

Raport dla zadania zaliczeniowego

Uladzislau Sobal ws374078

9 stycznia 2017

Implementacja i używane optymalizacje

Kompresja wierzchołków

Za pomocą hash mapy kompresowane są indeksy wszystkich wierzchołków żeby one były w przedziale od 0 do $N - 1$, gdzie N - ilość wierzchołków. Dzięki tej kompresji, można używać zwykłych wektorów zamiast map przy obliczaniu. Za kompresję jest odpowiedzialna klasa `CoordinateCompressor`.

Podział pracy

Wierzchołki dla których trzeba obliczyć wartości są przechowywane w wektorze `importantVertices`, z którego procesy biorą indeksy kiedy kończą swoje poprzednie zadanie za pomocą funkcji `getNextVertexIndex`.

Przekazywanie wyników

Każdy proces wykonuje funkcję `verticesProcessor`. Ta funkcja dopóki może próbuje otrzymać wierzchołek dla obrobienia, dla którego wywołuje funkcję `processVertex`, która wykonuje jedną iterację algorytmu Brandesa dla danego wierzchołka. Żeby przyspieszyć działanie, ta funkcja przyjmuje jako argument wektor, do którego ona będzie dodawać wyniki. W końcu, kiedy proces skończył pracę, on blokuje wektor wyniku końcowego bc za pomocą muteksa `bcMutex` i zapisuje do `bc` swoje obliczone wartości.

Wyniki optymalizacji

W tabeli poniżej podane są czasy wykonania oraz przyspieszenie na danych z wikipedii na komputerze students:

Ilość wątków	Czas (sekundy)	Przyspieszenie
1	12.7	1
2	7.3	1.74
3	4.33	2.933
4	3.73	3.405
5	3.16	4.019
6	2.78	4.568
7	2.19	5.799
8	1.73	7.341