

Sprawozdanie z badania działania pętli for w OpenMP

Aleksandra Pasternak

Michał Sokół

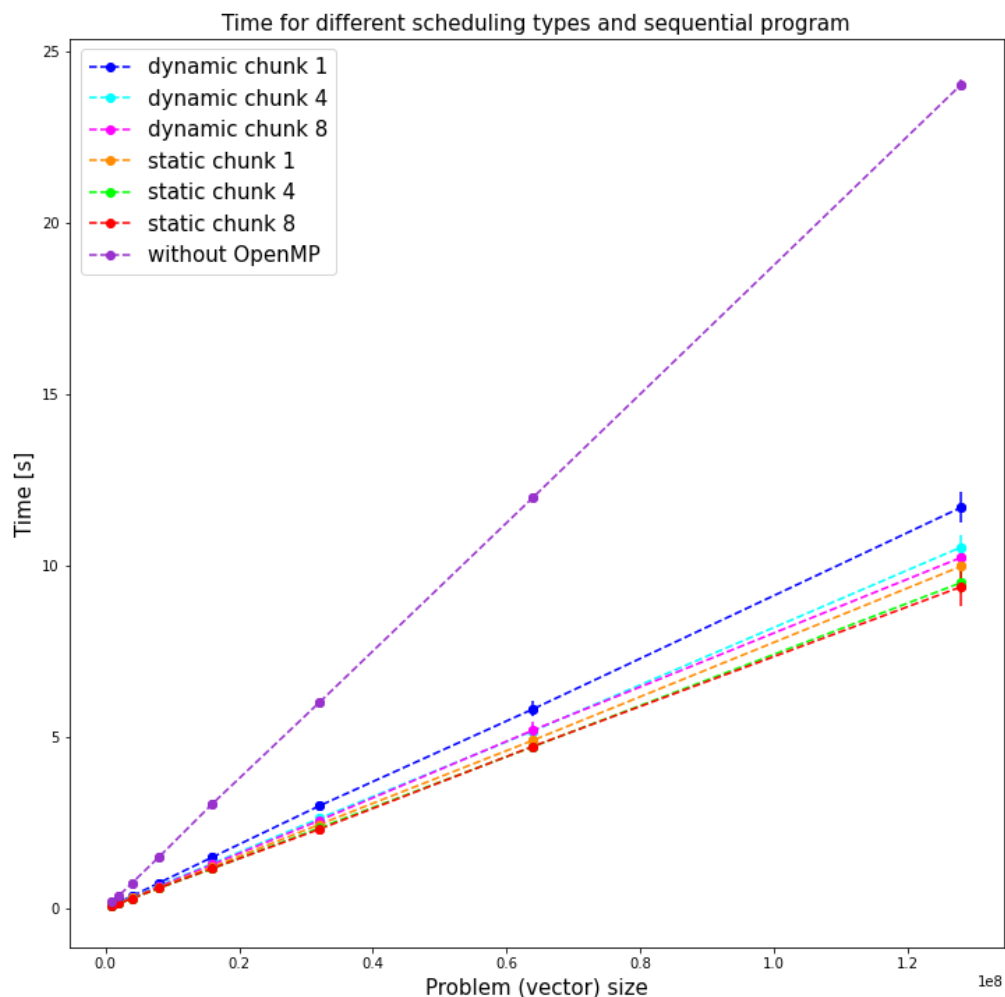
1. Konfiguracja

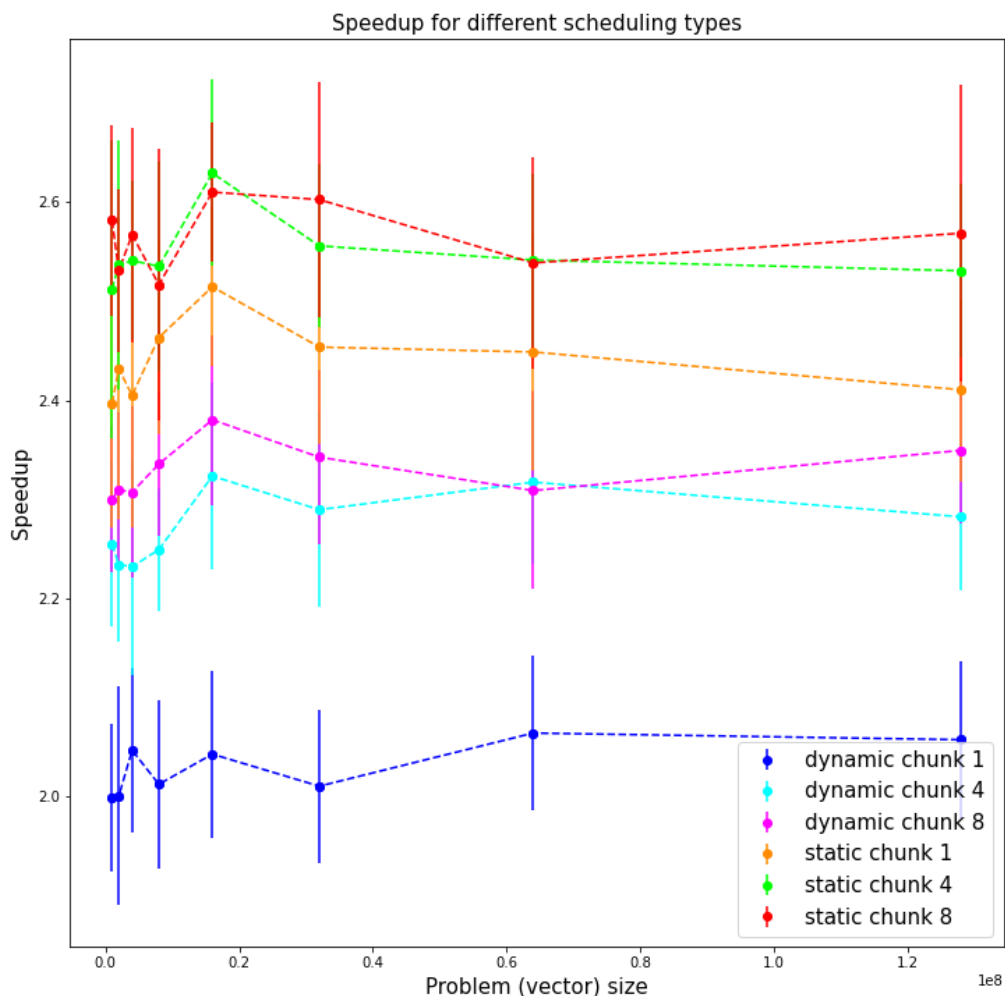
Testy wykonano w środowisku vCluster (vnode-04). Liczby losowano przy pomocy generatora std::mt19937 i zapisywano w std::vector.

Rozmiary wektora zaczynały się od miliona elementów, a kończyły na 128 milionach.

Porównano statyczne oraz dynamiczne schedulowanie z rozmiarami chunków równymi: 1, 4 oraz 8. Do obliczenia przyspieszenia wykonano także pętlę for w wersji sekwencyjnej bez użycia OpenMP.

2. Wyniki





3. Obserwacje

Na wykresie czasu widać, że zrównoleglenie obliczeń z pomocą OpenMP niezależnie od wybranej konfiguracji przyspieszało wykonanie programu w porównaniu do wersji sekwencyjnej.

Wykres przyspieszenia pokazuje, że dynamiczny podział iteracji pomiędzy wątki uzyskał gorszy wynik niż statyczna wersja.

Najlepsze wyniki dla obu sposobów schedulowania uzyskano przy rozmiarze chunka równym 8 i były one zbliżone do tych uzyskanych z chunkiem równym 4. Dla 64 milionów podział iteracji na chunki o rozmiarze 4 okazał się średnio lepszy niż chunk równy 8. Gorsze rezultaty uzyskano dla chunka o rozmiarze 1, przy czym większy wpływ wyboru rozmiaru chunka na przyspieszenie można zaobserwować przy podziale dynamicznym, ponieważ różnice w przyspieszeniu między implementacjami z chunkiem 1 oraz 4 są większe.

4. Wnioski

Równoważenie obciążenia oferowane przez dynamiczny podział daje dobre rezultaty przy dużych obliczeniach, jednak przy naszym problemie te korzyści nie są zauważalne, co widać przy porównaniu statycznego i dynamicznego podziału iteracji. Początkowy narzut danych związany ze sprawdzeniem jakie iteracje ma wykonać wątek w implementacji dynamicznej ma w naszym problemie duży wpływ na czas wykonywania. Lepszy w tym wypadku jest stały podział oferowany przez scheduling statyczny.

Obliczenia wykonywane przez poszczególne wątki są krótkie, dlatego bardziej optymalne jest przydzielenie wątkom większych chunków niż 1, jednak przy większych rozmiarach chunków (4 czy 8) te różnice nie są tak widoczne, więc takie rozmiary wydają się optymalne dla problemu.