

Liczba bloków na ścieżce indeksowej oraz liczba ścieżek indeksowych:

$$n_{bsi} = 1 + \left\lfloor \frac{3625 - (20 + 12 + 10)}{81 + \lceil 1,049 \cdot (12 + 10) \rceil} \right\rfloor = \left\lfloor \frac{3625 - 42}{105} \right\rfloor = 35,$$

$$n_{si} = \left\lceil \frac{16}{35} \right\rceil = 1.$$

Długość nie zajętej części ścieżki indeksowej oraz liczba bloków użytkownika w tej części:

$$l_{sr} = 3625 - 16 \cdot 105 = 1945 ,$$

$$n_{br} = 1 + \left\lfloor \frac{1945 - 632}{723} \right\rfloor = 2.$$

Liczba bloków użytkownika w cylindrze oraz liczba cylindrów obszaru podstawowego:

$$n_{bc} = 5 \cdot [10 - (1 + 2)] + 2 = 37,$$

$$n_{cp} = \left\lceil \frac{5000}{37} \right\rceil = 136.$$

Liczba cylindrów globalnego obszaru nadmiarowego oraz całkowita liczba cylindrów:

$$n_{cng} = \lceil (455 - 136 \cdot 2) / 10 \rceil = 19 ; n_c = 136 + 19 = 155.$$

Liczba ścieżek indeksu cylindrów:

$$n_{sic} = \left\lceil \frac{136}{35} \right\rceil = 4 .$$

10.3.2. Czasochłonność przetwarzania zbiorów indeksowo-sekwencyjnych

Dostęp do dowolnego zapisu w zbiorze indeksowo-sekwencyjnym, w celu np. jego aktualizacji, wymaga następujących operacji oraz następującego czasu trwania (przy założeniu, że zapis ten znajduje się w obszarze podstawowym):

1. Przeszukanie indeksu cylindrów w celu ustalenia, w którym cylindrze znajduje się szukany zapis; czas pomijalnie mały, ponieważ indeks ten znajduje się na ogół w pamięci operacyjnej.

2. Ustawienie głowic zapisu/odczytu na odpowiednim cylindrze (o ile głowice już się tam nie znajdują); czas przemieszczenia $t_p(k)$, $k \in \langle 0, n_c - 1 \rangle$.

Do obliczeń przyjmowany jest czas średni:

$$t_{p\bar{s}r} = t_p((n_c - 1)/2) .$$

3. Włączenie głowicy dla pierwszej ścieżki, na której znajduje się indeks ścieżek; czas pomijalnie mały.

4. Oczekiwanie, aż indeks ścieżek znajdzie się pod głowicą; średni czas oczekiwania $t_{oc\bar{s}r} = T/2$.

5. Wczytanie indeksu ścieżek do pamięci operacyjnej; czas transmisji t_{tri} .

6. Przeszukanie indeksu ścieżek w celu ustalenia, na której ścieżce znajduje się szukany zapis; czas pomijalnie mały.

7. Włączenie odpowiedniej głowicy i oczekiwanie na szukany blok; $t_{oc\bar{s}r} = T/2$.

8. Wczytanie bloku z szukany zapisem; czas transmisji t_{tr} .

9. Aktualizacja zapisu w pamięci operacyjnej; czas pomijalnie mały.

10. Jeśli głowice były przesuwane, ponowne ich ustawienie na tym samym cylindrze; czas jak w p.2.

11. Oczekiwanie na właściwą pozycję głowicy; $t_{oc\bar{s}r} = T/2$.

12. Zapisanie bloku na poprzednim miejscu; czas transmisji t_{tr} .

13. Powtórne odczytanie zapisanego bloku w celach kontrolnych; czas pełnego obrotu, czyli T .

Czas aktualizacji pojedynczego zapisu jest sumą następujących składników:

$$t_{zap}^{akt} = t_p(k) + 5T/2 + 2 \cdot t_{tr} + t_{tri} .$$

Jeśli zbiór indeksowo-sekwencyjny będzie przetwarzany w sposób bezpośredni (tzn. dostępy do zapisów nie będą wykonywa-

ne wg kolejności kluczy), wtedy istotną rolę będą odgrywać wszystkie powyższe składniki czasu. W szczególności, jeśli większość dostępów nie będzie dotyczyć pewnego ograniczonego fragmentu obszaru dysku zajmowanego przez zbiór, głowice będą często przemieszczać się o wiele cylindrów. Na przykład, jeśli 80% zamówień klientów dotyczy 1000 standardowych wyrobów, o których informacje przechowujemy w dwóch kolejnych cylindrach, natomiast pozostałe 20% zamówień dotyczy bardzo dużej liczby nietypowych wyrobów, ilość przemieszczeń głowic będzie niewielka.

Dlatego też warto czasami posortować transakcje aktualizujące wg wartości klucza, tzn. uporządkować je tak samo jak zbiór główny, aby głowice poruszały się tylko w jednym kierunku, "przeskakując" na kolejny cylinder, po przetworzeniu cylindra poprzedniego. Przy takim rozwiązaniu indeks ścieżek danego cylindra jest wczytywany tylko raz oraz tylko raz przesyłany jest blok danych.

Przykład 10.7

Rozważmy zbiór indeksowo-sekwencyjny składający się z 100 000 zapisów 50-znakowych, przechowywany na jednym pakiecie wg standardów ODRA/ICL. Jeden blok (porcja) zawiera 10 zapisów, zaś na ścieżce znajduje się 8 bloków. Pomijając przypadki nadmiarów oraz zakładając, że indeksy ścieżek zajmują mało miejsca, można obliczyć ilość cylindrów potrzebną dla tego zbioru:

$$n_c = \lceil 100\,000 / (10 \cdot 8 \cdot 10) \rceil = 125.$$

Zakładamy ponadto, że w przebiegu aktualizujemy 5% zapisów, równomiernie rozłożonych w rozważanym zbiorze. Czas aktualizacji wybranych zapisów dla dostępu bezpośredniego i sekwencyjnego będzie wynosił *):

*) W obliczeniach, podobnie jak poprzednio, uwzględniony jest tylko czas pamięci zewnętrznej.

Aktualizacja bezpośrednia zbioru

Czas przetwarzania każdego aktualizowanego zbioru będzie sumą następujących składników:

- średni czas przemieszczania głowic:

$$t_{p\text{śr}} = t_p((125-1)/2) = 70 \text{ ms},$$

- sumaryczny czas oczekiwania:

$$t_{oc\text{śr}} = 5 \cdot 20/2 = 50 \text{ ms},$$

- sumaryczny czas transmisji bloku:

$$2 \cdot T/8 = 5 \text{ ms}.$$

Liczba aktualizowanych zapisów = 5000.

Całkowity czas aktualizacji

$$t_{is}^{\text{akt}} = 5000 \cdot (70+50+5) \text{ ms} = 10,5$$

Aktualizacja sekwencyjna zbioru

Czas przemieszczenia głowic $t_p = t_p(1) \cdot (125-1) = 3,1 \text{ s}$.

Czas oczekiwania przy wczytywaniu indeksu ścieżek dla każdego cylindra: $125 \cdot 10 = 1,25 \text{ s}$.

Czas przetwarzania wszystkich aktualizowanych zapisów (transmisja i oczekiwanie) : $5000 \cdot (2 \cdot 1/8 + 2) \cdot 20 \text{ ms} = 225 \text{ s}$. Całkowity czas aktualizacji: $225 + 3,1 + 1,25 = 230 \text{ s} = 3,8 \text{ min}$. A więc warto (o ile jest to możliwe) posortować zbiór transakcji wg wartości kluczy.

W powyższym przykładzie przetwarzamy około 5000 bloków spośród 10 000 wszystkich bloków. Jeśli aktywność zbioru przekracza 10%, wtedy można spodziewać się, że będą przesłane wszystkie bloki, czyli można zorganizować zbiór sekwencyjnie (jeśli tylko można przetwarzać transakcje partiami).

W przedstawionym przykładzie zakładaliśmy, że tylko jedna transakcja aktualizuje każdy zapis, czyli mamy 5000 transak-

cji (np. aktualizacja zbioru rachunków klientów przez miesięczne płatności). Jednakże często zdarza się, że każdy zapis jest aktualizowany przez szereg transakcji, np. aktualizacją zbioru o stanie magazynu przez zamówienia klientów, przy czym poszczególne wyroby są zamawiane przez wielu klientów. Rozważmy poprzedni przykład przy założeniu, że "trafiony" zapis jest aktualizowany (średnio) przez 20 transakcji, tj. cały zbiór - przez 100 000 transakcji, czyli

- aktualizacja bezpośrednia spowoduje 100 000ostępów (zamiast 5000), a więc całkowity czas przetwarzania wyniesie: $100\ 000 (70 + 50 + 5) = 208\text{ min}$,

- aktualizacja sekwencyjna spowoduje tylko jeden dostęp do wyszukanego zapisu, czyli całkowity czas przetwarzania nie zmieni się i będzie wynosił 3,8 min.

10.4. Zbiory o organizacji losowej

W celu zlokalizowania zapisu w zbiorach o organizacji losowej, stosowane jest odwzorowanie wartości klucza tego zapisu w adres zapisu w pamięci. Odwzorowanie to nazywane jest algorytmem (lub funkcją) mieszającym (ang. hash function, randomizing f.). Adres otrzymany w wyniku działania algorytmu może być bezwzględnym adresem fizycznym (tzn. numer cylindra i głowicy lub nawet bloku na ścieżce) lub adresem względnym (na przykład kolejny numer bloku (porcji) w zbiorze), zamienianym na adres fizyczny przez standardowe programy zarządzania danymi.

Szczególnym przypadkiem funkcji mieszającej, zapewniającej jednoznaczne odwzorowanie zbioru wartości kluczy w równoliczny zbiór adresów jest funkcja liniowa. Metodę tę nazywa się adresowaniem bezpośrednim (ang. direct addressing) (lub samoindeksowaniem (ang. selfindexing)).