Задание по Cuda

Курс: Параллельное программирование и суперкомпьютерный кодизайн, 2023
Выполнил студент 530 группы, Демьянов Иван Сергеевич

Описание решения

Мой номер - 8, следовательно при выполнении данного задания мне досталась функция *cudaGetDeviceProperties*.

Ниже представлена организация проекта, который демонстрирует работу данной конструкции.

```
github.com/demyanov17/Cuda_task
├── main.cpp ..................................................Основная функция
├── cuda_device_checker.cu ..................Функция для проверки наличия Cuda
├── device_information.cu ..............Функция, выводящая характеристики Cuda
├── functions.h ..............................Файл с вспомогательными функциями
├── Makefile ..............................Файл для компиляции и запуска проекта
└── report.pdf ..........Отчёт по заданию, который Вы читатете в данный момент
```

Рис. 1: структура github репозитория с проектом

main.cpp

Основной модуль, в котором производится вызов вспомогательных функций:

- `int check_cuda_is_avialable(void)`
- `void print_device_properties_information(void)`

Первая проверяет наличие Cuda-устройств и в случае их отсутствия выдает ошибку, в противном случае - вызывается вторая функция, выводящая в стандартный поток вывода информацию об устройствах.

```
jovyan@demyanov-a100-0:~/demyanov17/Cuda_task$ make
g++ -c main.cpp
nvcc -c cuda_device_checker.cu
nvcc -c device_information.cu
nvcc main.o cuda_device_checker.o device_information.o -o Cuda_task
jovyan@demyanov-a100-0:~/demyanov17/Cuda_task$ ./Cuda_task
=====================================================
  Device 0: A100 80GB PCIe
  Compute Capability: 8.0
  Total Global Memory: 81252 MB
  Multiprocessors: 108
  Clock Rate: 1410 MHz
  Warp Size: 32
  Max Threads Per Block: 1024
  Max Threads Dimensions: (1024, 1024, 64)
  Max Grid Dimensions: (2147483647, 65535, 65535)
=====================================================
```

Рис. 2: Пример вывода программы

cuda_device_checker.cu

Модуль, содержащий вспомогательную функцию *check_cuda_is_avialable*. Эта функция устанавливает количество Cuda-устройств при помощи функции *cudaGetDeviceCount* и выдаёт сообщение об ошибке, если их нет.

device_information.cu

Модуль содержит вспомогательную функцию *print_device_properties_information*. Эта функция выводит на экран характеристику каждого Cuda-девайса (чаще всего он будет 1). Здесь и задействована демонстрация полезности конструкции *cudaGetDeviceProperties*: при помощи неё можно довольно элегантно получить интересующие характеристики видеокарты.

functions.h

Модуль, содержащий заголовки вспомогательных функций, описанных выше.

Makefile

Файл для более легкой сборки описанного проекта. Для запуска необходимо выполнить в терминале следующие команды:

1. `make`
2. `./Cuda_task`

Также можно чистить директорию от объектников, выполнив `make clean`.

Заключение

В данном отчете приведено описание структуры и модулей проекта. Стоит отметить, что данный проект носит учебный характер и направлен на ознакомление автора с устройством Cuda.

Автор 5 раз в неделю применяет его в индустрии и при необходимости получения характеристик видеокарты вводит что-то в этом духе в юпитер-ноутбуке

```
Ввод [3]: import torch

          torch.__version__

  Out[3]: '1.11.0+cu115'

Ввод [4]: !nvidia-smi

Wed Nov 29 00:31:14 2023
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 450.172.01   Driver Version: 450.172.01   CUDA Version: 11.5     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  A100 80GB PCIe      On   | 00000000:B9:00.0 Off |                    0 |
| N/A   32C    P0    41W / 300W |      0MiB / 81252MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Однако проделанная работа в рамках данного задания позволяет получить доступ к характеристикам вне уже реализованных терминальных команд. По правде говоря, вряд ли это может часто пригождаться, но в случае чего - буду иметь не только библиотечную информацию о рабочем Cuda-устройстве:)