

Współbieżny PageRank w CPP - Raport

Grzegorz Zaleski (418494)

22 lutego 2021

1 Wstęp

W tym raporcie przedstawię wyniki i analizę wykonania testów dostarczonych w poleceniu na mojej implementacji programu obliczającego współbieżnie wartości PageRank dla podanej sieci połączeń stron.

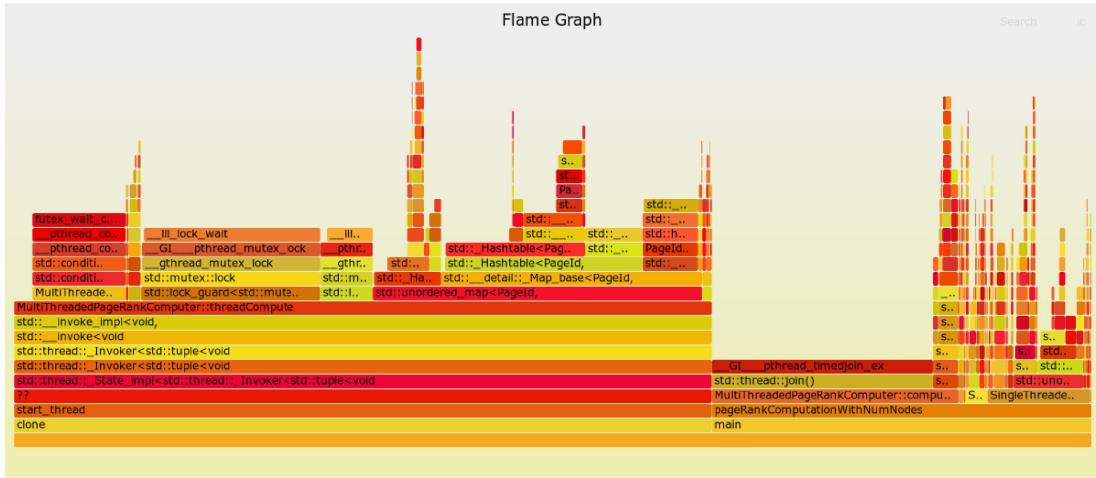
Należy wspomnieć że dla każdego testu program zwrócił **poprawny** wynik, **bez wycieków pamięci**, a także w wykonał powierzone zadanie **szyciej** niż dostarczona implementacja sekwenncyjna.

Testy wykonałem w dwóch środowiskach - na moim osobistym laptopie z systemem Debian - wyposażonym w 32 GB RAM-u oraz procesor Inter Core i7-6700HQ oraz udostępnionym nam przez uczelnię systemie systemie students.

2 Wynik testów

```
PageRank Performance Test [100 nodes, SingleThreadedPageRankComputer] took: 0.006191s
PageRank Performance Test [1000 nodes, SingleThreadedPageRankComputer] took: 1.27066s
PageRank Performance Test [2000 nodes, SingleThreadedPageRankComputer] took: 4.87456s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[1]] took: 3.65544s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[2]] took: 2.83799s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[3]] took: 2.36156s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[4]] took: 2.27087s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[8]] took: 2.2145s
PageRank Performance Test [500000 nodes, SingleThreadedPageRankComputer] took: 9.90874s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[1]] took: 8.29359s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[2]] took: 6.27394s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[3]] took: 5.44028s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[4]] took: 4.8932s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[8]] took: 4.65946s
E2E Test [SingleThreadedPageRankComputer] took: 9.64619s
E2E Test [MultiThreadedPageRankComputer[1]] took: 9.37962s
E2E Test [MultiThreadedPageRankComputer[2]] took: 2.69133s
E2E Test [MultiThreadedPageRankComputer[3]] took: 2.11258s
E2E Test [MultiThreadedPageRankComputer[4]] took: 1.95237s
E2E Test [MultiThreadedPageRankComputer[8]] took: 2.47298s
```

Czasy wykonywanie testów na moim komputerze.



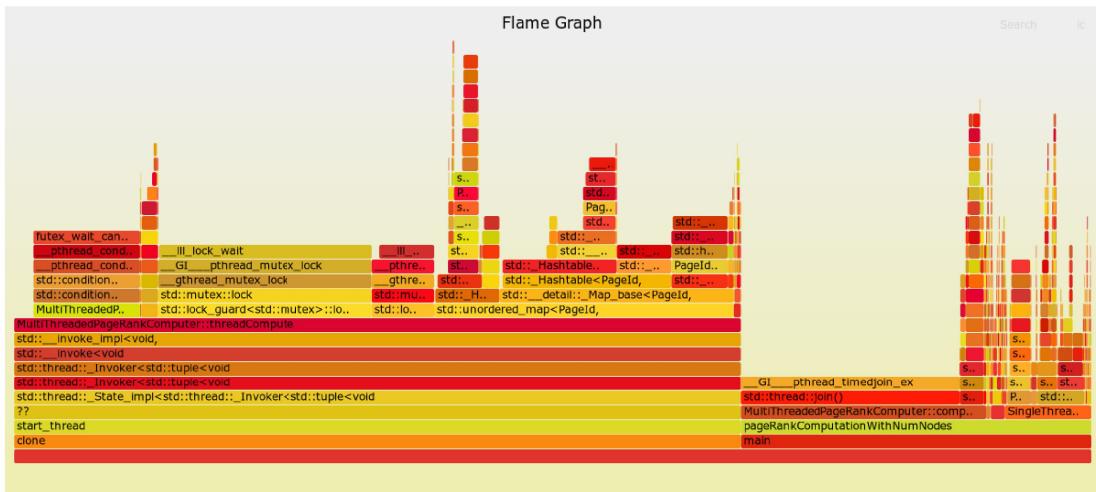
Wykres profilowania z wykonywania testów na moim komputerze.

```

PageRank Performance Test [100 nodes, SingleThreadedPageRankComputer] took: 0.0071144s
PageRank Performance Test [1000 nodes, SingleThreadedPageRankComputer] took: 1.60143s
PageRank Performance Test [2000 nodes, SingleThreadedPageRankComputer] took: 6.53697s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[1]] took: 5.94859s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[2]] took: 4.64073s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[3]] took: 4.47022s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[4]] took: 3.55791s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[8]] took: 4.1539s
PageRank Performance Test [500000 nodes, SingleThreadedPageRankComputer] took: 15.4084s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[1]] took: 12.8509s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[2]] took: 9.88027s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[3]] took: 8.91308s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[4]] took: 8.65884s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[8]] took: 8.87985s
E2E Test [SingleThreadedPageRankComputer] took: 14.1154s
E2E Test [MultiThreadedPageRankComputer[1]] took: 12.9076s
E2E Test [MultiThreadedPageRankComputer[2]] took: 7.27292s
E2E Test [MultiThreadedPageRankComputer[3]] took: 5.95831s
E2E Test [MultiThreadedPageRankComputer[4]] took: 5.29627s
E2E Test [MultiThreadedPageRankComputer[8]] took: 5.23686s

```

Czasy wykonywanie testów w systemie students.



Wykres profilowania z wykonywania testów w systemie students.

3 Analiza i wnioski

1. Czas wykonywania testów są proporcjonalne między sobą w obu przypadkach testowych, a więc można założyć że rozwiązanie jest w ogólności poprawne na każdej maszynie spełniającej wytyczne z polecenia.
2. Oba wykresy profilowania wyglądają dostatecznie podobne - można więc stwierdzić, że podział pracy między wątkami tj. procent czasu/mocy potrzebnych na wykonanie części zadania jest stały.
3. Wygląd wykresu przedstawia oczekiwane działanie programu i nie budzi żadnych zastrzeżeń. Znaczną większość zajmuje funkcja *MultiThreadedPageRankComputer::threadCompute()* która jest główną częścią programu, a także jedną częścią - poza inicjalizacją struktur i zapisywaniem wyniku w *MultiThreadedPageRankComputer::computeForNetwork()*. Lekki niepokój może budzić rozmiar udziału *lockguard*, jednak zważywszy na to że takie rzeczy jak przydzielanie wierzchołków do obliczania czy aktualizowanie obliczonych wartości nie mogą odbywać się wspólnie ta wartość nie budzi zastrzeżeń.
4. Elementem wartym uwagi jest także fakt że *MultiThreadedPageRankComputer* nawet przy jednym wątku który wykonuje wszystkie operacje - oraz dodatkowe musi zabierać i odkładać mutexy i bariery przechodzi testy szybciej niż *SingleThreadedPageRankComputer* - wynika to z kilku autorskich optymizacji do koncepcji obliczania wyniku takich jak np. tworzenie vectora z wynikami na końcu a nie co iteracje czy trzymaniu *danglingNodes* na vectorze a nie secie.
5. W każdym kolejnym wątkiem dane są przerobiane coraz szybciej, co jest zjawiskiem dosyć oczywistym i naturalnym, albowiem praca jest dzielona na więcej wątków to każda porcja pracy jest mniejsza.
6. Z każdym kolejnym wątkiem w *MultiThreadedPageRankComputer* czas przetworzenia testu maleje coraz wolniej - wynika to z faktu że więcej wątków musi naraz czekać na mutexy do struktur z ograniczoną współbieżnością czy na barierach, przez co nie wykonują wtedy żadnej pracy.