Programowanie w CUDA na NVIDIA GPU (PKG) rok akademicki 2023/2024, studenci kierunku Informatyka

Termin wykonania projektu 31.05.2024

Grupy projektowe: projekt jest wykonywany i zaliczany w grupach maksymalnie 2 osobowych. Skład grupy projektowej proszę zgłaszać przez email natychmiast po ustaleniu.

Koncepcja projektu

Celem ćwiczenia jest:

- zapoznanie Studentów praktyczne z zasadami programowania równoległego procesorów kart graficznych (PKG)
- zapoznanie z zasadami optymalizacji kodu dla PKG oraz
- ocena prędkości przetwarzania przy użyciu PKG oraz poznanie czynników warunkujących realizację efektywnego przetwarzania.

W ramach projektu należy:

- przygotować i wyjaśnić przebieg przetwarzania wymaganych wersji programów,
- wykonać wiarygodny test poprawności obliczeń dla danych niejednorodnych polegający na porównaniu wyników obliczeń na karcie graficznej, z obliczeniami wykonanymi za pomocą prostego programu sekwencyjnego,
- wykonać eksperyment obliczeniowy dla przygotowanych kodów z pomiarem czasu przetwarzania dla zadanego zakresu instancji i parametrów uruchomienia,
- wykorzystać uzyskane wyniki obliczeń i miary efektywności (wskazane w opisie projektu/wybrane przez autorów opracowania i kodu) do porównania jakości przetwarzania i udowodnienia, że wymagany zakres projektu został wykonany,
- przygotować sprawozdanie wg opisanych poniżej wymagań.

Różnorodność sprawozdań i dostępność opisu zadania

Ze względu na wąski zakres różnorodności tematów zadań, pewne podobieństwa w realizacji projektu przez poszczególne grupy realizujące ten sam temat są nieuniknione. Ze względu na szeroki zakres docelowego sprzętu, badanych instancji problemu i możliwych rozwiązań implementacyjnych liczę na różnorodne opracowania w ramach poprawnego wyjaśniania przyczyn i uzasadniania uzyskanych wyników. Tą drogą zwracam się z apelem o samodzielną pracę nad zadaniami, gdyż pozwoli ona na zrozumienie zagadnień programowania równoległego na PKG. Proszę odpowiednio wcześnie podjąć pracę nad projektem, aby można ją było wykonać w wymaganym terminie samodzielnie.

Temat projektu i warianty

Problem obliczeń szablonowych: Tablica Dwuwymiarowa TAB[N][N] (o wierszu długości N, słowo tablicy TAB[i][j] jest dostępne jako TAB[i*N+j]). Dla tablicy wejściowej TAB należy wyliczyć tablicę wyjściową OUT[N-2R][N-2R] (gdzie N>2R) zawierającą sumy elementów w „promieniu" R. Każdy element tablicy wyjściowej to suma (2*R+1)*(2*R+1) wartości. Przykładowo dla R=1 OUT[i][j]=TAB[i][j]+ TAB[i][j-1]+ TAB[i][j+1]+ TAB[i-1][j-1]+ TAB[i-1][j]+TAB[i-1][j+1]+ TAB[i+1][j]+ TAB[i+1][j-1]+ TAB[i+1][j+1]

Zadanie praktyczne:

Cechy kodów do przygotowania:

1. Przygotowanie kodu rozwiązującego problem sekwencyjnie (jeden wątek) dla głównego procesora komputera. Obliczenia mają dotyczyć niejednorodnych danych i są niezbędne na potrzeby testów poprawnościowych. W ramach testu poprawności kodu należy po wykonaniu obliczeń przez procesor i kartę graficzną dla typowej instancji problemu porównać parami odpowiadające sobie wyniki obliczeń GPU i CPU w celu sprawdzenia poprawności dostępów do danych w wersji dla GPU.
2. Przetwarzane tablice zawierają słowa zmiennoprzecinkowe typu float.
3. Różne wartości parametrów N, R, k i BS:

a. BS - rozmiar dwuwymiarowej tablicy bloku wątków, **zapewnienie możliwości poprawnego przetwarzania kodu również dla R>BS**,
b. Liczba wyników obliczanych przez jeden wątek – k,
c. Ilość obliczeń rośnie z wartością N i R,
d. Dla k=1 i R=1 należy przeprowadzić obliczenia dla zwiększającej się wielkość N (rozmiar 2 wymiarowej macierzy) do momentu uzyskania w systemie nasycenia obliczeniami dla tablicy o rozmiarze N_nas. Nasycenie obliczeniami występuje wtedy, gdy dalszy wzrost wielkości instancji (w tym przypadku powodowany wzrostem wartości N) nie powoduje wzrostu prędkości przetwarzania. W ramach tego eksperymentu wzrasta wielkość gridu i zajętość multiprocesora.
e. Dla macierzy **większych** od zapewniających nasycenie obliczeniami zbadać wpływ wartości parametrów k i R na prędkość obliczeń (należy przyjmować kolejno wartości k i R wynoszące 2,4,8,16… do momentu uzyskania maksymalnej wartości prędkości obliczeń). Wzrost wartości k i R powoduje adekwatny wzrost wielkości instancji problemu wg wzoru: N(k,R) = (N_nas-2)*k+2*R. W ramach tego eksperymentu wielkość gridu przy stałej wielkości bloku pozostaje stała.
f. Różne wielkości dwuwymiarowego bloku wątków BSxBS : 8x8, 16x16 i 32x32

4. Efektywne korzystanie z danych w **pamięci globalnej** karty – wątki realizują jednocześnie dostępy do sąsiednich elementów (w pamięci globalnej).
5. **Jest to wariant podstawowy kodu** bez użycia pamięci współdzielonej bloku wątków – użycie pamięci podręcznej karty do lokalnego czasowo dostępu do danych.

Pozostałe uwagi

Procedura kernela jest wywoływana w konfiguracji zależnej od wielkości każdego z 2 wymiarów macierzy N, parametru R i wielkości bloku BS; dwuwymiarowy kwadratowy blok wątków składa się z BS*BS wątków.

Parametrami wywołania kernela są:

- adres początku tablicy wejściowej w pamięci globalnej karty,
- adres początku tablicy wyjściowej w pamięci globalnej karty,
- wielkość parametrów N, R i k.

Prędkość przetwarzania to liczba operacji dodawania i mnożenia przypadająca na jednostkę czasu przetwarzania (oczekiwania na wynik). Obliczając prędkość należy uwzględnić operacje realizowane na przetwarzanych danych – liczba operacji jest zależna od wielkości danych – N, R i k, nie należy uwzględniać we wzorze operacji sterujących i operacji transferu danych.

Warianty projektu:

WARIANT 1 (podstawowy) – synchroniczne kopiowanie danych pomiędzy pamięciami - z CPU do GPU przed obliczeniami oraz wyników po obliczeniach z GPU do CPU.

Dokumentacja

Po wykonaniu zadania proszę dostarczyć materiały **POPRZEZ ZAKŁADKĘ PRZESYŁANIA OPRACOWAŃ** kursu PR PROJEKT 2. Należy dostarczyć:

- plik NR_INDEKSU1_NR_INDEKSU2.zip - archiwum z kodami źródłowymi
- plik NR_INDEKSU1_NR_INDEKSU2.pdf - sprawozdanie

Sprawozdanie powinno zawierać:

1. dane autorów (imiona i nazwiska, numery indeksu) i data oddania
2. nazwa przedmiotu i nazwa projektu,
3. **opis użytej karty graficznej** –
   - nazwa modelu karty i użyty w nim układ scalony,
   - nazwa technologii: Fermi, Maxwell, Volta itp,
   - parametr CC – możliwości obliczeniowe,
   - liczba jednostek wykonawczych przeznaczonych do obliczeń ogólnego przeznaczenia: liczba SM, liczba rdzeni, jednostki zmiennoprzecinkowe SP (pojedynczej precyzji) i DP, jednostki całkowitoliczbowe, SFU,
   - wielkości i rodzaje pamięci karty używanej podczas obliczeń,
   - ograniczenia posiadanego przez kartę parametru CC.

4. Opis: sposobu generowania wartości testowych (tablica wejściowa), testowania poprawności obliczeń oraz omówienie wyniku testu poprawności.
5. Opis zakresu zrealizowanego zadania (przeprowadzone analizy w ramach zadania), wyniki eksperymentów badania nasycenia obliczeniami oraz wpływu parametrów k, BS i R prędkość obliczeń.
6. Kluczowe fragmenty kodów kernela z **wyjaśnieniami** dotyczącymi:
    - znaczenia użytych instrukcji i zmiennych,
    - określenie i **uzasadnienie jakości występujących w kodzie dostępów do użytej pamięci** (odwołanie się w wyjaśnieniach do pojęć: łączenia dostępów do pamięci globalnej),
    - opis sposobu określania konfiguracji uruchomienia kernela.
7. **rysunki z opisem** określające:
    - miejsce dostępu i kolejność dostępu do danych realizowane przez poszczególne wątki, bloki i
    - wyznaczane przez wątki i bloki wartości wyników,
8. wzory zastosowane do obliczeń wszystkich prezentowanych miar efektywności przetwarzania wraz z wyjaśnieniem znaczenia tych miar,
9. wymagane wyniki jakości przetwarzania dla poszczególnych zbadanych instancji proszę przedstawić w postaci tabelarycznej (maksymalnie zwarta prezentacja w minimalnej liczbie tabel zawierających wszystkie parametry pomierzone lub wyliczone dla wszystkich wariantów uruchomień kodu dla tej samej wielkości bloku wątków)
10. tabele (i wykresy) należy ponumerować i podpisać w sposób unikalny i jednoznaczny definiujący prezentowaną zawartość,
11. wykres prezentujący zmianę prędkości przetwarzania w funkcji parametrów instancji problemu N,k,R,BS (wynikającą z cech karty graficznej - liczba dostępnych jednostek przetwarzających, wielkość dostępnej pamięci podręcznej)
12. wnioski z wykonanych eksperymentów z uzasadnieniem (użycie właściwie zdefiniowanych pojęć: łączenie dostępów do pamięci globalnej, ilość pracy wątku, liczba bloków wątków) obserwowanych wartości (konieczne czytelne odwołanie we wnioskach do wartości parametrów w tabeli (nr tabeli, kolumna i wiersz tabeli) wyników ze szczegółową informacją jakiego uruchomienia – parametry uruchomienia).

Zaliczanie projektu będzie wymagało wiadomości z zakresu przetwarzania równoległego na PKG, a w szczególności zagadnień związanych z projektem.

Przy tworzeniu kodu dla GPU proszę bazować na wskazanych podczas zajęć przykładowych projektach CUDA.

Podstawowe miary jakości przetwarzania

proponowane do wykorzystania podczas oceny jakości przetwarzania przygotowanych i uruchamianych kodów:
Czas obliczeń – Duration [sekundy] – czas pracy kernela zawierający lub nie czas komunikacji HOST-DEVICEI DEVICE-HOST w zależności od wariantu realizowanego zadania
Prędkość obliczeń [flop/s]
arithmetic intensity [flop/byte] – uwzględnia kod i użycie/brak użycia pamięci współdzielonej

Literatura do przygotowania projektu

Wykłady z przedmiotu z zakresu PKG/GPU
Dokumentacja SDK CUDA https://docs.nvidia.com/cuda/ a w szczególności:
CUDA_C_Programming_Guide
CUDA_C_Best_Practices_Guide
CC karty: https://developer.nvidia.com/cuda-gpus#compute
Przewodniki programowania dla poszczególnych rodzin kart graficznych:
https://docs.nvidia.com/cuda/#programming-guides
Przygotowano 5.05.2024
W przypadku pytań do treści, zadań projektu proszę o pytania na zajęciach lub kontakt przez email.