Scalanie przy użyciu buforów wielkich

Antoni Czuba 201096 Sem. V Grupa 5A

I. Opis algorytmu

W projekcie zastosowałem scalenie przy użyciu wielkich buforów, działa on w następujący sposób.

1. **Etap 1. (Tworzenie przebiegów)**
   a. Wczytaj pierwsze *nb* rekordów z pliku do buforów i posortuj je, używając wydajnej metody sortowania w pamięci (QuickSort, HeapSort, …), tworząc przebieg (run) o długości *nb*.
   b. Zapisz przebieg na dysk.
   c. Powtarzaj kroki a i b, aż do osiągnięcia końca pliku
2. **Etap 2. (Scalanie)**
   a. Scal pierwsze *n-1* przebiegów, używając *n*-tego bufora do stworzenia wyjściowego przebiegu, a następnie zapisz go na dysku.
   b. Powtarzaj krok b dla kolejnych przebiegów z pliku, aż do jego końca.
   c. Powtarzaj kroki a i b, aż pozostanie tylko jeden przebieg.

Gdzie
N – liczba rekordów w pliku
b – czynnik blokowania (liczba rekordów w jednym buforze dla jednej strony dyskowej w pamięci operacyjnej)
n – liczba buforów w pamięci operacyjnej dostępnych dla procesu sortowania

II. Implementacja

Jako zbiór danych posłużyły mi tablice rejestracyjne, przechowywałem je w postaci Stringów, o długości 8. Rekordy jakie generowałem miały następującą postać

XXXYYYYY

Wyróżnik miejsca oznaczony 2-3 znakami X składa się z samych liter alfabetu łacińskiego, w przypadku wyróżnika o długości 2 na końcu rekordu dodawana była spacja aby zachować stałą długość rekordu.

Natomiast wyróżnik pojazdu oznaczony 5 znakami Y składa głównie z cyfr arabskich, ale mogą pojawić się tam również litery alfabetu łacińskiego..

Rekordy przechowywałem w pliku .txt aby uprościć debugging i mieć prostszy dostęp do danych, a rekordu w nim były oddzielone znakiem nowej lini.

III. Eksperyment

Eksperyment przeprowadziłem na 4 zestawach danych dla b=5 oraz n=6.

Number of records: 30
Number of disk operations: 12
Number of disk read: 6
Number of disk write: 6
Number of merge phases: 0

Number of records: 31
Number of disk operations: 28
Number of disk read: 14
Number of disk write: 14
Number of merge phases: 1

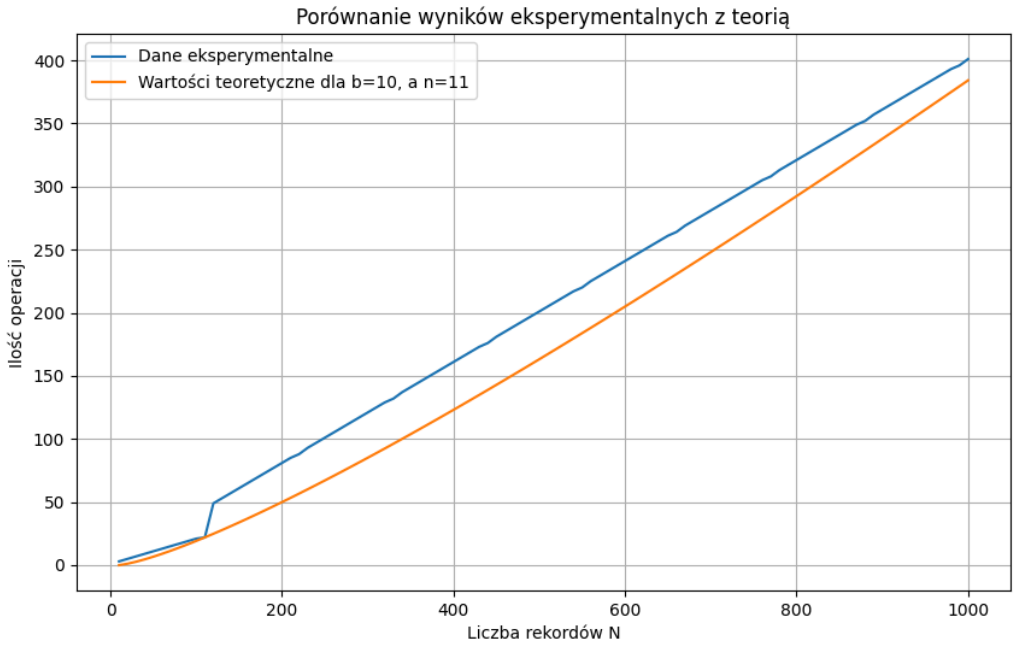
Pierwsze dwa zestawy danych pomimo bardzo małej różnicy w ilości rekordów dają nam znacznie inne rezultaty, wynika to z tego że pierwszy zestaw został posortowany w całości w pierwszym etapie czyli tworzenia runów, rekordów jest dokładnie tyle ile mieści się w buforach czyli b*n=5*6=30, natomiast w drugim przypadku rekordów jest 31 czyli przeprowadzane jest scalanie runów co daje znaczną dodatkowa ilość operacji odczytu i zapisu.

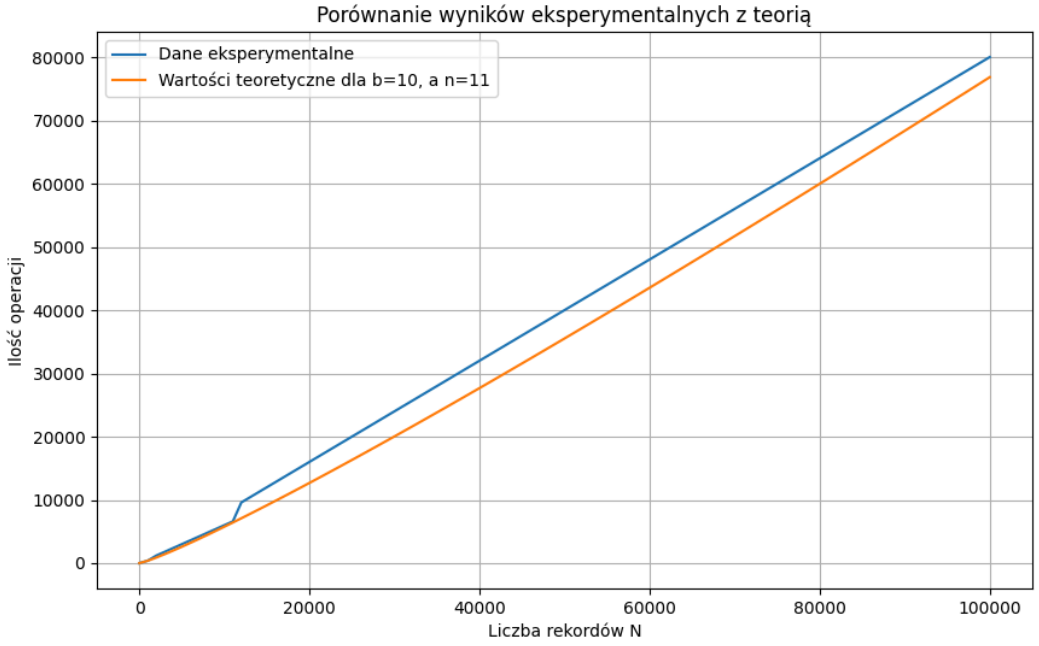
Number of records: 150
Number of disk operations: 120
Number of disk read: 60
Number of disk write: 60
Number of merge phases: 1

Number of records: 151
Number of disk operations: 186
Number of disk read: 93
Number of disk write: 93

Kolejne dwa zestawy danych ponownie pomimo bardzo małej różnicy w ilości rekordów dają nam znacznie inne rezultaty, wynika to z tego że pierwszy zestaw po pierwszym etapie składał się z n-1=5 runów bo n*b*(n-1)=6*5*5=150 co pozwoliło na tylko jeden przebieg scalania, z kolei następny zestaw wyszedł poza granice 151 rekordów, więc wymagał scalania dwóch przebiegów scalania runów,

Porównanie z teoria





Jak widać wzory szacujące ilość operacje dosyć wiernie oddają tendencje i kształt krzywej, natomiast widać wyraźne wzrosty ilości operacji w kluczowych punktach opisanych przeze mnie na wyżej opisanych danych, tj, na górnym wykresie widać wyraźny wzrost przy n*b, a na dolnym przy n*b*1000.