



Κινητός & Διάχυτος Υπολογισμός

Χειμερινό Εξάμηνο 2020-2021

Δημήτριος Κατσαρός

Σειρά προβλημάτων: 1^η: ΟΜΑΔΙΚΕΣ ΕΡΓΑΣΙΕΣ

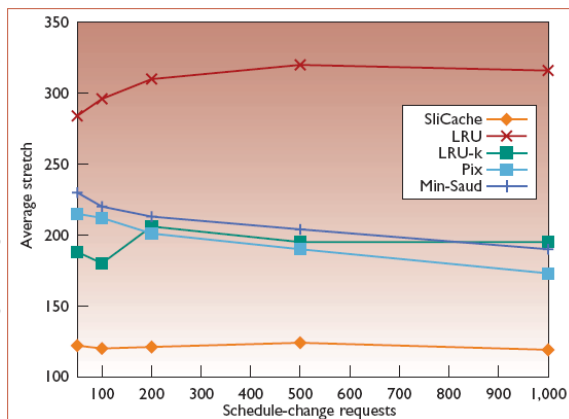
Ημέρα ανακοίνωσης: Monday, November 02, 2020

Προθεσμία παράδοσης: Κυριακή, Δεκέμβριος 13, 2020



Πρόβλημα-01

Έχουμε δει ότι ο αλγόριθμος Min_SAUD είναι optimal για την μετρική stretch. Στο διπλανό γράφημα βλέπουμε την επίδοση του Min_SAUD για έναν mobile client που κινείται σε κελιά όπου οι server έχουν διαφορετικά προγράμματα εκπομπής. Καταγράψτε για ποιους λόγους ο Min_SAUD στο γράφημα δεν εμφανίζεται ο καλύτερος ως αναφορά στην ελαχιστοποίηση του stretch. [Οι τιμές στον x-άξονα είναι ο αριθμός των αιτήσεων του client μέχρι να μεταβεί στο επόμενο κελί.



Μην σας απασχολεί η ερμηνεία της επίδοσης των άλλων αλγορίθμων που εικονίζονται στο γράφημα.]



Πρόβλημα-02

Θυμηθείτε τους δυο αλγορίθμους LTWF (ή LWF) και RxW για on-demand broadcast. Θεωρήστε την ακραία περίπτωση ότι ο request arrival rate κάθε στοιχείου παραμένει σταθερός και ίσος με D_i για κάθε στοιχείο i . Τα στοιχεία είναι ισομεγέθη.

Ποιος από τους δυο αλγορίθμους – ο RxW ή ο LWF – αποκλίνει από την απαίτηση του θεωρήματος Square-Root για την σχέση μεταξύ δημοφιλίας και ρυθμού εκπομπής (δηλαδή, μήκους interarrival διαστήματος);



Πρόβλημα-03

Δίνονται τα παρακάτω στοιχεία με τις αντίστοιχες σχετικές πιθανότητες προσπέλασης. Να δείχτει πώς θα εκπαιφθεί το δυαδικό ευρετήριο (να είναι search tree) καθώς και τα δεδομένα στο κανάλι εκπομπής.

Τα ζεύγη στοιχείο-πιθανότητα είναι: (A 13), (B 24), (C 2), (D 15), (E 7), (F 11), (G 15), (H 20), (I 19), (J 9), (K 12), (L 8), (M 12), (N 14), (O 10).



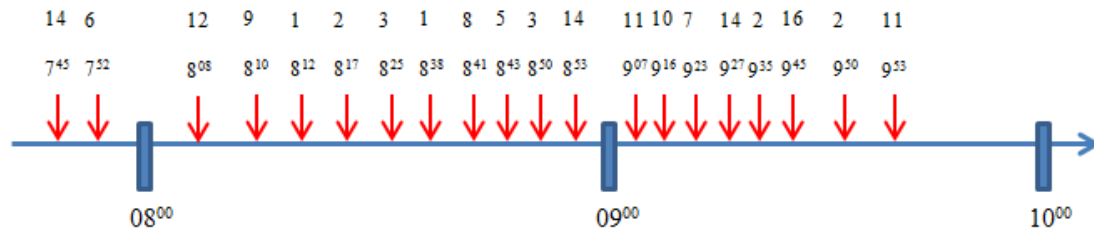
Πρόβλημα-04

Περιγράψτε πώς θα πρέπει να οργανώσει ο server τις BitSequences-based IR, ώστε οι clients να μην χρειάζεται να συντονιστούν και να λάβουν ολόκληρη την πληροφορία (την πλήρη IR), παρά να συντονίζονται επιλεκτικά για να πάρουν μόνο εκείνο το κομμάτι που τους ενδιαφέρει. Δηλαδή προτείνετε ένα tuning-optimized BitSequence-based IR σχήμα.



Πρόβλημα-05

Θεωρήστε έναν broadcast server που μπορεί να εκπέμψει προς τους κινητούς πελάτες του τα ακόλουθα 16 ισομεγέθη στοιχεία, d1, d2, d3, d4, d5, d6, d7, d8, d9, d10, d11, d12, d13, d14, d15, και d16. Έστω ότι η τρέχουσα χρονική στιγμή είναι 10:00, και ότι οι τροποποιήσεις που έχουν λάβει χώρα για τα στοιχεία αυτά εικονίζονται στο παρακάτω σχήμα (ο επάνω ακέραιος είναι το ID του στοιχείου, και πάνω αναγράφεται η χρονική στιγμή τροποποίησης).



Την τρέχουσα χρονική στιγμή θα εκπεμφθεί Invalidation Report με την μορφή BitSequences. Οι IRs εκπέμπρονται ανά δέκα λεπτά. Έστωσαν οι παρακάτω τρεις πελάτες στο κελί που έχουν caches μεγέθους 4, εκκρεμείς αιτήσεις pending_queries, και έχουν λάβει τελευταία φορά (π.χ., επειδή έκλεισαν το radio) Invalidation Report την χρονική στιγμή Tlb και έχουν ενημερώσει κατάλληλα τις caches τους.

Client-1: Cache= {8, 14, 7, 2} Pending_reqs {5, 14} Tlb= 08:30

Client-2: Cache= {5, 9, 12, 1} Pending_reqs {5, 9, 15} Tlb= 08:50

Client-3: Cache= {2, 7, 10, 6} Pending_reqs {2, 4, 7, 10} Tlb= 09:30

- Να κατασκευάσετε την IR Με μορφή BitSequences που θα πρέπει να εκπέμψει ο server.
- Να βρείτε ποια από τα στοιχεία της cache του μπορεί να χρησιμοποιήσει ο κάθε client κάνοντας ρητά χρήση της λογικής των BitSequences που έχετε κατασκευάσει στο παραπάνω βήμα. [Ο client δεν γνωρίζει πότε έγιναν τροποποιήσεις στα στοιχεία, όπως βλέπετε εσείς στο διάγραμμα. Η μόνη πληροφορία που διαθέτει είναι αυτή που υπάρχει στις BitSequences.]



Πρόβλημα-06

Υπό ποιες συνθήκες/παραδοχές ο MinSAUD ταυτίζεται με τον PIX στις αποφάσεις αντικαταστάσεως στοιχείων;



Πρόβλημα-07

Ένας server έχει συγκεντρώσει τα ακόλουθα στοιχεία δημοφιλίας για τα ισομεγέθη στοιχεία που πρόκειται να βάλει σε ένα pure push-based πρόγραμμα εκπομπής:

a: 0.40 b: 0.25 c: 0.15 d: 0.10 e: 0.10

Έστω ότι το προηγούμενο πρόγραμμα που είχε τελειώσει με τις εξής εκπομπές στοιχείων: abcadbea και πλέον θα φτιάξει το νέο πρόγραμμα με τις δεδομένες δημοφιλίες.

Θεωρούμε ότι η χρονική στιγμή 'μηδέν' είναι η στιγμή που ξεκινά το νέο πρόγραμμα εκπομπής.

Ένας πελάτης με μέγεθος cache ίση με 2 θέσεις, που περιέχει ήδη τα στοιχεία c και d, μπαίνει στο κελί του server αυτού, και αρχίζει να κάνει αιτήσεις από τη χρονική στιγμή 'μηδέν' με βάση το εξής ρεύμα αιτήσεων: abedaabbd. Ο πελάτης κάνει συνεχώς αιτήματα αλλά είναι blocking, δηλαδή κάνει ένα νέο αίτημα μόνο αφού έχει πάρει το στοιχείο που ζήτησε στο προηγούμενο αίτημά του.

Θεωρώντας δεδομένο ότι ο κινητός πελάτης γνωρίζει το προηγούμενο καθώς και το νέο πρόγραμμα εκπομπής του server, να υπολογίσετε – για το δεδομένο ρεύμα αιτήσεων του κινητού – το hit ratio της cache καθώς και τη μέση καθυστέρηση για τα αιτήματα του πελάτη, εάν η δημοφιλία των στοιχείων στον πελάτη είναι η εξής (σχετική, και όχι normalized στο 1):

a: 5 b: 4 c: 3 d: 2 e: 1

και η πολιτική αντικατάστασης στην cache του πελάτη είναι η PIX.



Πρόβλημα-08

Να βρείτε με ποια σειρά πρέπει να κάνει paging στα κελιά το δίκτυο, ώστε να ελαχιστοποιήσει το κόστος paging, όταν είναι γνωστό ότι ο κινητός έχει διαγράψει την εξής τροχιά: `aab|bca|aaa|cda|abb|abc|baa|`. (Οι κάθετες γραμμές δεν είναι κομμάτι του ιστορικού, αλλά βοηθούν στην ευαναγνωσιμότητα).



Πρόβλημα-09

Υπό ποιες συνθήκες/παράδοχές η πολιτική prefetching CBP ταυτίζεται με την PT? Ο αλγόριθμος CBP (Correlation-Based Prefetcher) έχει ως εξής:

1. Κάθε φορά που εκπέμπεται ένα στοιχείο, ανατίθεται μια τιμή CBP (την τιμή αυτή τη συμβολίζουμε G) σε κάθε στοιχείο που βρίσκεται στην cache καθώς και στο στοιχείο που εκπέμπεται. Έστω ότι τ_i είναι ο χρόνος που απομένει μέχρι το στοιχείο i να εκπεμφθεί ξανά και P_{ij} είναι η πιθανότητα ότι τα στοιχεία i και j θα ζητηθούν ταυτόχρονα. Επίσης, έστω ότι με C συμβολίζουμε το σύνολο στοιχείων που βρίσκονται στην cache και με Q_i το σύνολο των στοιχείων που θα εκπεμφθούν πριν το στοιχείο i εκπεμφθεί ξανά και τα οποία στοιχεία δεν περιέχονται στο C . Η CBP τιμή του στοιχείου i , G_i , ορίζεται ως εξής:

$$G_i = \tau_i \cdot \sum_{k \in C} P_{ik} + \sum_{k \in Q_i} (\tau_i - \tau_k) \cdot P_{ik}$$

2. G_j , που είναι η τιμή CBP του μόλις εκπεμφθέντος στοιχείου j , συγκρίνεται με τη μικρότερη τιμή CBP, G_{\min} , των στοιχείων της cache.
3. Εάν η G_j είναι μεγαλύτερη από την G_{\min} , το στοιχείο με τη μικρότερη τιμή CBT απομακρύνεται από την cache, και το j εισάγεται στην cache. Αλλιώς, δεν λαμβάνει χώρα αντικατάσταση στην cache.

Χρηστικές πληροφορίες:

Η προθεσμία παράδοσης είναι αυστηρή. Είναι δυνατή η παροχή παράτασης (μέχρι 3 ημέρες), αλλά μόνο αφού δώσει ο διδάσκων την έγκρισή του, και αυτή η παράταση στοιχίζει 15% ποινή στον τελικό βαθμό της συγκεκριμένης Σειράς Προβλημάτων. Η παράδοση γίνεται με email στο dkatsar@e-ce.uth.gr του αρχείου λύσεων σε μορφή pdf (typeset ή εξαιρετικής ποιότητας scanning/φωτογραφίας). Το subject του μηνύματος πρέπει να είναι: ECE515-Problem set 01: AEMx-AEMy

Ερμηνεία συμβόλων:



Δεν απαιτεί την χρήση υπολογιστή ή/και την ανάπτυξη κώδικα.



Απαιτεί την χρήση του Web για ανεύρεση πληροφοριών ή διεξαγωγή πειράματος.



Απαιτεί την ανάπτυξη κώδικα σε κάποια γλώσσα προγραμματισμού. Το παραδοτέο θα περιέχει:

- ❖ Τον ψευδοκώδικα υλοποίησης.
- ❖ Ένα παράδειγμα εκτέλεσης (με μικρή είσοδο).
- ❖ Τον πραγματικό πηγαίο κώδικα υλοποίησης.