

Προγραμματισμός και Συστήματα στον Παγκόσμιο Ιστό

Κεφάλαιο 5: Πρωτόκολλα και Αλγόριθμοι Διαχείρισης Κρυφών Μνημών στον Ιστό

Συνέπεια Αντιγράφων

- Το κρύψιμο δεδομένων (data caching) δημιουργεί **αντίγραφα**.
- Όταν η πηγή ενημερώνεται, μπορεί τα αντίγραφα να είναι **έωλα (stale)**.
- Αυτό είναι το **πρόβλημα συνέπειας**: εγγύηση ότι οι περιηγητές και αντιπρόσωποι χρησιμοποιούν μόνο ενημερωμένα αντικείμενα.
- Θεμελιώδες πρόβλημα στην επιστήμη Η/Υ
 - Πιο δύσκολο στον Ιστό, εξαιτίας της χαλαρής σχέσης ανάμεσα σε περιηγητές, αντιπροσώπους και εξυπηρετές-πηγές.
- Αφορά και περιηγητές και αντιπροσώπους → πελάτες

There are only two hard things in Computer Science: cache invalidation and naming things
– Phil Karlton

Πρωτόκολλα Συνέπειας

- Δυο θεμελιώδεις προσεγγίσεις:
 - Επικύρωση (validation)
 - Ακύρωση (invalidation)
- Επικύρωση γίνεται με **πρωτοβουλία** πελάτη
 - Ζητούν από την πηγή να στείλει, αν χρειάζεται, το πλέον ενημερωμένο αντίγραφο.
- Ακύρωση γίνεται με πρωτοβουλία πηγής
 - Ενημερώνει πελάτες για το ποιά αντικείμενά τους έχουν ενημερωθεί.
 - → πελάτες υποθέτουν ότι όλα τα αντίγραφά τους είναι έγκυρα, μέχρι να λάβουν μήνυμα Ακύρωσης.

Πρωτόκολλα συνέπειας

	Ισχυρή συνέπεια	Χαλαρή συνέπεια
Επικύρωση (από τον πελάτη)	Poll-every-time	TTL, Piggyback Cache Validation
Ακύρωση (από την πηγή)	Ακύρωση	Piggyback Server Invalidation, Μισθώσεις, Συνδρομές

Πρωτόκολλα Συνέπειας: Διεγκυστίδες

- Επικύρωση
 - Καταλήγει σε χαλαρή / **αδύναμη συνέπεια** (loose / weak consistency).
 - Μπορεί να εκτελείται **περιοδικά**, για να μειώσει το **κόστος** της
 - έξτρα μηνύματα προς στην πηγή
 - Έξτρα καθυστέρηση σε κάθε εύστοχο χτύπημα της ΚΜ.
 - Γενικά, είναι **σύγχρονη** ή **ασύγχρονη**
 - Σύγχρονη: γίνεται με κάθε προσπέλαση αντικειμένου
- Ακύρωση
 - Μπορεί να επιτύχει **ισχυρή συνέπεια** (strong consistency) αν γίνει άμεσα μετά μια ενημέρωση
 - **Ασύγχρονη** σε σχέση με αιτήσεις πελατών
 - Δεν επηρεάζει άμεσα την **καθυστέρηση** αιτήσεων πελατών.
 - Το **κόστος** της είναι κυρίως τα μηνύματα προς τους πελάτες που (μπορεί να)έχουν αντίγραφα ενός ενημερωμένου αντικειμένου.

Πρωτόκολλα Συνέπειας: Διεγκυνοστίδες

- **Κατάσταση** / Πληροφορία στην πηγή:
 - Ακύρωση: πηγές κρατούν **λίστες πελατών** ανά αντικείμενο που το έχουν ζητήσει → ενδέχεται να είναι στην ΚΜ τους.
 - Επικύρωση: πηγές δεν κρατούν καμμία πληροφορία
- **Περιττή κίνηση** στο δίκτυο
 - Επικύρωση: πελάτες δεν μπορούν να γνωρίζουν ποιά αντικείμενα έχουν ενημερωθεί → ζητούν επικύρωση για αντικείμενα που δεν έχουν ενημερωθεί
 - Ακύρωση: πηγές στέλνουν ακυρώσεις/ενημερώσεις για αντικείμενα που οι πελάτες δεν έχουν στην ΚΜ τους.
- Στους αντιπροσώπους μέχρι τώρα προτιμάται η επικύρωση
 - Κυρίως για λόγους μικρότερης μικρότερης εμπλοκής της πηγής
- Σε δίκτυα διανομής (δηλ. για θετούς εξυπηρετές/αντιπρόσωπους), προτιμάται η ακύρωση.

Πρωτόκολλα Συνέπειας: Επικύρωση

- Κύριο ερώτημα: Σύγχρονη ή ασύγχρονη;
- Ακραίες επιλογές
 - Επικύρωση με κάθε πρόσβαση στην ΚΜ (Poll-every-time)
 - → καθυστέρηση + έξτρα μηνύματα, ίσως περιττά
 - → ισχυρή συνέπεια
 - Επικύρωση κάθε περίοδο $\rightarrow \infty$
 - → μηδενικό κόστος
 - → χαλαρή συνέπεια
 - Γενικά, ασύγχρονη πολιτική επικύρωσης \rightarrow **χαραμισμένες εγκυροποιήσεις** όταν εγκυροποιούνται αντικείμενα που δε θα προσπελαστούν στην ΚΜ ή που θα διαγραφούν σύντομα από την ΚΜ.

Πρωτόκολλα Συνέπειας: Επικύρωση

- Ποικίλλα πρωτόκολλα Επικύρωσης
 - Με βάση την έννοια **TTL** (time to live)
- Βασικό πρωτόκολλο με τη χρήση **HTTP**
 - Πηγή: Επιστρέφοντας ένα αντικείμενο,
 - δίνει μια **TTL** τιμή, κάνοντας χρήση των κεφαλίδων **expires** ή **max-age**
 - Θέτει την κεφαλίδα **last-modified** ίση με το χρόνο τελευταίας ενημέρωσης του αντικειμένου.
 - Πελάτης: Λαμβάνοντας ένα αντικείμενο,
 - Το βάζει στην ΚΜ του και το χρησιμοποιεί μέχρι τη λήξη του **TTL**.
 - Μετά, πρέπει να επικυρώσει το αντικείμενο στέλνοντας μια αίτηση **GET** με την κεφαλίδα **if-modified-since** να έχει την τιμή που έλαβε στην **last-modified**.

Πρωτόκολλα Συνέπειας: Επικύρωση

- **Άστοχία:** 2 ειδών
 - **Αναγκαστική:** δεν έχει το αντικείμενο η ΚΜ
 - **Συνέπειας:** το έχει η ΚΜ, αλλά έχει λήξει το TTL, και το αντίγραφο είναι έωλο
 - Και στις 2 περιπτώσεις το αντικείμενο πρέπει να μεταφερθεί από την πηγή.
- **Ευστοχία:** 2 ειδών
 - **Γρήγορη:** είναι στην ΚΜ και δεν έχει λήξει το TTL
 - **Αργή:** είναι στην ΚΜ και έχει λήξει το TTL, αλλά μετά τον έλεγχο προκύπτει ότι συνεχίζει να είναι έγκυρο.
 - Και στις 2 περιπτώσεις το αντικείμενο ΔΕΝ πρέπει να μεταφερθεί από την πηγή.
- Μελέτες έχουν δείξει ότι το μεγάλο κόστος προκύπτει από την ανάγκη επικοινωνίας με την πηγή.
 - ➔ τα αργά HITS είναι οριακά μόνο καλύτερα από τις αστοχίες !

Πρωτόκολλα Συνέπειας: Επικύρωση

- **Αόριστα TTL:** Σχεδιαστές συχνά δεν καθορίζουν TTL αντικειμένων
 - ➔ πελάτες πρέπει να ορίσουν τα TTL
- **Πάγια TTL:** π.χ. Κάθε 2 ώρες
 - Μπορούν να διαφέρουν ανάλογα με τον τύπο αντικειμένων (π.χ. Εικόνες σπάνια ενημερώνονται ➔ μεγαλύτερα TTL
 - Και πάλι: πολλά περιττά μηνύματα και πολλές έωλες ανακτήσεις
- **Προσαρμοζόμενα TTL:** παραδοχή: όσο πιο πολύς χρόνος έχει μεσολαβήσει από την τελευταία ενημέρωση ενός αντικειμένου, τόσο πιο πολύ το αντικείμενο θα παραμείνει ανενημέρωτο.
 - $TTL = \min (k \times (\text{send_time} - \text{last_modified}), \text{max_threshold})$
 - max_threshold για να επικυρώνονται ακόμα και πολύ παλιά αντικείμενα.
 - $k = 0.2$ ή 0.1
- Το μόνο που χρειάζεται ο πελάτης είναι το `last_modified` που του το στέλνει πάντα η πηγή σε κάθε απάντηση!
- Μελέτες έδειξαν ότι το ποσοστό έωλων απαντήσεων με $k = 0,2$ είναι πάρα πολύ μικρό ($< 1\%$), αλλά $>50\%$ των αιτήσεων για επικύρωση ήταν περιττές.

Πρωτόκολλα Συνέπειας: Επικύρωση

- Συχνά, πελάτες και πηγές πρέπει να υπερβούν το βασικό πρωτόκολλο.
- Πελάτης:
 - Max-stale = seconds → ληγμένα TTL είναι αποδεκτό, αν νεώτερο από Max-stale
 - Max-age = seconds → μη-ληγμένα TTL δεν είναι αποδεκτό, αν παλιότερο από την τιμή Max-age
 - Min-fresh = seconds → μη-ληγμένα TTL δεν είναι αποδεκτό, αν το αντικείμενο δεν θα είναι φρέσκο για Min-fresh ακόμα.
 - No-cache: → ανεξάρτητα TTL, πάντα πήγαινε στην πηγή.
- Πηγή:
 - Max-age = 0 → πάντα πήγαινε στην πηγή
 - Must-revalidate: ακόμα και αν ο πελάτης μέσω Max-stale αποδέχεται ένα αντικείμενο με ληγμένα TTL, πάντα πήγαινε στην πηγή.
 - No-cache: → απαγορεύεται να μπει στην ΚΜ.

Πρωτόκολλα Συνέπειας: Επικύρωση

Βελτιστοποίηση Επικύρωσης

- **Τυφλές IMS** αιτήσεις για αντικείμενα με ληγμένα TTL έχουν μεγάλο κόστος
 - ➔ επικύρωσε μόνο τα ‘σημαντικά’ με IMS αιτήσεις
- για τα άλλα, στείλε **HEAD** αιτήσεις
 - Επιστρέφουν μόνο μεταπληροφορία (πχ τρέχουσες τιμές για last-modified)
 - ➔ πελάτης μπορεί να παρατείνει το TTL ή να ακυρώσει τα σχετικά αντικείμενα στην ΚΜ του
- ➔ μειώνονται οι απαιτήσεις για εύρος ζώνης
- ➔ παρόλα αυτά, ο αριθμός των μηνυμάτων ΔΕΝ μειώνεται
 - Μια λύση για αυτό είναι το **piggyback cache validation (PCV)**.

Πρωτόκολλα Συνέπειας: Επικύρωση

- **Ιδέα:** στείλε τα μηνύματα Επικύρωσης πάνω από ανεξάρτητη κίνηση HTTP μεταξύ πελάτη-πηγής.
- **PCV** : (Piggyback Cache Validation)
 - Κάθε φορά που ένας πελάτης στέλνει μια HTTP αίτηση, εξετάζει αν έχει στην ΚΜ κάποια αντικείμενα από αυτήν την πηγή και αν έχει λήξει (ή πλησιάζει να λήξει) το TTL τους.
 - Αν ναι, τότε στέλνει HEAD IMS αιτήσεις για αυτά με το μήνυμα που μεταφέρει την αρχική άσχετη αίτηση.
 - Αυτό γίνεται με **HTTP pipelining** (επιτρέπει στον πελάτη να εκδώσει πολλές αιτήσεις -- μέσω μιας ήδη υπάρχουσας TCP σύνδεσης -- σε έναν εξυπηρετή, χωρίς να περιμένει για τις ενδιάμεσες απαντήσεις).
- Μελέτες έχουν δείξει ότι συχνά υπάρχει αρκετή κίνηση HTTP μεταξύ πελατών-πηγών, ώστε να προκύπτουν σοβαρά ωφέλη

20/12/19 → **αμελητέες περιττές εγκυροποιήσεις.**

Πρωτόκολλα Συνέπειας: **Ακύρωση**

- Το HTTP μέσω των POST, PUT, DELETE επιτρέπει σε έναν εξυπηρετή να ακυρώσει τα περιεχόμενα της ΚΜ
- Στην πράξη, όμως, οι αντιπρόσωποι πίσω από firewalls δεν είναι προσπελάσιμοι από εξωτερικούς εξυπηρετές-πηγές, π.χ. Βλέπε εταιρικούς αντιπροσώπους.
- Αλλά, πρωτόκολλα ακύρωσης
 - μπορούν να υλοποιηθούν σε εταιρικά δίκτυα και
 - συζητιέται να υλοποιηθούν και για αντιπροσώπους
 - Συνδέονται με σημαντικούς μηχανισμούς/αλγόριθμους που απαντώνται σε πολλά Συστήματα Λογισμικού.
- Γιαυτό, θα τα εξετάσουμε...

Πρωτόκολλα Συνέπειας: Ακύρωση

- Ακύρωση ΚΜ πρέπει να αντιμετωπίσει 2 θεμελιώδη προβλήματα:
 - **Πρόβλημα client-list:**
 - Η πηγή πρέπει να καταγράφει όλες τις συναλλαγές με πελάτες
 - Δεν ξέρει πότε να σβήσει πεδία της λίστας – δεν ενημερώνεται πότε οι πελάτες σβήνουν αντικείμενα από την ΚΜ τους.
 - **Πρόβλημα delayed-updates:** Πρέπει να παρασχεθεί ισχυρή συνέπεια →
 - τί γίνεται αν η πηγή δεν μπορεί να επικοινωνήσει με έναν πελάτη;
 - δεν παρέχεται ισχυρή συνέπεια κατά την διάρκεια αποσυνδέσεων!
 - Επίσης, όταν καταφθάνει μια ενημέρωση στην πηγή, **πρέπει να καθυστερήσει** μέχρι
 - να ακυρωθούν όλες οι ΚΜ με αντίγραφα και
 - μετά να γίνει η ενημέρωση στην πηγή
 - ΜΟΝΟ αν ΟΛΕΣ οι ΚΜ ακυρώθηκαν.
- Τί γίνεται αν κάποιος πελάτης δεν απαντά;
- Μέχρι πότε θα περιμένει η ενημέρωση;

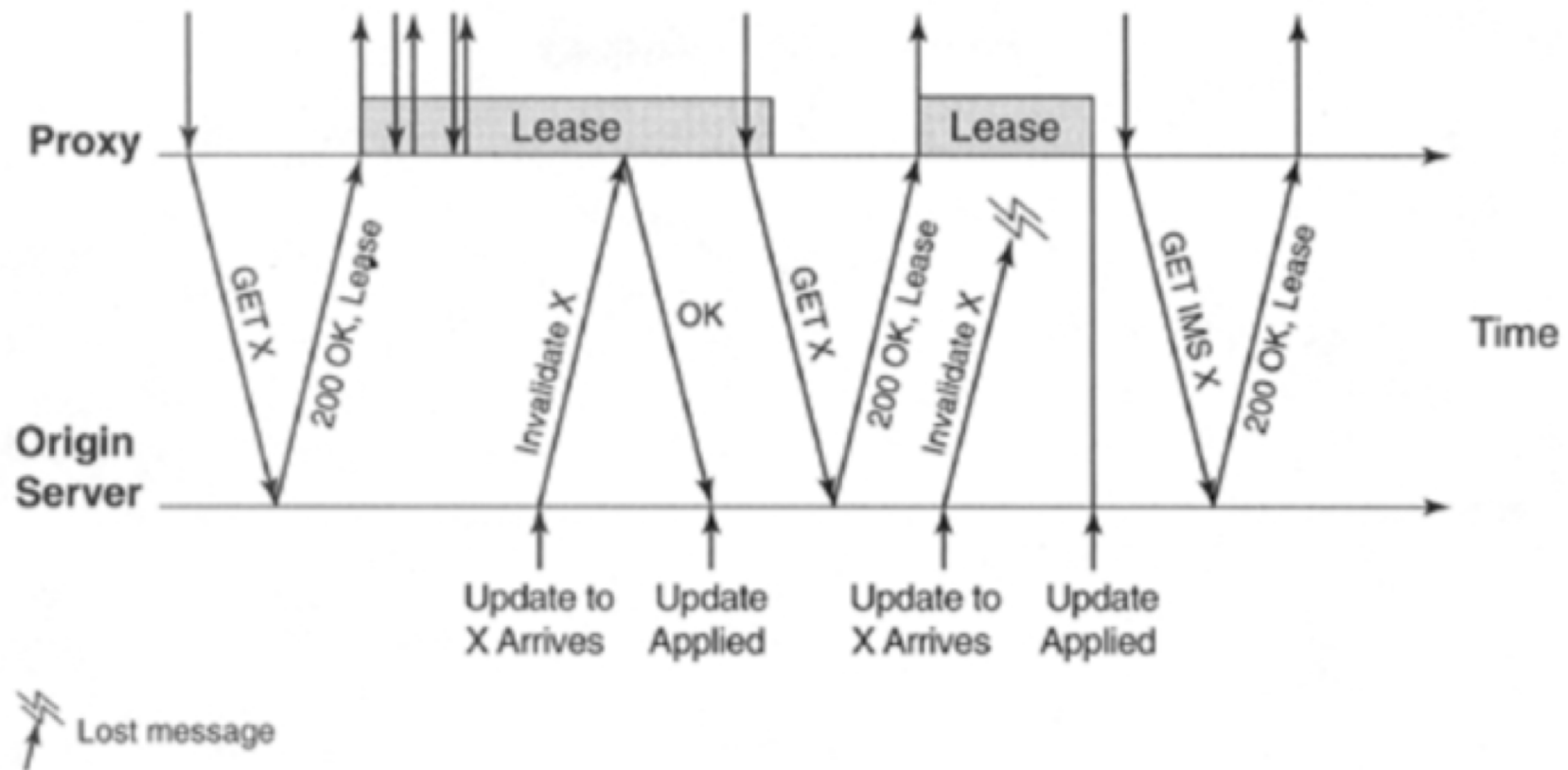
Πρωτόκολλα Συνέπειας: Ακύρωση με Μίσθωση

- Μίσθωση (**leases**): ένας μηχανισμός επίλυσης των προβλημάτων
 - client-lists
 - και delayed-updates.
- Μια μίσθωση, είναι μια υπόσχεση από την πηγή προς τον πελάτη ότι, για όλη τη διάρκεια της μίσθωσης, ο εξυπηρέτης θα τον ενημερώνει για την αλλαγή του αντικείμενου μέχρι να λήξει η μίσθωση.
- **client-lists**: με μισθωμένη συνέπεια, η πηγή κρατά στη λίστα της ένα πελάτη **μόνο μέχρι να λήξει η μίσθωση**. Μετά μπορεί με ασφάλεια να τον διαγράψει.
- **delayed-updates**: μια ενημέρωση στην πηγή μπορεί να καθυστερήσει από κάποιον πελάτη εκτός επικοινωνίας, μόνο μέχρι να λήξει η μίσθωσή του.

Πρωτόκολλα Συνέπειας: Ακύρωση με Μίσθωση

- Ο εξυπηρέτης-πηγή, όταν λαμβάνει μια ενημέρωση, πριν κάνει την ενημέρωση, ο στέλνει ένα μήνυμα ακύρωσης στους αντιπροσώπους για το αντικείμενο (το αντίγραφο διαγράφεται ή ακυρώνεται).
 - Αν όλοι οι αντιπρόσωποι επιβεβαιώσουν ότι έλαβαν το μήνυμα ακύρωσης, τότε ο εξυπηρέτης περνάει την ενημέρωση.
 - Αν κάποιος δεν απαντήσει, ο εξυπηρέτης πρέπει να περιμένει ΤΟ ΠΟΛΥ μέχρι να λήξουν οι μισθώσεις από κάθε ΚΜ αντιπροσώπου που δεν απάντησε.
- Ο αντιπρόσωπος πρέπει να ξανα-επικυρώσει ένα αντικείμενο στην ΚΜ του στην πρώτη αίτηση για το αντικείμενο μετά τη λήξη της μίσθωσης.
 - Με αυτή την επικύρωση, η **μίσθωση ανανεώνεται**.
- Συνεπώς, ο μηχανισμός αυτός είναι ένας **συνδυασμός ακύρωσης και επικύρωσης**.
- Κομβικό σημείο: η διάρκεια της μίσθωσης
 - Εξαρτάται κυρίως από την εφαρμογή: πόση είναι η μέγιστη αποδεκτή καθυστέρηση για ενημερώσεις;
 - Μεγάλη διάρκεια σημαίνει μεγάλη καθυστέρηση στην ανανέωση
 - Συνήθως επιλέγεται μηχανισμός προσαρμοζόμενης διάρκειας μίσθωσης
 - Αντίστοιχος του προσαρμοζόμενου TTL.

Πρωτόκολλα Συνέπειας: Ακύρωση με Μίσθωση



Πρωτόκολλα Συνέπειας: Ακύρωση με Μίσθωση

- Επέκταση: Volume leases
 - Μίσθωση σε μεμονωμένα αντικείμενα και σύνολο συσχετιζόμενων αντικειμένων
 - Ανάγνωση από την ΚΜ μόνο αν η μίσθωση του αντικειμένου ΚΑΙ η μίσθωση του volume δεν έχει παρέλθει
 - Ο server μπορεί να περάσει ενημερώσεις όταν λήξει η μίσθωση αντικειμένου, ή, η μίσθωση ενός volume
 - Βάζοντας μεγάλης διάρκειας μίσθωση στα αντικείμενα και μικρή στα volumes, αποφεύγουμε τα μειονεκτήματα της απλής μίσθωσης
 - Μικρό overhead για τα μεμονωμένα αντικείμενα (μεγάλη διάρκεια μίσθωσης αντικείμενου)
 - Ο server μπορεί να «περάσει» αλλαγές χωρίς μεγάλες καθυστερήσεις (μικρή διάρκεια μίσθωσης volume).

Πρωτόκολλα Συνέπειας:

Ακύρωση με **Συνδρομές**

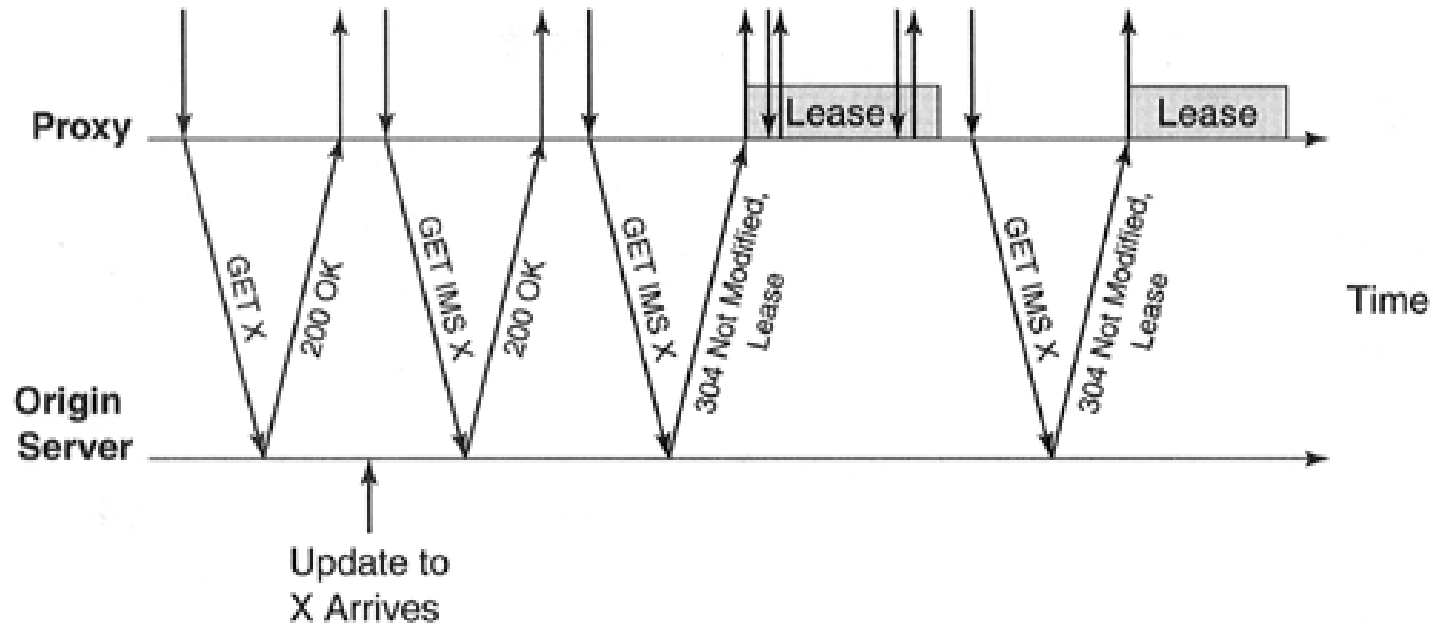
- Για να μειώσουμε περαιτέρω το μέγεθος των client lists (εκτός από σύντομες μισθώσεις), υπάρχει και η τεχνική των συνδρομών.
- Οι πελάτες ζητούν να γίνονται **συνδρομητές σε ακυρώσεις αντικειμένων** (αντί να τους δίνεται αυτόματα μια μίσθωση)
 - Όταν πελάτες ζητούν ένα αντικείμενο με ένα GET, μπορούν να ζητήσουν συνδρομή για ενημερώσεις
 - Οι εξυπηρέτες εγγράφουν σε λίστες συνδρομητών τους πελάτες (με κριτήρια)
 - Όταν προκύψει ενημέρωση, οι εξυπηρέτες ακυρώνουν τα αντικείμενα στις ΚΜ των συνδρομητών τους.
- Παρατήρηση: είναι γνωστό πως πολλά αντικείμενα που ζητιούνται από πελάτες, ΔΕΝ ξαναζητιούνται (~60% όλων των αντικειμένων).
- ➔ για αυτά τα αντικείμενα καλό θα ήταν να μη ζητηθούν (ή να μη γίνουν δεκτές από τον εξυπηρέτη) συνδρομές, γιατί οι ακυρώσεις από τον εξυπηρέτη θα ήταν χαραμισμένες!
- ➔ εξυπηρέτης δέχεται συνδρομές ΜΟΝΟ από πελάτες που **ήδη έχουν** ένα **έγκυρο** αντίγραφο στην ΚΜ τους.
 - Την πρώτη φορά που ζητιέται ένα αντικείμενο από ένα πελάτη, ο εξυπηρέτης δεν το δέχεται ως συνδρομητή.
 - Την επόμενη φορά που ο ίδιος πελάτης στέλνει ένα GET IMS μήνυμα, ο εξυπηρέτης τον εγγράφει ως συνδρομητή, ΜΟΝΟ αν ο πελάτης ήταν έγκυρος δηλ. το αντικείμενο δεν είχε ενημερωθεί!
 - Γιατί;

Πρωτόκολλα Συνέπειας:

Ακύρωση με Μισθωμένες Συνδρομές

- Συνδυάζοντας μισθώσεις με συνδρομές:
 - Η συνδρομή έχει μια διάρκεια ίση με τη διάρκεια της μίσθωσης, αλλά:
 - ➔ δεν υπάρχει μίσθωση με κάθε αίτηση πελάτη
 - ➔ δεν ανανεώνεται κάθε μίσθωση με νέα αίτηση
- Πχ.
 - Η 1^η αίτηση πελάτη για ένα αντικ. δεν μισθώνεται
 - Η 2^η αίτηση μπορεί να μην μισθωθεί αν το αντικ. Στην ΚΜ του πελάτη δεν είναι έγκυρο
 - Δηλ. σε αυτές τις περιπτώσεις δεν υπάρχει συνδρομή!
 - Επόμενες αιτήσεις του πελάτη για το αντικ. Μισθώνονται, αν η ΚΜ του πελάτη έχει έγκυρο αντίγραφο...κλπ

Παράδειγμα μισθωμένης συνδρομής



Πρωτόκολλα Συνέπειας:

Ακύρωση με καθυστερημένες ενημερώσεις

- Το πρωτόκολλο ακύρωσης επιβάλλει την καθυστέρηση της ενημέρωσης στην πηγή μέχρι όλες οι ΚΜ να στείλουν ACK ακύρωσης.
- Αυτό σε πολλές περιπτώσεις δεν έχει νόημα
 - Π.χ. Στον ηλεκτρονικό κατάλογο του Χρυσού Οδηγού πρέπει να εγγραφεί ένα νέο πεδίο για ένα νέο χρήστη τηλεφώνου.
 - Η τεχνητή καθυστέρηση της ενημέρωσης στον ιστότοπο του ΧΟ μέχρι να ακυρωθούν έωλες σελίδες από τις ΚΜ δε λύνει το πρόβλημα συνέπειας!
 - Το θέμα είναι: **ο νέος χρήστης υπάρχει** και οι πελάτες δεν τον βλέπουν ανεξάρτητα από το γεγονός ότι η ενημέρωση δε φαίνεται τεχνητά ούτε στην πηγή.
- Υπάρχουν όμως άλλες περιπτώσεις όπου έχει νόημα
 - Πχ αυτός που ενημερώνει απαιτεί από τους επισκέπτες της σελίδας να είναι ενημερωμένοι. Σκεφτείτε εφαρμογές!
- ➔ η επιλογή μεταξύ άμεσης ή καθυστερημένης ενημέρωσης εξαρτάται από την εφαρμογή!

Πρωτόκολλα Συνέπειας:

Ακύρωση με καθυστερημένες ενημερώσεις

- Οι καθυστερημένες ενημερώσεις οδηγούν σε μια πιο χαλαρή αντιμετώπιση της συνέπειας → **best-effort consistency**
- Επέκταση του πρωτοκόλλου ακύρωσης: Μαζί με την ακύρωση υπάρχει ένα flag (ack). Αν ο εξυπηρέτης το θέσει σε 1 σημαίνει ότι περιμένει ack αλλιώς 0 = best-effort
 - Στη 2^η περίπτωση το αποτέλεσμα δεν προσφέρει καλύτερη εγγύηση συνέπειας από την κεφαλίδα expires
- Με αυτήν τη σκέψη, οι πηγές θα μπορούσαν να σβήνουν κάποιους πελάτες από τη λίστα τους, όταν η μνήμη στερεύει (πχ σβήνοντας πελάτες που δεν είναι πολύ ενεργοί) ή όταν δεν απαντούν.

Ακύρωση: Piggyback Server Invalidation

- Ο εξυπηρέτης χωρίζει τους πόρους ενός ιστοτόπου σε volumes
 - Ο εξυπηρέτης διατηρεί ιστορικό τροποποίησης και «έκδοση» (version) για κάθε volume
- Η ΚΜ κρατάει το volume identifier και το version με κάθε αίτηση που κάνει για κάποιο αντικείμενο
 - Σε επόμενες αιτήσεις, ο εξυπηρέτης απαντά με μια λίστα πόρων για που έχουν τροποποιηθεί σε κάθε volume σε σχέση με την έκδοση που έχει η ΚΜ
 - Η ΚΜ ακυρώνει τα έωλα αντίγραφα ή επεκτείνει το χρόνο ζωής των αντιγράφων που είναι έγκυρα.
- Οδηγεί σε αυξημένη συνέπεια σε σχέση με άλλες προσεγγίσεις
 - Μειώνεται το κόστος επικυρώσεων
 - Λιγότερες τυφλές IMS αιτήσεις
 - Μπορεί να συνδυαστεί και με PCV για καλύτερα αποτελέσματα.