Теория парралелизма

Отчёт

Уравнение теплоправодности на Cuda

с несколькими видеокартами

05.26.2023

Гольцев Никита Сергеевич, 21932

1. Цели работы

Реализовать решение уравнения теплопроводности, переписав программу на Cuda, при том добавить реализацию вычислений для нескольких видеокарт с использованием MPI.

1. Компиляция

Компилятор используемый при работе:

mpicc

Так же в процессе, для замера времени, использовалась библеотека chrone.

1. Используемый профилировщик

nsys (NVIDIA Nsight Systems)

1. Этапы оптимизации

В данной работе в качестве дополнительного способараспараллеливания

вычислений при помощи видеокарт используется интерфейс MPI. При

помощи него мы можем задействовать сразу несколько устройств и при

помощи функции MPI\_Sendrecv передавать данные между устройствами.

Для дополнительного распараллеливания вычислений интерполяция границ

и внутренних составляющих матрицы происходят в разных асинхронных

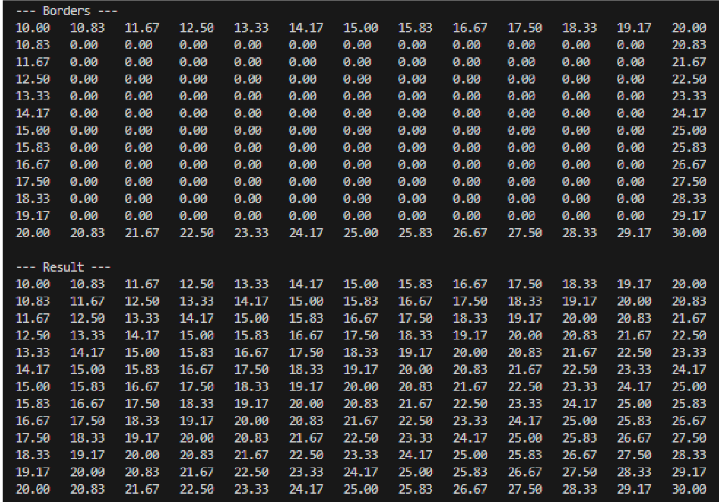
потоках. Чтобы обезопасить код, все вычисления помещены в блок try..catch,

при помощи которого отлавливаются ошибки и выводятся причины их

возникновения.

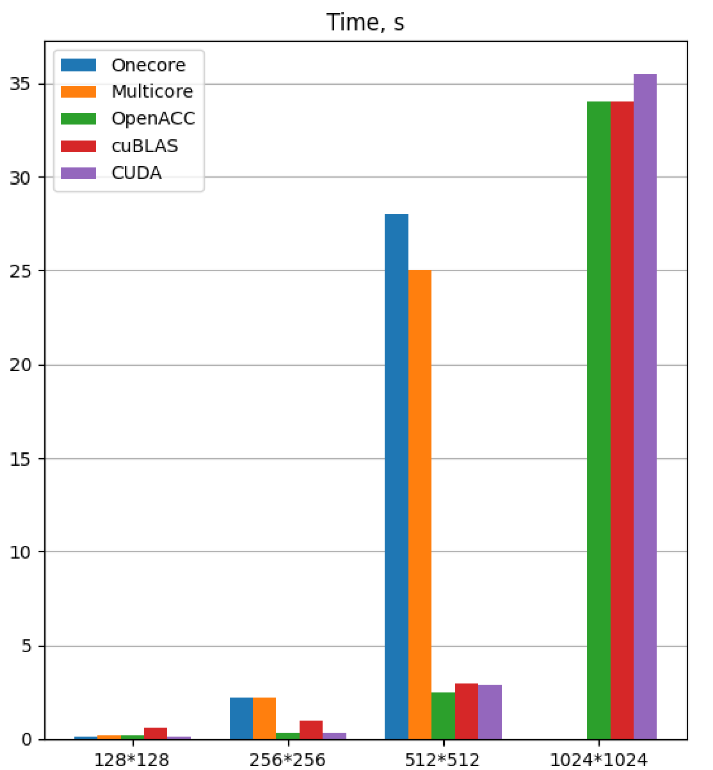
1. Точность вычиллений

При выводе после создания и по завершению вычислений матрица совпадает с аналогичными проверками прошлого задания.



1. Сравнение времени работы, как с пред идущими версиями, так и при вычисление на для разного количества видеокарт

Во всех случаях количество операций совподало, завися от того, какой был размер сетки.



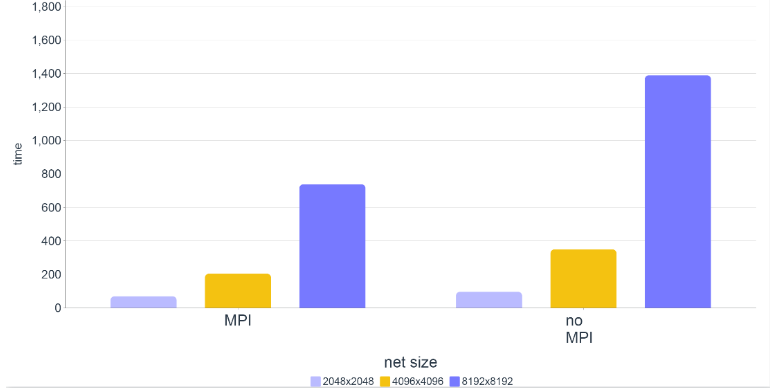
Выше представленно сравнение вычисления на одной видеокарте cuda и прошлыми версиями программ.

|  |  |  |  |
| --- | --- | --- | --- |
| Размер стеки | Время, мс | Точность | Итерации |
| 128\*128 | 160 мс | 9.9998-e7 | 31081 |
| 256\*256 | 340 мс | 9.9998-e7 | 37913 |
| 512\*512 | 2298 мс | 9.9998-e7 | 120657 |
| 1024\*1024 | 27890 мс | 1.373-e6 | 367890 |

Для начала проведём сравнения между одной и двумя видеокартами(MPI с 2мя видеокартами).

Явно видно приимущество по скорости вычисления для варианта с использованием mpi.

Теперь, для сравнения, добавим время работы с тремя видеокартами и наглядно покажем время работы на диограмме.



Из диограммы выше можно сделать вывод, что от количества используемых ресурсов явно зависит скорость выполнения задачи.

Ниже представленна полная таблица для 1, 2 и 4 видио карт.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Размер | 1gpu | 2gpu | 4gpu | Время, с | Коэф.  Эффективности  Для 4 gpu | Точность | Кол-во итераций |
| 128\*128 | 0.16 | 0.9 | 1.69 | - | 0.03 | 9.9e-07 | 31081 |
| 256\*256 | 0.75 | 3.2 | 5.3 | - | 0.035 | 9.9e-07 | 37913 |
| 512\*512 | 7.1 | 12.9 | 18.5 | - | 0.096 | 9.9e-07 | 120657 |
| 1024\*1024 | 64.7 | 217 | 54.8 | - | 0.29 | 1.4e-06 | 367890 |
| 2048\*2048 | 246 | 134.0 | 96.7 | - | 0.65 | 1.4e-06 | 1000000 |
| 4096\*4096 | 1056 | 538 | 277 | - | 0.95 | 9.8e-06 | 1000000 |

1. Вывод

Наглядно видно, что mpi очень полезен при сложных вычислениях(т.к. чем больше размер сетки, тем более явно видна его эффективность), а так же, что с увиличением количества видеокарт, на которых проводится вычисления, увиличивается и скорость работы программы.

Но более расширенные тесты показали, что на маленьких размерах сетки он проигрывает.

Можно сделать вывод, что вычисление на нескольких видеокартах оказывает значительный, положительный эффект на скорость работы программы, если у данной задачи достаточно высокая сложность.