

Teoria Współbieżności - ćwiczenie

Mateusz Szarek

1 Zadanie

Proszę zaimplementować przy użyciu mechanizmów Executor i Future program wykonujący obliczanie zbioru Mandelbrota w puli wątków. Jako podstawę implementacji proszę wykorzystać kod w Javie. Uwaga: pojedyncze zadanie powinno obliczać podzbiór całego zbioru (np. kilka wierszy). Problem proszę podzielić tak, żeby liczba zadań była rzędu 10 x liczba wątków.

Proszę przetestować szybkość działania programu w zależności od implementacji Executora i jego parametrów (np. liczba wątków w puli). Czas obliczeń można zwiększać manipulując parametrami problemu.

2 Wyniki

Parametry

W celu wykonania ćwiczenia, przyjąłem następujące wartości: wysokość (600), szerokość (800), grid (200x200).

Wykonanie

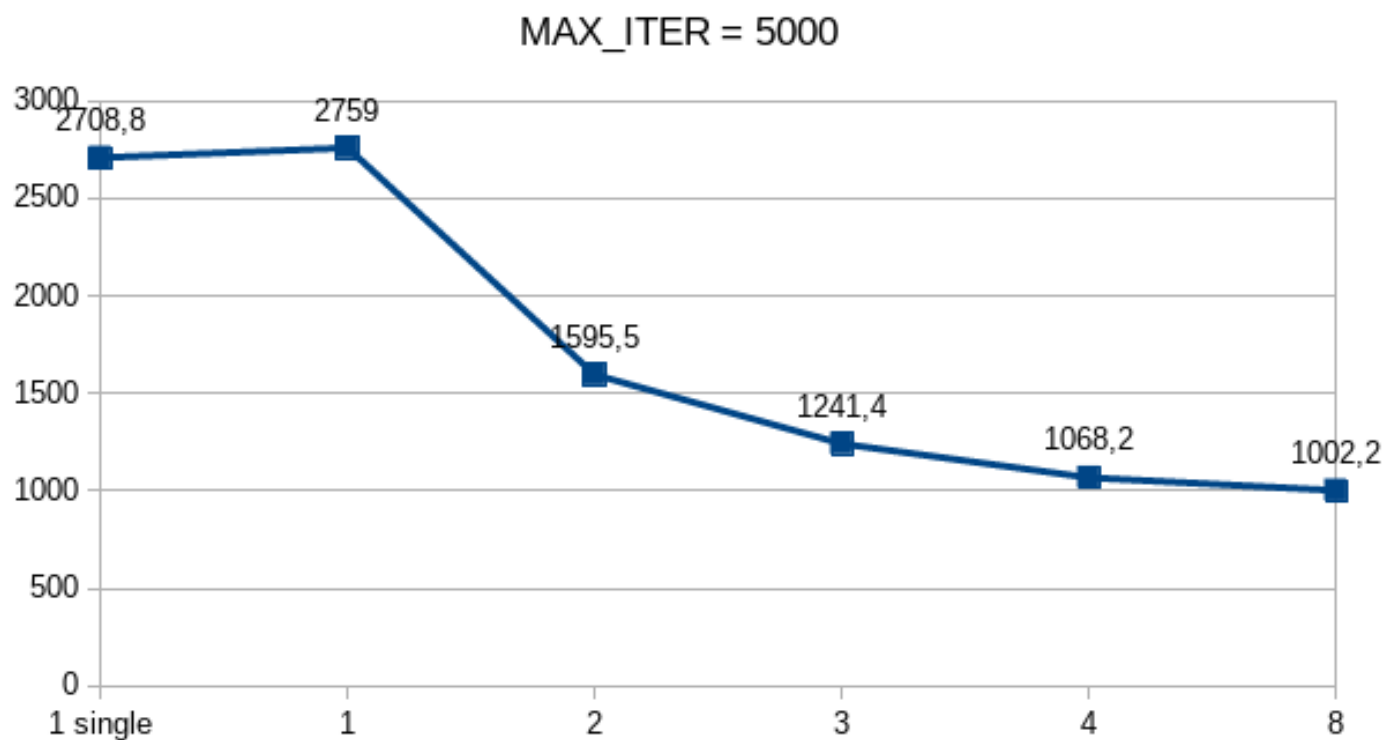
Pomiary wykonywałem na kodzie wykonanym własnoręcznie, testuję czas dla następującej ilości wątków:

1. SingleThreadedExecutor - 1 rdzeń
2. MultiThreadedExecutor - 1 rdzeń
3. MultiThreadedExecutor- 2 rdzenie
4. MultiThreadedExecutor - 3 rdzenie
5. MultiThreadedExecutor - 4 rdzenie
6. MultiThreadedExecutor - 8 rdzeni

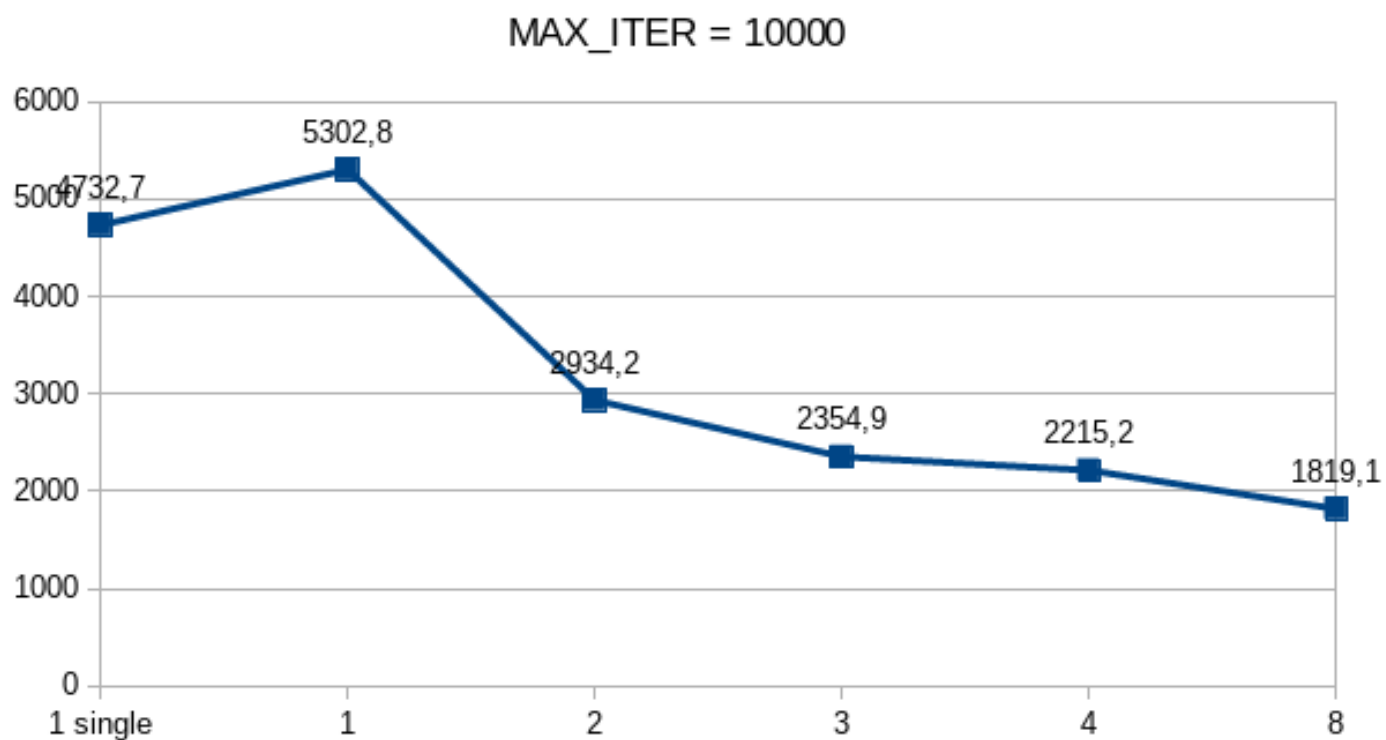
Wszystkie pomiary przedstawiłem na wykresach. Na osi OX znajduje się ilość wątków biorących udział w pomiarach, na osi OY znajduje się czas wykonania. Dla każdej ilości wątków wykonałem po 10 pomiarów. Dane posiadam na lokalnym dysku.

Wyniki

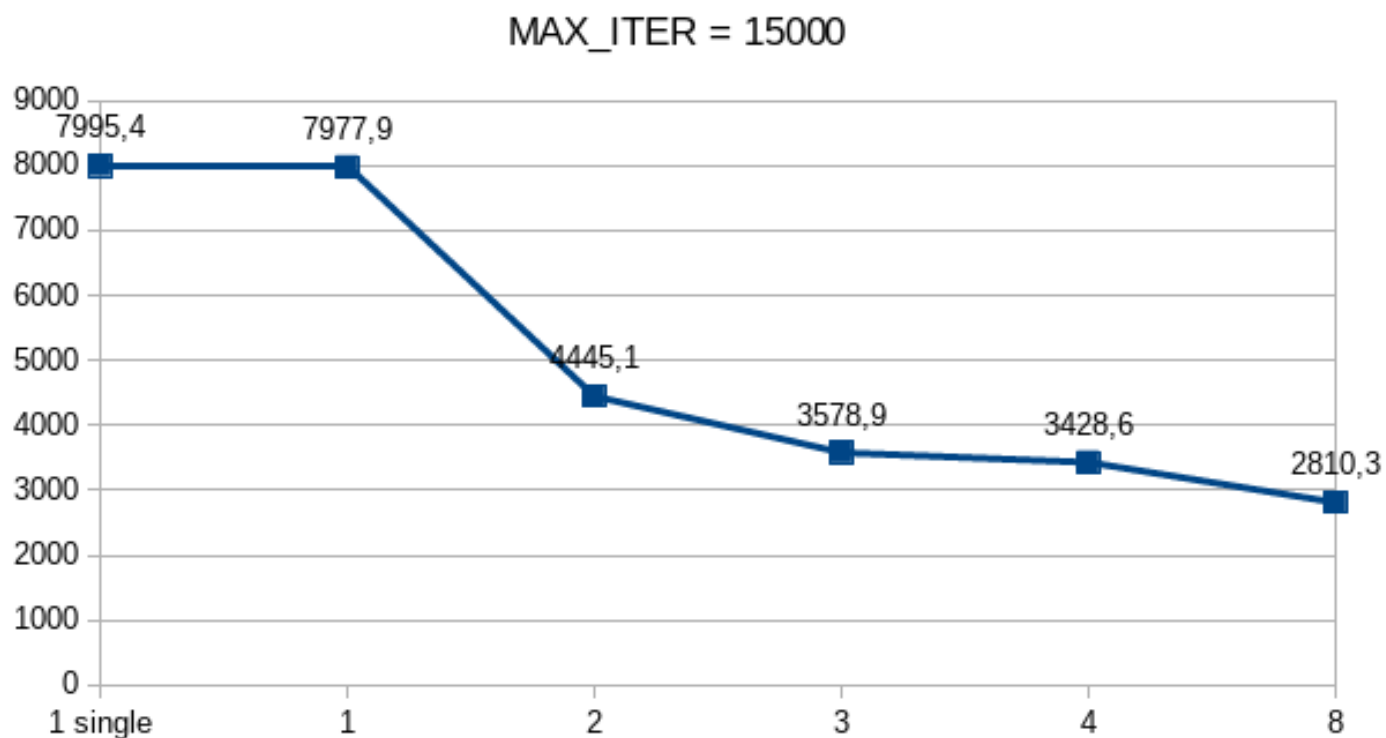
2.1 MAX_ITER = 5000



2.2 MAX_ITER = 10000



2.3 MAX_ITER = 15000



Podsumowanie

Jak widzimy, im więcej pracy do wykonania tym większy zysk otrzymujemy na Współbieżności. Niestety dla mniejszej ilości rdzeni, a także często dla naszego prywatnego komputera, aplikacje współbieżne nie mają sensu być instalowane.