**SPRAWOZDANIE**

Przedmiot: Programowanie komputerów wielordzeniowych

Laboratorium: 3

Student: Artur Ziemba

Grupa: C

Celem sprawozdania było sprawdzenie czasu oraz sposobu działania programu generującego fraktal w zależności od sposobu podziału iteracji przy użyciu języka c lub c++ oraz biblioteki Openmp przy użyciu klauzury schedule.

Wyniki zostały uzyskane na komputerze z systemem Windows 7 (64bit), 16GB RAM oraz procesorze Intel(R) Core(TM) i7-7700K CPU 4.20 GHz, 4 korowym, (8 procesorów logicznych). Do kompilacji programu wykorzystano kompilator g++. Wygenerowane obrazy, rozmiaru 1600x1600 pixeli, zostały stworzone przez 8 wątków. Każdy test został powtórzony 50 razy, a uzyskane wyniki czasowe prezentują wartość średnią.

1. Współdzielenie zmiennych między wątkami

Aby algorytm tworzenia fraktala mógł zostać wykonany poprawnie na wielu wątkach musimy zapewnić odpowiedni podział zmiennych między wątkami. W tym celu używamy klauzur shared oraz private, które określają odpowiednio jakie zmienne są współdzielone oraz prywatne.

1. Opis sposobów rozdziału iteracji

Do zarządzania sposobem rozdziału iteracji równoległej funkcji for służy schedule. Jako parametry przyjmuje ona typ strategii oraz opcjonalnie stała liczbę chunk. Dostępne strategie to:

**Static** – strategia domyślna, dzieląca po równo ilość iteracji między wątkami (ilość iteracji dla określonego wątku = ilość wszystkich iteracji / ilość wątków). Jeśli podamy wartość chunk, każdy wątek będzie wykonywać chunk iteracji, aż do wyczerpania całkowitej ich ilości. (dla 100 iteracji, 4 wątków i chunk równego 15 – 2 wątki wykonają 15x2, jeden 15 oraz 10, a ostatni 15 iteracji). Ilość iteracji jest statycznie dzielona między wątki, rozkładając je po równo bez wiedzy który wykona się pierwszy – przez rozpoczęciem wykonywania pętli. Może dojść do sytuacji gdzie jeden wątek skończy prace wcześniej i mimo że mógłby zabrać kolejną ilość iteracji nie zrobi tego w celu równego podziału iteracji.

**Dynamic** – Działa podobnie jak static ale rozkład kolejnej partii iteracji jest rozkładany na bieżąco. Dla 100 iteracji, 4 wątków i chunk 15 może dojść do sytuacji gdzie 1 wątek wykona 3x15 i 10 iteracji, a pozostałe po 15. Mała ilość chunk gwarantuje wysoka dynamiczność i szybkie działanie.

**Guided** – ilość iteracji dla wątków jest zmienna w trakcie działania programu, automatycznie dobierana tak aby jak najlepiej rozłożyć wszystkie iteracje. Jeśli podamy wartość chunk określamy najmniejszą możliwą ilość iteracji dla wątków.

**Auto** – kompilator wybiera jedno z powyższych estymując najlepszą możliwość

**Runtime** - pozwala zmienić opcje w czasie działanie programu przez użycie zmiennej OMP\_SCHEDULE

1. Wyniki dla poszczególnych sposobów podziału iteracji

|  |  |  |  |
| --- | --- | --- | --- |
| Tryb  Chunk | Static | Dynamic | Guided |
| 500 | C:\Users\aziemba\Desktop\moje\OpenMP\3\dobre\static-500-0.140226-.png  Czas: 0.140226 sec | C:\Users\aziemba\Desktop\moje\OpenMP\3\dobre\dynamic-500-0.143182-.png  Czas: 0.143182 sec | C:\Users\aziemba\Desktop\moje\OpenMP\3\dobre\guided-500-0.142635-.png  Czas: 0.142635 sec |
| 100 | C:\Users\aziemba\Desktop\moje\OpenMP\3\dobre\static-100-0.0444141-.png  Czas: 0.0444141 sec | C:\Users\aziemba\Desktop\moje\OpenMP\3\dobre\dynamic-100-0.0416591-.png  Czas: 0.0416591 sec | C:\Users\aziemba\Desktop\moje\OpenMP\3\dobre\guided-100-0.0428568-.png  Czas: 0.0428568 sec |
| 50 | C:\Users\aziemba\Desktop\moje\OpenMP\3\dobre\static-chunk-50-0.029402-.png  Czas: 0.029402 sec | C:\Users\aziemba\Desktop\moje\OpenMP\3\dobre\dynamic-50-0.0269659-.png  Czas: 0.0269659 sec | C:\Users\aziemba\Desktop\moje\OpenMP\3\dobre\guided-50-0.042769-.png  Czas: 0.042769 sec |

|  |  |  |  |
| --- | --- | --- | --- |
| 10 | C:\Users\aziemba\Desktop\moje\OpenMP\3\dobre\static-chunk-10-0.0288804-.png  Czas: 0.02888804 sec | C:\Users\aziemba\Desktop\moje\OpenMP\3\dobre\dynamic-10-0.0244079-.png  Czas: 0.0244079 sec | C:\Users\aziemba\Desktop\moje\OpenMP\3\dobre\guided-10-0.0441393-.png  Czas: 0.0441393 sec |
| Nie ustawiono | C:\Users\aziemba\Desktop\moje\OpenMP\3\dobre\8-watek-private-and-share-(static-default)-0.066403-.png  Czas: 0.066403 sec | C:\Users\aziemba\Desktop\moje\OpenMP\3\dobre\dynamic default 0.0249977-.png  Czas: 0.024359 sec | C:\Users\aziemba\Desktop\moje\OpenMP\3\dobre\guided-0.0430087-.png  Czas: 0.0430087 sec |

|  |  |
| --- | --- |
| Jednowątkowo | Auto |
| C:\Users\aziemba\Desktop\moje\OpenMP\3\dobre\rownolegly-1-watek0.160882-.png  Czas 0.160882 sec | C:\Users\aziemba\Desktop\moje\OpenMP\3\dobre\auto-0.0660879-.png  Czas: 0.0660879 sec |

1. Podsumowanie

Z przeprowadzonych badań wynika, że w celu osiągnięcia najlepszej wydajności należy odpowiednio skonfigurować wartość chunk dla dynamicznego typu podziału iteracji. Jeśli nie jest możliwe wyliczenie najlepszej wartości możemy użyć strategi guided. Z pewnością należy uważać aby nie ustawić zbyt wysokiej wartości, ponieważ możemy zbytnio obciążyć któryś z pracujących wątków.