Struktury Baz Danych

Projekt 1: Implementacja sortowania

Rafał Kajomof 193322

26 Listopada 2024

1. Cel projektu

Celem projektu jest implementacja programu sortującego dane zapisane w formie pliku na pamięci masowej przy użyciu wybranej metody sortowania oraz przeprowadzenie eksperymentu porównującego zmierzone wielkości z teoretycznymi dla różnych wielkości danych wejściowych. Program symuluje odczyty i zapisy blokowe między pamięcią operacyjną i masową, aby pokazać jak zachowa się w przypadku pracy z dużymi zbiorami danych w rzeczywistych warunkach. Wyniki eksperymentu zostaną przedstawione na wykresach i poddane analizie.

2. Opis zastosowanej metody

W tym projekcie zaimplementowano program sortujący dane metodą scalania naturalnego z użyciem wielkich buforów. Algorytm sortujący został zaprojektowany, żeby sortować ilości danych większe niż dostępna pamięć operacyjna. Sortowanie metodą scalania naturalnego polega na dzieleniu pliku wejściowego na serie do posortowania w pamięci operacyjnej, które następnie iteracyjnie zostają łączone aż do uzyskania jednej posortowanej serii, która zostaje zapisana do pliku wyjściowego. Każda iteracja, podczas której serie są scalane, nazywa się fazą, i ich ilość wzrasta w tempie logarytmicznym wraz z wzrostem ilości danych do posortowania.

3. Implementacja

Projekt został zaimplementowany w całości w języku C++. Program wykorzystuje 4 taśmy w schemacie 2+2. Rekordy do posortowania są zbiorami liczb całkowitych o rozmiarze nie większym niż 15, które mają być uporządkowane według największej liczby ze zbioru. Program uporządkowuje te rekordy rosnąco.

Możliwa jest generacja losowych danych wejściowych o wprowadzonej wielkości, a także wskazanie istniejącego pliku wejściowego do posortowania.

4. Specyfikacja formatu pliku wejściowego

Plik wejściowy składa się z serii rekordów zapisanych w postaci binarnej. Każdy rekord posiada nagłówek zawierający wielkość zbioru liczb zawartych w rekordzie oraz kolejno liczby do niego należące. Nagłówek ma rozmiar 8 bajtów a każda liczba całkowita 4 bajty.

5. Opis sposobu prezentacji wyników działania programu

Program wypisuje na standardowy strumień wyjściowy zawartość pliku wejściowego przed sortowaniem oraz po sortowaniu z odpowiednimi oznaczeniami. Po wybraniu odpowiedniej opcji wypisuje także zawartość po każdej fazie sortowania z oznaczeniem fazy. Dodatkowo program wyświetla liczbę faz sortowania, odczytów, zapisów, łącznych operacji na dysku i oczekiwaną liczbę operacji na dysku.

6. Opis eksperymentu

Eksperyment miał na celu zbadanie wydajności algorytmu w zależności od liczby sortowanych rekordów. W eksperymencie przeprowadzono następujące kroki:

1. Generowanie danych
2. Sortowanie danych
3. Porównanie wyników

Dla różnych liczb rekordów wygenerowano losowe rekordy i zapisano je do pliku wejściowego, a następnie posortowano, mierząc liczbę faz, odczytów i zapisów stron dyskowych. Tak zebrane dane porównano z teoretycznymi wartościami, obliczonymi na podstawie zastosowanych w implementacji algorytmu wielkości.

Teoretyczna liczba operacji na dysku wynosi w przybliżeniu

$$2 * \frac{N}{b * \log(n)} * \log\left(\frac{N}{b}\right)$$

gdzie:

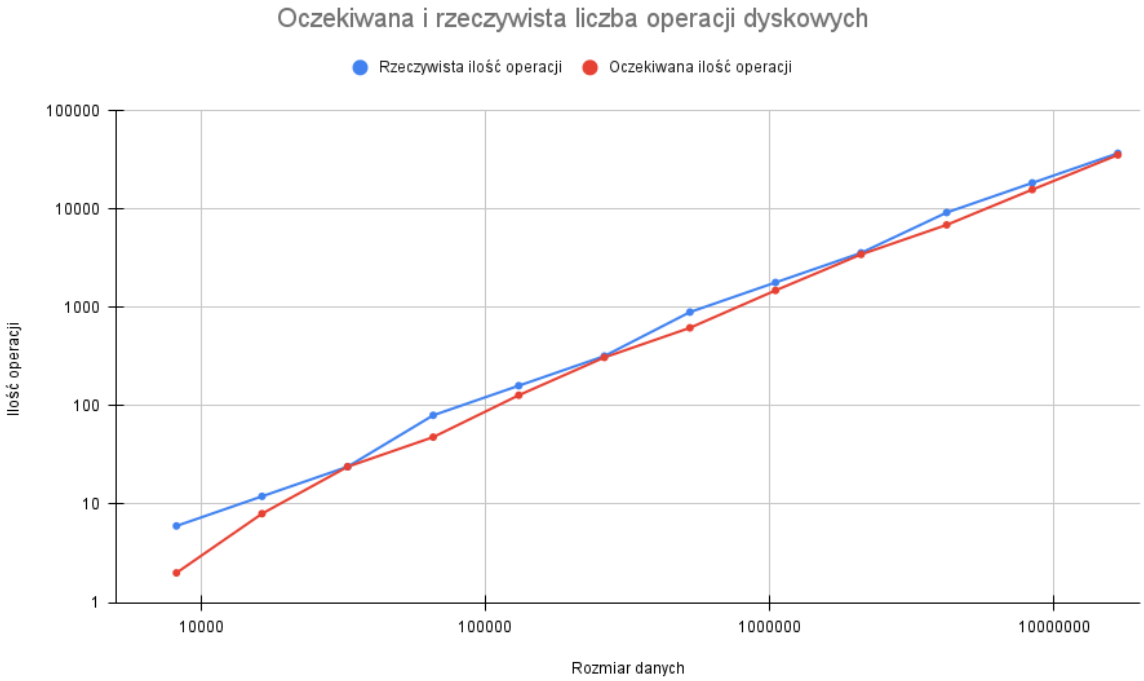
- N oznacza liczbę rekordów,
- b oznacza rozmiar bufora w rekordach, które może pomieścić,
- n oznacza liczbę buforów.

Liczby teoretycznych operacji na dysku zostały zaokrąglone w górę do najbliższej liczby całkowitej.

Wyniki eksperymentu zostały przedstawione w formie tabeli oraz na wykresie.

Rozmiar danych	Teoretyczna liczba operacji	Rzeczywista liczba operacji
8192	2	6
16384	8	12
32768	24	24
65536	48	80
131072	128	160
262144	310	320
524288	620	896
1048576	1488	1792
2097152	3458	3584
4194304	6902	9216
8388608	15760	18432
16777216	35460	36864

Tabela 1: Teoretyczne i rzeczywiste liczby operacji na dysku



Rysunek 1: Oczekiwana i rzeczywista liczba operacji dyskowych

Zmierzone wartości wielkości odbiegają od teoretycznych z powodu dodatkowych odczytów i zapisów zastosowanych w implementacji algorytmu i są zależne od wielkości buforów, ich ilości oraz rozmiaru stron dyskowych. Teoretyczna liczba operacji na dysku nie uwzględnia też mniej efektywnego wykorzystania buforów przez implementację algorytmu niż zakłada wzór teoretyczny. Rzeczywista liczba operacji na dysku jest także zależna od zmiennego rozmiaru rekordów. Wartości teoretyczne i praktyczne są jednak do siebie zbliżone i zachowują takie same tempo wzrostu zależnie od ilości danych.

7. Podsumowanie

Cele projektu zostały zrealizowane. Symulacja operacji blokowych została przeprowadzona poprawnie, a implementacja algorytmu sortowania zewnętrznego skutecznie sortuje duże ilości danych w ograniczonej pamięci operacyjnej. Analiza wyników eksperymentu potwierdziła teoretyczne modele.