CUDA Driver API

Романенко А.А.
arom@ccfit.nsu.ru

Новосибирский государственный университет

- Привязано ли ядро к исполняемому коду?
- Можно ли запускать ядро не используя расширения языка Си?
- Можно ли программировать на CUDA не на Си/Си++?

Объекты в CUDA driver API

- **Device** — CUDA-совместимое устройство
- **Context** — «эквивалент» процессу для CPU
- **Module** — «эквивалент» динамической библиотеки
- **Function** — ядро
- **Heap memory** — указатель на память устройства
- **CUDA Array** — контейнер для 1D или 2D массивов на устройстве, доступных через текстуру
- **Texture reference** — объект для описания данных в текстуре

Инициализация устройства

- CUresult **cuInit**(unsigned int flag);
  - Flag = 0
- CUT_DEVICE_INIT_DRV(cuDevice,
  ARGC, ARGV)
- Без инициализации все функции будут возвращать CUDA_ERROR_NOT_INITIALIZED

Управление устройствами (1)

- CUresult **cuDeviceGetCount**(int ∗count)
- CUresult **cuDeviceGet**(CUdevice ∗device, int ordinal)
- CUresult **cuDeviceComputeCapability**(int ∗major, int ∗minor, CUdevice dev)
- CUresult **cuDeviceTotalMem**(unsigned int ∗bytes, CUdevice dev)
- CUresult **cuDeviceGetAttribute**(int ∗pi, CUdevice_attribute attrib, CUdevice dev)

Атрибуты устройства (1)

- CU_DEVICE_ATTRIBUTE_MAX_THREADS_PER_BLOCK
- CU_DEVICE_ATTRIBUTE_MAX_BLOCK_DIM_X
- CU_DEVICE_ATTRIBUTE_MAX_BLOCK_DIM_Y
- CU_DEVICE_ATTRIBUTE_MAX_BLOCK_DIM_Z
- CU_DEVICE_ATTRIBUTE_MAX_GRID_DIM_X
- CU_DEVICE_ATTRIBUTE_MAX_GRID_DIM_Y
- CU_DEVICE_ATTRIBUTE_MAX_GRID_DIM_Z
- CU_DEVICE_ATTRIBUTE_MAX_SHARED_MEMORY_PER_BLOCK
- CU_DEVICE_ATTRIBUTE_TOTAL_CONSTANT_MEMORY
- CU_DEVICE_ATTRIBUTE_WARP_SIZE
- CU_DEVICE_ATTRIBUTE_MAX_PITCH

Атрибуты устройства (2)

- CU_DEVICE_ATTRIBUTE_MAX_REGISTERS_PER_BLOCK
- CU_DEVICE_ATTRIBUTE_CLOCK_RATE
- CU_DEVICE_ATTRIBUTE_TEXTURE_ALIGNMENT
- CU_DEVICE_ATTRIBUTE_GPU_OVERLAP
- CU_DEVICE_ATTRIBUTE_MULTIPROCESSOR_COUNT
- CU_DEVICE_ATTRIBUTE_KERNEL_EXEC_TIMEOUT
- CU_DEVICE_ATTRIBUTE_INTEGRATED

  CU_DEVICE_ATTRIBUTE_CAN_MAP_HOST_MEMORY
- CU_DEVICE_ATTRIBUTE_COMPUTE_MODE
  - CU_COMPUTEMODE_DEFAULT
  - CU_COMPUTEMODE_EXCLUSIVE
  - CU_COMPUTEMODE_PROHIBITED

Управление устройствами (2)

- CUresult cuDeviceGetProperties (CUdevprop ∗prop, CUdevice dev)
  - typedef struct CUdevprop_st {
    int maxThreadsPerBlock;
    int maxThreadsDim[3];
    int maxGridSize[3];
    int sharedMemPerBlock;
    int totalConstantMemory;
    int SIMDWidth;
    int memPitch;
    int regsPerBlock;
    int clockRate;
    int textureAlign
    } CUdevprop;

Контекст CUDA

- Контекст CUDA — аналог процесса для CPU
- В рамках потока может быть только один контекст CUDA
- При создании контекста счетчик использования равен 1
- **cuCtxAttach()** увеличивает счетчик, **cuCtxDetatch()** уменьшает счетчик на 1
- Контекст разрушается когда счетчик использования становиться равным 0

Управление контекстом CUDA

- CUresult **cuCtxAttach** (CUcontext ∗pCtx, unsigned int Flags)
  - Flag = 0
- CUresult **cuCtxCreate** (CUcontext ∗pCtx, unsigned int Flags, CUdevice dev)
  - Flag = 0 (CUDA 1.1); Schedule types for the thread
- CUresult **cuCtxDestroy** (CUcontext ctx)
- CUresult **cuCtxDetach** (CUcontext pCtx)
- CUresult **cuCtxGetDevice** (CUdevice ∗device)
- CUresult **cuCtxPopCurrent** (CUcontext ∗pCtx) — CUDA v2.2
- CUresult **cuCtxPushCurrent** (CUcontext nCtx) — CUDA v2.2
- CUresult **cuCtxSynchronize** (void)

Модули в CUDA

- Модуль — динамически подгружаемый объект с ядрами (kernel). Аналог DLL файлов
- Модули собираются с помощью nvcc. Могут распространятся независимо.
- CUmodule cuModule;
  cuModuleLoad(&cuModule, "module.cubin");
  CUfunction cuFunc;
  cuModuleGetFunction(&cuFunc, cuModule, "myKernel");

Управление модулями

- CUresult **cuModuleGetFunction** (CUfunction ∗func, CUmodule mod, const char ∗name)
- CUresult **cuModuleGetGlobal** (CUdeviceptr ∗ret_dptr, unsigned int ∗ret_bytes, CUmodule mod, const char ∗name)
- CUresult **cuModuleGetTexRef** (CUtexref ∗ppTexRef, CUmodule mod, const char ∗name)
- CUresult **cuModuleLoad**(CUmodule ∗phMod, const char ∗fname)
- CUresult **cuModuleLoadData**(CUmodule ∗phMod, const void ∗p)
- CUresult **cuModuleLoadDataEx** (CUmodule ∗phMod, const void ∗p, unsigned int numOptions, CUjit_option ∗options, void ∗∗optionValues) — CUDA 2.2
- CUresult **cuModuleUnload** (CUmodule mod)

Управление исполнением

- Задание конфигурации потокового блока
- Задание конфигурации сети
- Передача параметров функций
- Установка размеров разделяемой памяти

Управление исполнением

- CUresult cuFuncGetAttribute (int *pi, CUfunction_attribute attrib, CUfunction func)
- CUresult cuFuncSetBlockShape (CUfunction func, int x, int y, int z)
- CUresult cuFuncSetSharedSize(CUfunction func, unsigned int bytes)
- CUresult cuLaunchGrid (CUfunction func, int grid_w, int grid_h)
- CUresult cuLaunchGridAsync(CUfunction func, int grid_w, int grid_h, CUstream hStream)
- CUresult cuParamSetf (CUfunction func, int offset, float value)
- CUresult cuParamSeti(CUfunction func, int offset, unsigned int value)
- CUresult cuParamSetSize (CUfunction func, unsigned int numbytes)
- CUresult cuParamSetTexRef(CUfunction func, int texunit, CUtexref pTexRef)
- CUresult cuParamSetv(CUfunction func, int offset, void *ptr, unsigned int Nbytes)

Управление памятью

- CUresult cuMemAlloc (CUdeviceptr ∗dptr, unsigned int size)
- CUresult cuMemAllocHost (void ∗∗pp, unsigned int bytesize)
- CUresult cuMemAllocPitch (CUdeviceptr ∗dptr, unsigned int ∗pPitch, unsigned int WidthInBytes, unsigned int Height, unsigned int ElementSizeBytes)
- CUresult cuMemFree (CUdeviceptr dptr)
- CUresult cuMemFreeHost (void ∗p)
- CUresult cuMemcpy* - функции копирования между массивами, памятью GPU и CPU

Управление текстурами

- CUresult **cuTexRefCreate** (CUtexref ∗phTexRef)
- CUresult **cuTexRefDestroy** (CUtexref hTexRef)
- CUresult **cuTexRefSetAddress** (unsigned int ∗pOffset, CUtexref hTexRef, CUdeviceptr dptr, unsigned int bytes)
- CUresult **cuTexRefSetFilterMode** (CUtexref hTexRef, CUfilter_mode fm)
- CUresult **cuTexRefSetAddressMode** (CUtexref hTexRef, int Dim, CUaddress_mode am)
- CUresult **cuTexRefSetFlags** (CUtexref hTexRef, unsigned int Flags)