

Moje zadanie wykonałem na putty, zatem moja liczba procesorów logicznych jak i fizycznych jest równa 4 i nie mogłem wykonać wszystkich 3 eksperymentów zapisanych w zadaniu.

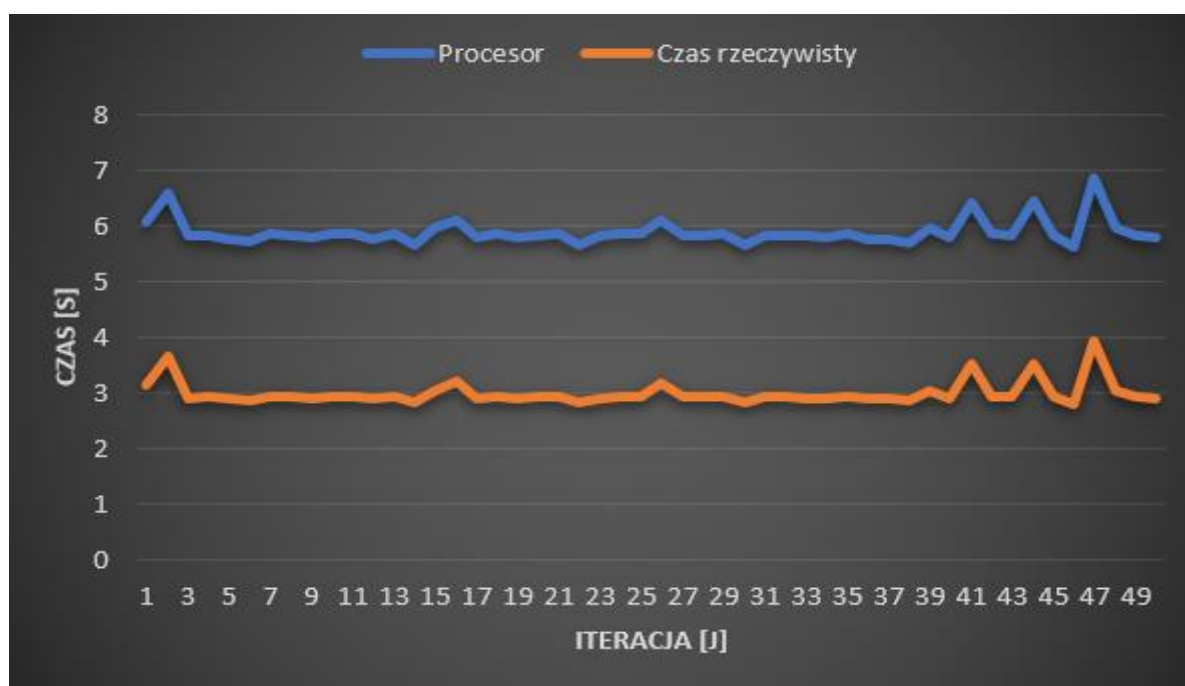
Liczba wątków i rdzeni: 4

| Kod: | Czas przetwarzania na procesorach sekwencyjnego: | Czas przetwarzania na procesorach równoległego: | Czas trwania obliczeń sekwencyjnych: | Czas trwania obliczeń równoległe: | Przyspieszenie: |
|------|--|---|--------------------------------------|-----------------------------------|-----------------|
| PI 1 | 9.250329   | --  | 9.262945                             | --                                | --              |
| PI 2 | 9.235898   | 9.141371  | 9.262408                             | 2.460498                          | 3.764           |
| PI 3 | 9.243321   | 227.724559                                      | 9.252329                             | 57.531186                         | 0.161           |
| PI 4 | 8.721350   | 8.880388  | 8.733892                             | 2.241296                          | 3.897           |
| PI 5 | 8.797293   | 8.897276  | 8.809750                             | 2.233498                          | 3.944           |
| PI 6 | 8.728044   | 9.430517  | 8.737214                             | 3.090640                          | 2.827           |

Liczba wątków: 2

| Kod: | Czas przetwarzania na procesorach sekwencyjnego: | Czas przetwarzania na procesorach równoległego: | Czas trwania obliczeń sekwencyjnych: | Czas trwania obliczeń równoległe: | Ratio: |
|------|--|---|--------------------------------------|-----------------------------------|--------|
| PI 1 | 9.250329   | --  | 9.262945                             | --                                | --     |
| PI 2 | 8.834548   | 8.924928  | 8.835472                             | 2.237025                          | 3.950  |
| PI 3 | 8.986689   | 87.758001                                       | 8.986695                             | 43.895145                         | 0.205  |
| PI 4 | 8.720074   | 8.809097  | 8.726152                             | 4.408371                          | 1.979  |
| PI 5 | 8.743338   | 8.797840  | 8.754957                             | 4.415465                          | 1.983  |
| PI 6 | 8.697036   | 8.872371  | 8.714387                             | 4.456754                          | 1.955  |

Zadanie PI7:



Krótszy czas (słabo widoczne na wykresie) znalazły się na iteracji 6, 14, 22, 30, 38, 46. Co oznacza, że różnica pomiędzy kolejnymi (cyklicznymi) czasami wynosi 8. Zatem długość linii pamięci wynosi  $8B(\text{słowo typu double}) \cdot 8 = 64B$ .

PI1:

Jest to kod sekwencyjny, do którego się odnosimy w następnych kodach.

PI2:

Zrównoleglenie przyspiesza obliczenia jednak wynik jest niepoprawny ze względu na wyścig spowodowany współdzieloną zmienną sum, która jest nadpisywana przez wszystkie wątki w programie. Czas dla wersji z połową wątków nie różni się znacząco z tego względu iż każdy wątek ma swój rdzeń (4 wątki na 4 rdzeniach) oraz przez to, że w tym programie nie występuje synchronizacja (każdy wątek w swoim czasie nadpisuje zmienną sum) zatem ich zmniejszenie nie koreluje ze wzrostem wydajności.

PI3:

Dyrektywa `atomic` poprawia działanie programu (wynik jest poprawny) jednak znacząco spowalnia obliczenia ze względu na jej działanie (jest ona swoistym semaforem pozwalająca na zapisywanie i odczytywanie zmiennej sum tylko przez jeden wątek- 4 wątki a zatem 1 milionach prób zapisu i odczytu zmiennej sum i każda próba blokuje pozostałe).

Czas dla dwóch wątków jest mniejszy (spowodowany 1 milionami zapisów zmiennej sum ale „lepszym” zrównolegleniem, ponieważ blokowany jest tylko jeden wątek a nie 3).

PI4:

Jest to rozszerzenie kodu PI3 o dodanie zmiennych lokalnych „`local_sum`” przez co jedynym momentem spowolnienia spowodowanej dyrektywą `opm atomic` jest dodanie do zmiennej globalnej sum sum lokalnych (na samym końcu przetwarzania). Czasy są szybkie i wynik jest poprawny. Przy dwóch wątkach czas jest 2 razy dłuższy (ponieważ 2 wątki muszą wykonać 1 milion iteracji tak samo jak 4 wątki).

PI5:

Jest to dokładnie ten sam kod co w przypadku kodu PI4 jednak przy użyciu narzędzia `reduction`, który właśnie dokładnie polega na utworzeniu lokalnych zmiennych sum a następnie na końcu przetwarzania zsynchronizowanie zmiennych. Przy dwóch wątkach czas jest 2 razy dłuższy (ponieważ 2 wątki muszą wykonać 1 milion iteracji tak samo jak 4 wątki).

PI6:

Wątki pracują na jednej tablicy, która prawdopodobnie w całości nie jest załadowana do pamięci cache. Zatem jeśli, któryś z wątków, będzie próbował odwołać się do komórki pamięci nie znajdującej się w pamięci cache, procesor będzie musiał ją przywołać. Z kolei inny wątek będzie próbował dostać się do danych, które uprzednio znajdowały się w pamięci, a teraz zostały zastąpione i procesor znów musi przywołać poprzednią linię pamięci. Jest to zjawisko `false sharingu` - wynik poprawny jednak czas stosunkowo wolniejszy od poprzednich wersji. Przy dwóch wątkach czas jest około 2 razy dłuższy (ponieważ 2 wątki muszą wykonać 1 milion iteracji tak samo jak 4 wątki).