

Środowiska testowe

System students:

- Linux students 4.19.0-10-amd64 #1 SMP Debian 4.19.132-1 (2020-07-24) x86_64 GNU/Linux
- Debian 10 buster
- GCC 8.3.0
- Procesor: Intel Xeon Processor (Skylake, IBRS), model nieznany, włączona wirtualizacja, reportowane 64 rdzenie przy 2GHz

Mój komputer stacjonarny:

- Windows 10 Home + WSL Ubuntu 18.04.2 LTS
- GCC (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0
- Procesor : Intel(R) Core(TM) i5-8400 CPU @ 2.80GHz 2.81 GHz

Wyniki wersji DEBUG

System students:

```
PageRank Performance Test [100 nodes, SingleThreadedPageRankComputer] took: 0.0195784s
PageRank Performance Test [1000 nodes, SingleThreadedPageRankComputer] took: 2.92121s
PageRank Performance Test [2000 nodes, SingleThreadedPageRankComputer] took: 10.9237s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[1]] took: 11.8581s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[2]] took: 7.29567s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[3]] took: 5.5261s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[4]] took: 4.50754s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[8]] took: 3.56227s
PageRank Performance Test [500000 nodes, SingleThreadedPageRankComputer] took: 22.1974s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[1]] took: 21.4349s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[2]] took: 17.0847s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[3]] took: 14.0905s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[4]] took: 12.1691s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[8]] took: 10.1103s
E2E Test [SingleThreadedPageRankComputer] took: 19.4758s
E2E Test [MultiThreadedPageRankComputer[1]] took: 17.1058s
E2E Test [MultiThreadedPageRankComputer[2]] took: 9.35183s
E2E Test [MultiThreadedPageRankComputer[7]] took: 5.55958s
```

Mój komputer stacjonarny:

```
PageRank Performance Test [100 nodes, SingleThreadedPageRankComputer] took: 0.0150387s
PageRank Performance Test [1000 nodes, SingleThreadedPageRankComputer] took: 1.91165s
PageRank Performance Test [2000 nodes, SingleThreadedPageRankComputer] took: 7.01655s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[1]] took: 7.13343s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[2]] took: 4.37246s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[3]] took: 3.78326s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[4]] took: 2.91893s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[8]] took: 2.4824s
PageRank Performance Test [500000 nodes, SingleThreadedPageRankComputer] took: 12.0329s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[1]] took: 11.5576s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[2]] took: 7.59065s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[3]] took: 7.09416s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[4]] took: 6.5759s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[8]] took: 6.37191s
E2E Test [SingleThreadedPageRankComputer] took: 20.6378s
E2E Test [MultiThreadedPageRankComputer[1]] took: 20.3834s
E2E Test [MultiThreadedPageRankComputer[2]] took: 10.7944s
E2E Test [MultiThreadedPageRankComputer[7]] took: 8.67018s
```

Wyniki wersji RELEASE

System students:

```
PageRank Performance Test [100 nodes, SingleThreadedPageRankComputer] took: 0.00607534s
PageRank Performance Test [1000 nodes, SingleThreadedPageRankComputer] took: 1.49248s
PageRank Performance Test [2000 nodes, SingleThreadedPageRankComputer] took: 7.5373s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[1]] took: 8.00292s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[2]] took: 4.66865s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[3]] took: 3.95254s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[4]] took: 3.44476s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[8]] took: 2.63506s
PageRank Performance Test [500000 nodes, SingleThreadedPageRankComputer] took: 15.3453s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[1]] took: 14.2171s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[2]] took: 11.0089s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[3]] took: 9.12297s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[4]] took: 8.01639s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[8]] took: 7.08782s
E2E Test [SingleThreadedPageRankComputer] took: 16.7526s
E2E Test [MultiThreadedPageRankComputer[1]] took: 14.3045s
E2E Test [MultiThreadedPageRankComputer[2]] took: 7.44391s
E2E Test [MultiThreadedPageRankComputer[3]] took: 5.80818s
E2E Test [MultiThreadedPageRankComputer[4]] took: 4.81201s
E2E Test [MultiThreadedPageRankComputer[8]] took: 4.83322s
```

Mój komputer stacjonarny:

```
PageRank Performance Test [100 nodes, SingleThreadedPageRankComputer] took: 0.0057064s
PageRank Performance Test [1000 nodes, SingleThreadedPageRankComputer] took: 0.959262s
PageRank Performance Test [2000 nodes, SingleThreadedPageRankComputer] took: 3.83136s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[1]] took: 3.98374s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[2]] took: 2.41189s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[3]] took: 1.9064s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[4]] took: 1.76755s
PageRank Performance Test [2000 nodes, MultiThreadedPageRankComputer[8]] took: 1.50974s
PageRank Performance Test [500000 nodes, SingleThreadedPageRankComputer] took: 6.39708s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[1]] took: 6.15382s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[2]] took: 4.31226s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[3]] took: 4.18963s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[4]] took: 4.07623s
PageRank Performance Test [500000 nodes, MultiThreadedPageRankComputer[8]] took: 3.53221s
E2E Test [SingleThreadedPageRankComputer] took: 18.5532s
E2E Test [MultiThreadedPageRankComputer[1]] took: 18.2604s
E2E Test [MultiThreadedPageRankComputer[2]] took: 9.52894s
E2E Test [MultiThreadedPageRankComputer[3]] took: 7.96711s
E2E Test [MultiThreadedPageRankComputer[4]] took: 7.84611s
E2E Test [MultiThreadedPageRankComputer[8]] took: 7.95514s
```

Wnioski dotyczące wydajności

Program w wersji jednowątkowej oraz wersji wielowątkowej był optymalizowany w paru miejscach m.in. unikanie wielokrotnego użycia `getid()` przez każdy wątek, unikanie kopii identyfikatorów, liczenie wyniku tylko w ostatniej iteracji. Mimo, że oczekiwano spowolnienie wersji wielowątkowej przy jednym wątku, jest on albo nieznacznie wolniejszy albo nieznacznie szybszy. Jest to spowodowane losowością oraz różnicami implementacji obu programów.

Testy wydajnościowe bez E2E:

W wersji wielowątkowej, uruchomienie na dwóch wątkach daje około 30-35% krótszy czas wykonania na większym teście i około 40-45% na mniejszym teście.

Na czterech : 40-50% na większym teście oraz 55-60% na mniejszym teście.

Na ośmiu : 50-55% na większym teście oraz 60-65% na mniejszym teście.

Testy E2E [release] na moim systemie: 2 wątki – 50%, 4 wątki – 55%, 8 wątków – 55%

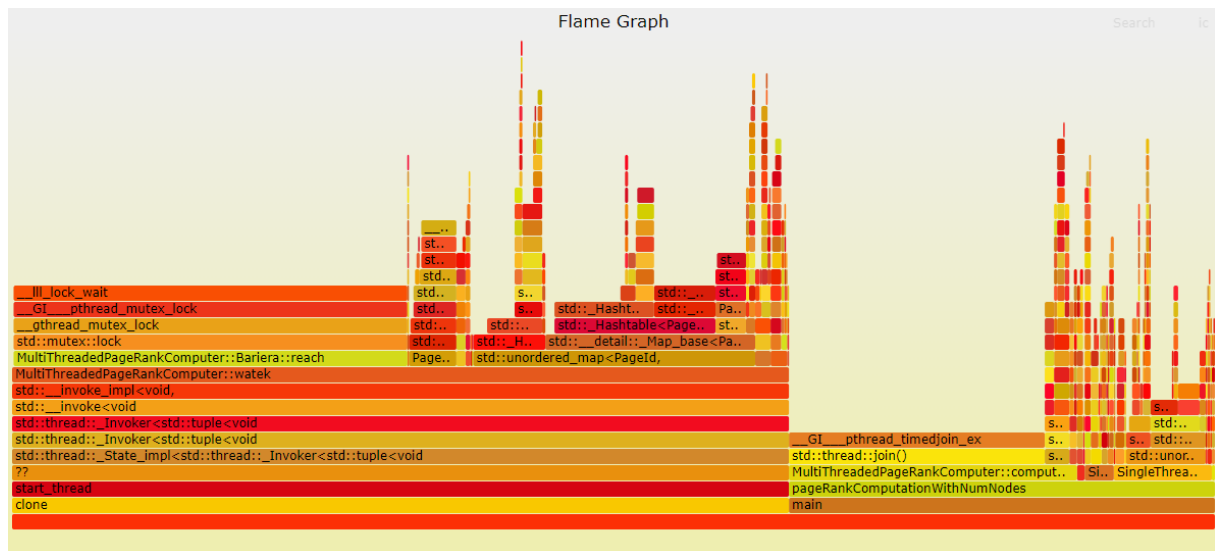
Testy E2E [release] na students: 2 wątki – 50%, 4 wątki – 65%, 8 wątków – 65%

Testy E2E są jedynymi testami, których czasy 'zbiegają' do pewnej wartości. Prawdopodobnie jest to spowodowane implementacją `sha256sum`.

Wyniki E2E na moim systemie są gorsze, niż te na students, prawdopodobnie spowodowane różnicą środowisk.

Zachowanie programu wg. Flame Graph

Wykres FlameGraph wygenerowany na systemie students dla załączonego rozwiązania wygląda następująco (dla pageRankPerformanceTest):



Wykres wygląda w ten sposób, ponieważ w głównej metodzie `computeForNetwork` tylko ustawiamy wątki i je joinujemy, podczas gdy całe obliczenia wykonują się w wątkach. W wątkach dominuje oczekiwanie na barierę, ponieważ wątki po skończeniu jakiegoś etapu muszą poczekać na inne wątki by przejść zrobić następny etap, bądź czekają na lidera (wątek odpowiedzialny za całą resztę, w kodzie zaznaczone że to jest wątek o id równym 0), który tworzy preprocessing, bądź sprawdza wynik z odpowiedziami.