

BFS na dużych grafach

Bartosz Maj

1 Podsumowanie wyników

Przeprowadzona analiza obejmowała cztery duże grafy: roadNet-CA, roadNet-PA, web-BerkStan i WikiTalk. Algorytm BFS (Breadth-First Search) został zaimplementowany w różnych wersjach, w tym BFSQueueVersion, BFSLayersVersion, BFSAtomicOppVersion, BFSAtomicOppGlobalVersion oraz wersji sekwencyjnej. Celem analizy było porównanie wydajności tych wersji w kontekście dużych grafów. Poniżej przedstawiona jest różnica pomiędzy algorytmami na CPU i GPU. Testy zostały przeprowadzone na karcie graficznej NVIDIA GeForce RTX 3070, jest to nowa karta graficzna 3 letnia, stąd atomowe operacje są szybsze niż na starszych.

Plik	Init	BFSQueue	BFSLayers	BFSAtomic	BFSAtomicGlobal	Seq. BFS
roadNet-CA.txt	27.74 ms	140.96 ms	80.46 ms	71.21 ms	88.35 ms	683.99 ms
roadNet-PA.txt	16.88 ms	53.89 ms	41.07 ms	24.43 ms	30.42 ms	276.13 ms
web-BerkStan.txt	30.25 ms	19.31 ms	14.31 ms	n/a	15.18 ms	491.31 ms
WikiTalk.txt	31.06 ms	97.43 ms	75.22 ms	n/a	84.69 ms	254.27 ms

Tabela 1: Wyniki BFS dla różnych plików grafów

2 Inicjalizacja

Czas inicjalizacji był zróżnicowany dla poszczególnych grafów, co może świadczyć o zależności od struktury grafu, liczby krawędzi oraz wierzchołków.

3 Wydajność Algorytmów BFS

3.1 BFS Queue

Ta wersja algorytmu była generalnie wolniejsza niż inne, z najdłuższym czasem dla grafu. Świadczy to o możliwych ograniczeniach tej metody spowodowanych dwoma funkcjami kernelowymi.

3.2 BFS Layers

Ta metoda prezentowała lepszą wydajność, szczególnie dla grafów gęstych. Jest to najprostsza, a zarazem najszybsza metoda.

3.3 BFS Atomic

Ta metoda nie zadziałała dla dwóch grafów, ponieważ jest tworzona lokalna kolejka, która może pomieścić maksymalnie 8192 wierzchołki. Dla grafów gęstych metoda ta nie działa. Dla grafów rzadkich daje ona zadowalające wyniki.

3.4 BFS Atomic Global

Jest to najwolniejsza metoda na GPU (w przypadku starszej karty graficznej np. 1070), ponieważ jest tworzona kolejka globalna, stąd jest dużo atomowych operacji na pamięci globalnej.

3.5 Sequential BFS

Jak oczekiwano, wersja sekwencyjna była najwolniejsza we wszystkich przypadkach, co podkreśla znaczenie równoległego przetwarzania w dużych grafach.

4 Wnioski

4.1 Znaczenie Struktury Grafu

Czas inicjalizacji i wydajność różnych wersji algorytmu BFS wyraźnie różniły się w zależności od struktury grafu. W szczególności gęstość grafu i liczba wierzchołków miały duży wpływ na efektywność poszczególnych metod. Wnosi to, że wybór optymalnej metody BFS powinien uwzględniać charakterystykę konkretnego grafu.

4.2 Ograniczenia i Wydajność Algorytmów

- Algorytmy **BFSAtomic** i **BFSAtomicGlobal** wykazały swoje ograniczenia, szczególnie w kontekście grafów o dużej gęstości, co może być związane z ograniczeniami pamięciowymi i zwiększoną liczbą operacji atomowych.
- Wersja **Sequential BFS** była konsekwentnie najwolniejsza, co potwierdza przewagę przetwarzania równoległego nad sekwencyjnym w dużych grafach.
- Metody **BFSQueue** i **BFSLayers** zapewniały lepszą wydajność, co sugeruje ich użyteczność w różnych scenariuszach.