

PW - zadanie zaliczeniowe 3: Problem Collatza

Michał Napiórkowski

11 lutego 2022

1 Środowiska

- Huawei MateBook 13 (2019)
Pop!_OS 20.04 LTS
Intel® Core™ i5-10210U
(4 rdzenie, 8 wątków, 1.60-4.20 GHz, 6MB cache)
RAM 8 GB (DDR3, 2133 MHz)
- maszyna students

2 Metodologia

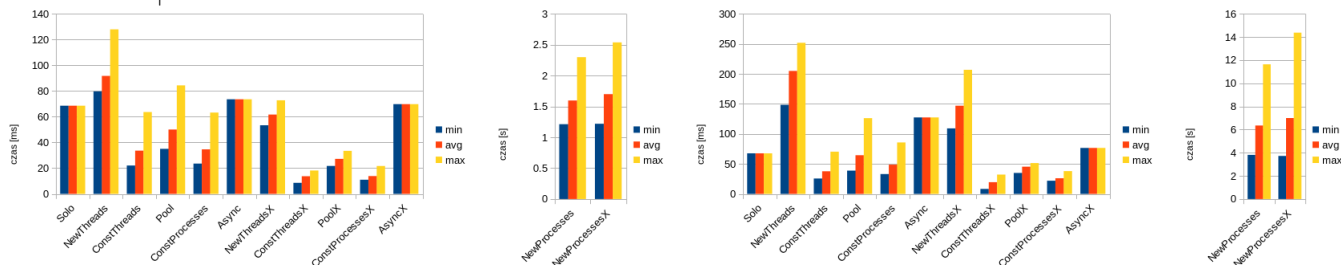
W celu zniwelowania różnic w wynikach przy wielokrotnym uruchamianiu programu, w każdym ze środowisk przeprowadzono po 5 uruchomień, a następnie uśredniono wyniki dla każdego zespołu w danym konkursie.

Poniżej znajdują się wykresy ilustrujące wyniki konkursów. W niektórych przypadkach wykres został rozbity na dwie części, ze względu na znaczną różnicę między TeamNewProcesses i TeamNewProcessesX a pozostałymi zespołami. W każdym z konkursów lewe wykresy dotyczą wyników z mojego laptopa, a prawe z maszyny students.

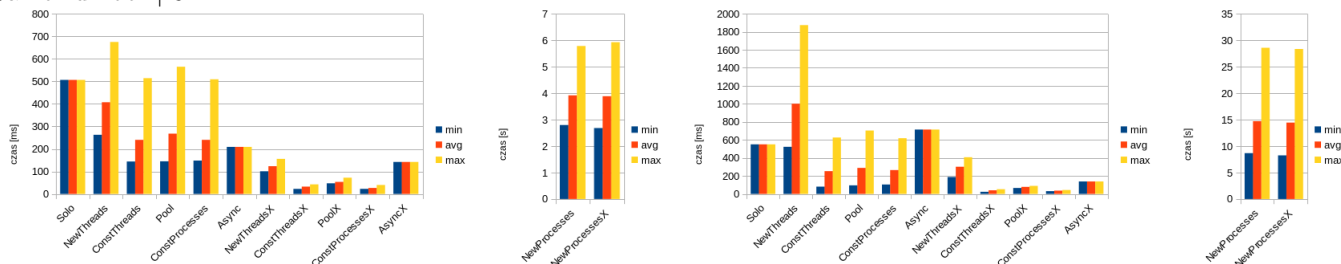
Dla zespołów innych niż TeamSolo, TeamAsync i TeamAsyncX konkursy przeprowadzono w wariantach dla 1, 2, 3, 4, 7 i 10 wątków/procesów. Górny słupek na wykresie wskazuje największy z czasów uzyskanych w tych wariantach, środkowy to średnia arytmetyczna, a dolny to najmniejszy z uzyskanych czasów.

3 Wyniki

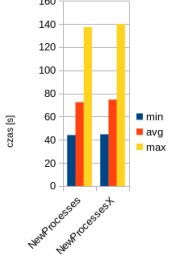
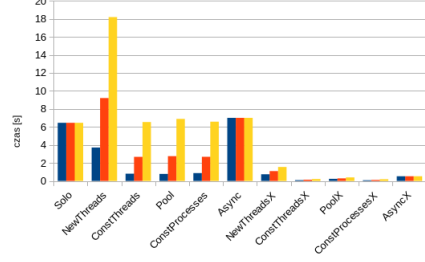
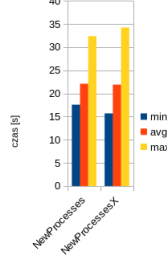
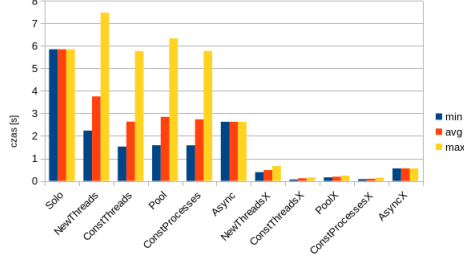
- SameNumber | 2



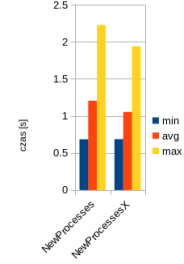
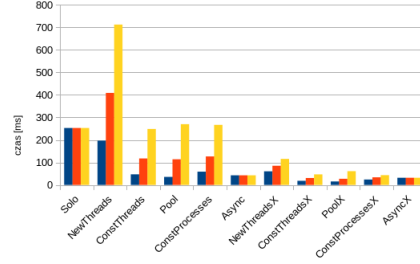
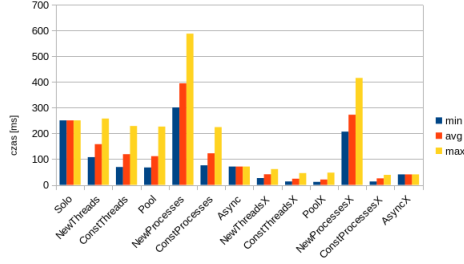
- SameNumber | 5



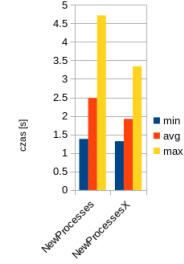
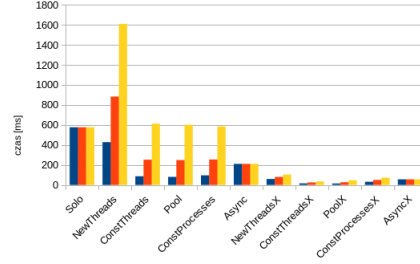
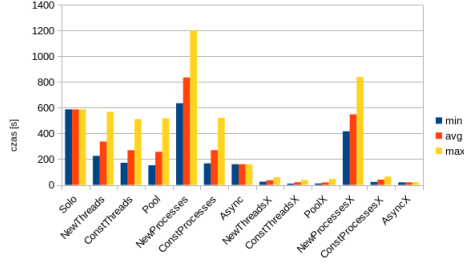
- SameNumber | 23



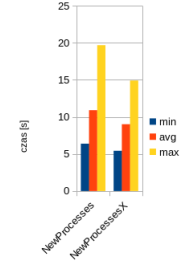
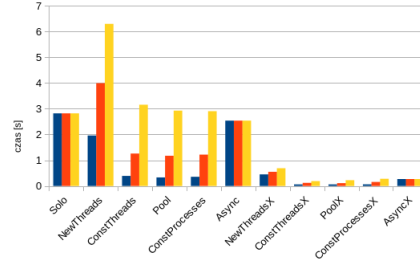
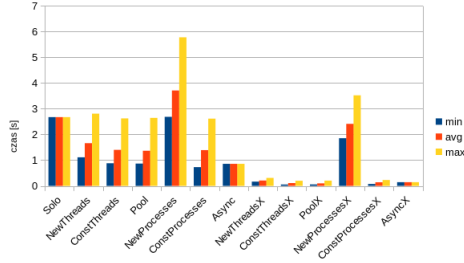
- ShortNumber | 2



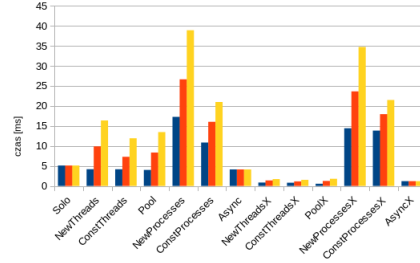
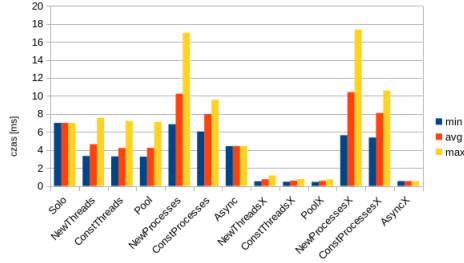
- ShortNumber | 5



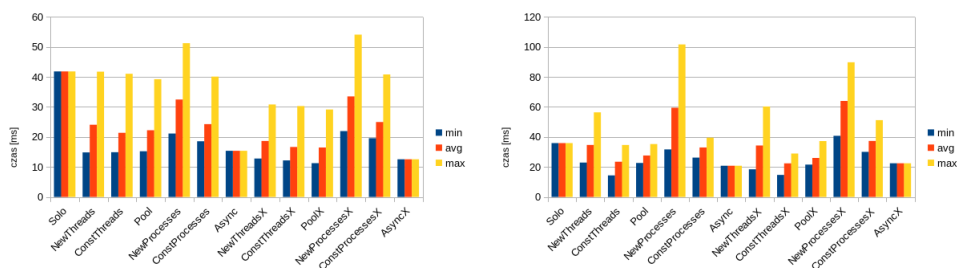
- ShortNumber | 23



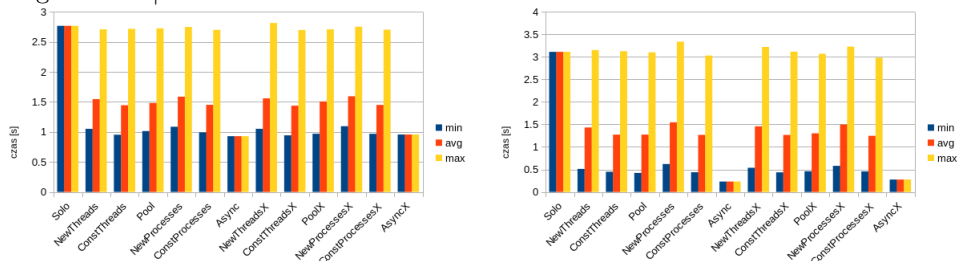
- LongNumber | 2



- LongNumber | 5



• LongNumber | 23



4 Wnioski

Uwaga: kiedy używam wyrażeń takich jak „na początku”, „im dalej” i podobnych, to dotyczą one kolejności konkursów danego rodzaju, czyli wg rosnącego contestId.

• SameNumber

CalcCollatzSoloTimer wskazuje na bardzo dużo bardzo krótkich wywołań.

Konkurs wygrywają ConstThreadsX, ConstProcessesX i PoolX, ale im dalej, tym lepiej w stosunku do reszty radzą sobie też AsyncX i NewThreadsX.

NewProcesses i NewProcessesX są znacznie wolniejsze od pozostałych zespołów. Reszta drużyn (z wyjątkiem NewThreads na maszynie students) jest szybsza lub porównywalna z Solo.

Na maszynie students NewProcesses, NewProcessesX, NewThreads i NewThreadsX są znacznie wolniejsze niż na moim laptopie. Reszta zespołów w obu środowiskach ma zbliżone wyniki.

• ShortNumber

CalcCollatzSoloTimer wskazuje na dużo krótkich wywołań.

Wnioski są podobne do tych z SameNumber, z tą różnicą, że od początku AsyncX i NewThreadsX toczą wyrównaną walkę z ConstThreadsX, ConstProcessesX i PoolX.

• LongNumber

CalcCollatzSoloTimer wskazuje na mało długich wywołań.

Na początku NewThreadsX, ConstThreadsX, PoolX i AsyncX znacznie wygrywają, ale im dalej, tym bardziej wyrównane są wyniki wszystkich zespołów. Nawet NewProcesses i NewProcessesX wreszcie zrównują się z resztą.

Na końcu wszystkie zespoły działają praktycznie tak samo szybko (maksimum na poziomie Solo, minimum na poziomie Async i AsyncX, średnie wyrównane).

Podsumowując, najmniej efektywne okazało się ciągle tworzenie nowych procesów. Przy wielu wywołaniach funkcji calcCollatz dużym usprawnieniem jest spamiętywanie wyników częściowych w pamięci dzielonej między wątkami/procesami. Na maszynie students wielokrotne tworzenie nowych wątków/procesów jest znacznie wolniejsze niż na moim laptopie.