HTTP Prefetching

Μετά την εγκατάσταση μίας TCP σύνδεσης, ένας client εκπέμπει μία HTTP αίτηση προς τον WWW server. Ακόμη και στην περίπτωση που ο client έχει ένα cached αντίγραφο του ζητούμενου πόρου, η σύνδεση με τον server πρέπει να εγκατασταθεί για να πραγματοποιηθεί ο έλεγχος της ορθότητας (revalidation) του πόρου. Η καθυστέρηση στην λήψη της HTTP απάντησης εξαρτάται από πολλούς παράγοντες όπως π.χ., ο χρόνος παραγωγής της απάντησης στον server, το μέγεθος της απάντησης καθώς και το εύρος ζώνης από τον server στον client. Ο client μπορεί να αποκρύψει την καθυστέρηση από τον χρήστη παράγοντας εκ των προτέρων μία HTTP αίτηση και αποθηκεύοντας την απάντηση στην cache. Εάν ο χρήστης αιτηθεί τον πόρο, ο client παρέχει την απάντηση άμεσα από την cache. Με την υπόθεση ότι το περιεχόμενο του πόρου είναι επίκαιρο, ο client μπορεί να εξυπηρετήσει την αίτηση χωρίς να επιλύσει την IP διεύθυνση του server, να εγκαταστήσει την σύνδεση και να αποστείλει την HTTP αίτηση. Η απάντηση θα είναι άμεση. Επίσης, το prefetching μπορεί να εκμεταλλευτεί αποδοτικότερα τις TCP συνδέσεις απ'ότι οι συνήθεις μεταφορές WWW πόρων που ενεργοποιούνται ως αποτέλεσμα των ενεργειών του χρήστη. Ο prefetching client μπορεί να διασωληνώσει (pipeline) ένα σύνολο αιτήσεων σε μία, μόνιμη (persistent) σύνδεση. Οι συνεχείς (back-to-back) μεταφορές αυτές των απαντήσεων μπορούν να συντελέσουν στην αποφυγή της φάσης slow-start του ελέγχου συμφόρησης.

Το prefetching των HTTP απαντήσεων έχει αποτελέσει αντικείμενο πολλών ερευνητικών προσπαθειών από το 1995 έως σήμερα. Παρά την δυνητική μείωση της υστέρησης που εκλαμβάνεται ο χρήστης, η προ-ανάκτηση WWW πόρων μπορεί να επιφέρει ένα σημαντικότατο φόρτο στον WWW server και το δίκτυο. Το prefetching δεν είναι χρήσιμο αν οι πόροι δεν ζητούνται από τον χρήστη και το περιεχόμενο τους είναι επίκαιρο. Επίσης, η προ-ανάκτηση απαντήσεων έρχεται σε ανταγωνισμό με τις εξελισσόμενες μεταφορές πόρων για το εύρος ζώνης του δικτύου.

Αντί της προ-ανάκτησης του ιδίου του πόρου, ο client μπορεί να προ-ανακτήσει την μεταπληροφορία για τον πόρο. Για παράδειγμα, ο client μπορεί να στείλει ένα αίτημα HEAD προς τον server. Η HTTP απάντηση θα πρέπει να έχει τις ίδιες επικεφαλίδες

όπως η απάντηση σε ένα GET αίτημα. Αυτή η πληροφορία είναι ιδιαίτερα χρήσιμη εάν ο client έχει ήδη ένα αποθηκευμένο αντίγραφο του πόρου. Ανάλογα με το αν ο πόρος έχει μεταβληθεί ή όχι στον origin server, ο client θα μπορούσε να ακυρώσει τον αποθηκευμένο πόρο ή να ανανεώσει τον χρόνο εγκυρότητας του (freshness lifetime). Οι επικεφαλίδες του μηνύματος απάντησης μπορούν να φανούν χρήσιμες ακόμη και όταν ο server δεν έχει αποθηκευμένο αντίγραφο του πόρου. Για παράδειγμα, ο client μπορεί να ελέγξει τις επικεφαλίδες που σχετίζονται με το caching όπως οι Last-Modified, Cache-Control ή Content-Type και να αποφασίσει εάν αξίζει να προχωρήσει με το prefetching ολοκλήρου του πόρου. Ο client μπορεί να αποφασίσει να μην κάνει prefetch ένα πόρο που δεν επιδέχεται caching ή αλλάζει πολύ τακτικά. Επίσης, ο client θα μπορούσε να εξετάσει το πεδίο Content-Length για να προσδιορίσει το ακριβές μέγεθος του πόρου. Ο client μπορεί να αποφασίσει να προ-ανακτήσει κάποιο πόρο αν η τιμή του πεδίου Content-Length βρίσκεται κάτω από συγκεκριμένο κατώφλι. Ετσι αποφεύγονται πόροι που εισάγουν σημαντική δικτυακή επιβάρυνση.

Αντί να μεταδώσει ένα αίτημα HEAD προς τον server για να προσδιορίσει το μέγεθος του πόρου, ο client μπορεί να ελέγξει άμεσα τον όγκο της προ-ανακτώμενης πληροφορίας παράγοντας μία GET ερώτηση με το πεδίο Range. Για παράδειγμα, ο client θα μπορούσε να αιτηθεί τα πρώτα 1000 bytes από τον πόρο. Αυτή η τακτική επιτρέπει στον client να αντλήσει πλήρως ένα μικρό πόρο ή το αρχικό τμήμα ενός μεγάλου πόρου. Σε ορισμένες περιπτώσεις η προ-ανάκτηση του αρχικού τμήματος (prefix) ενός πόρου μπορεί να αποβεί ιδιαίτερα χρήσιμη. Για παράδειγμα, τα πρώτα bytes ενός GIF αρχείου περιέχουν το μέγεθος του αρχείου και μπορούν να περιέχουν ένα αντίγραφο της εικόνας χαμηλής ευκρίνειας. Η εκ των προτέρων διαθεσιμότητα του prefix της εικόνας επιτρέπει στον client να ξεκινήσει την γραφική απόδοσης της (graphic rendering). Το prefix ενός αρχείου video ή audio περιέχει επαρκή αριθμό πλαισίων (frames) ώστε να ξεκινήσει το playback. Οταν ο χρήστης αιτηθεί τον πόρο, ένα δεύτερο αίτημα με το πεδίο range προκαλεί την ανάκτηση του υπολοίπου του πόρου.

Ανάλυση των αλγορίθμων pre-fetching

Top-10

Ο αλγόριθμος Top-10 προτάθηκε στις εργασίες [Markatos, 1999], [Markatos, 1996] και βασίζεται στην γνώση από την πλευρά του server για τον εντοπισμό των πλέον δημοφιλών εγγράφων, την χρήση proxies για την συνάθροιση αιτήσεων προς τους servers καθώς και τον επιμερισμό του κόστους prefetching σε ένα μεγάλο αριθμό από clients καθώς και στην προσαρμογή (adaptation) στα εξελισσόμενα πρότυπα πρόσβασης (access patterns) του χρήστη. Ειδικότερα, τα παραπάνω συστατικά του αλγορίθμου έχουν ως εξής:

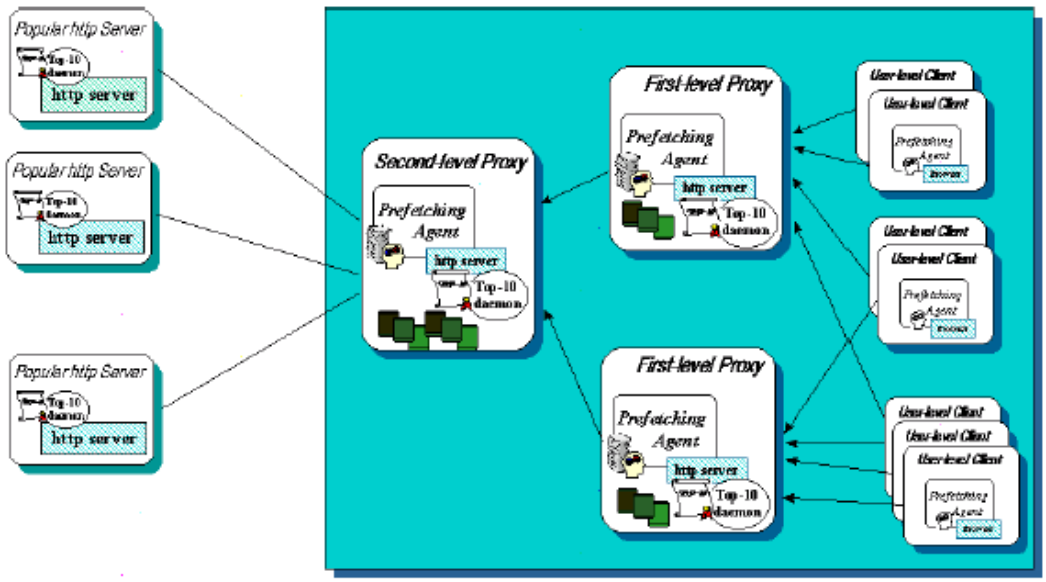
- **Server knowledge**: Στο WWW τα δημοφιλή έγγραφα είναι πολύ δημοφιλή. Σε κάθε server ένα μικρό ποσοστό των αρχείων του αντιστοιχεί στο μεγαλύτερο ποσοστό των αιτήσεων σε αυτόν τον server. Αυτό το σύνολο αρχείων καλείται Top-10. Μόνο τα έγγραφα που αντιστοιχούν στο Top-10 λαμβάνονται υπόψη για το prefetching. Ο πραγματικός αριθμός των αρχείων που περιέχεται στο Top-10 ρυθμίζεται βάσει των profiles των χρηστών λαμβάνοντας υπόψη τον όγκο των αιτήσεων που πραγματοποιήθηκαν στο πρόσφατο παρελθόν από τον client και το χώρο δευτερεύουσας αποθήκευσης που είναι διαθέσιμος για prefetched πόρους.
- **Proxying**: Ο τυπικός WWW χρήστης παράγει ένα μικρό αριθμό αιτήσεων προς ένα server. Στην εργασία [Arlitt, 1996] αναφέρεται ότι το 30% των clients ενός server πραγματοποιούν μόνο μία αίτηση και δεν ζητούν κάτι από τον ίδιο server στην συνέχεια. Εάν κάποιος client πρόκειται να ζητήσει μόνο ένα έγγραφο από τον server, ο μηχανισμός του prefetching δεν έχει ιδιαίτερο νόημα. Εάν, όμως οι clients επικοινωνούν μέσω κάποιου proxy, το prefetching των εγγράφων στον client μπορεί να βελτιώσει σημαντικά τις επιδόσεις του συστήματος εφόσον τα έγγραφα που γίνονται prefetch για λογαριασμό κάποιου client μπορούν να χρησιμοποιηθούν και από άλλους.
- **Adaptation:** Ο όγκος των αιτήσεων που παράγονται από τους διάφορους www clients ποικίλει. Ορισμένοι παράγουν ιδιαίτερα υψηλούς όγκους ενώ άλλοι αιτούνται μόνο λίγα αντικείμενα. Ο αλγόριθμος Top-10 έχει σχεδιαστεί ώστε να προσαρμόζεται σε αυτές τις περιπτώσεις. Επιτρέπει έτσι σε χρήστες που ζητούν συχνά πόρους (frequent users) να κάνουν prefetch πολλά έγγραφα ενώ οι περιστασιακοί χρήστες αφήνονται να κάνουν prefetch σημαντικά λιγότερους

πόρους ενδεχομένως και καθόλου. Το prefetching ελέγχεται από το access profile του χρήστη. Το profile αυτό περιέχει τον αριθμό των εγγράφων που έχουν ζητηθεί από τον κάθε client στο πρόσφατο παρελθόν. Βάσει αυτού προσδιορίζεται το πλήθος των πόρων που θα πρέπει να γίνουν prefetch από τον server στο μέλλον. Εάν το πρότυπο προσπέλασης (access pattern) αλλάξει, το Top-10 θα προσαρμοστεί και θα ξεκινήσει prefetching από τους πλέον δημοφιλείς κόμβους.

Οπως αναφέρθηκε παραπάνω, ο αλγόριθμος βασίζεται στην συνεργασία client – server για την επιτυχή υλοποίηση prefetch ενεργειών. Η πλευρά του server είναι υπεύθυνη για τον περιοδικό υπολογισμό της λίστας με τους πλέον δημοφιλείς πόρους (Top-10) και την προώθηση της στους clients. Για να είναι βέβαιο ότι οι πόροι γίνονται prefetch μόνο στους clients που μπορούν να τα χρησιμοποιήσουν, το Top-10 δεν αντιμετωπίζει όλους τους clients ενιαία. Ο χρόνος θεωρείται διαιρεμένος σε διαστήματα (intervals) και το prefetching από κάθε server ενεργοποιείται μόνο αφού ο συγκεκριμένος client έχει πραγματοποιήσει επαρκή αριθμό αιτήσεων στον συγκεκριμένο server (Access Threshold). Ετσι, σε περιστασιακούς clients δεν γίνονται prefetch έγγραφα ενώ οι «συχνοί» clients εντοπίζονται και λαμβάνονται σημαντικά υπόψη κατά το prefetching.

Ορισμένοι clients π.χ., proxies, παράγουν σημαντικά περισσότερες αιτήσεις από άλλους και ένας σωστά σχεδιασμένο αλγόριθμος θα πρέπει να κάνει prefetch διαφορετικούς αριθμούς documents στους διαφορετικούς clients. Για παράδειγμα, δεν έχει νόημα να γίνουν prefetch τα 500 πιο δημοφιλή έγγραφα σε ένα client που έκανε μόνο 10 αίτησεις κατά το προηγούμενο χρονικό διάστημα. Λαμβάνοντας το πρόσφατο παρελθόν ως ένδειξη για το άμεσο μέλλον, ο πελάτης θα πραγματοποιήσει περίπου 10 αιτήσεις στο επόμενο χρονικό διάστημα και κατά το μέγιστο 10/500 = 2% των prefetched εγγράφων θα χρησιμοποιηθούν. Ενας άλλος client ο οποίος πραγματοποίησε 20000 αιτήσεις στο προηγούμενο χρονικό διάστημα μπορεί να ωφεληθεί σημαντικά από τα 500 prefetched έγγραφα. Ο μηχανισμός Top-10 προσαρμόζει τον όγκο του prefetching στους διαφόρους clients με βάση τον αριθμό των αιτήσεων που έχουν παραχθεί στο παρελθόν. Ενας client δεν μπορεί να προχωρήσει σε prefetching περισσότερων εγγράφων από αυτά που ζήτησε στο πρόσφατο παρελθόν. Τέλος, για να διασφαλιστεί ότι η πολιτική Top-10 θα είναι λιγότερο ή περισσότερο επιθετική όπως απαιτείται, η παράμετρος Top-10 καθορίζει

τον μέγιστο αριθμό πόρων που μπορούν να ανακτηθούν σε κάθε χρονικό διάστημα από οποιοδήποτε κόμβο. Ενας client δεν μπορεί να ζητήσει prefetching περισσότερων από Top-10 πόρων ακόμη και αν έχει προσπελάσει σημαντικότατο αριθμό πόρων στο προηγούμενο διάστημα.

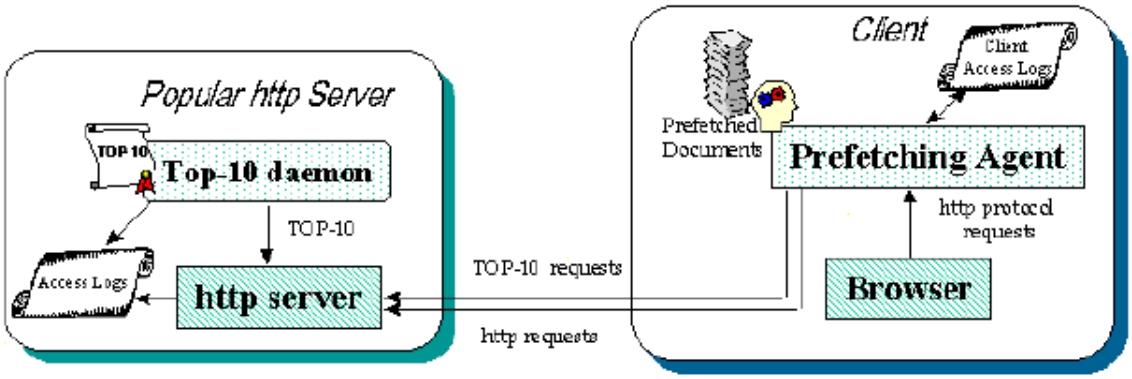


Σχήμα 1. Top-10 prefetching στο πλαίσιο client-proxy-server

Το prefetching μπορεί να υλοποιηθεί τόσο στο επίπεδο του client όσο και σε επίπεδο proxy (Σχήμα 1). Οι clients επιπέδου χρήστη ενεργοποιούν το prefetching από τους proxies πρώτου επιπέδου λαμβάνοντας πρόνοια για τους χρήστες που εξυπηρετούν. Τα proxies πρώτου και δευτέρου επιπέδου παίζουν επίσης τους ρόλους του client και του server. Οι proxies 1^{ου} επιπέδου αποτελούν clients για τα proxies 2^{ου} επιπέδου και κάνουν prefetch και αποθηκεύουν έγγραφα για λογαριασμό των clients επιπέδου χρήστη. Τα proxies 2^{ου} επιπέδου αποτελούν clients σε δημοφιλείς servers από τους οποίους κάνουν prefetch διάφορα έγγραφα τα οποία προωθούν στους clients τους.

Η υλοποίηση του μηχανισμού Top-10 βασίζεται στην συνεργασία των οντοτήτων του client και του server (Σχήμα 2). Στη πλευρά του server, ο Top–10 daemon επεξεργάζεται όλα τα logs και διαμορφώνει την Top-10 λίστα, δηλαδή την λίστα των πλέον δημοφιλών πόρων στον συγκεκριμένο server. Ενημερώνει μία συγκεκριμένη web σελίδα ώστε η πληροφορία Top-10 να εμφανίζεται ως ένας άλλος πόρος στον

server. Η συχνότητα εκτίμησης του Top-10 εξαρτάται από το πόσο συχνά μεταβάλλεται το περιεχόμενο του server.



Σχήμα 2. Συνεργασία client-side prefetching πράκτορα & http servers

Στην πλευρά του client, ο prefetching πράκτορας καταγράφει όλες τις http αιτήσεις του client και προσαρμόζει την prefetching δραστηριότητα του σε αυτές. Ο prefetching agent συνεργάζεται με ένα proxy που φιλτράρει τις http αιτήσεις που υποβάλλονται από τον client. Εάν μία http αίτηση μπορεί να εξυπηρετηθεί από την τοπική cache με τα prefetched έγγραφα, ο proxy διεκπεραιώνει την αίτηση από την cache. Σε διαφορετική περίπτωση, η αίτηση προωθείται στο proxy server του επόμενου επιπέδου. Σε καθημερινή ή εβδομαδιαία βάση, ανάλογα με την ρύθμιση, ο prefetching πράκτορας διατρέχει τα client access logs τα οποία περιέχουν όλες τις http αιτήσεις που έχει υποβάλλει ο client και διαμορφώνει το prefetching profile αυτού του χρήστη (δηλαδή, την λίστα των servers από τους οποίους πρέπει να γίνουν prefetch πόροι). Ο αριθμός των πόρων που έχουν ζητηθεί από τους συγκεκριμένους servers κατά την προηγούμενο χρονικό διάστημα υπερβαίνει το όριο Access_Threshold. Τελικά, βασιζόμενος στο prefetching profile αυτού του χρήστη, ο prefetching πράκτορας ζητάει τα πιο δημοφιλή έγγραφα από τους servers που έχουν ενεργοποιηθεί για την διαδικασία του prefetching. Ο αριθμός των εγγράφων που ζητούνται από κάθε server ισούται με το ελάχιστο του αριθμού των αιτήσεων σε αυτόν τον server ή το πλήθος της λίστας Top-10.

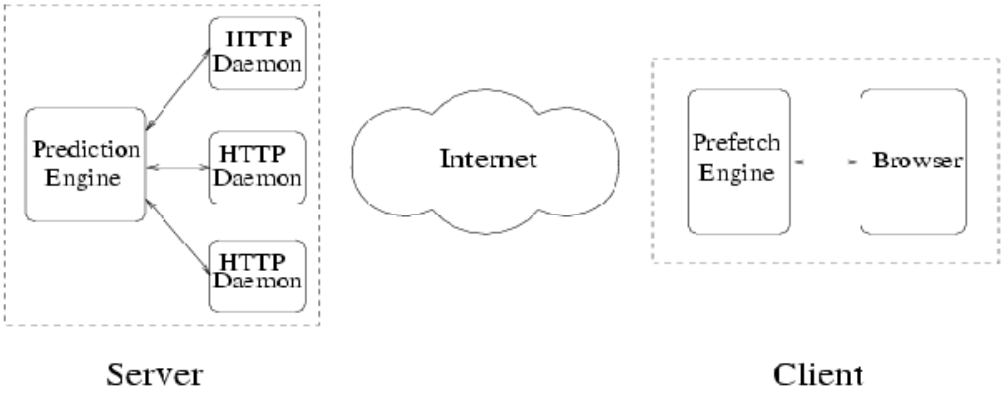
Predictive Prefetching

Στην εργασία [Padmanabhan, 1996] περιγράφεται ένας αλγόριθμος για την πρόβλεψη και την προ-ανάκτηση (prefetching) πόρων που είναι πολύ πιθανόν να ζητηθούν

σύντομα. Συγκεκριμένα, ο server υπολογίζει την πιθανότητα να ζητηθεί ένας συγκεκριμένος πόρος στο άμεσο μέλλον και προωθεί αυτή την πληροφορία στον client. Ο client αποφασίζει αν θα προχωρήσει στην προ-ανάκτηση του πόρου. Ο server έχει την δυνατότητα παρατήρησης των προτύπων πρόσβασης από διάφορους clients και χρησιμοποιεί αυτή την πληροφορία για να προβεί σε ευφυείς προβλέψεις. Ο client είναι σε καλύτερη θέση να αποφασίσει αν θα εκμεταλλευτεί την παρεχόμενη πληροφορία για την προ-ανάκτηση των πόρων ανάλογα με την κατάσταση της τοπικής cache και το εκτιμώμενο κόστος ανάκτησης.

Αρχιτεκτονική του συστήματος prefetching

Στην πλευρά του server υπάρχουν δύο τύποι user-level διεργασιών. Το ένα αφορά ένα σύνολο HTTP deamon διεργασιών με υποστήριξη persistent συνδέσεων. Μία http διεργασία γεννάται για να διεκπεραιώσει το αίτημα ενός client. Εφόσον υποστηρίζονται persistent συνδέσεις, εκτελείται μία διεργασία ανά client και όχι ανά αίτημα. Η άλλη διεργασία είναι ο prediction deamon (predictd) ο οποίος παράγει τις προβλέψεις. Υπάρχει μόνο μία διεργασία prediction deamon ανά server. Ο predictd επικοινωνεί μόνο με τον server και όχι απευθείας με τους clients.



Σχήμα 3. Αρχιτεκτονική συστήματος prefetching

Οταν λάβει μία αίτηση από ένα client, το http περνάει την ταυτότητα του client και τα URL των αιτούμενων πόρων στον predictd. Ο predictd ενδιαφέρεται μόνο για τα αιτήματα που υποβάλλονται με την μέθοδο GET (ή παραλλαγές της). Ο predictd εφαρμόζει τον αλγόριθμο πρόβλεψης που αναλύεται στην επόμενη παράγραφο για να προσδιορίσει ποιά αρχεία είναι υποψήφια για prefetching βάσει της πιθανότητας

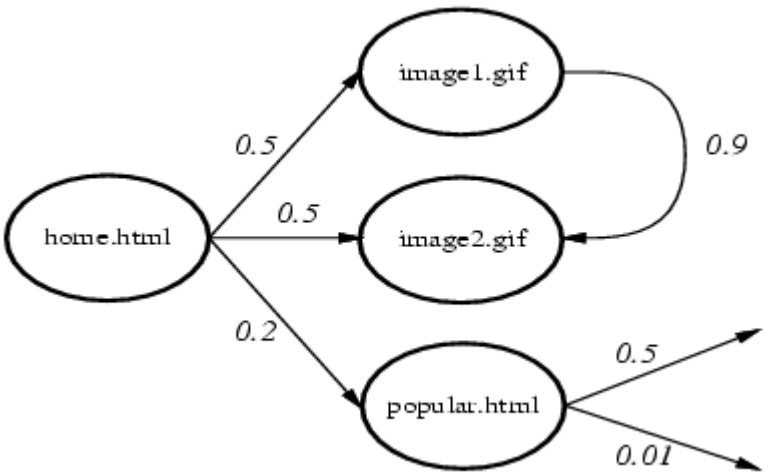
ανάκτησης τους στο άμεσο μέλλον. Η πληροφορία αυτή μεταδίδεται στον client, piggy-backed στις απαντήσεις που στέλνονται από τον server, σε ειδικό πεδίο.

Στον client εκτελούνται ένας συμβατικός browser και ένα prefetching engine. Η μηχανή prefetching χρησιμοποιεί την πληροφορία πρόβλεψης που έχει μεταδόσει ο server για να αποφασίσει για την προ-ανάκτηση των αρχείων. Η απόφαση επηρεάζεται από πολλές παραμέτρους όπως τα περιεχόμενα της τοπικής cache, τον φόρτο του συστήματος, την τρέχουσα κατάσταση λειτουργίας του browser, κλπ.

Μόλις αποφασιστεί η προ-ανάκτηση κάποιων αρχείων από το prefetch engine, κατάλληλες αιτήσεις μεταδίδονται προς τον server. Στις αιτήσεις αυτές προσδιορίζεται ότι δεν πρόκειται για την συνήθη ανάκτηση πόρων αλλά για prefetching. Η πληροφορία αυτή μπορεί να φανεί χρήσιμη στον server. Ο predictd αποφασίζει να μην συμπεριλάβει τις αιτήσεις αυτές στον υπολογισμό των προβλέψεων του αφού πρόκειται για prefetchig αιτήσεις. Επίσης, σε περίπτωση που το σύστημα διεκπεραιώνει ήδη κάποιες αιτήσεις, οι αιτήσεις prefetching λαμβάνουν χαμηλή προτεραιότητα.

Αλγόριθμος πρόβλεψης

Ο αλγόριθμος πρόβλεψης βασίζεται στον αλγόριθμο που έχει προταθεί στην εργασία [Griffioen, 1994]. Συγκεκριμένα κατασκευάζεται ένας γράφος εξαρτήσεων (dependency graph) που αποτυπώνει το πρότυπο πρόσβασης (access pattern) στα διαφορετικά αρχεία του server. Ο γράφος έχει ένα κόμβο για κάθε αρχείο για το οποίο το σύστημα έχει δεχθεί αιτήσεις ανάκτησης. Υπάρχει μία ακμή από το κόμβο Α στον Β μόνο και μόνο αν κάποια χρονική στιγμή υπήρξε προσπέλαση στο Β μετά από w προσπελάσεις μετά την επίσκεψη του Α. Το w καλείται lookahead window size. Το βάρος της ακμής είναι το πηλίκο του αριθμού αναφορών το Β μετά από ένα παράθυρο από την επίσκεψη στο Α προς τον αριθμό αιτήσεων για το Α. Αυτό το βάρος δεν είναι στην πραγματικότητα η πιθανότητα επίσκεψης του Β αμέσως μετά το Α. Ετσι, τα βάρη των ακμών που ξεκινούν από ένα συγκεκριμένο κόμβο δεν απαιτείται να έχουν άθροισμα το 1. Στο Σχήμα 4 παρουσιάζεται ένα τμήμα ενός υποθετικού γράφου εξαρτήσεων.



Σχήμα 4. Τμήμα του γράφου εξαρτήσεων

Ο γράφος εξαρτήσεων ενημερώνεται δυναμικά καθώς ο server δέχεται νέες αιτήσεις. Υπεύθυνη για την ενημέρωση είναι η διεργασία predictd, η οποία δέχεται πληροφορία από τις httpd διεργασίες που διεκπεραιώνουν αιτήσεις clients. Διατηρεί ένα κυκλικό (ring) buffer, μεγέθους ίσου με το μέγεθος του παραθύρου, w, για κάθε client που είναι συνδεδέμενος με το σύστημα (υπόθεση: persistent συνδέσεις). Οταν λάβει μία νέα αίτηση από μία από τις httpd διεργασίες εισάγει τον προσδιοριστή (ID) του αρχείου που ζητήθηκε στον αντίστοιχο κυκλικό (ring) buffer. Μόνο οι καταχωρήσεις μέσα στο ίδιο ring buffer θεωρούνται συσχετιζόμενες, έτσι μόνο οι αντίστοιχες ακμές του γράφου εξαρτήσεων ενημερώνονται. Ετσι διαχωρίζονται οι αιτήσεις από διαφορετικούς clients και αποφεύγονται προβλήματα λανθασμένων συσχετίσεων.

Το predictd βασίζει τις προβλέψεις του στον γράφο εξαρτήσεων. Οταν το Α προσπελαστεί, θα έχει νόημα να γίνει προ-ανάκτηση του Β εφόσον το βάρος της ακμής από το Α στο Β είναι υψηλό (δηλαδή υπάρχει μεγάλη πιθανότητα προσπέλασης του Β στο άμεσο μέλλον). Γενικά, το predictd ανακηρύσσει το Β υποψήφιο για prefetching εάν η ακμή από το Α στο Β έχει βάρος υψηλότερο από ένα όριο prefetch (prefetch threshold), p. Είναι δυνατόν η μεταβλητή p να λαμβάνει διαφορετικές τιμές για κάθε client και να μεταβάλλεται δυναμικά.

Web prefetching σε περιβάλλοντα κινητών επικοινωνιών

Στην εργασία [Jiang, 1998] συζητείται ένα προσαρμοστικό σύστημα το οποίο βασίζεται σε ένα τμήμα πρόβλεψης (prediction module) και σε ένα τμήμα

οριοθέτησης (threshold module). Ο αλγόριθμος πρόβλεψης είναι ανάλογος με αυτόν της εργασίας [Padmanabhan, 1996] αλλά βασίζεται στην παρατήρηση ότι οι web σελίδες δομούνται συνήθως για να συνδέσουν συσχετιζόμενες σελίδες και είναι πολύ πιθανότερο ο χρήστης να ακολουθήσει συνδέσμους παρά να πληκτρολογήσει στον browser το URL του πόρου που τον ενδιαφέρει. Περιστασιακά χρησιμοποιούνται bookmarks για να επιτρέψουν την μετακίνηση σε ένα άλλο site με τελείως διαφορετικό περιεχόμενο. Ο αλγόριθμος πρόβλεψης υποθέτει ότι μόνο οι σελίδες που είναι συνδεδεμένες με την τρέχουσα, παρουσιαζόμενη σελίδα μπορούν να έχουν μη μηδενικές πιθανότητες πρόσβασης. Για να υπολογιστούν οι πιθανότητες πρόσβασης κρατούνται ιστορικά στοιχεία για το ποίες σελίδες έχουν προσπελαστεί και πώς. Συγκεκριμένα, για μία σελίδα A και το σύνδεσμο B_i στην σελίδα αυτή, χρησιμοποιείται ο μετρητής C_A για να καθορίσει τον αριθμό των φορών που η σελίδα A έγινε download και ο μετρητής $C_{A,Bi}$ για να καθορίσει πόσες φορές επιλέχθηκε η B_i μέσω του συνδέσμου στην σελίδα A. Υπάρχουν ακριβώς ένας μετρητής για κάθε σελίδα και ένα μετρητής για κάθε σύνδεσμο στην συγκεκριμένη σελίδα. Οι σχετικοί μετρητές ενημερώνονται κάθε φορά που ο χρήστης διαμορφώνει ένα νέο αίτημα. Η πιθανότητα προσπέλασης $P\{B_i|A\}$, δηλαδή η δεσμευμένη πιθανότητα προσπέλασης του συνδέσμου B_i δεδομένης της παρουσίασης της σελίδας A, υπολογίζεται ως εξής:

$$P\{B_i \mid A\} = \begin{cases} \min(1, \dfrac{C_{A,B_i}}{C_A}), \text{for } C_A \geq 5 \\ 0, \text{for } C_A < 5 \end{cases} \qquad (1)$$

Η τιμή CA = 5 έχει επιλεγεί αυθαίρετα στην εξίσωση (1) γιατί εάν ο χρήστης δεν έχει επισκεφτεί μία σελίδα αρκετές φορές, η ιστορία πρόσβασης δεν υποδεικνύει τα πραγματικά του ενδιαφέροντα. Εάν μία σελίδα D δεν έχει σύνδεσμο στην A η πιθανότητα $P\{D|A\}$ είναι μηδενική.

Ο αλγόριθμος μπορεί να εκτελεστεί τόσο στον client όσο και στον server. Στην δεύτερη περίπτωση, ο server συναθροίζει τα πρότυπα πρόσβασης προσφάτων επισκέψεων για να υπολογίσει τις πιθανότητες πρόσβασης μελλοντικών χρηστών. Οι πιθανότητες προσπέλασης των διαφόρων συνδέσμων σε μία σελίδα μεταδίδονται στους clients μαζί με την ίδια την σελίδα για την ελαχιστοποίηση της καθυστέρησης.

Το επόμενο βήμα μετά τον υπολογισμό της πιθανότητας προσπέλασης ενός συνδέσμου είναι η απόφαση αν αξίζει να προ-ανακτηθεί ο συγκεκριμένος πόρος. Υπάρχουν τρείς παράγοντες οι οποίοι λαμβάνονται υπόψη για την επιλογή των αρχείων προς prefetching:

- το χρονικό όφελος από την προ-ανάκτηση του πόρου για τον χρήστη που ενδεχομένως τον ζητήσει,
- το εύρος ζώνης το οποίο πρόκειται να σπαταληθεί σε περίπτωση που ο πόρος δεν ζητηθεί, και,
- η επίδραση της αίτησης prefetch σε άλλους χρήστες, των οποίων οι κανονικές αιτήσεις μπορεί να καθυστερήσουν λόγω της αίτησης προ-ανάκτησης.

Με βάση ένα μοντέλο πρόσβασης πολλαπλών χρηστών (multi-user) σε ένα ενιαίο server μέσω κοινής δικτυακής σύνδεσης, το βέλτιστο όριο (threshold) της πιθανότητας προσπέλασης, για την ελαχιστοποίηση του κόστους, υπολογίζεται ως εξής:

$$H = 1 - \frac{(1-\rho)\frac{a_T}{a_B}}{1-\rho)^2 b + \frac{a_T}{a_B}} \qquad (2)$$

Η παράμετρος b αφορά την δυνατότητα του συστήματος σε kbits/sec, ρ είναι η μέση χρήση του συστήματος, α_T το χρονικό κόστος της απάντησης ($/sec) α_B το κόστος εύρους ζώνης. Η εξίσωση (2) ουσιαστικά υποδεικνύει ότι, άσχετα από την κατανομή των πιθανοτήτων προσπέλασης, αν ο κάθε χρήστης προβαίνει σε prefetching με πιθανότητα μεγαλύτερη από το παραπάνω όριο, ο μέσο κόστος ανά χρήστη θα είναι μικρότερο απ'ότι χωρίς καθόλου prefetching.

Αναφορές

[Markatos, 1996] E. Markatos, and C. Chronaki, "A Top-10 Approach to Prefetching on the Web", Technical Report 173, ICS-Forth, August 1996.

[Markatos, 1999] E. Markatos, and C. E. Chronaki, "A Top-10 Approach to Prefetching the web", Proceedings of INET '98 Geneva, Switzerland, July 1999.

[Padmanabhan, 1996] V. Padmanabhan, and J.C.Mogul, "Using Predictive Prefetching to Improve WWW Latency", ACM SIGCOMM Computer Communications Review, Vol. 26, No.3, July 1996.

[Jiang, 1998] Z. Jiang, and L. Kleinrock, «Web Prefetching in a Mobile Environment», IEEE Personal Communications Magazine, October 1998.