

Zaawansowane zastosowania kart graficznych

Laboratorium 3

Zadania demonstrujące wykorzystanie podstawowych klocków, z których buduje się algorytmy równoległe.

1. Napisz program znajdujący dla każdego elementu największą wartość znajdującą się przed nim w tablicy. Wykorzystaj operację scan.
2. Napisz program zliczający jedynki (bity) w tablicy intów (osobno dla każdego inta w tablicy). Wykorzystaj a) trik bitowy (parz slajdy z wykładu), b) instrukcję `__popc()`.
3. Zmodyfikuj powyższy program aby podawał sumaryczną liczbę wartości. Wykorzystaj operację reduce.
4. Napisz program obliczający liczbę liczb parzystych znajdujących się w tablicy. Wykorzystaj operację reduce.
5. Napisz program znajdujący długość wspólnego prefiksu dwóch łańcuchów za pomocą operacji scan i reduce.
6. Napisz program znajdujący medianę w tablicy wejściowej. Wykorzystaj sort.
7. Napisz program wyszukujący wszystkie liczby większe od 10 i zwracający ich pozycje w tablicy wejściowej. Wykorzystaj operację compact.
8. Poczytaj o iteratorze `counting_iterator` (https://thrust.github.io/doc/classthrust_1_1counting_iterator.html) i wykonaj poprzednie zadanie z jego użyciem.
9. Napisz program wyszukujący zadany podłańcuch w większym łańcuchu. Zwróć położenia wszystkich wystąpień w osobnej tablicy. Wykorzystaj compact.
10. Napisz program wykonujący kompresję RLE za pomocą `reduce_by_key`.
11. Poczytaj o iteratorze `constant_iterator` (https://thrust.github.io/doc/classthrust_1_1constant_iterator.html) i wykonaj poprzednie zadanie z jego użyciem.
12. Zaimplementuj algorytm compact.
13. Dana jest posortowana tablica liczb. Część z nich to liczby kolejne a część nie. Przykładowo: 2,3,4,8,9,10,11,12,17,20,21,22,23,24. Oblicz tablicę zawierającą długości ciągów kolejnych liczb. Dla przykładowych danych wejściowych powinno to być: 3,3,2,1,5.
14. Zaimplementuj algorytm usuwania duplikatów z posortowanej tablicy.
15. Zaimplementuj algorytm, który dla danej tablicy dodatnich liczb x , np. 3,5,2,6, zwróci tablicę o długości równej sumie liczb x i zawierającą kolejne liczby od 0 do wartości $x-1$. Dla przykładowej tablicy wejściowej byłoby to 0,1,2(3),0,1,2,3,4(5), 0,1(2),0,1,2,3,4,5(6).
16. Niech będzie dana tablica kluczy, np. 1,1,1,2,2,3,3,3 oraz tablica odpowiadających im wartości, np.: a,b,c,a,d,e,f,k,l. W celu wytłumaczenia zadania wprowadzono następujące oznaczenie. Niech $N(k)$ oznacza liczbę wystąpień klucza, np. $N(3)$ wynosi 4. W wyniku powinny powstać trzy tablice K , $V1$ i $V2$. Obydwie tablice powinny mieć rozmiar równy sumie liczby kombinacji po 2 z $N(k)$ po wszystkich kluczach występujących w tablicy wejściowej. W przykładowej tablicy $N(1)=3$, $N(2)=2$ i $N(3)=4$. Stąd $C_2^{N(1)} = 3$, $C_2^{N(2)} = 1$ i $C_2^{N(4)} = 6$. Tablice wynikowe powinny zawierać zatem $3+1+6=10$ pozycji. W tablicach wynikowych powinny

pojawić się wszystkie kombinacje dwuelementowe wartości z tablicy wejściowej. W niniejszym przykładzie wynik powinien wyglądać następująco:

Dane wejściowe	
Klucze	Wartości
1	A
1	B
1	C
2	A
2	D
3	E
3	F
3	K
3	L

Wynik			
	K	V1	V2
Kombinacje wartości A,B,C	1	A	B
	1	A	C
	1	B	C
Kombinacje wartości A,D	2	A	D
Kombinacje wartości E,F,K,L	3	E	F
	3	E	K
	3	E	L
	3	F	K
	3	F	L
	3	K	L