Analiza Wydajności Mnożenia Macierzy z Wykorzystaniem Wątków

Cel zadania

Głównym celem zadania było wykonanie mnożenia macierzy z wykorzystaniem C++11 Threads. Należało dobrać odpowiedni wymiar macierzy, który w tym przypadku wynosił 3000 wierszy na 3000 kolumn. Macierze należało zadeklarować poprzez pamięć dynamiczną liniową, a następnie podzielić liczbę iteracji wykonywanych podczas mnożenia pomiędzy wątki. Ze względu na parametry procesora badania wykonano na 1, 2, 4 oraz 8 wątkach. Pomiar czasu wykonania należało wykonać dla wersji równoległej oraz wersji równoległej z macierzą transponowaną.

Wyniki

Liczba wątków	*Wersja równoległa*	*Wersja równoległa z BT*
1	*130.03s*	*57.48s*
2	*67.18s*	*35.53s*
4	*43.97s*	*23.43s*
8	*33.72s*	*18.28s*

Specyfikacja techniczna procesora

AMD Ryzen 5 2500U	
Szybkość podstawowa	*2.00 GHZ*
Rdzenie	*4*
Procesory logiczne	*8*
Architektura	*Zen*
Pamięć podręczna poziomu 1	*384KB*
Pamięć podręczna poziomu 2	*2.0 MB*
Pamięć podręczna poziomu 3	*4.0 MB*

Wnioski

Podczas badania pomiarów czasu wykonania z wykorzystaniem wielowątkowości w C++ w celu wymnożenia macierzy o dużych wymiarach można wysunąć kilka wniosków. Korzystanie z wielowątkowości znacznie polepsza czas wykonania obliczeń na macierzach. Użycie wątków pozwala na wykonywanie obliczeń niezależnie, dzięki czemu w jednej chwili może być wykonywane kilka obliczeń. Umożliwia to wykorzystanie pełnego potencjału procesora. W tabeli z wynikami można prześledzić jak wraz z dodawaniem coraz większej ilości wątków, zmniejsza się czas wykonania obliczeń. Ten przykład pokazuje, że obliczenia można znacznie skrócić pod warunkiem dodawania kolejnych wątków. W badaniu przeprowadzono obliczenia dla wersji równoległej oraz wersji równoległej z wykorzystaniem macierzy transponowanej. Po analizie wyników można zauważyć, że czas wykonania obliczeń jest znacznie krótszy, jeżeli przy wykonywaniu mnożenia wykorzystywana jest macierz transponowana. Dzieje się tak, ponieważ po wykonaniu transpozycji kolumny macierzy stają się wierszami, a wiersze kolumnami. Przez to dostęp do elementów macierzy jest bardziej efektywny i pozwala to na wykonanie wszystkich operacji na konkretnym wierszu lub kolumnie. W przypadku obliczeń dla macierzy bez transpozycji odbywa się to w sposób mniej efektywny i bardziej rozproszony, ponieważ, żeby obliczyć jedną wartość macierzy wynikowej, należy uzyskać dostęp do całego wiersza pierwszej macierzy i całej kolumny drugiej macierzy.