

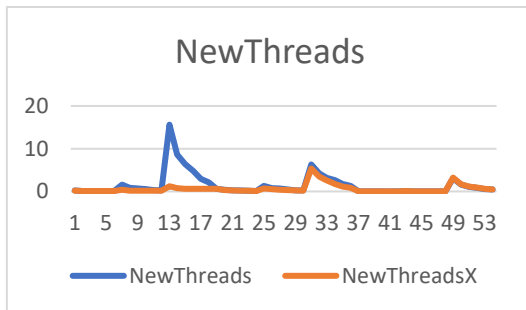
Wiktor Grzankowski, nr indeksu wg429211

Raport z rozwiązania trzeciego zadania

zaliczeniowego z programowania współbieżnego

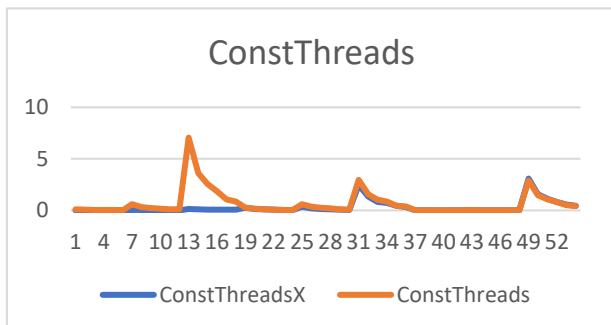
Zadanie polegało na zaimplementowaniu tzw. zespołów, które obliczały liczbę kroków potrzebnych do zakończenia wykonywania obliczeń dla wzoru Collatza podanego w jego hipotezie.

Przyjrzyjmy się kolejnym zespołom i im odpowiednikom X, oraz temu, jak radziły sobie z wykonywaniem obliczeń.



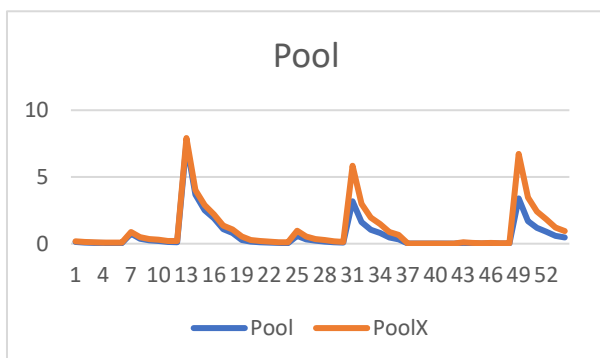
TeamNewThreads – tworzy nowy wątek dla każdego jednego obliczenia liczby w hipotezie Collatza, ale

jednocześnie nie więcej, niż dozwolona liczba ustalana przy wywołaniu pracy zespołu. Problemem tego zespołu jest koszt tworzenia niepotrzebnie dużej liczby wątków, co trwa znacząco dłużej w porównaniu z czasem dokonywania faktycznych obliczeń. Naturalnie dla pracy z maksymalnie jednym wątkiem naraz jest zauważalnie wolniejsze od sekwencyjnego wykonania zadania. Zgodnie z oczekiwaniami, przy zwiększeniu liczby dopuszczanych wątków, w większości przypadków prędkość wykonania rośnie niemal liniowo. Łączny czas wykonania wszystkich testów z *main.cpp* zajmuje na students około 127 sekund.



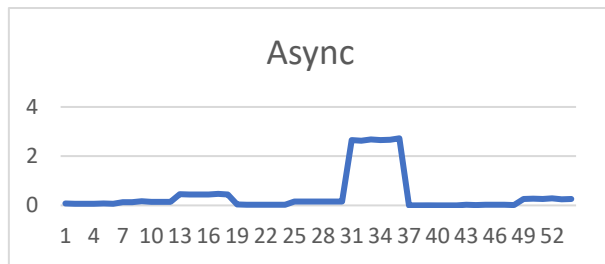
TeamConstThreads – tworzy z góry określoną liczbę wątków, które mają wcześniej przydzielone konkretne podzadania do wykonania. Zespół wykonuje zadanie bardzo szybko w każdym z rozpatrywanych przypadków. W sytuacji, gdy działa tylko jeden wątek, zespół działa sekwencyjnie z niewielkim

dodanym kosztem na obsługę jednego wątku. Zwiększenie liczby dopuszczanych wątków również powoduje przyspieszenie programu. Wykonanie wszystkich testów z *main.cpp* zajmuje na students około 64 sekund.



TeamPool – wykorzystuje pulę wątków zaimplementowaną w załączonej do programu biblioteki. Zespół działa bardzo podobnie do zespołu ConstThreads. Ponownie zwiększenie dopuszczanej liczby wątków zauważalnie przyspiesza wykonywanie programu. Wykonanie wszystkich testów załączonych w *main.cpp*

zajmuje na students około 67 sekund.



TeamAsync – wykorzystuje mechanizm `std::async`. Nie korzysta z ręcznie tworzonych wątków, ani z ich puli. Zespół działa bardzo dobrze dla najbardziej skomplikowanych obliczeń, w szczególności jest najlepszym zespołem dla ostatniego testu (*long number 23*).

O wiele mniej efektywnie działa na prostych testach, np. *same number 2*. Wykonanie wszystkich testów zajmuje około 27 sekund, przy czym jest uruchamiany mniejsza liczbę razy. Przy przeskalowaniu liczby wywołań zespołu do takiej, jak pozostałych zespołów, zajmuje około 90 sekund.

TeamNewProcesses – tworzy nowy proces dla każdego z obliczeń, przy czym nie więcej, niż z góry dopuszczona liczba. Zespół jest zdecydowanie najwolniejszy niezależnie od zadania. Tworzenie procesów, plików tymczasowych i ich obsługa zajmuje nieporównywalnie więcej czasu, niż same obliczenia. W szczególności, rozpatrując zespół w ramach zadań z *main.cpp*, tworzenie blisko 3000 procesów, aby te każdorazowo dla *same number 2* dokonywały tylko jednego dzielenia całkowitoliczbowego, jest bardzo bezsensowne. Rozwiązanie stosowane przez zespół jest wybitnie nieefektywne i wykonanie wszystkich testów na maszynie students trwa blisko godziny.

Każdy z zespołów, za wyjątkiem TeamNewProcesses, miał odpowiadający mu zespół X. Przed wykonaniem pętli i obliczeniu wyniku, program sprawdza, czy ten wynik nie jest już znany. Zespoły X spisują się lepiej od swoich standardowych odpowiedników na testach z rodziny *same number*, gdyż tylko jeden raz wykonywane są faktyczne obliczenia. Największa różnica jest obserwowalna dla *same number 23*. Natomiast dla testów z rodziny *short number* zespoły X są wolniejsze, gdyż mniej opłacalna jest obsługa struktury współdzielonej, niż dokonanie samych, prostych obliczeń. Dla testów *long number* różnice między zespołami X a ich odpowiednikami są niewielkie.

W rozwiązaniu nie zostały zaimplementowane zespoły NewProcessesX, ConstProcesses oraz ConstProcessesX.

Program wykonywany jest w wersji 20 języka C++. Uargumentowane jest to wygodą z korzystania z nowszych funkcji oraz brakiem zauważalnych wad takiego wyboru.

Do raportu załączony jest plik txt z wynikami wszystkich drużyn. Program wykonywany był na komputerze HP Pavilion 15 o procesorze Intel Core i5-8300H, 16 gb RAMu i 4 rdzeniach, system: Linux 5.10.93-1-MANJARO #1 SMP PREEMPT x86_64 GNU/Linux.

