ΔΙΑΤΑΞΕΙΣ ΑΙΤΗΣΕΩΝ ΣΕ ΔΙΣΚΟΥΣ ΜΕ ΔΥΟ ΚΕΦΑΛΕΣ ΑΝΑ ΕΠΙΦΑΝΕΙΑ

ΓΙΑΝΝΗΣ ΜΑΝΩΛΟΠΟΥΛΟΣ ΑΘΗΝΑ ΒΑΚΑΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ,

ΤΟΜΕΑΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ,

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ,

54006 ΘΕΣΣΑΛΟΝΙΚΗ.

ΠΕΡΙΛΗΨΗ

Οι κεφαλές των εμπορικών συστημάτων μαγνητικών δίσκων με δύο κεφαλες εγγραφής/ανάγνωσης ανά επιφάνεια βρίσκονται σε σταθερή απόσταση μεταξύ τους και δεν υπερβαίνουν τα όρια της επιφάνειας του δίσκου. Η ικανοποίηση των απαιτήσεων του συστήματος κοστολογείται με δύο τρόπους: με το "κόστος απάντησης" που σχετίζεται με το εύρος των κυλίνδρων που σαρώνεται για κάθε επιμέρους ερώτηση και το "κόστος μετάβασης" που εκφράζει τον αριθμό των κυλίνδρων που σαρώνονται κατά τη μετάβαση για την εξυπηρέτηση της επόμενης απαίτησης. Στόχος είναι η εύρεση της βέλτιστης διάταξης των απαιτήσεων ώστε να ελαχιστοποιηθεί το κόστους μετάβασης. Το πρόβλημα αντιμετωπίζεται ως μία ειδική περίπτωση του προβλήματος του "περιοδεύοντος πωλητή". Δίνονται αναλυτικοί τύποι για το κόστος απάντησης και το κόστος μετάβασης καθώς επίσης γίνεται και σύγκριση με συστήματα μίας κεφαλής.

1. ΕΙΣΑΓΩΓΗ

Συμβατικά συστήματα δίσκων με μία κεφαλή εγγραφής/αναγνωσης ανά επιφάνεια διατίθενται εμπορικά από δεκαετίες κι έχουν μελετηθεί αναλυτικά και πειραματικά. Τα τελευταία χρόνια διατιθενται στο εμπόριο και συστήματα με δύο κεφαλές ανά επιφάνεια (πχ. μοντέλα BURROUGHS FD120, DEC RA81, IBM 3380, SPERRY UNIVAC 8450). Οι κεφαλές αυτές βρίσκονται σε σταθερή απόσταση μεταξύ τους. Ταυτόχρονα γίνεται μελέτη για τη δυνατότητα κατασκευής συστηματων δίσκων με δύο κεφαλές ανά επιφάνεια που κινούνται ανεξάρτητα η μία απο τήν άλλη οδηγούμενες από δύο ελεγκτές.

Εστω ένα σύστημα αρχείων λειτουργικού συστήματος ή αρχεία μίας βάσης δεδομένων που είναι αποθηκευμένη στο συγκεκριμένο σύστημα δίσκου. Κάθε σύστημα διοίκησης βάσεων δεδομένων διαθέτει μηχανισμούς, όπως πχ. δευτερεύοντες κατάλογους, για τον εντοπισμό των διευθύνσεων των εγγραφών που ενδιαφέρουν το χρήστη. Μετα τον εντοπισμό των εγγραφών η απαιτηση μεταβιβάζεται στο λειτουργικό σύστημα για εκτέλεση. Εστω, τώρα ότι έρχεται ένα σύνολο T απαιτήσεων του συστήματος ή του χρηστη της βάσης. Οι απαιτησεις αυτες μπορεί να είναι πολλών ειδών: να αφορούν προσπέλαση σε ένα κύλινδρο, σε διαδοχικούς κυλίνδρους ή σε μη διαδοχικούς κυλίνδρους που επιλέγονται σύμφωνα με ένα πρότυπο προσπελασης απο το σύνολο των κυλίνδρων του δίσκου που συμβολίζεται με N (για ευκολία χωρίς μείωση της γενικότητας, έστω ότι το N είναι άρτιο). Συνεπώς οι απαιτήσεις διαφέρουν ως προς την πολυπλοκότητα, το περιεχόμενο και την έκταση τους.

Η βέλτιστη διάταξη ενός συνόλου ερωτήσεων σε μορφή μαζικής επεξεργασίας σε ένα δίσκο είναι παλιότερο και συζητημένο πρόβλημα [BuKo79, BiWo79, Ko78, MaKo88, Wo83]. Με την αντικατάσταση των συστημάτων αρχείων με την πιο εξελιγμένη τεχνολογία των βάσεων δεδομένων το πρόβλημα εμφανίζεται με νέες μορφές [Pa88, Se88, KoMaPa90]. Κοινός τόπος όλων αυτών των εργασιών είναι ότι αναφέρονται σε συμβατικά συστήματα δίσκων με μία κεφαλή ανά επιφάνεια.

Οι μέχρι στιγμής εργασίες για συστήματα δίσκων με δύο κεφαλές ανά επιφάνεια μελετούν συγκεκριμένες τεχνικές εξυπηρέτησης [Ho83, MaKo89, MaVa90, PaWo81], βέλτιστης τοποθέτησης δεδομένων [MaKo90, CaCoFl88] ή βέλτιστης απόστασης των κεφαλών [CaCoFl84, MaKo89 CaCoFl90]. Στην παρούσα εργασία μελετώνται αλγόριθμοι διάταξης ενος συνόλου ερωτησεων σε συστήματα δίσκων με δύο κεφαλές ανά επιφάνεια.

Το πρόβλημα εδώ αντιμετωπίζεται ως ειδική περίπτωση του προβλήματος του "περιοδεύοντος πωλητή" [LaLeKaSh86] και μπορεί να τοποθετηθεί ως εξής. Κάθε απαίτηση αφορά ένα σύνολο κυλίνδρων (υποσυνολο των N θέσεων) που ξεκινούν από τη θέση L και φτάνουν μέχρι τη θέση R. Για τη συνεχεια υιοθετούνται οι εξής συνθήκες:
(α) οι κεφαλές απέχουν σταθερή απόσταση $d=N/2$ [MaKo89] (Ετσι οι κύλινδροι που βρίσκονται στο διάστημα $[1,d]$ εξυπηρετούνται από τη μία κεφαλή και οι κύλινδροι που βρίσκονται στο διάστημα $[d+1,N]$ εξυπηρετούνται από την άλλη κεφαλή.),
(β) οι κεφαλές αρχικά είναι τοποθετημένεες επάνω από τις θέσεις 1 και $d+1$ αντίστοιχα,

(γ) ο αλγόριθμος δρομολόγησης των κεφαλών είναι ο CSCAN, σύμφωνα με τον οποίο οι κεφαλές εξυπηρετούν τις αιτήσεις κινούμενοι μόνο κατά τη μία κατεύθυνση (από αριστερα προς δεξιά) [ΠαΜπιΤσα86],
(δ) η εξυπηρέτηση μίας απαίτησης αρχίζει μόλις τελειώσει η εξυπηρέτηση της προηγούμενης, και
(ε) μετά την εξυπηρέτηση της τελευταίας απαίτησης οι κεφαλές δεν επιστρεφουν στη θέση 0.

Ο κυριότερος παράγοντας κόστους κατά την κίνηση των κεφαλών του δίσκου είναι ο λεγόμενος χρόνος αναζήτησης (seek time). Για το πρόβλημα μας το κόστος αυτό διακρίνεται σε δύο επιμέρους: το "κόστος απαντησης" που είναι ο χρόνος σαρωσης του απαιτούμενου εύρους των κυλίνδρων για την εξυπηρετηση καθε επιμέρους ερώτησης και το "κόστος μετάβασης" που είναι ο απαιτούμενος χρόνος κατά τη μετάβαση για την εξυπηρετηση της επόμενης ερώτησης. Και τα δύο κόστη πρακτικά μετρώνται με αριθμό των κυλίνδρων που σαρώνονται. Στόχος είναι να βρεθεί η βέλτιστη διάταξη των απαιτήσεων για την ελαχιστοποίηση του κόστους μετάβασης. Στη συνέχεια γίνεται αναφορα και στο κόστος απάντησης αλλά όχι με σκοπό την ελαχιστοποίηση του. Ούτως ή άλλως αυτό είναι αναπόφευκτο και σταθερό.

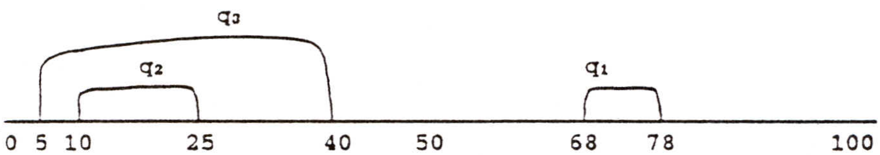
Το υπόλοιπο της εργασίας έχει οργανωθεί ως εξής. Στο δεύτερο κεφάλαιο εξετάζονται συστήματα με μία κεφαλή ανά επιφάνεια. Περιγραφεται ο αλγόριθμος εύρεσης της βέλτιστης διάταξης που ακολουθεί τη μέθοδο της "ενωσης υποδιαδρομών" του προβλήματος του περιοδεύοντος πωλητή [KoMaPa90]. Στο τρίτο κεφάλαιο περιγράφονται αλγόριθμοι διάταξης των απαιτήσεων σε συστήματα με δύο κεφαλές ανά επιφάνεια και δίνονται αναλυτικοί τύποι για το κόστος απάντησης και το κόστος μετάβασης. Ειδικότερα εξετάζονται τρεις περιπτώσεις:
(α) οι απαιτήσεις αφορούν μόνο στο διάστημα [1,d] ή μόνο στο διάστημα [d+1,N],
(β) οι απαιτήσεις ξεκινούν από το διάστημα [1,d] και καταλήγουν στο διάστημα [d+1,N],
(γ) οι απαιτήσεις ανήκουν και στις δύο παραπάνω κατηγορίες.
Τα συμπερασματα και προοπτικές για μελλοντική μελέτη περιέχονται στο τελευταίο καφάλαιο.

2. ΔΙΑΤΑΞΗ ΣΕ ΣΥΣΤΗΜΑΤΑ ΔΙΣΚΩΝ ΜΕ ΜΙΑ ΚΕΦΑΛΗ

Αρχικά περιγράφεται ο σχετικός αλγόριθμος για συστήματα δίσκων με μία κεφαλή ανά επιφάνεια [KoMaPa90]. Κάθε κύλινδρος του δίσκου παρομοιάζεται με μία πόλη όπου ο πωλητής μπορεί να σταματήσει. Συμπληρωματικά θεωρείται η πόλη c(0) για τη θέση 0 όπου αρχικά είναι η πρώτη κεφαλή. Η i-οστή απαίτηση ορίζεται από τις ακραίες πόλεις-κυλίνδρους L_i και R_i. Το κόστος απάντησης της ερώτησης αυτης είναι $R_i - L_i$. Το κόστος μετάβασης από την πόλη c(i) στην πόλη c(j) είναι $| R_i - L_j |$, το κόστος μετάβασης από τη θέση c(0) στη θέση c(i) είναι L_i, ενώ δεν υπάρχει κοστος για την επιστροφή στη θέση c(0). Ετσι το πρόβλημα της διαταξης των απαιτησεων μετασχηματίζεται στο πρόβλημα της συντομότερης διαδρομής ξεκινώντας από τη θέση c(0), περνώντας από όλες τις πόλεις και καταλήγοντας στο c(0).

Σύμφωνα με τη μέθοδο τα L_i και R_i αρχικά ταξινομούνται σε αύξουσα σειρά. Οι μετασχηματισμοί που προκύπτουν ονομάζονται P_1 και P_2. Κατόπιν σχηματίζεται η βέλτιστη ανάθεση όπου ανήκουν τα τόξα $(P_2(i), P_1(i+1))$, όπου $i=1,2,...$ Ετσι δημιουργούνται μερικοί κύκλοι, οι λεγόμενες υποδιαδρομές (subtours), με τόξα (c_i, c_j) που μπορεί να είναι και ασύνδετοι μεταξύ τους. Ο αλγόριθμος ενώνει διαδοχικά υποδιαδρομές μεταξύ τους μέχρι να καταλήξει σε μία μοναδικη, δηλαδή στη βέλτιστη διαδρομή. Η ένωση των υποδιαδρομών (subtour patching) γίνεται σε δύο φάσεις: στην πρώτη φάση ενώνονται υποδιαδρομές που έχουν κοινές περιοχές κυλίνδρων (δηλ. χωρίς αύξηση του κόστους), ενώ στη δεύτερη φάση γίνεται ένωση υποδιαδρομών ξένων μεταξύ τους με βάση τη μικρότερη απόσταση μεταξύ των κυλίνδρων τους. Ο αλγόριθμος αυτός βρίσκει τη βέλτιστη διαδρομή σε χρόνο O(nlogn). Ακολουθεί ένα παράδειγμα.

ΑΠΑΙΤΗΣΗ	L_i	R_i	ΚΟΣΤΟΣ ΑΠΑΝΤΗΣΗΣ
q_1	68	78	10
q_2	10	25	15
q_3	5	40	35



Σχ.1: Παράδειγμα αλγόριθμου για μία κεφαλή ανά επιφάνεια

Η εφαρμογή των μετασχηματισμών του αλγορίθμου στο παράδειγμα του Σχήματος 1 έχει ως εξής:

	P_1	P_2	
(c_3)	5	25	(c_2)
(c_2)	10	40	(c_3)
(c_1)	68	78	(c_1)

άρα

ΤΟΞΑ ΔΙΑΔΡΟΜΗΣ
c_0 --> c_3
c_2 --> c_2
c_3 --> c_1
c_1 --> c_0

Προκύπτουν επομένως δύο υποδιαδρομές:
η K_0 = [c_0 --> c_3 --> c_1 --> c_0] και η K_1 = [c_2 --> c_2].

Κατά την ένωση των υποδιαδρομών επειδή δεν υπάρχουν τεμνόμενες περιοχές κυλίνδρων η σύνδεση γίνεται με βάση τη δεύτερη φάση. Ετσι ενώνεται η πόλη c_0 με τη c_2 γιατί οι δύο αυτές βρίσκονται στη μικρότερη δυνατή απόσταση και προκύπτει η βέλτιστη διαδρομή:

c_0 --> c_2 --> c_3 --> c_1

Πράγματι για την περίπτωση της μίας κεφαλής αυτή η διαδρομή είναι η βέλτιστη δυνατή με συνολικό κόστος μετάβασης ίσο με 10 + 20 + 28 = 58 κυλίνδρους που πρέπει να διανυθούν για να βρεθεί η κεφαλή στην κατάλληλη θέση εξυπηρέτησης κάθε πόλης. Σημειώνεται όμως ότι εκτός απο το κόστος μετάβασης (58) υπαρχει και το κόστος απάντησης (60). Στην πραγματικότητα δηλ. η κεφαλή διανύει αριθμό κυλίνδρων ίσο με το άθροισμα του κόστους μετάβασης και του κοστους απάντησης.

Είναι εύκολο να αποδειχθεί ότι για κάθε πιθανή διάταξη των απαιτησεων το κόστος απαντησης είναι:

$$\sum_{i=1}^{T} | R_i - L_i | \tag{1}$$

ενώ το κοστος μετάβασης είναι:

$$L_1 + \sum_{i=1}^{T-1} | R_i - L_{i+1} | \tag{2}$$

Απο τον τύπο (1) φαίνεται ότι το κόστος απάντησης δεν επιδέχεται ελάττωση σε σχέση με τις πιθανές διατάξεις.

3. ΔΙΑΤΑΞΗ ΑΠΑΙΤΗΣΕΩΝ ΣΕ ΣΥΣΤΗΜΑΤΑ ΔΥΟ ΚΕΦΑΛΩΝ

Εστω τώρα συστήματα δίσκων με δύο κεφαλές που (α) βρίσκονται σε σταθερή απόσταση d=N/2 μεταξύ τους και (β) δεν μπορούν να μετακινηθούν έξω από τα όρια της επιφάνειας του δίσκου. Ετσι η πρώτη (δεύτερη) κεφαλή μετακινείται εξυπηρετεί κυλίνδρους στο διάστημα [1,d] ([d+1,N]). Σε περίπτωση που μία απαίτηση περιλαμβανει τις θέσεις i και i+d (όπου i=1,2,...,N/2), τότε οι κεφαλές τοποθετούνται μαζί επάνω από τις δύο θέσεις, τις εξυπηρετούν διαδοχικά και δεν υποχρεώνονται σε επαναφορά στις ίδιες θέσεις με αύξηση του κόστους σε επιπλέον διαδρομές.

Το πρόβλημα ανάγεται σε μία ειδική περίπτωση του προβλήματος του περιοδεύοντος πωλητή κι αντιμετωπίζεται με διαφορετικό τρόπο ανάλογα με τις θέσεις όπου αναφέρονται οι απαιτήσεις. Εξετάζονται λοιπόν οι παρακατω τρεις περιπτώσεις. Απαραίτητο είναι να αναφερθεί ότι πρεπει να διατηρείται και μία τιμή (0 ή 1) που να δηλώνει την κεφαλή, πρώτη ή δεύτερη.

- A - ΑΠΑΙΤΗΣΕΙΣ ΑΠΟ ΤΟ ΔΙΑΣΤΗΜΑ [1,d] ή ΤΟ ΔΙΑΣΤΗΜΑ [d+1,N]

Εστω ότι κάθε απαίτηση αναφέρεται σε θέσεις κυλίνδρων που ανηκουν στο ίδιο διαστημα, είτε αυτό είναι το [1,d] είτε το [d+1,N]. Ο αλγόριθμος εχει ως εξής :

Τα L_i και R_i αναδιοργανώνονται έτσι ώστε να δημιουργηθούν μετασχηματισμοί P_1 και P_2 από όπου θα προκυψουν τα τόξα σύνδεσης των πόλεων. Γιά τη δημιουργία των P_1 και P_2 τα L_i και R_i του διαστηματος [d+1,N] ανάγονται στα αντίστοιχα τους στο διάστημα

[1,d]. Μία τιμή 0 ή 1 δηλώνει την κεφαλή εξυπηρέτησης και εξασφαλίζει την ενεργοποίηση της κατάλληλης κεφαλής. Η αντιστοιχιση αυτή γίνεται ως εξής:

Οι τιμες L_1 > d ανάγονται στις τιμές L_1 - d και αντίστοιχα
" " R_1 > d " " " R_1 - d.

Κατόπιν όλα τα L_1 και R_1 (που έχουν προκύψει από την παραπανω αναγωγη) ταξινομούνται οπότε το πρόβλημα λύνεται με τον αλγοριθμο που περιγράφεται στην παραγραφο 2. Δηλαδή θα σχηματισθούν τα τόξα (P_2(i),P_1(i+1)) και κατόπιν οι υποδιαδρομές θα ενωθούν στις δύο φάσεις που περιγράφηκαν. Ετσι το παραδειγμα της παραγραφου 2 αντιμετωπίζεται ως εξής :

	L_1	R_1	ΑΝΑΓΩΓΗ		L_1	R_1	ΤΑΞΙΝΟΜΗΣΗ		P_1	P_2	
			-------->				------ -->				
(q_1)	68	78		(q_1)	18	28		(c_3)	5	25	(c_2)
(q_2)	10	25		(q_2)	10	25		(c_2)	10	28	(c_1)
(q_3)	5	40		(q_3)	5	40		(c_1)	18	40	(c_3)

Τα τόξα είναι: $c_0 \rightarrow c_3$, $c_2 \rightarrow c_2$, $c_1 \rightarrow c_1$ και $c_3 \rightarrow c_0$.

Οι υποδιαδρομές αντίστοιχα είναι: K_0 = [$c_0 \rightarrow c_3 \rightarrow c_0$],
K_1 = [$c_2 \rightarrow c_2$] και
K_2 = [$c_1 \rightarrow c_1$].

ΠΡΩΤΗ ΦΑΣΗ

Αρχικά ενώνονται οι υποδιαδρομές K_1 και K_2 που έχουν τεμνομενες περιοχες και προκύπτει:

$$\begin{matrix} c_2 \rightarrow c_2 \\ c_1 \rightarrow c_1 \end{matrix} \longrightarrow \begin{matrix} c_2 \rightarrow c_1 \\ c_1 \rightarrow c_2 \end{matrix}$$

Αρα παράγεται μία νέα υποδιαδρομή: K' = [$c_2 \rightarrow c_1 \rightarrow c_2$]

ΔΕΥΤΕΡΗ ΦΑΣΗ

Η βασική υποδιαδρομή K_0 ενώνεται με τη νέα υποδιαδρομή K' λόγω μικρότερης απόστασης και προκύπτει:

$$\begin{matrix} c_0 \rightarrow c_3 \\ c_1 \rightarrow c_2 \end{matrix} \longrightarrow \begin{matrix} c_0 \rightarrow c_2 \\ c_1 \rightarrow c_3 \end{matrix}$$

Τελικά προκύπτει: K = [$c_0 \rightarrow c_2 \rightarrow c_1 \rightarrow c_3 \rightarrow c_0$]

Αρα σ'αυτή την περίπτωση η βέλτιστη διαδρομή είναι:

$$c_0 \rightarrow c_2 \rightarrow c_1 \rightarrow c_3$$

Πράγματι για το παραπάνω παράδειγμα η διαδρομή αυτή είναι η βέλτιστη δυνατή με συνολικό κόστος μετάβασης των δύο κεφαλών ίσο με 10 + 7 + 23 = 40 κυλίνδρους που πρέπει να διανυθούν για να βρεθεί κεφαλή στην κατάλληλη θέση εξυπηρέτησης. Ακόμη σημειώνεται ότι σε σύγκριση με το κόστος μετάβασης του παραδείγματος του προηγουμένου κεφαλαίου υπάρχει μείωση από 58 σε 40, μείωση δηλαδή της τάξεως του 23%, ενώ το κόστος απάντησης παραμένει σταθερό και στις δύο περιπτώσεις (60). Αυτό συμβαίνει γιατί γίνεται κατάλληλη αξιοποίηση της ιδιαιτερότητας του υλικού.

Και πάλι εύκολα αποδεικνύεται ότι το κόστος απάντησης δίνεται από τον τύπο δίνεται από τον τύπο (1), ενώ το κόστος μετάβασης ισούται με:

$$f(L_1) + \sum_{i=1}^{T-1} | f(R_i) - f(L_{i+1}) | \tag{3}$$

όπου η συνάρτηση f(.) ορίζεται από τη σχέση:

$$f(a) = \begin{cases} a & \text{αν } a \leq d \\ a-d & \text{αν } a > d \end{cases} \tag{4}$$

Με άλγεβρα αποδεικνύεται ότι παντοτε για τη βέλτιστη διαταξη το αποτελεσμα του τύπου (3) είναι μικροτερο από το αποτέλεσμα του τύπου (2).

- Β - ΑΠΑΙΤΗΣΕΙΣ ΠΟΥ ΞΕΚΙΝΟΥΝ ΣΤΟ ΔΙΑΣΤΗΜΑ [1,d] ΚΑΙ ΛΗΓΟΥΝ ΣΤΟ ΔΙΑΣΤΗΜΑ [d+1,N]

Εστω ότι οι απαιτήσεις αναφέρονται σε θέσεις κυλίνδρων που ξεκινούν στο διάστημα [1,d] (θέσεις L) και λήγουν στο διάστημα [d+1,N] (θέσεις R). Είναι βέβαιο ότι όλες οι θέσεις κυλίνδρων του διαστήματος [L,d] θα εξυπηρετηθούν από την πρώτη κεφαλή, ενώ οι θέσεις κυλίνδρων του διαστήματος [d+1,R] θα εξυπηρετηθούν από τη δεύτερη κεφαλή. Σημαντική, επίσης, είναι η παρατήρηση ότι για να εξυπηρετηθεί τμήμα απαίτησης του διαστήματος [d+1,R] θα πρέπει οι δύο κεφαλές να βρίσκονται στις θέσεις 1 και (d+1) αντίστοιχα. Αρα αν η μεθοδος αντιμετώπισης των απαιτήσεων αυτών γίνεται με την εξυπηρέτηση πρώτα του διαστήματος [L,d] και κατόπιν του διαστήματος [d+1,R], τότε στο κόστος μετάβασης για κάθε απαιτηση θα συμπεριλαμβάνεται η απόσταση d. Για να μειωθεί λοιπόν το κοστος αυτό ακολουθείται ο εξής τρόπος εξυπηρέτησης. Πρώτα εξυπηρετούνται οι θέσεις του διαστήματος [d+1,R] και καθώς γίνεται αυτή η εξυπηρέτηση αν συναντηθεί η θέση L τότε γίνεται ταυτοχρονα και η εξυπηρετηση του διαστήματος [L,d]. Σε διαφορετική περίπτωση η εξυπηρέτηση τελειώνει στη θέση R και οι κεφαλές μετακινούνται στις θέσεις L και L+d αντίστοιχα. Πάντοτε, λοιπόν, κάθε εξυπηρέτηση τελειώνει σε θέσεις κεφαλών d και N αντίστοιχα. Αν συναντηθεί το L τοτε το κοστος μετάβασης δεν θα αυξηθεί, διαφορετικά το κόστος θα ισούται με (L-(R-d)). Επομένως με τη λογική αυτή παρουσιάζεται μείωση του κόστους μετάβασης γιατί η εξυπηρετηση είναι άμεση και εξοικονομείται μία διαδρομή d.

Αρα, τελικά, το κόστος απάντησης ισούται με:

$$T * d \tag{5}$$

ενώ το κοστος μετάβασης ισούται με

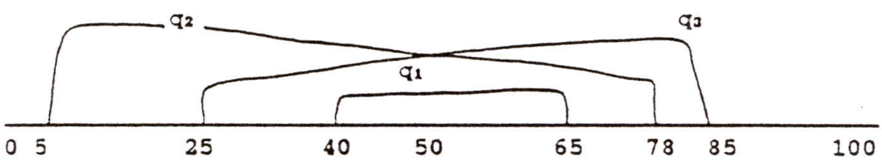
$$(T-1) * d + \sum_{i=1}^{T-1} g(L_i - R_i + d) \qquad (6)$$

όπου η συνάρτηση $g(.)$ είναι μία συνάρτηση ράμπας και ορίζεται απο τη σχεση:

$$g(a) = \begin{cases} 0 & \text{αν } a \le 0 \\ a & \text{αν } a > 0 \end{cases} \qquad (7)$$

Επίσης είναι σημαντικό το ότι σύμφωνα με τον τύπο (6) δεν έχει σημασια η σειρά εξυπηρέτησης των απαιτήσεων. Τα παραπανω περιγραφονται με τη βοήθεια ενός παραδείγματος:

ΑΠΑΙΤΗΣΗ	L_i	R_i	ΚΟΣΤΟΣ ΑΠΑΝΤΗΣΗΣ
q_1	40	65	50
q_2	5	78	50
q_3	25	85	50



Σχ.2: Παράδειγμα αλγόριθμου με 2 κεφαλές ανά επιφάνεια και για απαιτήσεις με $L_i < d$ και $R_i > d$.

Αν στο παραπάνω παράδειγμα ακολουθηθεί η αρχική λογική της εξυπηρέτησης πρώτα του διαστήματος [L,d] και μετά του διαστήματος [d+1,R], τότε το κόστος μετάβασης για τη βέλτιστη διαδρομή του παραδείγματος $c_2 \rightarrow c_3 \rightarrow c_1$ είναι ίσο με 5 + 50 + 3 + 50 + 5 + 50 = 163. Με τη μέθοδο που περιγράφηκε προηγούμενα το κόστος μετάβασης είναι ίσο με 25 + 50 + 50 = 125, δηλαδή μία διαφορά 38 κυλίνδρων (-23%).

Σε σχέση με τα συστήματα δίσκων με μία κεφαλή ανά επιφάνεια μπορούν να γίνουν οι εξής παρατηρήσεις. Αν οι ίδιες απαιτήσεις του προηγουμένου σχήματος εξυπηρετούνταν από συμβατικό σύστημα τότε το κόστος απάντησης θα ήταν 158, ενώ το κόστος μετάβασης θα ήταν 83. Σημειώνεται, δηλαδή, ότι εδώ το κόστος απάντησης (μετάβασης) είναι μικρότερο (μεγαλύτερο). Εικάζεται ότι (α) όσον αφορά στο κόστος απάντησης επειδή εξαρτάται από τη φύση των απαιτήσεων δεν μπορεί να εξαχθεί κάποιο γενικότερο συμπέρασμα και (β) όσον αφορά στο κόστος μετάβασης αυτό είναι πάντα μικρότερο στα συμβατικά συστήματα. Η εικασία αυτή θα σχολιασθεί στο επόμενο κεφάλαιο.

- Γ - ΑΠΑΙΤΗΣΕΙΣ ΠΟΥ ΑΝΗΚΟΥΝ ΣΤΙΣ ΚΑΤΗΓΟΡΙΕΣ Α ΚΑΙ Β

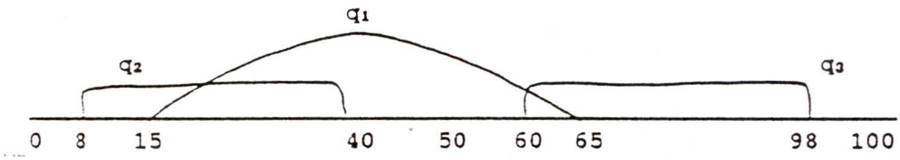
Στην περίπτωση αυτή εξετάζονται απαιτήσεις που αναφερονται σε θεσεις κυλίνδρων που μπορεί να ξεκινούν και να καταλήγουν σε οποιοδήποτε από τα διαστήματα [L,d] και [d+1,R].

Ας θεωρηθεί για τη συνέχεια ότι οι απαιτήσεις διαχωρίζονται σε δύο κατηγορίες σύμφωνα με τις παραπάνω περιπτώσεις Α και Β. Σε κάθε μία από τις κατηγορίες Α και Β ακολουθείται η λογική των αλγορίθμων εξυπηρέτησης που περιγράφηκαν προηγούμενα. Ετσι στην κατηγορία Α προκύπτει μία βέλτιστη διαδρομή επίσκεψης. Στην κατηγορία Β ακολουθείται η λογική εξυπηρέτησης πρώτα του διαστήματος [L,d] και μετά του διαστήματος [d+1,R] χωρίς ιδιαίτερη διάταξη των απαιτήσεων. Ετσι δημιουργούνται δύο ακολουθίες από τα Α και Β που ενώνονται με τη μέθοδο του δένδρου ελάχιστου κόστους (minimum cost spanning tree) [KoMaPa90]. Δηλαδή, ισχύει:

$$\begin{matrix} c(i) \longrightarrow c(j) \\ c(k) \longrightarrow c(l) \end{matrix} \quad \longrightarrow \quad \begin{matrix} c(i) \longrightarrow c(l) \\ c(k) \longrightarrow c(j) \end{matrix}$$

έτσι ώστε τα R και L να ικανοποιούν την ελάχιστη απόσταση. Στα παρακάτω σχήματα απεικονίζονται δύο περιπτώσεις όπου εξετάζεται η μεθοδος αυτή.

ΑΠΑΙΤΗΣΗ	L_i	R_i	ΚΟΣΤΟΣ ΑΠΑΝΤΗΣΗΣ
q_1	15	65	50
q_2	8	40	32
q_3	60	98	50



Σχ.3: Πρώτο παράδειγμα αλγόριθμου με 2 κεφαλές ανά επιφάνεια και για απαιτήσεις με L_i, R_i σε οποιοδήποτε διάστημα.

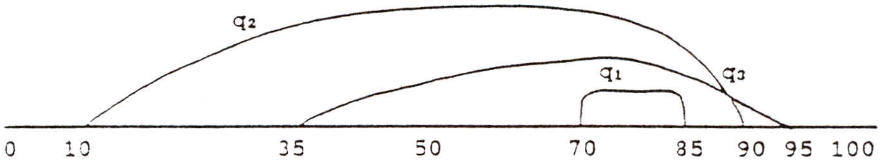
Στο πρώτο παράδειγμα αν ακολουθηθεί η μέθοδος αυτή προκύπτει η διαδρομή :

$c_2 \rightarrow c_1 \rightarrow c_3$ με κόστος 8 + 25 + 50 + 5 = 88.

ενώ στο δεύτερο παράδειγμα από τη μέθοδο αυτή προκύπτει:

$c_2 \rightarrow c_1 \rightarrow c_3$ με κόστος 10 + 50 + 20 + 50 = 130

ΑΠΑΙΤΗΣΗ	L_i	R_i	ΚΟΣΤΟΣ ΑΠΑΝΤΗΣΗΣ
q_1	70	85	15
q_2	10	90	50
q_3	35	95	50



Σχ.4: Δεύτερο παράδειγμα αλγόριθμου με 2 κεφαλές ανά επιφάνεια και για απαιτήσεις με L_i, R_i σε οποιοδήποτε διάστημα.

Ετσι ο αλγόριθμος καταλήγει σε μία πιο ραφιναρισμένη μορφή. Και πάλι οι απαιτήσεις διαχωρίζονται σε δύο κατηγορίες A και B. Στην κατηγορία A προκύπτει μία βέλτιστη διαδρομή επίσκεψης. Στην κατηγορία B όπως αναφέραμε δεν έχει σημασία η σειρά εξυπηρέτησης. Ετσι στον αλγόριθμο αυτό η αρχική εξυπηρέτηση αφορά πόλη της κατηγορίας B, στη συνέχεια εξυπηρετούνται όλες οι πόλεις της κατηγορίας A (με σειρά που έχει προκύψει από τη βέλτιστη τους διαδρομή) και στο τέλος εξυπηρετούνται οι υπόλοιπες πόλεις της κατηγοριας B (με σειρα κόστους απάντησης).

Στα ίδια παραδείγματα των δύο σχημάτων εξετάζεται η μέθοδος αυτή και προκύπτει η διαδρομή:

c_1 --> c_2 --> c_3 με κόστος 42 + 30 = 72.

από το πρώτο και από το δεύτερο αντίστοιχα:

c_2 --> c_1 --> c_3 με κόστος 30 + 35 = 65.

Ας σημειωθεί ότι το κόστος μετάβασης σε συμβατικό σύστημα για τις ίδιες απαιτήσεις θα ήταν 38 και 80 κύλινδροι αντίστοιχα, ενώ το κόστος απάντησης θα ήταν 120 και 155 κύλινδροι αντίστοιχα. Συνεπώς δεν μπορούν να εξαχθούν γενικότερα συμπεράσματα από τη σύγκριση των δύο συστημάτων. Εικάζεται, ωστόσο, ότι αυτή η διάταξη είναι η βέλτιστη στο δεδομένο περιβάλλον (συστήματα δύο κεφαλών, αλγόριθμος CSCAN).

4. ΣΥΜΠΕΡΑΣΜΑΤΑ - ΠΡΟΤΑΣΕΙΣ

Στην εργασία αυτή εξετάσθηκαν συστήματα δίσκων με δύο κεφαλές εγγραφής/ανάγνωσης ανά επιφάνεια που βρίσκονται σε σταθερή απόσταση μεταξύ τους και δόθηκαν νέοι αλγόριθμοι διάταξης απαιτήσεων. Οι αλγόριθμοι αυτοί βασίζονται σε προηγούμενη μέθοδο για συμβατικά συστήματα με μία κεφαλή ανά επιφάνεια που βασίζεται στο πρόβλημα του "περιοδεύοντος πωλητή".

Το πρόβλημα αντιμετωπίσθηκε για τρεις περιπτώσεις. Στην πρώτη περίπτωση θεωρήθηκε ότι τα όρια κάθε απαίτησης βρίσκονται στο ίδιο ημιδιάστημα και εξυπηρετούνται από την ίδια κεφαλή. Το συμπέρασμα είναι ότι με αλγόριθμο O(nlogn) βρίσκεται η βέλτιστη διάταξη και σε σχέση με συμβατικά συστηματα το κόστος μετάβασης (απάντησης) μειώνεται (μένει σταθερό). Στη δεύτερη περίπτωση θεωρήθηκε ότι τα όρια κάθε απαίτησης δεν βρίσκονται στο ίδιο ημιδιάστημα και επομένως δεν εξυπηρετούνται από την ίδια κεφαλή. Εδώ, το συμπέρασμα είναι ότι δεν απαιτείται ιδιαίτερη διάταξη των απαιτήσεων γιατί το κόστος παραμένει σταθερό. Επίσης εξάγεται το συμπερασμα ότι το κόστος μετάβασης ειναι μεγαλύτερο από το αντίστοιχο κόστος των συμβατικών συστημάτων λόγω της σταθερής απόστασης που έχουν οι κεφαλες μεταξύ τους και λόγω του ότι η μετακίνηση των κεφαλών γίνεται προς μία κατεύθυνση (αριστερά προς τα δεξιά) σύμφωνα με την πολιτική δρομολόγησης κεφαλών CSCAN. Τέλος, τρίτη είναι η περίπτωση που οι απαιτήσεις ανήκουν και στις δύο προαναφερθείσες περιπτώσεις. Και για την περίπτωση αυτή δόθηκε αλγόριθμος ελαχιστοποίησης του κοστους μετάβασης.

Η μελλοντική μελέτη μπορεί να στραφεί στις εξής κατευθύνσεις:

- Α - Αυστηρή μαθηματική απόδειξη ότι οι αλγόριθμοι που παρουσιάσθηκαν δίνουν πράγματι τη βέλτιστη διάταξη των απαιτησεων.

- Β - Αναπτυξη αλγορίθμων θεωρώντας ότι οι απαιτήσεις ικανοποιούνται ενώ οι κεφαλές κινούνται και κατά τις δύο κατευθύνσεις (αλγόριθμος SCAN). Εικάζεται ότι στην περίπτωση αυτή θα υπάρχει βέβαιο κέρδος στο κόστος μετάβασης σε σχέση με τα συμβατικά συστήματα.

- Γ - Ανάπτυξη αλγορίθμων για συστήματα με δύο ανεξάρτητες κεφαλές που μπορούν να κινηθούν αυτόνομα.

ΑΝΑΦΟΡΕΣ

[BuKo79] Burton F.W. and Kollias J.G.: Optimising disk head movement in secondary key retrievals, The Computer Journal, Vol.22, No.3, pp.206-208, 1979.

[BiWo79] Bitner F.W. and Wong C.K.: Optimal and near optimal algorithms for batched processing in linear storage, SIAM Journal on Computing, Vol.8, No.4, pp.479-498, 1979.

[CaCoFl84] Calderbank A.R., Coffman E.G.Jr. and Flatto L.: Optimum head separation in a disk with two read/write heads, Journal of the ACM, Vol.31, No.4, pp.433-446, 1984.

[CaCoFl88] Calderbank A.R., Coffman E.G.Jr. and Flatto L.: Optimal directory placement on disk storage devices, Journal of the ACM, Vol.35, No.2, pp.433-446, 1988.

[CaCoFl89] Calderbank A.R., Coffman E.G.Jr. and Flatto L.: A Note Extending the Analysis of Two-Headed Disk Systems to More General Seek-Time Characteristics, IEEE Transactions on Computers, Vol.38, No.11, pp.1594-1586, 1989.

[Ho83] Hofri M.: Should the two-headed disk be gready? Yes, it should, Information Processing Letters, Vol.16, pp.83-85, 1983.

[Ko78] Kollias J.G.: An estimate of seek time for batched searching of random or indexed sequential files, The Computer Journal, Vol.21, No.2, pp.132-133, 1978.

[KoMaPa90] Kollias J.G., Manolopoulos Y. and Papadimitriou C.H.: The optimum execution order of queries in linear storage, Information Processing Letters, Vol.36, pp.141-145, 1990.

[LaLeRiSh1986] Lawler E.L., Lenstra J.K., Kan A.R. and Shmoys D. (editors): The traveling salesman problem, Prentice Hall, 1986.

[MaKo88] Manolopoulos Y. and Kollias J.G.: Estimating disk head movement in batched searching, BIT, Vol.28, pp.27-36, 1988.

[MaKo89] Manolopoulos Y. and Kollias J.G.: Perfomance of a two-headed disk system when serving database queries under the SCAN policy, ACM Transactions on Database Systems, Vol.14, No.3, pp. 425-442, 1989.

[MaKo90] Manolopoulos Y. and Kollias J.G.: Optimal data placement in two-headed disk systems, BIT, Vol.30, pp.216-219, 1990.

[Ma90] Manolopoulos Y.: Probability distributions for seek time evaluation, Information Sciences, to appear, 1991.

[MaVa90] Manolopoulos Y. and Vakali A.: Seek distances in disks with two independent heads per surface, Information Processing Letters, to appear, 1991.

[Pa88] Palvia P.: Batched searching in database organizations, Information Sciences, Vol.45, pp.23-37, 1988.

[PaWo81] Page I.P. and Wood R.T.: Empirical analysis of a moving disk head model with two heads separated by a fixed number of tracks, The Computer Journal, Vol.24, No.4, pp.339-341, 1981.

[ΠαΜπιΤσα86] Παπακωνσταντίνου Γ., Μπιλάλης Ν. και Τσανάκας Π.: Λειτουργικά συστήματα, Μέρος πρώτο, 1986.

[PaSt82] Papadimitriou C.H. and Steiglitz K.: Combinatorial optimization: Algorithms and complexity, Prentice Hall, 1982.

[Se88] Sellis T.: Multiple query optimization, ACM Transactions on Database Systems, Vol.13, No.1, pp.23-52, 1988.

[Wo83] Wong C.K.: Algorithmic studies in mass storage systems, Computer Science Press, 1983.