



Προχωρημένη Κατανεμημένη Υπολογιστική
Χειμερινό Εξάμηνο 2019-2020
Δημήτριος Κατσαρός

Σειρά προβλημάτων: 2^η: ΑΤΟΜΙΚΕΣ & ΟΜΑΔΙΚΕΣ ΕΡΓΑΣΙΕΣ

Ημέρα ανακοίνωσης: Monday, November 11, 2019

Προθεσμία παράδοσης: Κυριακή, Δεκέμβριος 15, 2019



Πρόβλημα-01

Δίνονται οι τρεις φυσικές μηχανές P1, P2 και P3, που στεγάζουν τις παρακάτω εικονικές μηχανές:

P1: VM₁(0.3, 0.3), VM₂(0.2, 0.4), VM₃(0.3, 0.25)

P2: VM₄(0.45, 0.1)

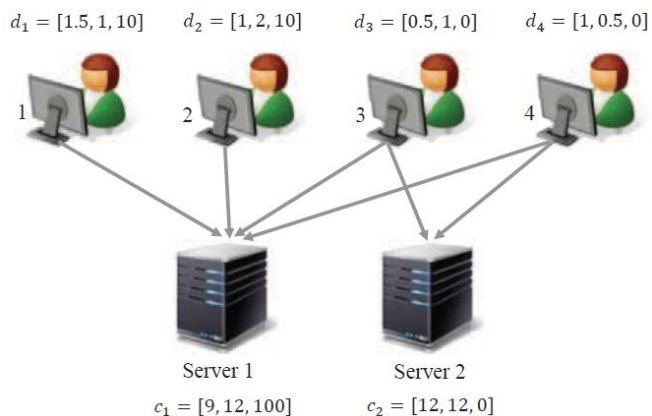
P3: VM₅(0.35, 0.25)

Οι αριθμοί εντός των παρενθέσεων είναι οι απαιτήσεις σε cpu και μνήμη, αντίστοιχα. Έστω ότι η P1 βρέθηκε ότι είναι hot spot. Πώς πρέπει να γίνει το migration κάποιας (ποιας) VM της P1 σύμφωνα με τον αλγόριθμο VSR, και πώς σε σχέση με τον αλγόριθμο dot-product; [Να φανούν τα γραφήματα τελικής ανάθεσης.]



Πρόβλημα-02

Δίνεται ένα cluster από ετερογενείς servers το οποίο παρέχει τρεις πόρους, π.χ., cpu, memory και network bandwidth. Ένας αριθμός χρηστών του cluster παρουσιάζουν τις απαιτήσεις τους προς τους πόρους, όπως οι χρήστες που εικονίζονται στο διπλανό σχήμα. Π.χ., ο πρώτος χρήστης χρειάζεται 1.5 μονάδα cpu, 1 μονάδα memory, και 10 μονάδες bandwidth.



Είναι πιθανόν κάποιος χρήστης να μην έχει απαιτήσεις για κάποιον πόρο, όπως για παράδειγμα ο τρίτος και το τέταρτος χρήστης για το bandwidth. Το βέλος από κάποιον χρήστη προς κάποιον server σημαίνει ότι υπάρχουν placement constraints, δηλαδή ότι οι tasks αυτού του χρήστη μπορούν να εκτελεστούν σε εκείνον τον server. Για παράδειγμα οι tasks του πρώτου χρήστη μπορούν να εκτελεστούν μόνο στον πρώτο server.

Επινοήστε μια γενίκευση του DRF (ή αναζητήστε κάποια στην βιβλιογραφία) η οποία να μπορεί να λειτουργήσει σε περιβάλλον ετερογενών servers (δηλαδή, δεν είναι όλοι οι πόροι σε ένα ενιαίο pool) και να σέβεται τα placement constraints. Ο αλγόριθμος που θα επινοήσετε δεν είναι ανάγκη να κατέχει τις ιδιότητες που έχει ο DRF τις οποίες είδαμε στο μάθημα, ούτε φυσικά ν' αποδειξτε ποιες ιδιότητες κατέχει. Κατόπιν, εκτελέστε τον αλγόριθμο που επινοήσατε στο περιβάλλον του σχήματος για να βρείτε την τελική ανάθεση.



Πρόβλημα-03

Δίνεται ένα cluster που διαθέτει 19 CPUs και 21 GBs, και δυο χρήστες των οποίων οι tasks έχουν απαιτήσεις σε πόρους $\langle 1,2 \rangle$ και $\langle 3,1 \rangle$, αντίστοιχα. Να διατυπωθεί το πρόβλημα DRF ως πρόβλημα γραμμικού προγραμματισμού, και να αποδοθεί και επιλυθεί γραφικά. Εάν η λύση δεν είναι ακέραια, τότε να χρησιμοποιήσετε την μέθοδο του tableau (progressive filling) που εξηγήθηκε στην τάξη για να βρείτε την ακέραια λύση του.



Πρόβλημα-04

Ένα αποθηκευτικό σύστημα αποτελείται από μαγνητικό δίσκο (HD) και δίσκο στερεάς κατάστασης (SSD). Κάθε πελάτης κάνει μια σειρά αιτήσεων. Κάθε αίτηση απευθύνεται είτε στον HD είτε στον SSD. Μια προσπέλαση στον SSD θα καλείται hit και κάθε προσπέλαση στον HD θα καλείται miss.



Το hit (miss) ratio του πελάτη i είναι το ποσοστό των IO αιτήσεων του προς τον SSD (HD) και συμβολίζεται ως h_i (m_i). Υποθέτουμε εφαρμογές του κάθε πελάτη με σταθερό h_i (m_i). Υποθέστε ότι έχουμε HD και SSD με “χωρητικότητες” 10 IOPS και 50 IOPS, αντίστοιχα. Στο αποθηκευτικό αυτό σύστημα υπάρχουν δυο πελάτες με hit ratios $h_1 = 0.5$ και $h_2 = 0.9$, αντίστοιχα. Βρείτε πώς ο DRF θα καταναίμει το bandwidth στο σύστημα αυτό. Στην ουσία πρέπει να βρείτε πόσα IOPS από τον HD και πόσα IOPS από τον SSD θα πάρει κάθε πελάτης.

[Αναγνωρίστε πρώτα την dominant resource για κάθε πελάτη.] [Στο τέλος θα βρείτε κλασματικά IOPS.]

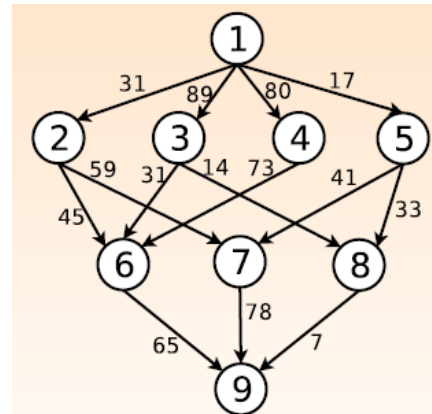


Πρόβλημα-05

Δίνεται το παρακάτω γράφημα δεξιά που περιγράφει τις εξαρτήσεις μεταξύ εργασιών και τον όγκο δεδομένων που απαιτείται να μεταφέρουν μεταξύ τους, και ο παρακάτω πίνακας αριστερά που περιγράφει το χρόνο εκτέλεσης κάθε εργασίας σε 3 ετερογενείς υπολογιστές. Να βρεθεί η ανάθεση των εργασιών στους υπολογιστές κατά HEFT (Heterogeneous Earliest Finish Time).



	P1	P2	P3
1	19	41	34
2	28	46	20
3	36	34	62
4	15	25	37
5	30	50	54
6	33	35	59
7	12	20	21
8	13	22	24
9	41	68	73



Πρόβλημα-06

Σε ένα γράφημα που εικονίζει τις προτεραιότητες μεταξύ εργασιών, και τα δεδομένα που πρέπει να μεταφερθούν μεταξύ αυτών, παράξτε μια διάταξη με βάση τον αλγόριθμο PageRank. Το PageRank ενός πατέρα να μοιράζεται στα παιδιά του ανάλογα με τον όγκο των δεδομένων που πρέπει να μεταφερθεί. Διατάξτε τις εργασίες από εκείνη με το μικρότερο PageRank προς εκείνη με το μεγαλύτερο, και κατόπιν χρησιμοποιήστε αυτήν την διάταξη για να αναθέσετε εργασίες σε επεξεργαστές. Τα δεδομένα εισόδου είναι αυτά της προηγούμενης άσκησης.



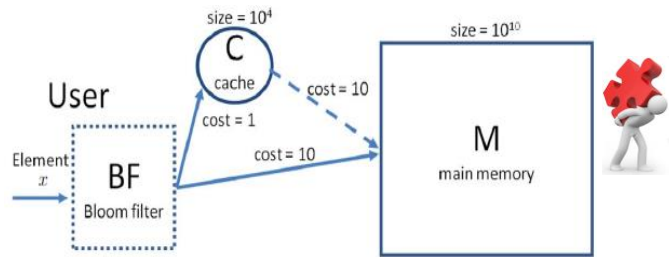
Εάν η τελική διάταξη δεν σέβεται τους περιορισμούς προτεραιότητας, επινοήστε μια λύση.

- Να βρεθεί η ανάθεση των εργασιών στους επεξεργαστές.
 - Παράγει μικρότερο makespan από το αντίστοιχο του HEFT;
- Η τιμή του damping factor για τον υπολογισμό του PageRank είναι $d=0.15$.



Πρόβλημα-07

Θεωρήστε ένα γενικό σύστημα που περιλαμβάνει έναν χρήστη, μια main memory η οποία περιέχει όλα τα δεδομένα, και μια cache που κρατά ένα υποσύνολο των δεδομένων. (Θα μπορούσε αυτό να είναι ένα key-value store σύστημα, όπου ως “main memory” λογίζεται ο δίσκος, και ως cache λογίζεται η memcached).



Όταν ο χρήστης επιθυμεί να διαβάσει κάποιο δεδομένο, μπορεί να προσπελάσει άμεσα την main memory με κόστος 10. Εναλλακτικά, μπορεί να προσπελάσει πρώτα την cache με κόστος 1. Εάν η cache περιέχει το δεδομένο, τότε δεν υπάρχει επιπρόσθετο κόστος. Αλλιώς, θα πρέπει να προσπελάσει κατόπιν την main memory με επιπρόσθετο κόστος 10. Υποθέστε ότι ο χρήστης έχει ένα Bloom Filter (BF) το οποίο “δείχνει” ποια στοιχεία βρίσκονται στην cache, και αυτό το BF έχει false positive probability (fpp) ίση με 10^{-3} .

Υποθέστε ότι τα περιεχόμενα της cache (10^4 στοιχεία) επιλέγονται uniformly at random από τα περιεχόμενα της main memory (10^{10} στοιχεία), και έχουν την ίδια δημοφιλία για τον χρήστη. Προφανώς τα στοιχεία της cache είναι ανά δυο διακριτά.

- Ποιο είναι το μέσο κόστος προσπέλασης, εάν δεν χρησιμοποιήσουμε καθόλου την cache και προσπελάζουμε απευθείας την main memory;
- Ποιο είναι το μέσο κόστος προσπέλασης εάν πρώτα χρησιμοποιήσουμε το BF για να ελέγξουμε την cache, και κατόπιν προσπελάσουμε την main memory (εάν χρειαστεί);

Ποιο από τα δυο είναι υψηλότερο; Γιατί; Είναι φυσιολογικό γεγονός ή παράδοξο;



Πρόβλημα-08

Έστω ένα Bloom filter που φτιάχνεται για να στεγάσει n στοιχεία, έχει μήκος m bits και θα χρησιμοποιήσει k hash functions, όπου η καθμία τους μπορεί να ενεργοποιήσει (set) οποιοδήποτε από τα m bits. Στις διαλέξεις του μαθήματος έχουμε υπολογίσει την πιθανότητα εμφάνισης ενός false positive για αυτό το Bloom filter. Έστω τώρα ότι το Bloom filter (δηλαδή τα m bits) χωρίζεται σε k ισομεγέθη κομμάτια (ένα για κάθε μια hash function) και κάθεμιά τους μπορεί να ενεργοποιήσει κάποιο bit που όμως μπορεί να βρίσκεται μόνο στο δικό της κομμάτι. Να υπολογίσετε την πιθανότητα των false positives.



Πρόβλημα-09

Θεωρήστε δυο σύνολα ακεραίων, $S = \{s_1, s_2, \dots, s_m\}$ και $T = \{t_1, t_2, \dots, t_n\}$, $m \leq n$.

Υποθέστε ότι έχουμε κατασκευάσει ένα Bloom filter μεγέθους r , και k hash functions $h_1, h_2, \dots, h_k : U \rightarrow [0, r-1]$ (U είναι το σύνολο όλων των πιθανών keys).

- Πειράξτε 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 υποθέσετε ότι $k = O(1)$.]

Για τις επόμενες 3 ασκήσεις μελετήστε σχολαστικά την ακόλουθη μεταπτυχιακή διατριβή: http://inf-server.inf.uth.gr/courses/CE623/noexternalweb/Join_Algorithms_using_MapReduce.pdf. Τα δεδομένα δυο σχεσιακών πινάκων που θα χρησιμοποιήσετε βρίσκονται στα δυο φύλλα του εξής αρχείου: http://inf-server.inf.uth.gr/courses/CE623/adc_fall17/assignments/twotables.xlsx.



Πρόβλημα-10

Να υλοποιηθεί το Reduce-side Join. Ισότητα στο join attribute. Είναι two-way join.





Πρόβλημα-11

Να υλοποιηθεί το Map-side Join. Ισότητα στο join attribute. Είναι two-way join. Ερμηνεύεται στις 2 τελευταίες παραγράφους της σελίδας 27 του προαναφερθέντος άρθρου.



Πρόβλημα-12

Να υλοποιηθεί το Broadcast Join. Ισότητα στο join attribute. Είναι two-way join. Περιγράφεται εδώ : “Join Query Processing in MapReduce Environment”, International Conference on Advances in Communication, Network, and Computing, 2012. https://link.springer.com/chapter/10.1007/978-3-642-35615-5_42 (sec. 3.2)



Χρησιμικές 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 10% 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). Το subject του μηνύματος πρέπει να είναι: CE623-Problem set 02: 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, εάν φυσικά το επιθυμείτε να εργαστείτε σε ομάδα. Κάθε μέλος της ομάδας παραδίδει ξεχωριστά το αποτέλεσμα της κοινής εργασίας.



Ατομική εργασία.