Домашнее задание №2 (CUDA)
Умножение матриц на GPU средствами CUDA.

Постановка задачи

$$\boldsymbol{C = A \cdot B}$$

Элементы матриц A, B и C имеют тип double; матрицы имеют размер $N \times N$

Матрицы A, B и C хранятся в одномерных массивах в column-major порядке

Задания

1. *Реализовать перемножение матриц с использованием глобальной памяти на CUDA.*

   Ваша задача состоит в том, чтобы написать программу, которая на CPU генерирует произвольные матрицы, затем отправляет их на GPU и там перемножает с использованием двумерной системы блоков и потоков.

2. *Ускорить передачу матрицы из CPU в GPU за счет использования Pinned памяти.*

   Ваша задача состоит в том, чтобы реализовать изначально создавать и хранить матрицы используя Pinned память RAM. Сравните общее время на перемножение (учитывающее передачу матрицы на GPU и обратно) используя средства **CudaEventElapsedTime**.

3. *Ускорить передачу матрицы из CPU в GPU за счет использования CUDA потоков.*

Ваша задача состоит в том, чтобы реализовать перенос матрицы из памяти CPU в память GPU параллельным путем используя CUDA потоки (**cudaMemcpyAsync**).

4. **Написать перемножение матриц с использованием Разделяемой памяти.*

   Используя метод блочного перемножения матрицы, реализовать перемножение матрицы с использованием общей памяти блоков **(__shared__)**, рассмотреть различные варианты реализации и ускорения перемножения. Обратить внимание на работу с банками разделяемой памяти.

Оценивание

Задание 1: 4 балла

Задание 2: 2 балла

Задание 3: 2 балла

Задание 4*: 2 балла

Сдача будет происходить на семинаре.
Рекомендуется сдавать задания последовательно, иначе у вас и у ваших одногруппников может не остаться времени на её сдачу. Файл программы нужно называть “cumm_ %Фамилия%_%группа%.cu”,
например “cumm _Ivanov_211.cu”.

Файлы необходимо отправить на корпоративную почту семинариста с названием вида "ВВ_дз2_cumm_Ivanov_211"

Примечание

Последнее задание является исследовательским и направлено на наиболее заинтересованных студентов, которые хотят более подробно разобраться в теме высокопроизводительных вычислений. Таким образом за основную часть задания вы можете получить максимум 8 баллов.

Работающий пример по подсчету значения функции на GPU находится в папке /tmp/hpc_course/ (файл main_f_x.cu) на Харизме. Для компиляции рекомендуем подгрузить модуль nvhpc.

module load nvidia_sdk/nvhpc/22.7

Компиляция: *nvcc main_f_x.cu*

Также напоминаем, что задачи на расчет на кластере нужно ставить ТОЛЬКО с использованием системы SLURM (srun, sbatch):

- *srun -n 1 --gpus=1 -A proj_N ./a.out*
- *sbatch -n 1 --gpus=1 -A proj_N --wrap="./a.out"*
- *salloc -n 1 --gpus=1 -A proj_N,* после выделения узла, можете запускать программу сколько угодно раз на нем (*srun ./a.out*). Чтобы снять выделенный узел нажмите Ctrl+D.

Рекомендуется придерживаться стандартов написания кода.

Дедлайн

01.11.2022

При сдаче после дедлайна оценка будет снижена на 2 балла за каждую прошедшую неделю от дедлайна.