

## Лабораторная работа № 7

# Измерение пропускной способности памяти видеокарт с поддержкой технологии CUDA

**Цель работы:** научиться измерять пропускную способность памяти GPU при различных типах и направлениях передачи данных.

Работа с памятью является одним из краеугольных камней при разработке эффективных программ с использованием технологии CUDA. Ее правильное использование зачастую позволяет увеличить скорость обработки данных до 10 и более раз.

Видеокарта не имеет доступа к оперативной памяти компьютера (host), поэтому все обрабатываемые данные должны быть загружены в динамическую память GPU (device). В данной работе используется глобальная память, для обмена данными с ней используется функция

```
cudaError_t cudaMemcpy(void *dst, void *src, size_t count, cudaMemcpyKind kind);
```

которая осуществляет копирование count байт из источника src в приемник dst. Направление копирования задается параметром kind и может принимать следующие значения:

```
cudaMemcpyHostToHost    /* Host    -> Host */  
cudaMemcpyHostToDevice  /* Host    -> Device */  
cudaMemcpyDeviceToHost  /* Device -> Host */  
cudaMemcpyDeviceToDevice /* Device -> Device */
```

В первом случае копирование производится в пределах оперативной памяти (функция работает как memcpy()), во втором – из оперативной памяти в глобальную память GPU, в третьем – из глобальной памяти GPU в оперативную память, и в четвертом – в пределах глобальной памяти GPU.

Для выделения области памяти в глобальной памяти GPU используется функция

```
cudaError_t cudaMalloc(void **devPtr, size_t size);
```

возвращающая в первом параметре указатель на выделенную область памяти размером size байт. Для выделения области динамической памяти в оперативной памяти можно использовать функции

```
void * malloc(size_t size);
```

и

```
cudaError_t cudaMallocHost(void **ptr, size_t size);
```

Функция malloc() входит в стандартную библиотеку, она выделяет блок динамической памяти размера size и возвращает указатель на его начало. Функция cudaMallocHost() отличается от нее тем, что помечает выделяемые страницы виртуальной памяти так, что они гарантированно находятся в оперативной памяти и не могут быть вытеснены из нее в файл подкачки (т.н. page locking). Это упрощает процедуру наблюдения за данной памятью со стороны драйвера видеокарты и увеличивает скорость

обмена. Однако пользоваться данной функцией необходимо с осторожностью, т.к. при выделении большого объема данных производительность системы в целом может существенно ухудшиться, могут появиться системные ошибки и т.п.

Копирование в рамках глобальной памяти GPU осуществляется асинхронно: функция `cudaMemcpy()` возвращает управление раньше, чем завершается копирование. Чтобы учесть это, можно добавить в код вызов функции

```
cudaError_t cudaThreadSynchronize();
```

которая ожидает завершения текущей асинхронной операции и лишь затем возвращает управление. Фактически таким образом происходит синхронизация потока CPU с GPU.

### Задание.

1. Выделить блоки памяти одинакового размера в оперативной памяти и глобальной памяти видеокарты. Скопировать содержимое блоков между:
  - двумя буферами в оперативной памяти;
  - между оперативной памятью и глобальной памятью видеокарты в направлении CPU→GPU и GPU→CPU с использованием обычной и page-locked памяти (должно быть 4 различных варианта копирования);
  - между двумя буферами в глобальной памяти видеокарты.
2. Убедиться в том, что копирование происходит корректно.
3. Измерить время копирования и, зная размер копируемого блока, определить пропускную способность (в ГБ/с) для каждого типа копирования. Сделать выводы о скорости копирования в различных режимах.
4. В отчет включить краткое описание типа и параметров видеокарты и процессора.

```
C:\Projects\CUDA\06 GPU_bandwidth_measure\Time_measure\debug\Time_measure.exe
CUDA memory bandwidth test (block size = 100 MB)
(c) Eduard I. Vatutin
WWW: http://evatutin.narod.ru
e-mail: evatutin@rambler.ru
ICQ: 203-229-391

1 CUDA device(s) found
GPU 0: GeForce GTX 450
RAM allocating... OK
GPU global RAM allocating... OK
Copying Host -> Device
Average bandwidth = 1.21082 GB/s
Copying Device -> Host
Average bandwidth = 0.834231 GB/s
RAM allocating... OK
Copying Host -> Host
Average bandwidth = 1.27087 GB/s
GPU global RAM allocating... OK
Copying Device -> Device
Average bandwidth = 20.0366 GB/s
Copying Host -> Device (using page-locked)
Average bandwidth = 2.46627 GB/s
Copying Device -> Host (using page-locked)
Average bandwidth = 1.69953 GB/s
Done
```

Рис. Результаты измерения пропускной способности

### Содержание отчета.

1. Титульный лист
2. Цель работы
3. Задание
4. Листинг программы
5. Результаты измерения пропускной способности

## 6. Выводы