

Теория Параллелизма

Отчет

Оптимизированные библиотеки

21933, Круковский Василий Сергеевич
04.23

Цель работы:

Изменить предыдущую реализацию решения уравнения теплопроводности под библиотеку CuBLAS. Оптимизировать код, базируясь на данных профилировщика.

Используемый компилятор:

pgc++

Используемый профилировщик:

nsys (NVIDIA Night System) с OpenACC

Как производился замер времени работы:

Замер производился с использованием библиотеки chrono

Команды для компиляции, профилирования и запуска.

Тип	Команда
GPU	pgc++ -acc -Minfo=accel -fast code.cpp -o gpu
GPU+cublas	pgc++ -acc -Minfo=accel -Mcudalib=cublas -fast code.cpp -o cublas
CPU(multicore)	pgc++ -acc=multicore -Minfo=accel -fast code.cpp -o cpu_multi
CPU(one core)	pgc++ -acc code.cpp -o cpu
Execute	./name_prog
Профилирование	nsys profile -t openacc ./gpu
Профилирование + cublas	nsys profile -t openacc,cublas ./cublas

Прошлые результаты выполнения на CPU:

1.1 Onecore

Размер сетки	Время выполнения, сек	Точность	Количество итераций
128*128	2.62	9.98e-07	30080
256*256	64.21	9.98e-07	102912
512*512	2067.725	9.93e-07	339968

1.2 Multicore

Размер сетки	Время выполнения	Точность	Количество итераций
128*128	1.5	9.98e-07	30080
256*256	12.8	9.98e-07	102912
512*512	76.17	9.93e-07	339968
1024*1024	689.576	1.37e-06	1000000

1.3 GPU

Размер сетки	Время выполнения, сек	Точность	Количество итераций
128*128	0.2	9.98e-07	30080
256*256	0.72	9.98e-07	102912
512*512	6.3	9.93e-07	339968
1024*1024	66.5	1.37e-06	1000000

Этапы оптимизации программы:

Этап 0:

Изменение кода для GPU под библиотеку cublas в том числе:

Удаление некоторых прагм и циклов. Циклы в основном алгоритме переставлены местами.

Размер сетки	Время выполнения, сек	Точность	Количество итераций
128*128	0.69	9.98e-07	30080
256*256	1.22	9.98e-07	102912
512*512	3.88	9.93e-07	339968
1024*1024	38.16	1.37e-06	1000000

Этап 1:

Грамотное использование асинхронности.

Размер сетки	Время выполнения, сек	Точность	Количество итераций
128*128	0.6	9.98e-07	30080
256*256	0.86	9.98e-07	102912
512*512	2.92	9.93e-07	339968
1024*1024	34.2	1.37e-06	1000000

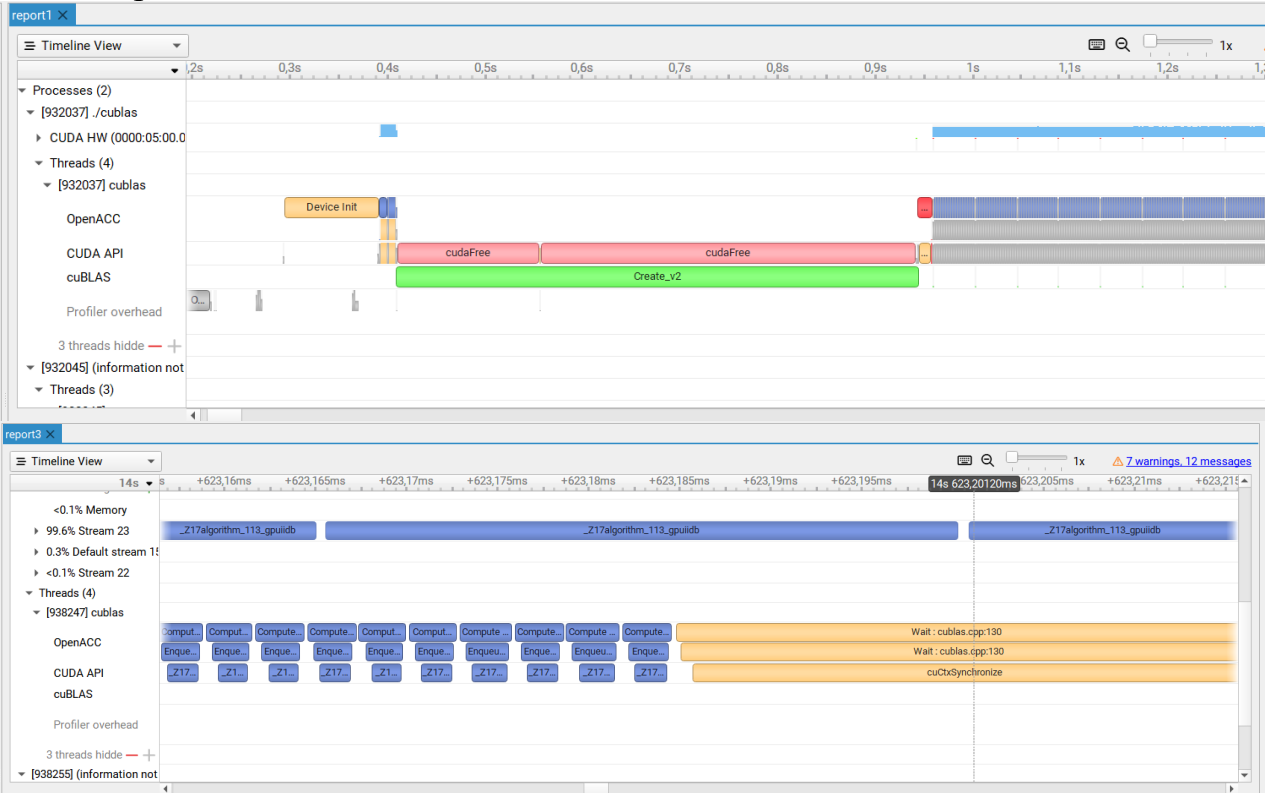
Вывод:

В сравнении с GPU, программа замедлилась на небольших сетках, но на больших, библиотека cublas достаточно хорошо оптимизирована. Если сравнивать скорость выполнения в профилировщике, то ускорение по времени почти в два раза.

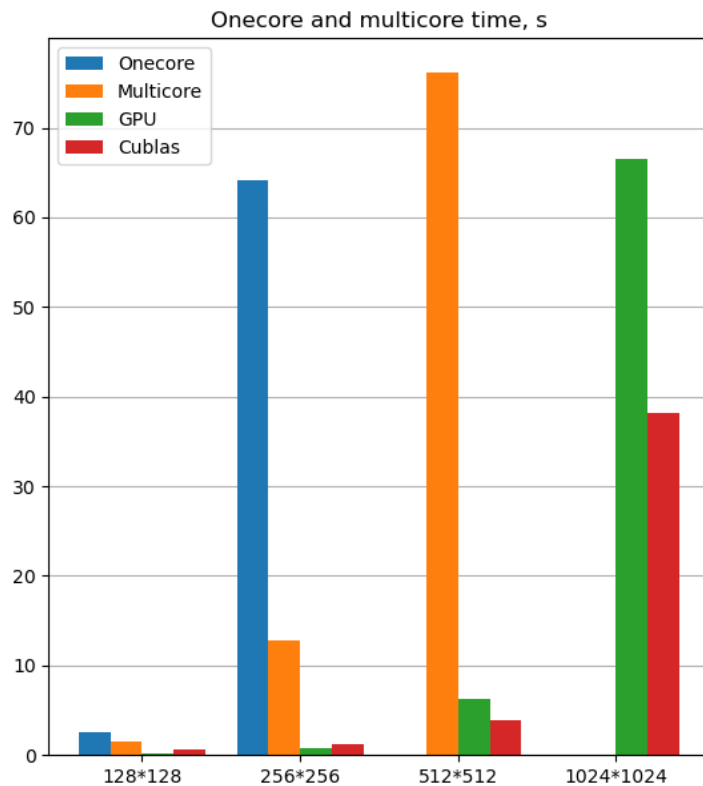
Результат выполнения на GPU с использованием CuBLAS:

Размер сетки	Время выполнения, сек	Точность	Количество итераций
128*128	0.6	9.98e-07	30080
256*256	0.86	9.98e-07	102912
512*512	2.92	9.93e-07	339968
1024*1024	34.2	1.37e-06	1000000

Стоит заметить, что некоторое время уходит на Create_v2 в CuBLAS, а именно около нескольких десятых долей секунд. Что непременно сказывается на общем времени вычислений.



Сравнение скорости выполнения программ:



Для удобства сравнения на диаграмме не приведены некоторые данные.

Вывод:

Нет особого смысла применять Cublas для сетей небольшой размерности, потому что инициализация handle и create занимает долгое время

Приложение:

Ссылка на GitHub: <https://github.com/MegaSear/parallelism/tree/master/task3>