

Інтеграція прискорення на графічному процесорі у середовище виконання для блоково- рекурсивних матричних алгоритмів DAR

Студент спеціальності Комп'ютерні науки МП-2:
Комонов Кирило Максимович
Науковий керівник:
доктор фіз.-мат. наук, професор
Малашонок Геннадій Іванович

Вступ

- Колегами з НаУКМА було реалізовано середовище виконання блоково-рекурсивних матричних алгоритмів DAR.
- Воно запускається на кластері і здатне виконувати матричні задачі великого розміру у паралельний спосіб.
- Раніше, DAR не міг використовувати прискорення від графічних процесорів.
- В рамках цієї роботи, до DAR надано можливість виконувати множення щільних матриць на графічному процесорі.

Постановка задачі

- Оглянути існуючі програмні інструменти для виконання обчислень на відеокарті.
- Реалізувати виконання необхідних алгоритмів на відеокарті.
- Інтегрувати це рішення у середовище DAP.
- Провести експерименти і заміряти метрики, що нас цікавлять.
- На основі результатів експериментів з'ясувати межі використання прискорення на відеокартах для задачі матричного множення.
- Визначити, для якого діапазону розмірів вхідних даних використання графічного процесора надає перевагу у швидкодії

Наукова новизна

- Середовище блоково-рекурсивних матричних алгоритмів DAP отримало підтримку прискорення на графічних процесорах для алгоритмів множення щільних матриць.
- Порівняно швидкодію роботи DAP з прискоренням і без.

Алгоритм Штрассена

- Алгоритм Штрассена є блоково-рекурсивним алгоритмом множення матриць.
- Він має асимптотичну складність $O(n^{\log_2 7})$ - замість $O(n^3)$ для наївного алгоритму.

$$X = \begin{bmatrix} A & B \\ C & D \end{bmatrix}, \quad Y = \begin{bmatrix} E & F \\ G & H \end{bmatrix}$$

Рисунок 1 - Розділення матриць на чотири підматриці

$$XY = \begin{bmatrix} P_5 + P_4 - P_2 + P_6 & P_1 + P_2 \\ P_3 + P_4 & P_1 + P_5 - P_3 - P_7 \end{bmatrix}$$

$$P_1 = A(F - H)$$

$$P_2 = (A + B)H$$

$$P_3 = (C + D)E$$

$$P_4 = D(G - E)$$

$$P_5 = (A + D)(E + H)$$

$$P_6 = (B - D)(G + H)$$

$$P_7 = (A - C)(E + F)$$

Рисунок 2 - Схема алгоритму множення матриць Штрассена

Координація використання GPU 1

На кожному вузлі у кластері ми маємо N ядер процесора і M відеокарт ($N > M$).

Отже, процесам запущеним на різних ядрах необхідно ділити доступ до графічного процесора (GPU).

Для ефективного використання GPU, процесорам на різних ядрах слід використовувати GPU по черзі.

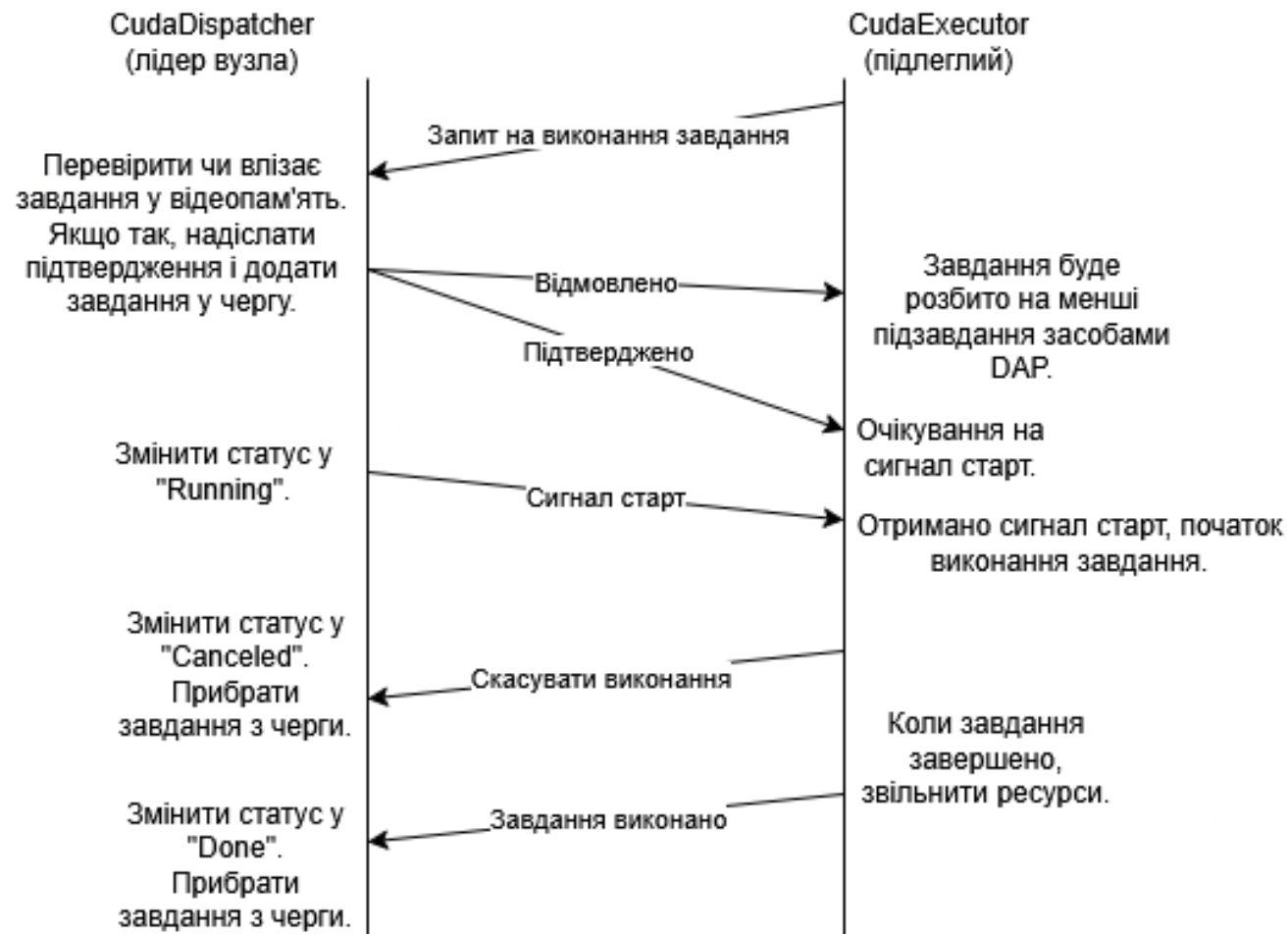
Для координації використання GPU було створено `CudaDispatcher` клас.

Об'єкт класу `CudaDispatcher` присутній єдиний на вузол, всі процеси запущені на цьому вузлі комунікують з ним засобами MPI або API (для процесу що містить об'єкт класу `CudaDispatcher`).

Координація використання GPU 2

CudaDispatcher дозволяє чи забороняє виконання завдання (наприклад, на основі кількості відеопам'яті необхідної для виконання обчислень).

CudaDispatcher керує тим коли починається виконуватись кожне завдання



Експериментальні запуски

| Алгоритм | Вузол | Чи є GPU | Кількість процесорів | Розмір вхідних матриць | Час виконання (сек) | Час обслуговування GPU (сек) | RAM (mb) |
|---------------------|-------|----------|----------------------|------------------------|--------------------------------|------------------------------|----------|
| MatrDMultStrassWin7 | g4301 | Ні | 1 | 512 | 47.954 | - | 113 |
| MatrDMultStrassWin7 | g4301 | Ні | 1 | 1024 | 1025.409 | - | 253 |
| MatrDMultStrassWin7 | g4301 | Ні | 1 | 2048 | 15425.986 | - | 935 |
| MatrDMultStrassWin7 | g4301 | Ні | 12 | 512 | 13.504 | - | 52 |
| MatrDMultStrassWin7 | g4301 | Ні | 12 | 1024 | 173.593 | - | 239 |
| MatrDMultStrassWin7 | g4301 | Ні | 12 | 2048 | 3854.414 | - | 881 |
| MatrDMultStrassWin7 | g4301 | Так | 1 | 512 | 0.394 | 0.167 | 32 |
| MatrDMultStrassWin7 | g4301 | Так | 1 | 1024 | 0.655 | 0.247 | 95 |
| MatrDMultStrassWin7 | g4301 | Так | 1 | 2048 | 1.657 | 0.639 | 347 |
| MatrDMultStrassWin7 | g4301 | Так | 1 | 4096 | 8.849 | 5.789 | 1363 |
| MatrDMultStrassWin7 | g4301 | Так | 1 | 8192 | 30.859 | 16.259 | 5445 |
| MatrDMultStrassWin7 | g4301 | Так | 1 | 16384 | Закінчилась оперативна пам'ять | | |
| MatrDMultStrassWin7 | g4301 | Так | 12 | 512 | 0.452 | 0.249 | 29 |
| MatrDMultStrassWin7 | g4301 | Так | 12 | 1024 | 0.61 | 0.287 | 92 |
| MatrDMultStrassWin7 | g4301 | Так | 12 | 2048 | 1.714 | 0.971 | 344 |
| MatrDMultStrassWin7 | g4301 | Так | 12 | 4096 | 6.249 | 3.39 | 1353 |
| MatrDMultStrassWin7 | g4301 | Так | 12 | 8192 | 25.809 | 14.566 | 5385 |
| MatrDMultStrassWin7 | g4301 | Так | 12 | 16384 | Закінчилась оперативна пам'ять | | |

Час виконання програми для алгоритму Штрассена на вузлі g4301

Досягнуті коефіцієнти прискорення

| Алгоритм | Вузол | Кількість процесів | Розмір вхідних матриць | Коефіцієнт прискорення (округлений до одиниць) |
|---------------------|-------|--------------------|------------------------|--|
| MatrDMult4 | g4301 | 1 | 512 | 137 |
| MatrDMult4 | g4301 | 1 | 1024 | 1815 |
| MatrDMult4 | g4301 | 1 | 2048 | 20723 |
| MatrDMult4 | g4301 | 12 | 512 | 161 |
| MatrDMult4 | g4301 | 12 | 1024 | 1745 |
| MatrDMult4 | g4301 | 12 | 2048 | 6445 |
| MatrDMultStrassWin7 | g4301 | 1 | 512 | 122 |
| MatrDMultStrassWin7 | g4301 | 1 | 1024 | 1566 |
| MatrDMultStrassWin7 | g4301 | 1 | 2048 | 9310 |
| MatrDMultStrassWin7 | g4301 | 12 | 512 | 30 |
| MatrDMultStrassWin7 | g4301 | 12 | 1024 | 285 |
| MatrDMultStrassWin7 | g4301 | 12 | 2048 | 2249 |

Коефіцієнти прискорення швидкодії програми з впровадженням обчислення на графічному процесорі (на вузлі g4301)

ВИСНОВКИ

- Було реалізоване прискорення на графічному процесорі алгоритмів «розділяй і пануй» і Штрассена.
- Нова версія DAP з прискоренням на графічних процесорах працює швидше для задачі множення щільних матриць на всьому діапазоні можливих розмірів вхідних матриць, що наразі підтримується DAP.

Дякую за увагу!