Основы технологии CUDA

Семинар 6:Работа с текстурной памятью

Автор курса:

- Казённов Андрей

Типы памяти в CUDA

Тип памяти	Доступ	Уровень выделения	Скорость работы
Регистры	R/W	Per-thread	Высокая(on-chip)
Локальная	R/W	Per-thread	Низкая (DRAM)
Shared	R/W	Per-block	Высокая(on-chip)
Глобальная	R/W	Per-grid	Низкая (DRAM)
Constant	**R/O**	**Per-grid**	**Высокая(L1 cache)**
Texture	**R/O**	**Per-grid**	**Высокая(L1 cache)**

Применение константной памяти

```
__constant__ float contsData [256];
float              hostData  [256];

cudaMemcpyToSymbol ( constData, hostData, sizeof ( data ), 0,
                     cudaMemcpyHostToDevice );
////////////////////////////////////////////////////////
template <class T>
cudaError_t cudaMemcpyToSymbol   ( const T& symbol, const void *
   src,
                    size_t count, size_t offset, enum
   cudaMemcpyKind kind );

template <class T>
cudaError_t cudaMemcpyFromSymbol ( void * dst, const T& symbol,
                    size_t count, size_t offset, enum
   cudaMemcpyKind kind );
```

Применение текстурной памяти

- Латентность больше, чем у прямого обращения в память
  - Дополнительные стадии в конвеере:
    - Преобразование адресов
    - Фильтрация
    - Преобразование данных
- Кэшируется в кэшах на SM и на всем кристале
  - Разумно использовать, если:
    - Объем данных не влезает в shared память
    - Паттерн доступа хаотичный
    - Данные переиспользуются разными потоками

Применение текстурной памяти: виды

- Линейная
  - Базируется на простой глобальной памяти. Может хранить только линейные массивы
- cudaArray – особый контейнер

Применение текстурной памяти: объявление текстуры

- `texture< type , dim, tex_type> g_TexRef;`
  - Type – тип хранимых переменных
  - Dim – размерность текстуры (1, 2, 3)
  - Tex_type – тип возвращаемых значений
    - cudaReadModeNormalizedFloat
    - cudaReadModeElementType

Применение текстурной памяти: channelDesc

```
struct cudaChannelFormatDesc {
        int x, y, z, w;
        enum cudaChannelFormatKind f;
};
```

Задает формат возвращаемого значения

- int x, y, z, w; - число [0,32] проекция исходного значения по битам
- cudaChannelFormatKind – тип возвращаемого значения
  - cudaChannelFormatKindSigned – знаковые int
  - cudaChannelFormatKindUnsigned – беззнаковые int
  - cudaChannelFormatKindFloat - float

Применение текстурной памяти: линейная

Можно использовать обычную *глобальную* память

- Ограничения:
  - Только для одномерных массивов
  - Нет фильтрации
  - Доступ по целочисленным координтам
  - Обращение по адресу вне допустимого диапазона возвращает ноль

- **cudaBindTexture**(size_t shift, texref tex, &src, size_t size));

- **cudaBindTexture2D**(size_t shift, texref tex, &src, &channelDesc, int width,int height,int pitch);

- **cudaUnbindTexture(**texref tex**);**

Применение текстурной памяти: линейная

Можно использовать обычную *глобальную* память

- Ограничения:
  - Только для одномерных массивов
  - Нет фильтрации
  - Доступ по целочисленным координтам
  - Обращение по адресу вне допустимого диапазона возвращает ноль

- **cudaBindTexture**(size_t shift, texref tex, &src, size_t size));

- **cudaBindTexture2D**(size_t shift, texref tex, &src, &channelDesc, int width,int height,int pitch);

- **cudaUnbindTexture(**texref tex**);**

Применение текстурной памяти: линейная

Можно использовать обычную *глобальную* память

- Ограничения:
  - Только для одномерных массивов
  - Нет фильтрации
  - Доступ по целочисленным координтам
  - Обращение по адресу вне допустимого диапазона возвращает ноль
- Доступ: tex1Dfetch(tex, int)

- **cudaBindTexture**(size_t shift, texref tex, &src, size_t size));

- **cudaBindTexture2D**(size_t shift, texref tex, &src, &channelDesc, int width,int height,int pitch);

- **cudaUnbindTexture(**texref tex**);**

Применение текстурной памяти: cudaArray

- Позволяет организовывать данные в 1D/ 2D/3D массивы данных вида:
  - 1/2/4 компонентные векторы
  - 8/16/32 bit signed/unsigned integers
  - 32 bit float
  - 16 bit float (driver API)
- Доступ по семейству функций tex1D()/tex2D()/tex3D()

- `cudaArray *cuda_array;`

- `cudaBindTextureToArray(texRef, cuArray, &channelDesc);`

- `cudaUnbindTexture(texref tex);`

Применение текстурной памяти: возможности cudaArray

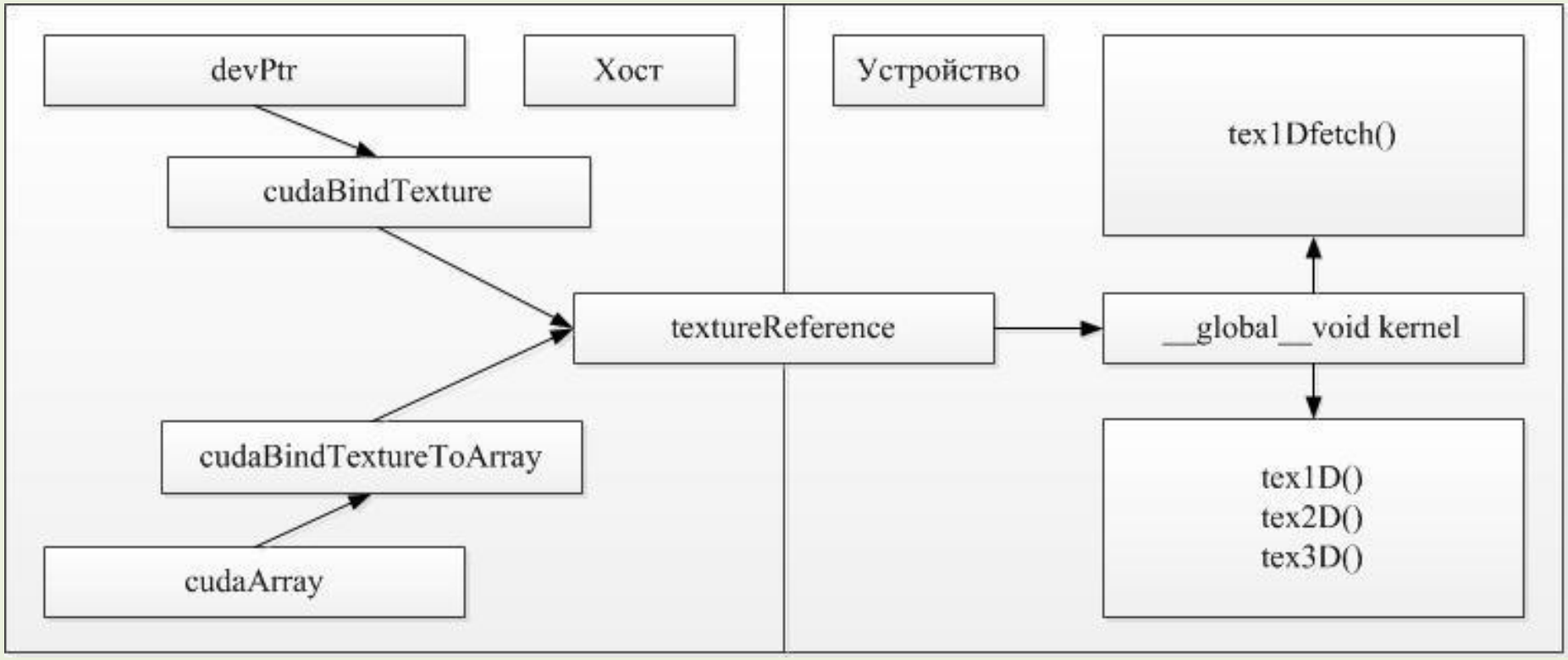
- Нормализация координат (перевод [W, H] => [0,1])

- Преобразование координат
  - Clamp – координата обрезается по границе
  - Wrap – координата заворачивается

- Фильтрация (при обращении по float координате)
  - Point – возвращается ближайшее заданное значение
  - Linear – производится билинейная интерполяция

Основы технологии CUDA

Подведем итоги

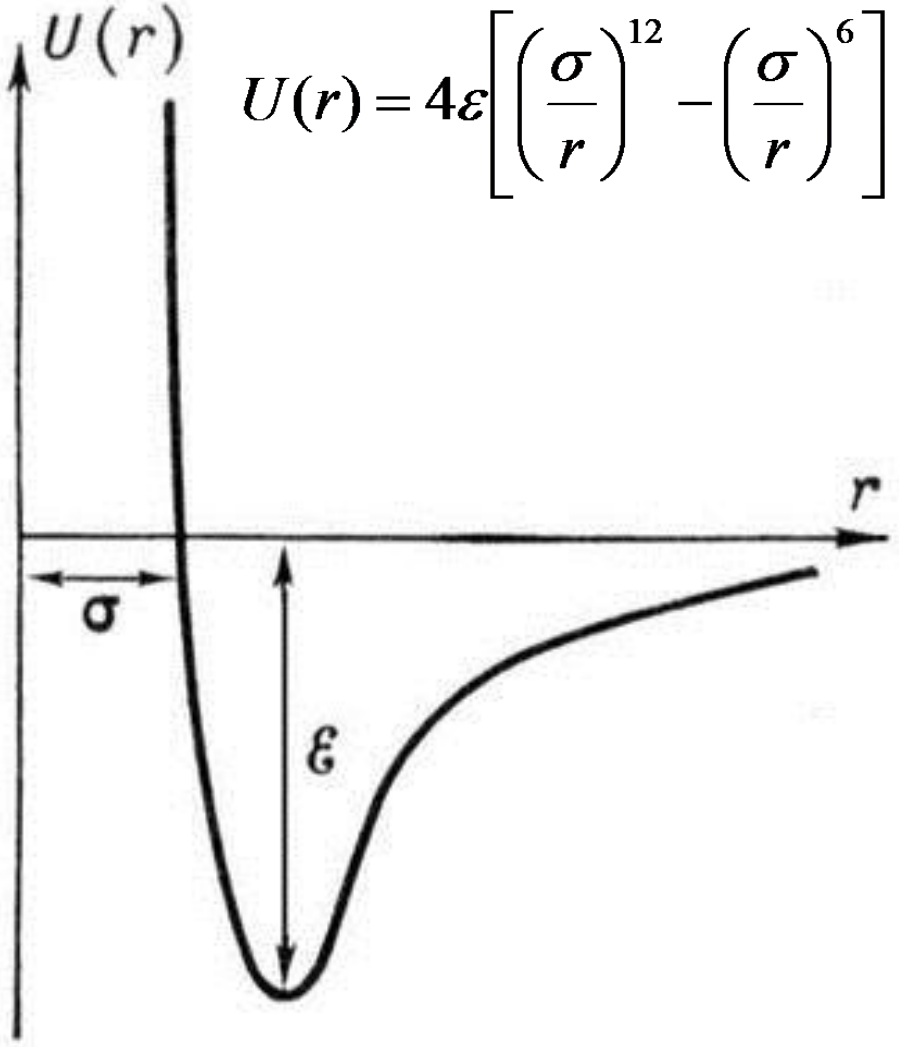


Основы технологии CUDA

Задача: Написать программу МД моделирования

- Входной файл :
  x y z
- Шаг по времени 1нс
- Количество шагов 1000
- Тип взаимодействия Леннард-Джонс
- Задать потенциал интерполяционной таблицей
- Принять радиус действия потенциала -- 5

$$U(r) = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$



Ресурсы курса

Сайт: HPC.MIPT.RU

Раздел образование

Автор курса:

Казённов Андрей

- E-mail: kazennov@gmail.com
- ICQ: 622774