ';'
       ;

ClientClassDistributionList:   ClientClassDistribution
                           |   ClientClassDistributionList
                               ClientClassDistribution
                           ;

ClientClassDistribution:   NUM_CLIENTS  OF  CLASS  ID  ':'  INTNUM
                           AT  ALL_SITES_TOKEN  ';'
```

```
                        |   NUM_CLIENTS  OF  CLASS  ID  ':'  INTNUM
                            AT  '{'  ID_List  '}'  ';'
                        ;

WorkloadLocalization:   DEFINE  SYNTHETIC  WORKLOAD_TOKEN  ID  AS  '{'
                        ClientClassDistributionList  '}'
                    ;

PerformanceGoalSpecificationList:   /* empty */
                                |   PerformanceGoalSpecificationList
                                    PerformanceGoalSpecification
                                ;

PerformanceGoalSpecification:   DEFINE  PERFORMANCE_GOAL_TOKEN  OF  CLASS
                                    ID  ':'  REALNUM  ';'
                            ;
```

Παράρτημα Β: Μοντέλο Διατυπωμένο στην Γλώσσα Προδιαγραφής

Στο παράρτημα αυτό παρουσιάζεται ένα πλήρες παράδειγμα μοντέλου προσομοίωσης διατυπωμένο στην γλώσσα προδιαγραφής του προσομοιωτή TPsim. Το μοντέλο αυτό μάλιστα χρησιμοποιήθηκε για την πειραματική μελέτη της επίδοσης αλγορίθμων χρονοπρογραμματισμού για σύνθετες μονάδες φόρτου, που παρουσιάστηκε στο κεφάλαιο 7.

```
**SYSTEM CONFIGURATION**

% definition of type of disk used for logging
DEFINE IO_DEVICE LogDisk WITH {
   IO_COPY_DELAY: 0.001;
   IO_LOAD_DELAY: 0.00075;
   IO_SEEK_DELAY: 0.00075;
   IO_ROTATIONAL_DELAY: 0.008333;
   NUM_HEADS: 8;
   NUM_SECTORS: 250000;
}

% definition of type of disk used for data storage
DEFINE IO_DEVICE DmDisk WITH {
   IO_COPY_DELAY: 0.001;
   IO_LOAD_DELAY: 0.00075;
   IO_SEEK_DELAY: 0.00075;
   IO_ROTATIONAL_DELAY: 0.008333;
   NUM_HEADS: 8;
   NUM_SECTORS: 250000;
}


% definition of transaction processing node type
DEFINE NODE_CLASS nodeType WITH {
   CPUcnt: 1;
   MPL: 200;
   CPUrate: 50.0;  % measured in MIPS
   % The following costs are measured in instruction counts ...
   ATTACH_TASK_COST: 15100.0;
   DM_INTERFACE_COST: 2000.0;
   DM_CALL_COST: 4000.0;
   DM_IO_COST: 10000.0;
   FUNCTION_SHIP_SEND_COST: 12600.0;
   FUNCTION_SHIP_RECV_COST: 12600.0;
   PRIMARY_PREPARE_COST: 7000.0;
   SEND_PREPARE_COST: 12600.0;
```

```
   SEND_COMMIT_COST: 12600.0;
   PRIMARY_COMMIT_COST: 14000.0;
   RECV_PREPARE_COST: 12600.0;
   RECV_COMMIT_COST: 12600.0;
   LOG_IO_COST: 5000.0;
   SECONDARY_PREPARE_COST: 12000.0;
   SECONDARY_COMMIT_COST: 12000.0;
   DETACH_TASK_COST: 15100.0;
   DM_BUFFER_SIZE: 20000; % total: 160 MBytes
   LOG_IO_DEVICE: LogDisk;  % name of I/O device used for logging
   DM_IO_DEVICE:  DmDisk;   % name of I/O device used for data storage
   numDisks: 2;             % # disks with DB files
}

% definition of front-end node type
DEFINE NODE_CLASS nodeType_FE WITH {
   CPUcnt: 1;
   MPL: 200;
   CPUrate: 50.0;  % measured in MIPS
   % The following costs are measured in instruction counts ...
   ATTACH_TASK_COST: 15100.0;
   DM_INTERFACE_COST: 2000.0;
   DM_CALL_COST: 4000.0;
   DM_IO_COST: 10000.0;
   FUNCTION_SHIP_SEND_COST: 12600.0;
   FUNCTION_SHIP_RECV_COST: 12600.0;
   PRIMARY_PREPARE_COST: 7000.0;
   SEND_PREPARE_COST: 12600.0;
   SEND_COMMIT_COST: 12600.0;
   PRIMARY_COMMIT_COST: 14000.0;
   RECV_PREPARE_COST: 12600.0;
   RECV_COMMIT_COST: 12600.0;
   LOG_IO_COST: 5000.0;
   SECONDARY_PREPARE_COST: 12000.0;
   SECONDARY_COMMIT_COST: 12000.0;
   DETACH_TASK_COST: 15100.0;
   DM_BUFFER_SIZE: 5000; % total: 40 MBytes
   LOG_IO_DEVICE: LogDisk;  % name of I/O device used for logging
   DM_IO_DEVICE:  DmDisk;   % name of I/O device used for data storage
   numDisks: 2;             % # disks with DB files
}

% definition of a front-end and a back-end (processing) node
DEFINE NODE TPS_node_1 OF CLASS nodeType_FE;
DEFINE NODE TPS_node_2 OF CLASS nodeType;

DEFINE NODE_CLUSTER TPS % Transaction Processing System
   WITH {
      numNodes: 2;
      packetSize: 1024; % measured in bytes
      transferRate: 500000; % measured in bytes per sec (4 MBit).
      MEMBER_NODES: { TPS_node_1, TPS_node_2}
   }

DEFINE NODE TPS_node_1 OF CLASS nodeType_FE (FRONT_END) ;

**DATA DISTRIBUTION**

DEFINE RELATION DB_HOT WITH {
   ATTRIBUTES {
           A_ID: NUMERIC;
```

```
            B_ID: NUMERIC;
             num: NUMERIC;
   }
   tuplesPerPage: 100.0;
   KEY = {A_ID};
   INDEX ON {A_ID, B_ID};
}

DEFINE RELATION DB_COLD WITH {
   ATTRIBUTES {
            A_ID: NUMERIC;
            B_ID: NUMERIC;
             num: NUMERIC;
   }
   tuplesPerPage: 100.0;
   KEY = {A_ID};
   INDEX ON {A_ID, B_ID};
}

% 'instance' definitions
DEFINE INSTANCE OF RELATION DB_HOT AT
   {TPS_node_2} WITH numItems: 10000;

DEFINE INSTANCE OF RELATION DB_COLD AT
   {TPS_node_2} WITH numItems: 40000;

% Total DB size: 50000 pages
% Buffer size: 20000 pages (40% of DB size)

**WORKLOAD DESCRIPTION**

% definition of transaction classes A, B, C, D
DEFINE TRANSACTION_CLASS  TX_CLASS_A AS {
   applicationBurst: 40000.0;
   blocksAccessed_min: 1;
   blocksAccessed_max: 4;
   accessProbability OF INSTANCE Account_COLD: 0.2;
   writeProbability OF INSTANCE  Account_COLD: 0.8;
   accessProbability OF INSTANCE Account_HOT: 0.8;
   writeProbability OF INSTANCE  Account_HOT: 0.8;
}

DEFINE TRANSACTION_CLASS  TX_CLASS_B AS {
   applicationBurst: 60000.0;
   blocksAccessed_min: 1;
   blocksAccessed_max: 8;
   accessProbability OF INSTANCE Account_COLD: 0.2;
   writeProbability OF INSTANCE  Account_COLD: 0.8;
   accessProbability OF INSTANCE Account_HOT: 0.8;
   writeProbability OF INSTANCE  Account_HOT: 0.8;
}

DEFINE TRANSACTION_CLASS  TX_CLASS_C AS {
   applicationBurst: 80000.0;
   blocksAccessed_min: 1;
   blocksAccessed_max: 12;
   accessProbability OF INSTANCE Account_COLD: 0.2;
   writeProbability OF INSTANCE  Account_COLD: 0.8;
   accessProbability OF INSTANCE Account_HOT: 0.8;
   writeProbability OF INSTANCE  Account_HOT: 0.8;
}
```

```
DEFINE TRANSACTION_CLASS  TX_CLASS_D AS {
   applicationBurst: 100000.0;
   blocksAccessed_min: 1;
   blocksAccessed_max: 16;
   accessProbability OF INSTANCE Account_COLD: 0.2;
   writeProbability OF INSTANCE  Account_COLD: 0.8;
   accessProbability OF INSTANCE Account_HOT: 0.8;
   writeProbability OF INSTANCE  Account_HOT: 0.8;
}

% definition of workflow classes WC1, WC2
DEFINE WORKFLOW_CLASS WC1 AS {
   program: TxChain;
   chain :  { TX_CLASS_A, TX_CLASS_B, TX_CLASS_D };
}

DEFINE WORKFLOW_CLASS WC2 AS {
   program: TxChain;
   chain :  { TX_CLASS_C, TX_CLASS_A, TX_CLASS_B, TX_CLASS_D, TX_CLASS_C };
}

% definition of client class
DEFINE CLIENT_CLASS clientClass AS {
   arrivalRate: 4.0;   % 'think time' for user (in sec's)
   SUBMIT WC1 WITH PROBABILITY 0.5;
   SUBMIT WC2 WITH PROBABILITY 0.5;
}

DEFINE SYNTHETIC WORKLOAD wrkLoad AS {
   numClients OF CLASS clientClass: 150 AT {TPS_node_1};
}

% definition of performance goals for workflow classes
DEFINE PERFORMANCE_GOAL OF CLASS WC1 : 3.0;
DEFINE PERFORMANCE_GOAL OF CLASS WC2 : 5.0;
```

Παράρτημα Γ: Παράδειγμα Προδιαγραφής Πειράματος

Στο παράρτημα αυτό δίδεται ένα παράδειγμα προδιαγραφής πειράματος προσομοίωσης στην τυπική γλώσσα που δέχεται ο διερμηνέας της διεργασίας διαχείρισης πειραμάτων που περιγράφηκε στο κεφάλαιο 6. Το παράδειγμα αυτό περιγράφει ένα από τα πειράματα προσομοίωσης για την αξιολόγηση των αλγορίθμων χρονοπρογραμματισμού για σύνθετες μονάδες φόρτου, που παρουσιάστηκαν στο κεφάλαιο 7.

```
Experiment Title:   EXP-S-STATIC

Simulation Model Specification:  workflowModel.1

Model Type:   closed

Simulation Time: 3600.0

Routing Algorithm: JSQ

Scheduling Algorithm: STATIC_GOAL

Number of Replications:  5

Independent Variable: clientClass.TPS_node_1

Number of Points: 4

BEGIN_DATA_POINT_SPEC
clientClass.TPS_node_1 : 100
WC1.goal: 3.0
WC2.goal: 5.0
END_DATA_POINT_SPEC

BEGIN_DATA_POINT_SPEC
clientClass.TPS_node_1 : 125
WC1.goal: 3.0
WC2.goal: 5.0
END_DATA_POINT_SPEC

BEGIN_DATA_POINT_SPEC
clientClass.TPS_node_1 : 150
WC1.goal: 3.0
WC2.goal: 5.0
END_DATA_POINT_SPEC

BEGIN_DATA_POINT_SPEC
```

```
clientClass.TPS_node_1 : 175
WC1.goal: 3.0
WC2.goal: 5.0
END_DATA_POINT_SPEC
```