

Analiza wydajności zrównoleżonego algorytmu Page Rank

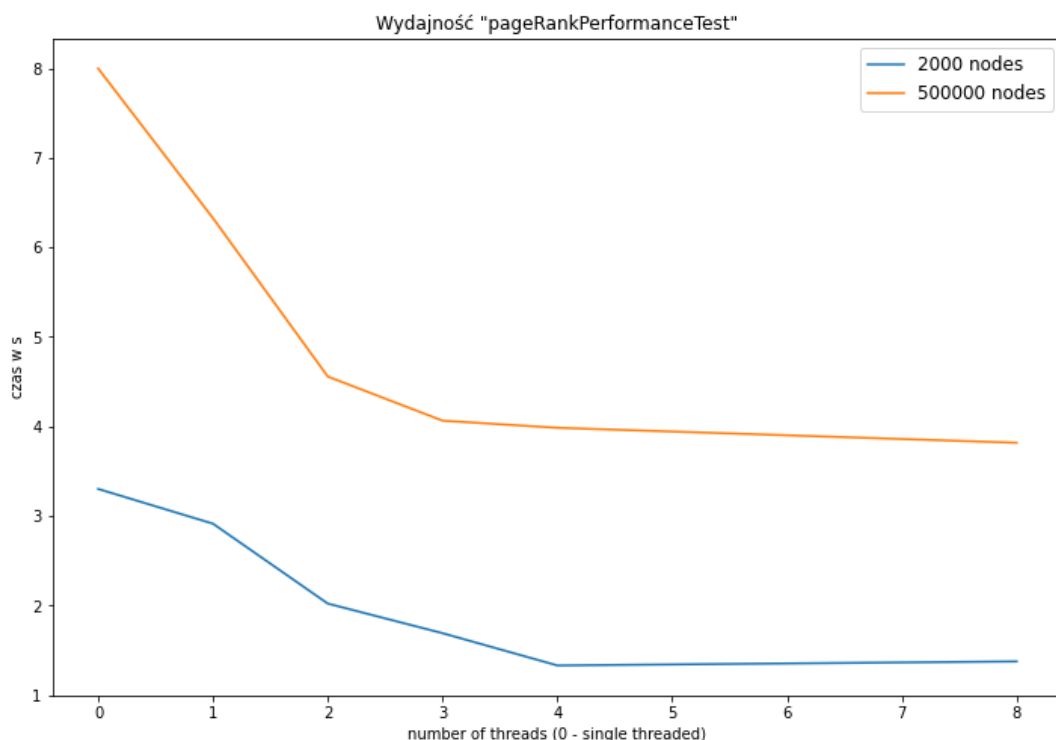
1. Środowisko I – komputer studenta:

Ubuntu 20.04.1 / Intel i5-9300H @ 2.4GHz, 4 rdzenie / 8GB RAM (postawiony na VBox)

a. Czasy po odpaleniu skryptu ./runTest.sh w wersji release:

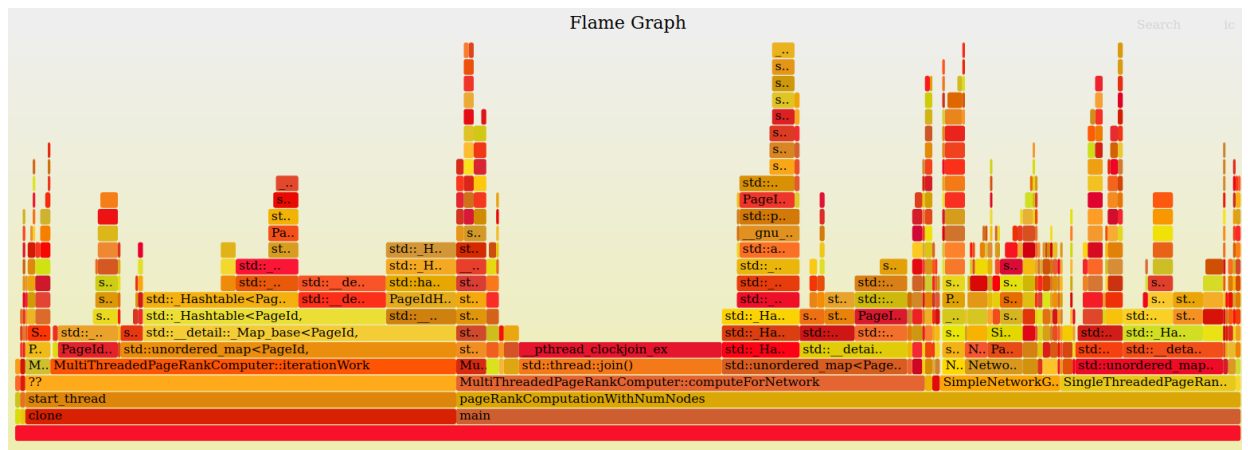
```
PageRank Performance Test [100 nodes, SingleThreadedPageRankComputer] took: 0.0074849s
PageRank Performance Test [1000 nodes, SingleThreadedPageRankComputer] took: 0.838753s
PageRank Performance Test [2000 nodes, SingleThreadedPageRankComputer] took: 3.30177s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[1]] took: 2.91361s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[2]] took: 2.02096s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[3]] took: 1.69002s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[4]] took: 1.32959s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[8]] took: 1.37461s
PageRank Performance Test [500000 nodes, SingleThreadedPageRankComputer] took: 7.99944s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[1]] took: 6.32712s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[2]] took: 4.55745s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[3]] took: 4.06442s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[4]] took: 3.9848s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[8]] took: 3.81769s
E2E Test [SingleThreadedPageRankComputer] took: 3.12913s
E2E Test [MultiThreadedPageRankComputer[1]] took: 2.94537s
E2E Test [MultiThreadedPageRankComputer[2]] took: 1.5819s
E2E Test [MultiThreadedPageRankComputer[3]] took: 1.17174s
E2E Test [MultiThreadedPageRankComputer[4]] took: 0.966399s
E2E Test [MultiThreadedPageRankComputer[8]] took: 0.974881s
```

b. pageRankPerformanceTest:



W obu przypadkach możemy zauważyć istotny wzrost efektywności obliczeń wraz ze wzrostem liczby operujących wątków (dla 2, 3, 4 wątków). Dalej wykres się wypłaszcza, jest to związane z tym, że mamy do dyspozycji tylko 4 rdzenie, dalej pojawiają się również dodatkowe narzuty związane z tworzeniem, synchronizowaniem i joinowaniem większej ilości wątków. Ponadto wzrost efektywności jest bardziej wyraźny, w przypadku kiedy mamy średnio więcej obliczeń wykonywanych dla każdej strony w naszej sieci (test dla 2k nodes) niż dla testu kiedy obliczeń

(krawędzi) jest niewiele. Wynika to ze sposobu implementacji iteracji obliczeń, gdzie każdy wątek dostaje do policzenia pageRank stron pomiędzy zadanymi iteratorami haszmapy.



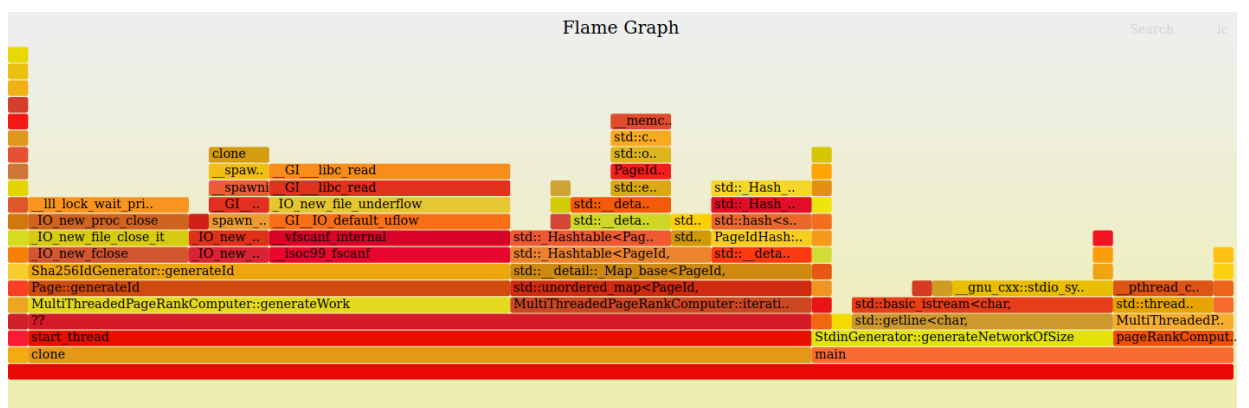
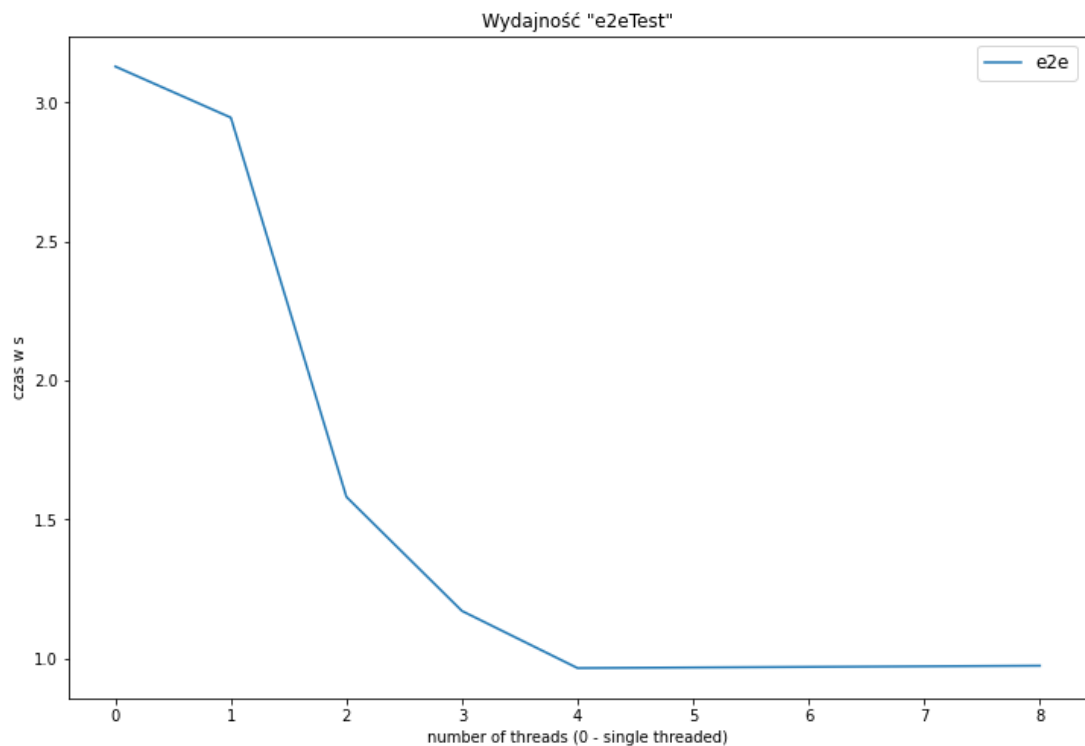
Z Flame Graphu w wersji interaktywnej (dla firefoxa) możemy odczytać instrukcje dominujące wykonywane przez wątek główny (prawa strona wykresu, nad mainem w drugim wierszu od dołu), są to:

- generowanie sieci (ok. 10% całkowitego czasu);
- obliczenia singleThreaded (ok. 15% całkowitego czasu);
- obliczenia multiThreaded (ok. 38% całkowitego czasu), i możemy wyróżnić:
 - joinowanie, czyli oczekiwanie na wątki pomocnicze (ok. 16% całkowitego czasu), co jest jak najbardziej uzasadnione, biorąc pod uwagę fakt, iż zdecydowałem się na implementację przy użyciu join loop, a nie np. barrier (wtedy sporo czasu zajmowałoby wiszenie na semaforach),
 - operacje na statycznych strukturach, unordered_map/set (ok. 15% całkowitego czasu), przygotowanie struktur do pierwszej iteracji, generowanie grafu zależności itd.,
 - dodatkowe składowe to np. pobieranie id z wektora, czyszczenie statycznych struktur.

A dla wątków pomocniczych:

- generowanie id (simpleGenerator) (ok. 2% całkowitego czasu);
- obliczenia w metodzie iterationWork (ok. 35% całkowitego czasu), gdzie:
 - operacjami dominującymi były te wykonywane na statycznych strukturach unordered map/set, czyli kopiowanie, aktualizacja i pobieranie wartości
 - oraz pobieranie identyfikatora strony.

c. e2eTest:



Wnioski co do wydajności e2eTest są w zasadzie takie same jak w przypadku performanceTest zwłaszcza dla >4 wątków pomocniczych. Warto jednak zwrócić uwagę na istotną różnicę pomiędzy tymi testami, w tym drugim generujemy id za pomocą unixowego sha256sum. Wyraźnie widać to na powyższym Flame Graphie, metoda generateId zajmuje prawie 40% całkowitego czasu. Na generateId składają się tworzenie pipe, forkowanie i wywołanie sha256sum (przy użyciu popen) oraz pisanie generowanych identyfikatorów na odpowiednie strumienie.

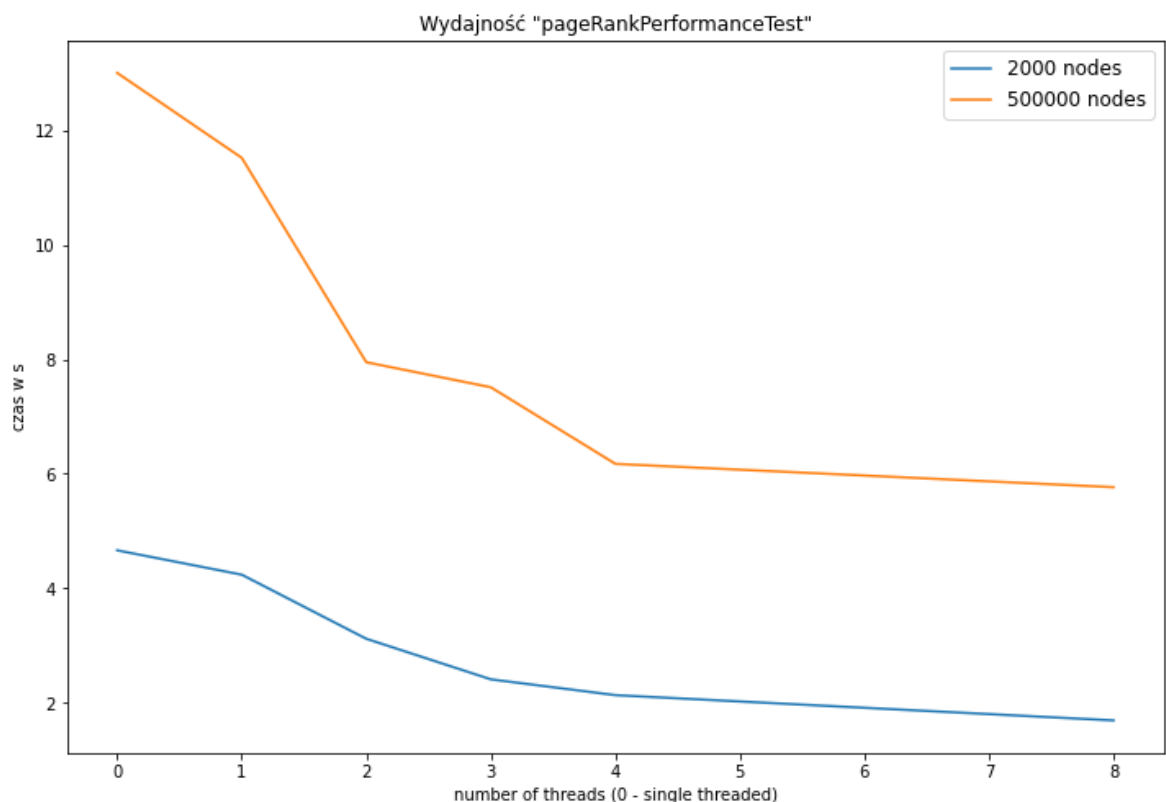
2. Środowisko II – students:

Debian 4.19.132-1 / Intel Xeon @ 2.1GHz, 64 rdzenie / 336 GB RAM

a. Czasy po odpaleniu skryptu ./runTest.sh:

```
PageRank Performance Test [100 nodes, SingleThreadedPageRankComputer] took: 0.00426761s
PageRank Performance Test [1000 nodes, SingleThreadedPageRankComputer] took: 1.09851s
PageRank Performance Test [2000 nodes, SingleThreadedPageRankComputer] took: 4.66309s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[1]] took: 4.23764s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[2]] took: 3.11835s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[3]] took: 2.41003s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[4]] took: 2.13399s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[8]] took: 1.69272s
PageRank Performance Test [500000 nodes, SingleThreadedPageRankComputer] took: 12.9999s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[1]] took: 11.517s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[2]] took: 7.94769s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[3]] took: 7.51038s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[4]] took: 6.17303s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[8]] took: 5.76546s
E2E Test [SingleThreadedPageRankComputer] took: 15.2301s
E2E Test [MultiThreadedPageRankComputer[1]] took: 11.7104s
E2E Test [MultiThreadedPageRankComputer[2]] took: 6.60997s
E2E Test [MultiThreadedPageRankComputer[3]] took: 4.63638s
E2E Test [MultiThreadedPageRankComputer[4]] took: 4.12243s
E2E Test [MultiThreadedPageRankComputer[8]] took: 3.98582s
```

b. pageRankPerformanceTest:



Możemy zauważyć, że czas działania programu na serwerze students jest istotnie różny względem tego, co mogliśmy zaobserwować na maszynie lokalnej (choćby porównując czas działania wersji singleThreaded). Lokalnie algorytm działa dużo szybciej. Z drugiej strony na studentsie mamy do dyspozycji aż 64 wątki – wykres dla pageRankPerformanceTest się nie wypłaszcza, czas działania jest wyraźnie mniejszy dla coraz większej liczby wątków. Aczkolwiek tempo przyspieszenia maleje (więcej wątków, to większe narzuty przy tworzeniu, joinowaniu i synchronizacji).

Warto jeszcze rozważyć różnice związane z generowaniem id stron w formie hasha SHA256. Widać, że te koszty w środowisku students są dużo wyższe (co jest bardzo sensownym wytłumaczeniem tego, że e2eTest idzie dużo wolniej, na lokalu 40% całkowitego czasu, tutaj 75% !, w szczególności spory narzut stanowi wywoływanie popen).