

Sprawozdanie przetwarzanie równoległe OpenMP

Sprzęt:

Procesor - AMD Ryzen 5 4600H with Radeon Graphics (6 rdzeni, 12 procesorów logicznych)

System operacyjny - Ubuntu

PI1.

Czas pracy procesorów	Uptyw czasu rzeczywistego	Liczba iteracji	Wynik
1.395016	1.395494	10^9	3.141592653590

PI2.

Liczba wątków	Czas pracy procesorów	Uptyw czasu rzeczywistego	Przyśpieszenie	Liczba iteracji	Wynik
6	2.074876	0.463947	3.756	10^9	0.626407509031
12	1.568427	0.349484	4.987	10^9	0.332564930202

Zmienne prywatne: zmienna „i” jest prywatna dla każdego wątku.

Zmienne współdzielone: zmienne „sum” i „x” (zmienna „sum” i „x” są współdzielone do odczytu jak i zapisu)

Przyczynami niepoprawnych wyników najprawdopodobniej jest to, że wątki mogą równocześnie modyfikować zmienne „sum” i „x”

Ocena liczby dostępów do pamięci: każdy wątek może równoległe aktualizować zmienne współdzielone bez oczekiwania na inne i bez oczekiwania na synchronizację.

PI3.

Liczba wątków	Czas pracy procesorów	Uptyw czasu rzeczywistego	Przyśpieszenie	Liczba iteracji	Wynik
6	153.460727	26.371609	0.053	10^9	3.141592653590
12	177.927164	29.951841	0.046	10^9	3.141592653590

Dyrektywa „#pragma omp atomic” jest używana do zapewnienia atomowości pojedynczych operacji które aktualizują zmienne współdzielone między wątkami, dzięki temu operacja taka nie może zostać przerwana.

Atomowość na poziomie wątku można także zapewnić dyrektywą „pragma omp critical” jest ona bardziej ogólna i może być stosowana do większych bloków kodu co spowoduje jednak jeszcze dłuższe czasy oczekiwania.

Ocena liczby dostępów do pamięci: przez to że każda operacja aktualizacji zmiennej „sum” ma przebiegać atomowo liczba dostępów do pamięci się zwiększy.

PI4.

Liczba wątków	Czas pracy procesorów	Uptyw czasu rzeczywistego	Przyśpieszenie	Liczba iteracji	Wynik
6	1.888387	0.360609	4.262	10 ⁹	3.141592653590
12	1.329417	0.258924	5.936	10 ⁹	3.141592653590

Ocena liczby dostępów do pamięci: liczba dostępów do pamięci względem poprzedniego punktu się zmniejszy gdyż operujemy tym razem na zmiennej prywatnej do zapamiętywania wyniku co robimy równolegle synchronizując go dopiero po wyjściu z pętli co zapewnia nam lepsze czasy jak i poprawny wynik.

PI5.

Liczba wątków	Czas pracy procesorów	Uptyw czasu rzeczywistego	Przyśpieszenie	Liczba iteracji	Wynik
6	1.595020	0.352526	4.227	10 ⁹	3.141592653590
12	1.328090	0.250030	5.960	10 ⁹	3.141592653590

Wyniki tego eksperymentów są praktycznie identyczne jak w punkcie 4.

PI6.

Liczba wątków	Czas pracy procesorów	Uptyw czasu rzeczywistego	Przyśpieszenie	Liczba iteracji	Wynik
6	22.780066	4.127149	0.356	10 ⁹	3.141592653590
12	13.078508	2.330870	0.63	10 ⁹	3.141592653590

PI7.

Podczas przeprowadzania eksperymentu można było dostrzec że krótsze czasy obliczeń występowały w równych odstępach (co 8 iteracji w przykładzie w którym używamy słów z tablicy znajdujących się obok siebie). Słowo w naszym eksperymencie jest typu double który ma wielkość 8 bajtów. Obliczenie długości linii pamięci „l” będzie równe: $l = 8 * 8B = 64B$.