Zewnętrzne Sortowanie Pliku Metodą Scalania z Użyciem Wielkich Buforów

Autor: Jakub Bronk 197965

---

1. Wprowadzenie

To sprawozdanie prezentuje implementację i analizę algorytmu **sortowania z wielkimi buforami*. Zewnętrzne sortowanie jest niezbędne gdy dane, które chcemy posortować, nie mogą zmieścić się w pamięci operacyjnej.

---

2. Opis metody

2.1 Opis Algorytmu

Algorytm ten działa w dwóch fazach:

1. Wytworzenie początkowych taśm

Plik wejściowy jest wczytywany w blokach, które mogą zmieścić się w pamięci. Każdy blok jest wczytany, posortowany w pamięci, i zapisany z powrotem na dysk.

2. Scalanie taśm

Posortowane taśmy są scalane przy pomocy min-heap, aż pozostanie jeden posortowany plik. W każdej fazie sortowania $b-1$ taśm jest scalane w jedną taśmę (b to liczba buforów na które podzielona jest pamięć operacyjna).

2.1 Konfiguracja Programu

Program rezerwuje duży blok pamięci operacyjnej, który jest dzielony na kilka mniejszych według potrzeb.
Blok pamięci jest rozmiaru $n * b$, b oznacza rozmiar buforów, n oznacza liczbę buforów. Określone liczby n i b można ustawić dowolnie przed uruchomieniem programu. Domyślne wartości to $n = 11; n = 10$

Do pierwszego etapu cały blok jest używany jako jeden duży bufor, co pozwala nam posortować więcej rekordów w pierszej fazie.

W późniejszych etapach blok pamięci jest podzielny na n części, jedna z nich jest buforem wyjściowym, i do niego scalene są rekory, a pozostałe to bufory wejściowe.

---

3. Format Plików Testowych

3.1 Struktura rekordów

Każdy rekord składa się z 6 zmiennych typu int reprezentujących wielomian:

$$a[0], a[1], a[2], a[3], a[4], x$$

Klucz do sortowania jest wyliczony następująco:

$$y = a[0] + a[1]x + a[2]x^2 + a[3]x^3 + a[4]x^4$$

3.2 Format Pliku

Rekordy zapisywane są w formie normalnego tekstu, jeden rekord na linię

---

4. Użycie programu

```
./build/main [-g count] [-e] [-i input] [-o output] [-d dir] [-n buffers] [-
b block_size]
```

Opcje:

- `-g count` : Generuje `count` losowych rekordów do pliku wyjściowego
- `-e` : Wyświetla cały plik wejściowy, z wyliczonymi kluczami
- `-i input` : Plik wejściowy
- `-o output` : Plik wyjściowy
- `-d dir` : Lokalizacja na dane tymczasowe
- `-n buffers` : Ilość buforów
- `-b block_size` : Ilość rekordów w buforach

---

5. Analiza Teoretyczna

Po pierwszym etapie mamy $\lceil N/(nb) \rceil$ taśm na dysku

Koszt pierwszego etapu to $2N$ operacji odczytu, lub zapisu na dysku (każdy rekord jest raz wczytywany i raz zapisywany)

Każdy cykl w drugim etami zmniejsza ilość taśm w przybliżeniu n-krotnie
Więc ilość cyklów sortowania to $\lceil \log_n(N/(nb)) \rceil$

Koszt każdego cyklu to $2N$ operacji dyskowych

---

6. Eksperyment

6.1. Konfiguracja eksperymentu

Do przeprowadzenia poniższego eksperymentu została zastosowana poniższa konfiguracja programu:

- Ilość buforów: 11
- Ilość rekordów w buforze: 10
- Całkowity rozmiar bloku pamięci: 110

Tak niskie liczby zostały wybrane, aby lepiej zaprezentować ilości faz sortowania, zależnie od danych wejściowych, bez konieczności zapełniania całego dysku.

6.2. Testowane przypadki

Osiem przypadków testowych z różnymi rozmiarami:

Test #	N (ilość rekordów)	Przewidywana ilość faz	Przewidywana ilość operacji dyskowych
1	100	1	200
2	500	2	2 000
3	1,000	2	4 000
4	5,000	3	30 000
5	10,000	3	60 000
6	50,000	4	200 000
7	100,000	4	800 000
8	500,000	5	5 000 000

6.3. Metodologia

1. Dla każdego przypadku testowego:

- Wygenerowano N losowych rekordów
- Uruchomiono program sortujący z odpowiednią konfiguracją
- Zapisano potrzebne dane
- Zweryfikowano poprawność z użyciem opcji `-e`

2. Policzenie przewidywanych wartości używając wzorów z sekcji 5
3. Porównanie otrzymanych wartości z oczekiwanymi

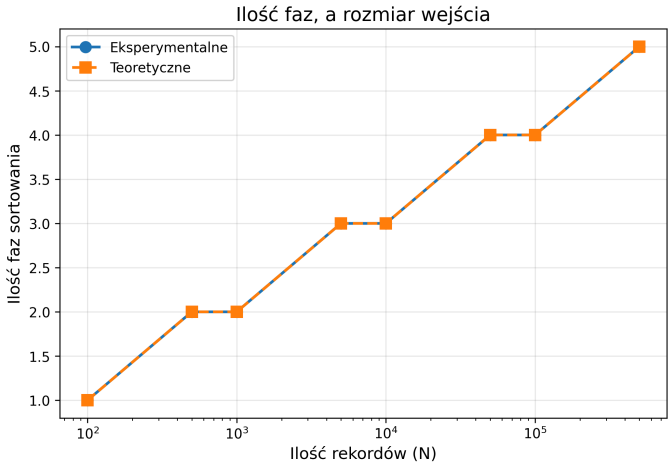
---

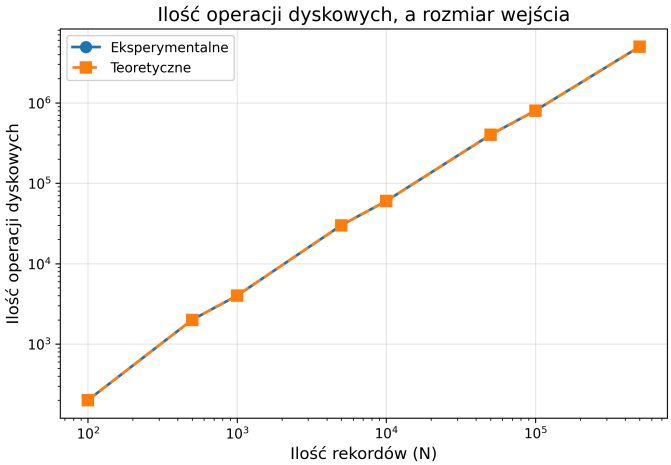
7. Wyniki eksperymentu

7.1. Wyniki numeryczne

Test #	N (ilość rekordów)	Przewidywana ilość faz	Przewidywana ilość operacji dyskowych	Otrzymana ilość faz	Otrzymana ilość operacji
1	100	1	200	1	200
2	500	2	2 000	2	2 000
3	1,000	2	4 000	3	4 000
4	5,000	3	30 000	3	30 000
5	10,000	3	60 000	3	60 000
6	50,000	4	200 000	4	200 000
7	100,000	4	800 000	4	800 000
8	500,000	5	5 000 000	5	5 000 000

7.2 Wykresy





7.3 Obserwacje

- Ilość faz rośnie logarytmicznie z rozmiarem wejścia zgodnie z przewidywaniami
- Ilość operacji dyskowych rośnie z wzorem $nlog(n)$, zgodnie z przewidywaniami

7.4 Wnioski

Metoda sortowania z wielkimi buforami daje rezultaty identyczne do teoretycznych