|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Лабораторная работа № \_\_**1**\_\_**

**Дисциплина Программирование специализированных вычислительных устройств**

|  |  |
| --- | --- |
| **Студент \_Брянская Е.В.\_\_\_\_\_\_\_\_**  **Группа \_ИУ7-41М\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель \_Ковтушенко А.П.** |  |

Москва

2024

**Задание**

Проанализировать зависимость времени работы процесса умножения матриц с помощью технологии CUDA в зависимости от размера матриц, соотношения их сторон и их расположения в памяти.

**Решение**

Рассматриваются 4 варианта расположения матриц в памяти – варианты отличаются транспонированием одной или обеих матриц. Предусмотрены следующие варианты:

|  |  |
| --- | --- |
| Флаг | Описание |
| 0 | «Классическое размещение» обеих матриц |
| 1 | Транспонируется первая матрица |
| 2 | Транспонируется вторая матрица |
| 3 | Транспонируются обе матрицы |

Первая матрица описывается следующими размерами: m x n.

Вторая матрица описывается размерами: n x p.

В ходе анализа производятся следующие замеры времени

1. GPU Total – мс на выделение памяти под матрицы на устройстве, копирование обеих матриц с хоста на устройство, выполнение операции умножения матриц и запись результата графическим процессором
2. GPU Clean – мс на выполнение операции умножения матриц и запись результата графическим процессором

Результаты замеров на квадратных матрицах приведены ниже в таблице:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Способ расположения** | **n** | **m** | **p** | **GPU Clean, мс** | **GPU Total, мс** | **CPU, мс** |
| 0 | 100 | 100 | 100 | 0,0594 | 1,0689 | 4,6420 |
| 0 | 300 | 300 | 300 | 0,1020 | 1,4322 | 72,0870 |
| 0 | 500 | 500 | 500 | 0,2622 | 3,0279 | 456,3160 |
| 0 | 1000 | 1000 | 1000 | 1,5398 | 10,6451 | 6336,6499 |
| 0 | 1500 | 1500 | 1500 | 4,6068 | 16,8870 | 25965,3164 |
| 0 | 2000 | 2000 | 2000 | 10,9951 | 33,6031 | 56354,3711 |
|  | | | | | | |
| 1 | 100 | 100 | 100 | 0,0594 | 0,7476 | 2,5410 |
| 1 | 300 | 300 | 300 | 0,1279 | 1,4473 | 71,3530 |
| 1 | 500 | 500 | 500 | 0,3284 | 3,8305 | 603,2300 |
| 1 | 1000 | 1000 | 1000 | 1,9994 | 9,6311 | 6576,2480 |
| 1 | 1500 | 1500 | 1500 | 6,2986 | 20,4611 | 24917,0449 |
| 1 | 2000 | 2000 | 2000 | 14,9002 | 37,8443 | 56507,4297 |
|  | | | | | | |
| 2 | 100 | 100 | 100 | 0,0748 | 0,7364 | 2,4980 |
| 2 | 300 | 300 | 300 | 0,2534 | 1,3265 | 71,7310 |
| 2 | 500 | 500 | 500 | 0,7853 | 3,3951 | 416,1940 |
| 2 | 1000 | 1000 | 1000 | 6,3643 | 13,4864 | 6468,1929 |
| 2 | 1500 | 1500 | 1500 | 15,7077 | 29,9343 | 25753,3828 |
| 2 | 2000 | 2000 | 2000 | 61,4829 | 83,0175 | 56799,5312 |
|  | | | | | | |
| 3 | 100 | 100 | 100 | 0,0789 | 0,7196 | 2,6741 |
| 3 | 300 | 300 | 300 | 0,2554 | 1,4600 | 71,0730 |
| 3 | 500 | 500 | 500 | 0,7690 | 3,1972 | 417,4530 |
| 3 | 1000 | 1000 | 1000 | 6,4012 | 13,8697 | 6132,7988 |
| 3 | 1500 | 1500 | 1500 | 15,6640 | 30,4613 | 26506,5273 |
| 3 | 2000 | 2000 | 2000 | 61,3386 | 84,8868 | 56552,8906 |

Результаты замеров на неквадратных матрицах приведены ниже в таблице:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Способ расположения** | **n** | **m** | **p** | **GPU Clean, мс** | **GPU Total, мс** |
| 0 | 100 | 50 | 25 | 0,0563 | 0,8353 |
| 0 | 400 | 200 | 100 | 0,0774 | 0,8076 |
| 0 | 800 | 400 | 200 | 0,2099 | 2,2944 |
| 0 | 1600 | 800 | 400 | 0,9128 | 5,6151 |
| 0 | 3200 | 1600 | 800 | 5,6488 | 22,1217 |
|  | | | | | |
| 1 | 100 | 50 | 25 | 0,0583 | 0,5017 |
| 1 | 400 | 200 | 100 | 0,0955 | 0,9780 |
| 1 | 800 | 400 | 200 | 0,2780 | 2,3649 |
| 1 | 1600 | 800 | 400 | 1,2352 | 5,8801 |
| 1 | 3200 | 1600 | 800 | 7,6388 | 21,6110 |
|  | | | | | |
| 2 | 100 | 50 | 25 | 0,0830 | 0,9998 |
| 2 | 400 | 200 | 100 | 0,3063 | 1,3542 |
| 2 | 800 | 400 | 200 | 1,4602 | 4,0852 |
| 2 | 1600 | 800 | 400 | 6,2705 | 11,7585 |
| 2 | 3200 | 1600 | 800 | 43,9999 | 60,3755 |
|  | | | | | |
| 3 | 100 | 50 | 25 | 0,0829 | 0,7875 |
| 3 | 400 | 200 | 100 | 0,2977 | 1,3372 |
| 3 | 800 | 400 | 200 | 1,4128 | 3,1076 |
| 3 | 1600 | 800 | 400 | 6,2505 | 11,5044 |
| 3 | 3200 | 1600 | 800 | 44,0753 | 59,3723 |

Полученные результаты можно визуализировать с помощью графиков, представленных ниже.

Зависимость времени работы GPU от размера и типа матриц

Ниже на рисунках 1 и 2 приведены графики зависимости времени работы GPU от размера как квадратных, так и прямоугольных матриц. Замеры сделаны для классического размещения всех матриц.

Рисунок 1

Рисунок 2

По представленным графикам можно сделать вывод о том, что на малых размерах время, приходящееся на выделение памяти под матрицы, копирование и т.д., занимает большую долю от всего затраченного времени обработки. Например, для квадратных матриц размера 100 на 100 GPU Total примерно в 18 раз больше GPU Clean. Однако с увеличением размера матриц эта разница ставится меньше, становясь менее значительной. Такой эффект можно наблюдать на матрицах размером 2000 на 2000 элементов – разница составляет примерно 3 раза.

Аналогичная ситуация наблюдается, если рассматривать прямоугольные матрицы.

Таким образом, можно сделать вывод о том, что привлечение GPU имеет место лишь на значительных размерах матриц, поскольку в таком случае накладные расходы окупаются за счёт ресурсоёмкости операции умножения.

Зависимость времени работы GPU и CPU от размера квадратных матриц

На рисунке 3 приведён график демонстрирующий временные затраты, приходящиеся на GPU и CPU соответственно для квадратных матриц с ориентацией 0 варианта.

Рисунок 3

По полученным результатам можно сделать вывод о том, что операция умножения матриц с учетом таких накладных расходов как выделение памяти, копирование данных и т.д. выполняется на GPU значительно быстрее, чем на CPU. С увеличением количества элементов разница становится всё существеннее.

Аналогичная ситуация наблюдается при рассмотрении других ориентаций как для квадратных, так и для прямоугольных матриц.

Зависимость времени GPU Clean от размера матриц на различных ориентациях

На рисунках 4-5 приведены графики сравнения времени умножения как квадратных, так и прямоугольных матриц при различных ориентациях.

Рисунок 4

По рисунку выше можно сделать вывод о том, что варианты ориентации 0 и 1 имеют лучшие показатели по сравнению с 2 и 3. Наилучший результат наблюдается в случае, когда ни одна матрица не была транспонирована.

Аналогичная ситуация наблюдается в случае, если рассматриваются прямоугольные матрицы.

Рисунок 5

**Вывод**

В результате проведённых исследований можно прийти к следующим заключениям:

1. Затраты, приходящиеся на выделение памяти, копирование данных, составляют значительную долю от всего времени, занимаемого операцией умножения, поэтому привлечение графического процессора лучше осуществлять на матрицах большого размера, поскольку операция умножения в таком случае становится преобладающей по ресурсам.
2. Вычисление произведения матриц на GPU происходит быстрее, чем на CPU.
3. Не смотря на множество вариаций размещения матриц в памяти, наиболее эффективным оказывается классический способ без какого-либо транспонирования матриц.