UNIVERSITY OF THESSALY
SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL
& COMPUTER ENGINEERING

Προχωρημένη Κατανεμημένη Υπολογιστική
Χειμερινό Εξάμηνο 2019-2020
Δημήτριος Κατσαρός

Σειρά προβλημάτων: 3η: **ΑΤΟΜΙΚΕΣ & ΟΜΑΔΙΚΕΣ ΕΡΓΑΣΙΕΣ**

Ημέρα ανακοίνωσης: Sunday, December 01, 2019

Προθεσμία παράδοσης: Κυριακή, Φεβρουάριος 02, 2020

Πρόβλημα-01

Η τεχνική consistent hashing βασίζεται στον ορισμό k hashing συναρτήσεων $\{h_0, ..., h_{k-1}\}$ που αντιστοιχίζουν το μοναδικό όνομα ενός κόμβους και τα object ids σε hashes. Υπάρχουν διάφορες κατασκευές γι' αυτές τις hash functions, με τις συνηθέστερες να είναι η iterative hashing και η salted hashing. Στην iterative hashing χρησιμοποιούμε μια hash function *h* και την εφαρμόζουμε επαναληπτικά, έτσι ώστε οι hashes ενός αντικειμένου id o ορίζεται ως εξής:

$$h_i(o) = \begin{cases} h(o) & \text{εάν } i=0 \\ h(h_{i-1}(o)) & \text{αλλιώς.} \end{cases}$$

Στην τεχνική του salted hashing το object id συνενώνεται με την hash function index i με συνέπεια τον ακόλουθο ορισμό:

$$h_i(o) = h(o \mid i).$$

Ποια από τις δυο τεχνικές είναι καλύτερη; Ποια είναι τα πλεονεκτήματά της;

Πρόβλημα-02

Στην σχετική διάλεξη στην τάξη, "αποδείξαμε" ότι το false positive probability (fpp) ενός Bloom Filter (BF) είναι ίσο με $fpp=(1-(1-1/m)^{kn})^k$, όπου m είναι το μέγεθος σε bits του BF, n είναι ο αριθμός των εισαχθέντων στοιχείων, και k είναι ο αριθμός των hash functions. Έστω ότι φτιάχνουμε ένα BF με τις εξής τιμές παραμέτρων: n = 1, k = 2, m = 2.

- Εάν χρησιμοποιήσουμε τον προαναφερθέντα τύπο, τότε βρίσκουμε ότι το false positive probability αυτού του BF είναι ίσο προς $fpp_{Bloom}=0.5625$.
- Κάνοντας εξαντλητική απαρίθμιση των πιθανών γεγονότων (δυο στοιχεία εμπλέονται στην ανάλυση: το εισαχθέν, και ένα ακόμη το οποίο είναι το queried), βρείτε το πραγματικό fpp_{real}. [Δεν χρειάζεται να προσδιοριστούν ποιες είναι οι hash functions.]
- Εάν υπάχει σημαντική διαφορά μεταξύ των δυο, αιτιολογήστε πού οφείλεται αυτή.

Πρόβλημα-03

Θεωρήστε δυο σύνολα ακεραίων, $S = \{s_1, s_2, ..., s_m\}$ και $T = \{t_1, t_2, ..., t_n\}$, $m \leq n$. Υποθέστε ότι έχουμε κατασκευάσει ένα Bloom filter μεγέθους r, και k hash functions $h_1, h_2, ...h_k : U \rightarrow [0, r-1]$ (U είναι το σύνολο όλων των πιθανών keys).

- Περιγράψτε έναν αλγόριθμο ο οποίος ελέγχει κατά πόσον το S είναι υποσύνολο του T σε O(n) worst time.
- Ποια είναι η πιθανότητα να πάρουμε θετική απάντηση παρόλο που το S ΔΕΝ είναι υποσύνολο του T;

  [Ως συνάρτηση των n, m, r και $|S \cap T|$, μπορείτε να υποθέστε ότι k = O(1).]

Πρόβλημα-04

Θεωρήστε δυο 2 Bloom filters ιδίου μεγέθους και με το ίδιο σύνολο hash functions. Απαντήστε στα ερωτήματα A, B, C επιλέγοντας την σωστή (ή σωστές απαντήσεις).

A. Είναι το bitwise OR των δυο Bloom filters, το ίδιο με το Bloom filter που θα προκύψει από την εισαγωγή των στοιχείων της ενώσεως των δυο συνόλων; Γιατί;
   i. Όχι, ένα Bloom filter αλλάζει εντελώς όταν αλλάξουμε το σύνολο των εισόδων.
   ii. Όχι, ένα Bloom filter εξαρτάται από την σειρά εισόδου των δεδομένων προς εισαγωγή.
   iii. Ναι, στην πραγματικότητα τα δυο Bloom filters μπορούν να συνδυαστούν με χρήση οποιασδήποτε δυαδικής λογικής λειτουργίας.
   iv. Ναι, η λειτουργία OR δουλεύει, επειδή δεν χάνει κανένα "1" bit που είχε ενεργοποιηθεί σε κάποιο από τα δυο αρχικά Bloom filters.

B. Έστω ότι εκτελούμε bitwise AND των δυο Bloom filters και χρησιμοποιούμε το αποτέλεσμα ως Bloom filter για την τομή των δυο συνόλων. Τι μπορούμε να πούμε για τα false negatives;
   i. Δεν θα υπάρχουν false negatives, αφού για κάθε μέλος οι τιμές των bit στις αντίστοιχες στήλες για τις δυο γραμμές θα είναι "1" και για τα δυο Bloom filters και η λειτουργία AND θα το διατηρήσει..
   ii. Δεν θα υπάρχουν false negatives, αφού τα Bloom filters μπορούν να συνδιαστούν με χρήση οποιασδήποτε δυαδικής λογικής λειτουργίας.
   iii. Το ποσοστό false negative θα αυξηθεί, γιατί θα υπάρχει πάντα μια θέση στο Bloom filter η οποία αντιστοιχούσε στην τιμή "1" στο ένα Bloom filter και στην τιμή "0" στο άλλο.
   iv. Τίποτε δεν μπορεί να ειπωθεί με βεβαιότητα.

C. Έστω ότι εκτελούμε bitwise AND των δυο Bloom filters και χρησιμοποιούμε το αποτέλεσμα ως Bloom filter για την τομή των δυο συνόλων. Τι μπορούμε να πούμε για τα false positives;
   (1). Η πιθανότητα των false positives στο προκύπτον Bloom filter είναι το πολύ ίση με την false-positive πιθανότητα των συνιστώντων Bloom filters.
   (2). Η πιθανότητα των false positives μπορεί να είναι μεγαλύτερη από την false positive πιθανότητα του Bloom filter το οποίο κατασκευάζεται από την αρχή χρησιμοποιώντας την τομή των δυο συνόλων.
   (3). Η πιθανότητα των false positives θα είναι το πολύ ίση με εκείνη του Bloom filter το οποίο κατασκευάζεται από την αρχή χρησιμοποιώντας την τομή των δυο συνόλων.
   i. 1 & 3
   ii. 1 & 2
   iii. 2
   iv. 3

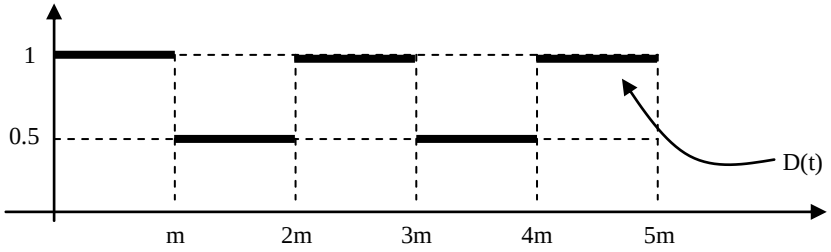
Πρόβλημα-05

Θεωρήστε ένα datacenter όπου η συνάρτηση ζήτησης μιας υπηρεσίας (demand function D(t)) εξελίσσεται εκθετικά και περιοδικά στο χρόνο ως εξής: Για τις χρονικές στιγμές [1...3), ακολουθεί την μορφή $D(t)=e^{t-1}$. Κατόπιν, την χρονική στιγμή 3 πέφτει στο μηδέν, και για το διάστημα [3...5) ακολουθεί την μορφή $D(t)=e^{t-3}$, κ.ο.κ. μέχρι την χρονική στιγμή 10. Το datacenter έχει τη δυνατότητα να παρακολουθεί <u>συνεχώς</u> τη μεταβολή της ζήτησης <u>χωρίς</u> να έχει δυνατότητα πρόβλεψής της. Χρειάζεται όμως χρόνο μιας χρονικής μονάδας για ν' ανταποκριθεί στη ζήτηση. Το μοναδιαίο κόστος overprovisioning (δηλαδή όταν demand >

resource) είναι c, ενώ το κόστος underprovisioning (δηλαδή όταν demand < resource) είναι επίσης ίσο προς c.
Συνηθίζουμε να λέμε: "*κάλλιο αργά, παρά ποτέ*". Ισχύει στην περίπτωσή μας αυτό το ρητό ή μήπως ισχύει το "κάλλιο ποτέ, παρά αργά" (δηλαδή να μην ανταποκριθούμε καθόλου); Για ποιες τιμές του c;
[Από τη στιγμή που το DC ανιχνεύει το επίπεδο μιας τιμής ζήτησης και δρομολογεί την εξυπηρέτησή της, δεν μπορεί να υπαναχωρήσει από αυτή του την απόφαση.]

Πρόβλημα-06

Θεωρήστε ένα datacenter όπου η συνάρτηση ζήτησης μιας υπηρεσίας (demand function D(t)) ακολουθεί τη μορφή που εικονίζεται στο σχήμα. Το datacenter έχει τη δυνατότητα να παρακολουθεί περιοδικά ανά m χρονικές στιγμές τη μεταβολή της ζήτησης χωρίς να έχει δυνατότητα πρόβλεψής της, και χρειάζεται χρόνο $t_p=m/2$ για ν' ανταποκριθεί στη ζήτηση.
Πόση είναι η ζημία που υφίσταται ο cloud provider σε χρόνο T=5m υιοθετώντας τη συνάρτηση απώλειας (loss function) που έχουμε παρουσιάσει στις διαλέξεις και κατά την οποία το μοναδιαίο κόστος overprovisioning (δηλαδή όταν demand > resource) είναι c, ενώ το κόστος underprovisioning (δηλαδή όταν demand < resource) είναι ίσο προς 3c;
[Από τη στιγμή που το DC ανιχνεύει το επίπεδο μιας τιμής ζήτησης και δρομολογεί την εξυπηρέτησή της, δεν μπορεί να υπαναχωρήσει από αυτή του την απόφαση.]



Πρόβλημα-07

Θεωρήστε ένα datacenter όπου η συνάρτηση ζήτησης μιας υπηρεσίας (demand D(t)) είναι ομοιόμορφα κατανεμημένη μεταξύ του μηδέν και μιας γνωστής μέγιστης τιμής *MAX*. Το κόστος του underprovisioning είναι γραμμική συνάρτηση του "ποσού" του underprovisioning με σταθερά αναλογίας k_1. Το κόστος του overprovisioning είναι γραμμική συνάρτηση του "ποσού" του overprovisioning με σταθερά αναλογίας k_2.
Να βρεθεί η βέλτιστη τιμή του προσφερόμενου resource $R_{opt}(t)$, ώστε να ελαχιστοποιηθεί το συνολικό κόστος, όταν:

- $k_1=2k_2$.
- $k_2=2k_1$.

Πρόβλημα-08

Θεωρήστε ένα datacenter όπου η συνάρτηση ζήτησης μιας υπηρεσίας (demand D(t)) είναι ομοιόμορφα κατανεμημένη μεταξύ του μηδέν και μιας γνωστής μέγιστης τιμής *M*. Το κόστος του underprovisioning είναι τετραγωνική συνάρτηση του "ποσού" του underprovisioning με σταθερά αναλογίας k_1. Το κόστος του overprovisioning είναι γραμμική συνάρτηση του "ποσού" του overprovisioning με σταθερά αναλογίας k_2.
Να βρεθεί η βέλτιστη τιμή του προσφερόμενου resource $R_{opt}(t)$, ώστε να ελαχιστοποιηθεί το συνολικό κόστος.

Πρόβλημα-09

Να γραφεί κώδικας MapReduce στο Hadoop για τον υπολογισμό του Kronecker product δυο πινάκων ακεραίων (https://www.geeksforgeeks.org/kronecker-product-two-matrices/). Να υποθέσετε ότι κάθε πίνακας είναι αποθηκευμένος σε ένα txt αρχείο, και σε κάθε γραμμή κάθε αρχείου υπάρχουν: ο αριθμός_γραμμής και κατόπιν μια σειρά ακεραίων χωρισμένων

με space. Δεν καταγράφεται πουθενά η διάσταση του πίνακα. Π.χ., ένα μικρό παράδειγμα των δυο πινάκων θα ήταν ως ακολούθως:
Αρχείο-1.txt
1 1 2
2 3 4
Αρχείο-2.txt
1 0 5
2 6 7

Το Kronecker product των δυο πινάκων θα πρέπει να είναι το ακόλουθο:
1 0 5 0 10
2 6 7 12 14
3 0 15 0 20
4 18 21 24 28

Πρόβλημα-10

Στην παρούσα άσκηση πρέπει να υλοποιήσετε equi-join algorithms στο Hadoop με την βοήθεια ενός Bloom filter. Ο αλγόριθμος που καλείστε να υλοποιήσετε είναι αυτός του άρθρου: https://dl.acm.org/citation.cfm?id=2401626. Επιπλέον βιβλιογραφία μπορείτε ν' αναζητήσετε στο άρθρο: https://link.springer.com/chapter/10.1007/978-3-662-49534-6_2.

Χρηστικές πληροφορίες:
Η προθεσμία παράδοσης είναι αυστηρή. Είναι δυνατή η παροχή παράτασης (μέχρι 2 ημέρες), αλλά μόνο αφού δώσει ο διδάσκων την έγκρισή του και αυτή η παράταση στοιχίζει 10% ποινή στον τελικό βαθμό της συγκεκριμένης Σειράς Προβλημάτων. Η παράδοση γίνεται με email του αρχείου λύσεων σε μορφή pdf (typeset). Το subject του μηνύματος πρέπει να είναι: CE623-Problem set 03: AEMx

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 MapReduce στο Hadoop. Το παραδοτέο θα περιέχει:
- Τον ψευδοκώδικα υλοποίησης (για όσα ζεύγη map-reduce απαιτούνται).
- Ένα παράδειγμα εκτέλεσης (με μικρή είσοδο) που θα παρουσιάζει τις φάσεις εκτέλεσης των map και reduce που σχεδιάσετε.
- Τον πραγματικό πηγαίο κώδικα (π.χ., Java, Python) υλοποίησης

Εργασία σε ομάδα των 2 ατόμων.
Κάθε μέλος της ομάδας παραδίδει ξεχωριστά το αποτέλεσμα της κοινής εργασίας, αναφέροντας φυσικά το ΑΕΜ του συνεργάτη του.

Ατομική εργασία.