Ημερομηνία παράδοσης: 01/04/2011

Εργασία 3

Εισαγωγή:

Οι εξυπηρετητές στον Παγκόσμιο Ιστό (Web Servers) είναι υπεύθυνοι να εξυπηρετούν καθημερινά ένα πολύ μεγάλο αριθμό από αιτήσεις (requests). Πολλές από αυτές τις αιτήσεις γίνονται παράλληλα από πολλαπλούς χρήστες (Web clients).
Θα ήταν πολύ δύσκολο αν όχι αδύνατο, οι εξυπηρετητές να εξυπηρετούσαν αυτό το μεγάλο αριθμό αιτήσεων χωρίς την χρήση κρυφής μνήμη (cache).
Η χρήση cache χρησιμοποιείται στον Παγκόσμιο Ιστό σε πολλά επίπεδα. Μπορεί να χρησιμοποιηθεί εσωτερικά από κάποιο εξυπηρετητή για να ανακτήσει πιο γρήγορα τις πληροφορίες που χρειάζεται ή μπορεί να χρησιμοποιηθεί εξωτερικά με τη χρήση κάποιου ενδιάμεσου εξυπηρετητή (proxy server), ο οποίος απαντά αυτόματα αιτήσεις χρηστών, σε περίπτωση που παρόμοιες αιτήσεις είχαν διεκπεραιωθεί πρόσφατα και τις είχε φυλάξει στην μνήμη του.
Πιο συγκεκριμένα, οι Web proxy servers χρησιμοποιούν cache για να αποθηκεύουν προηγούμενες αιτήσεις από εξυπηρετητές, κυρίως για ιστοσελίδες. Η τεχνική Web caching μειώνει το ποσό της πληροφορίας που πρέπει να μεταδοθεί στο δίκτυο, επαναχρησιμοποιώντας πληροφορίες που είναι ήδη αποθηκευμένες στο cache. Αυτό μειώνει τις ανάγκες σε επεξεργασία και bandwidth από τους εξυπηρετητές, αυξάνοντας την συνολική ανταπόκριση προς τους χρήστες (user responsiveness).

Βασική Εκφώνηση Εργασίας:

Στην προηγούμενη άσκηση είδαμε πως χρησιμοποιούμε threads και multithreading, για να υποστηρίξουμε πολλούς χρήστες ταυτόχρονα σε εξυπηρετητές του Παγκόσμιου Ιστού. Στην εργασία αυτή προχωρούμε ένα βήμα παραπέρα, χρησιμοποιώντας κρυφή μνήμη (cache) για να βελτιώσουμε την απόδοση του συστήματος μας.
Σκοπός αυτής της εργασίας είναι να επεκτείνετε τον εξυπηρετητή που δημιουργήσατε στην Εργασία 2, ώστε να χρησιμοποιεί την εσωτερική κρυφή μνήμη του, για να απαντά παρόμοιες αιτήσεις, που γίνονται από διάφορους χρήστες ταυτόχρονα ή με μικρή χρονική διαφορά. Οι αιτήσεις αυτές, όπως και στις προηγούμενες δύο εργασίες, θα αφορούν το διάβασμα διαφόρων αρχείων που βρίσκονται στον εξυπηρετητή.

Θα πρέπει να μελετήσετε τις διαφοροποιήσεις που υπάρχουν από τη χρήση κρυφής μνήμης. Επομένως, πρέπει να πάρετε δυο σύνολα μετρήσεων, ένα για λειτουργία χωρίς κρυφή μνήμη και ένα για λειτουργία με κρυφή μνήμη στην κύρια μνήμη του εξυπηρετητή. Θα πρέπει να εξετάσετε την εφαρμογή σας χρησιμοποιώντας διαφορετικό αριθμό (ταυτόχρονων) χρηστών, διαφορετικό αριθμό αιτήσεων καθώς και διαφορετικό αριθμό αρχείων. Ποιο είναι το cache hit σε κάθε περίπτωση; Πώς επηρεάζει το σύστημα η χρήση διαφορετικού μέγιστου χρόνου αποθήκευσης αρχείων στο cache;

Επιπλέον Λειτουργικότητα:

Σε επόμενο στάδιο, θα πρέπει να δημιουργήσετε κάποιο ενδιάμεσο εξυπηρετητή (proxy server), ο οποίος θα αναλαμβάνει να απαντά αιτήσεις χρηστών χωρίς να χρειάζεται να επικοινωνεί με τον υπεύθυνο εξυπηρετητή. Όπως προαναφέραμε, οι ενδιάμεσοι εξυπηρετητές (proxy servers), έχουν την ικανότητα να απαντούν αυτόματα αιτήσεις χρηστών, σε περίπτωση που παρόμοιες αιτήσεις είχαν διεκπεραιωθεί στο πρόσφατο παρελθόν και είχαν αποθηκευτεί στην μνήμη του proxy server. Όπως και πριν, θα πρέπει να πάρετε δυο σύνολα μετρήσεων, ένα για λειτουργία χωρίς τον proxy server και ένα για λειτουργία με τον proxy server.
Επίσης, θα πρέπει να απαντήσετε τις ερωτήσεις που απαντήσατε και πριν. Πως επηρεάζει ο αριθμός των χρηστών, ο αριθμός των αιτήσεων καθώς και ο αριθμός των αρχείων την απόδοση του συστήματος; Ποιο είναι το cache hit σε αυτή τη περίπτωση; Πώς επηρεάζει το σύστημα η χρήση διαφορετικού χρόνου αποθήκευσης αρχείων στο cache;

Σημειώσεις:

- Όπως και στην πρώτη άσκηση, το πρωτόκολλο επικοινωνίας θα πρέπει είναι στο επίπεδο του TCP/IP μέσω socket programming.
- Μπορείτε να θεωρήσετε ότι η κρυφή μνήμη του δικού σας εξυπηρετητή είναι η κύρια μνήμη (ενώ γενικά τα αρχεία βρίσκονται αποθηκευμένα στη δευτερεύουσα μνήμη).
- Θα πρέπει να χρησιμοποιήσετε κάποιο τυχαίο τρόπο (randomness), όσον αφορά την επιλογή αρχείων από τους clients κάθε φορά.

- Όσοι φοιτητές δεν έχουν καταφέρει να υλοποιήσουν την δεύτερη άσκηση, μπορούν να επικοινωνήσουν με το βοηθό του μαθήματος Αντρέα Καμηλάρη, για να τους στείλει ένα πρότυπο (template) πάνω στο οποίο να μπορούν να δουλέψουν για τις ανάγκες αυτής της εργασίας. Το πρότυπο αυτό θα είναι στη Java ή στη C (κατόπιν δικής σας προτίμησης) και θα είναι κάποια από τις πιο ολοκληρωμένες υλοποιήσεις της δεύτερης άσκησης.
- Σε αντίθεση με τις προηγούμενες ασκήσεις, σε αυτή την άσκηση θα δοθεί πολύ μεγάλη βαρύτητα στο report. Θα πρέπει να παρουσιάσετε τις μετρήσεις που θα πάρετε σε γραφικές παραστάσεις, τρέχοντας την υλοποίηση σας χρησιμοποιώντας διαφορετικά σενάρια.

Γενικές Πληροφορίες:

Για τη συγγραφή του προγράμματος θα χρησιμοποιηθεί η γλώσσα Java ή η γλώσσα C. Σε περίπτωση που επιλέξετε την γλώσσα C, θα πρέπει να βεβαιωθείτε ότι η εφαρμογή σας μπορεί να τρέξει στα εργαστήρια B103 (για σκοπούς συμβατότητας με θέματα compilation). Η ημερομηνία παράδοσης ορίζεται στις 1 Απριλίου 2011.

Τα παραδοτέα τις εργασίας αυτής είναι:

- Ο πηγαίος κώδικας του προγράμματος (source code)
- Ανάλυση και τεκμηρίωση (documentation) της υλοποίησης σας, και πιθανά σχόλια για δυσκολίες που αντιμετωπίσατε (3-6 σελίδες το μέγιστο).
- Γραφικές παραστάσεις με μετρήσεις που αφορούν την αποδοτικότητα χρήσης του cache στην κύρια μνήμη του εξυπηρετητή.
- Επίσης, όσοι φοιτητές υλοποιήσουν τον ενδιάμεσο εξυπηρετητή, θα πρέπει να συμπεριλάβουν γραφικές παραστάσεις με μετρήσεις που αφορούν την αποδοτικότητα χρήσης του cache μέσω του proxy server.

Τα παραδοτέα θα πρέπει να αποσταλούν με email στον βοηθό του μαθήματος Αντρέα Καμηλάρη (email: kami@cs.ucy.ac.cy), επισυναπτόμενα (attached) σε μορφή zip file.