

Założenia projektowe

2021/2022

1 Edycja i zmiany

Edytowano - 19-12-2021

2 Kategorie

A - Projekt należy wykonać zarówno na GPU i CPU oraz porównać czas wykonania zadania, tworząc odpowiednie charakterystyki. Czas wykonania można porównać w zależności od wielu metryk występujących w zadaniu projektowym.

B - Projekt powinien mieć przestrzenną wizualizację graficzną zjawisk/wyniku programu.

C - Do projektu można użyć gotowych bibliotek wspomagających, które realizują dane zagadnienie na CPU.

D - Do projektu można użyć gotowych bibliotek wspomagających rozwiązanie GPU.

E - Do projektu należy dołączyć sprawozdanie. Sprawozdanie zawiera wnioski i przemyślenia grupy/studenta, wykresy oraz dane wymagane z innych podpunktów. Należy skupić się nie tylko na osiągniętych rezultatach, ale także na tym czego nauczyła się grupa realizująca dany projekt wraz z własnymi przemyśleniami odnośnie zasadności użycia układu GPU do realizacji swojego projektu.

F - Wyznaczyć w przybliżeniu złożoności obliczeniowe i przedstawić je na charakterystykach. Dodatkowo można wyznaczyć złożoności pamięciowe.

G - Stworzyć własny generator danych dla zadania projektowego, tworzącego określony lub losowy zbiór danych w zależności od podanych parametrów.

H - Przygotować ustalony zbiór danych lub przypadków (nie ma potrzeby użycia generatora).

I - Wizualizacja wyniku i danych wejściowych projektu z użyciem wykresów i arkuszy.

J - Wizualizacja 2D grafu przed i po wykonaniu zaimplementowanego algorytmu, z zaznaczeniem wyniku pracy programu.

K - Przygotować prosty Api/interfejs w celu sprawdzenia poprawności i przetestowania działania programu.

L - Należy wygenerować taki dataset do zadania, który będzie zbyt wymagający dla algorytmu działającego na CPU. Wyznaczyć tą granicę. Następnie powtórzyć krok dla urządzenia GPU.

M - Projekt powinien działać w czasie rzeczywistym.

Kategoria przyznana grupie projektów przechodzi automatycznie na wszystkie podprojekty. Podprojektem można przypisać nową kategorię, lub pozbyć się kategorii nadrzędnych, pisząc przy tytule np. - A. Jeżeli jest to potrzebne, pod projektami dodano dodatkowe instrukcje.

3 Projekty

3.1 Grupa: Kryptografia : A, E, F, H

Należy wyznaczyć wielkości klucza, dla którego przy zastosowaniu optymalnego algorytmu na CPU, rozszyfrowanie treści staje się praktycznie niemożliwe. Następnie należy przenieść algorytm na urządzenie GPU i wyznaczyć nową górną granicę rozmiaru klucza dla którego nie uda się rozszyfrować wiadomości. Wyznaczyć złożoności obliczeniowe, speedup itp. Wiadomość tworzona jest we własnym zakresie, tak jak implementacja algorytmu. Można użyć dowolnych ataków kryptograficznych np.(Plaintext-Based Attacks).

3.1.1 RSA

Zmieniać należy długość klucza.

3.1.2 Krzywe Eliptyczne

Zmieniać należy długość klucza.

3.1.3 AES : C

Zmieniać należy liczbę cykli dla klucza 128 bitowego. Można spróbować od dwóch cykli.

3.2 Grupa: Silniki gier planszowych : E, F, H, K

Należy stworzyć własny silnik do podanych gier planszowych, który poziomem plasowałby się pośród 10% najlepszych graczy w danej grze. Wykorzystać GPU w celu minimalizacji czasu pracy silnika (przykładowo ustalić czas 5 sec na posunięcie). Opisać strukturę sztucznej inteligencji użytej w projekcie. Nazwy gier zawierają się w tytułach.

3.2.1 Szachy

3.2.2 Go

3.2.3 Warcaby (Kanadyjskie)

3.3 Grupa: AI : E, H

W kategorii sieć neuronowa, grupy mogą zgłosić jako projekt dowolne zagadnienie z obszaru sztucznej inteligencji. Warunkiem jest napisanie własnej sieci działającej na GPU. Dodatkowo sieć musi działać poprawnie dla danego zagadnienia. Można wykorzystać gotowe datasety i architektury.

3.3.1 Sieć neuronowa : D* (W zależności od skomplikowania architektury sieci)

3.3.2 Temat własny - Sieć neuronowa, pisanie utworów bazujących na tekstach znanego artysty : D*

3.3.3 Wyznaczanie minimum globalnych i lokalnych : A, B, F, L

Wybrać kilka algorytmów i porównać ich działanie. Należy przygotować różne popularne powierzchnie sprawdzające poprawność działania algorytmu optymalizującego (np. siódło, pułapki z minimami lokalnymi uniemożliwiającymi dojście do minimum globalnego, itp.).

3.4 Grupa: Symulacje :B, E

3.4.1 Przepływ wody :H

Przygotować kilka zaworów/przeszkód/rur w celu wizualizacji przepływu strumienia. Przedstawić kierunek i prędkość wody w poszczególnych punktach.

3.4.2 Grawitacja cząstek :M, G

Przygotować wizualizację wraz z symulacją w czasie oddziaływania grawitacyjnego na siebie wielu cząstek umieszczonych w przestrzeni. Dodać kilka modyfikacji wzoru na siłę grawitacji w celu porównania rezultatów.

3.4.3 Temat własny - Układ słoneczny :M, H

Przygotować symulację układu słonecznego. Przyjąć, że planety i księżycy są jednorodnymi kulami. Zderzenie ciał niebieskich powoduje zwiększenie się objętości jednej z kuli o objętość drugiej (obiekty mają różną gęstość).

3.4.4 Wizualizacja pola magnetycznego cewki :H,

3.4.5 Temat własny - Symulacja Populacji : -B, G, L*(Pominać krok dla GPU)

Należy przygotować symulację sprawdzającą rozwój danego gatunku. Populacja osobników składa się z ogromnej liczby agentów (tak dużej, by niemożliwa była praktyczna implementacja na CPU). Może istnieć kilka różnych populacji w ekosystemie. Symulacja analizuje rozwój populacji na długim przedziale czasowym. Symulację dzielimy na poszczególne dni, w których agenci podejmują kilka działań i spotykają się w sposób losowy. Zdarzeniami mogą być walka o pokarm, znalezienie partnera, walka z innymi osobnikami, odpoczynek itp. Analizie może być poddana końcowa i startowa wielkość populacji oraz występujące geny. Różni agenci o różnej puli genowej mogą mieć inną szansę, na rozmnażanie się, czy zdobycie pokarmu.

3.4.6 Temat własny - Symulacja rozprzestrzeniania się choroby : -B, G, L*(Pominać krok dla GPU)

Należy przygotować symulację sprawdzającą rozprzestrzenianie się choroby w ludzkiej populacji. Populacja osobników składa się z ogromnej liczby agentów (tak dużej, by niemożliwa była praktyczna implementacja na CPU). Symulacja analizuje rozwój choroby na długim przedziale czasowym. Symulację dzielimy na poszczególne dni, w których agenci podejmują kilka działań i spotykają się w sposób losowy. Przykładowo każdy agent podejmuje dziennie trzy podróże do trzech różnych miejsc, w których spotyka się z innymi agentami. Niektórzy agenci mogą unikać większych skupisk ludzi, jak np. koncerty czy obrzędy religijne. Każdy agent ma procentową szansę na zarażenie się, śmierć w wyniku choroby oraz zarażenie innej osoby. Dodatkowo należy wprowadzić dodatkowe czynniki jak noszenie maseczek czy szczepionki. Umożliwić zmianę parametrów symulacji. Dodatkowo choroba ma procentową szansę na mutację. Mutacje mogą mieć inną śmiertelność i szansę na rozprzestrzenianie się.

3.5 Grupa: Matematyka i optymalizacja : A, E, F, G, I, J, L

Nazwa problemów matematycznych/optymalizacyjnych znajduje się w tytułach.

3.5.1 Komiwojażer

3.5.2 K-center problem

3.5.3 Problem Plecakowy

3.5.4 Nakładanie filtrów obrazów: M, H, -G, -I, -J

Wczytywać obrazy z kamery lub plików z filmami i nałożyć jeden z kilku predefiniowanych filtrów w czasie rzeczywistym. Najlepiej, żeby program obsługiwał jak najlepszą jakość wideo (np. 4K, 60 FPS). Przygotować możliwość odtwarzania kilku filmów w tym samym czasie. Sprawdzić ile obrazów wideo zaimplementowany program jest w stanie obsłużyć w tym samym czasie i przy jakiej jakości obrazu.

3.5.5 Wyznaczanie drogi, Algorytm - A * graf ważony dodatni

3.5.6 Wyznaczanie drogi, Algorytm - Porównanie kilku algorytmów drogi

3.5.7 Wyznaczanie drogi, Algorytm - Algorytmy genetyczne

3.5.8 Drzewo filogenetyczne

Należy odtworzyć drzewo filogenetyczne na podstawie losowo wygenerowanych danych. Każdemu osobnikowi można przypisać litery alfabetu jako reprezentacje poszczególnych genów.

3.5.9 Kolorowanie grafów bi-partyjnych, krawędziowe, interwałowe

Należy zająć się zagadnieniem kolorowania interwałowego krawędziowego grafów bi-partyjnych dla n wierzchołków (w szczególności sprawdzić graf pełny).

3.6 Grupa : Inne

3.6.1 Temat własny - Ruch stronnictw po planszy : A, E, H, I, J, L

Należy wygenerować planszę stanowiącą mapę świata gry (może być to plansza składająca się z pól). W grze występuje kilka frakcji, z których każda może mieć przypisanych kilku agentów. Mając podanych kilku agentów, należy wyznaczyć ich drogę do określonych celów z założeniem, że mają być to drogi możliwie najszybsze i należy unikać agentów wrogich stronnictw jeżeli jest to agent silniejszy. Agenci mogą mieć różną prędkość przemieszczania jak i różną siłę. Agenci startują w tym samym czasie.