GPU (Graphics Processing Unit) Computing

Motivace

recept na pomalou paměť (memory latency):
- CPU: rychlá cache paměť
- GPU: stream processing (multithreading), data parallelism

Historie

– 1999 první GPU (NVIDIA)
– GPGPU (general purpose computing on GPUs): programování grafických procesorů pomocí DirectX, OpenGL aj.
– GPU Computing: programování pomocí standardních překladačů (C, Fortran aj.)
– konkurence: AMD (ATI) FireStream, Intel Larrabee aj.

NVIDIA

1. generace (2006): architektura G80 (GeForce 8800, Quadro FX 5600, Tesla C870), Compute Capability (CC) 1.0, 1.1
   GeForce (PC grafika), Quadro (profesionální stanice), Tesla (high-performance computing)
2. generace (2008): architektura G200 (GeForce GTX 280, Quadro FX 5800, Tesla T10), CC 1.3
3. generace (2010): architektura Fermi/GF100 (GeForce GTX 480, Tesla C2050), CC 2.0

Hardware
- GPU = zařízení (device) obsahuje několik multiprocesorů (1, 2, 4, …, 24, 30; Fermi 16) a globální paměť
  - multiprocesor (streaming multiprocessor, SM) obsahuje procesory (cores) a několik typů rychlé paměti
    - SM v G80/G200 obsahuje 8 procesorů pro integer a single-precision real, G200 navíc 1 procesor pro double-precision real (SM Fermi: 32 procesorů pro integer a SP/DP real)
    - procesory mají lokální soukromou paměť (1-2 tis. 32bitových registrů, tj. 32-64 KB/SM)
    - SM má lokální sdílenou paměť (16 KB), cache pro konstanty (8 KB) a cache pro textury (8 KB)
  - globální paměť (device memory, soukromá i sdílená, dostupná hostiteli) o velikosti 256 MB, 512 MB, …, 4 GB
  - fyzická karta může obsahovat více zařízení (2, 4), počítač může pojmout více fyzických karet

Software
- procedura přeložená pro GPU (Fortran: kernel subroutine, C: kernel function) se spustí v paralelních vláknech
- vlákna jsou sdružena v blocích (block), vlákna bloku běží na jednom SM a jsou synchronizovatelná
- bloky vláken jsou sdruženy v mřížce (grid), různé bloky mohou být prováděny na různých SM
- indexace vláken v blocích i bloků v mřížce může být 1D, 2D nebo 3D (mřížka efektivně nejvýše 2D)
- rozprostřením kernelu do bloků a mřížky se optimalizuje vytíženost SM a přístup do paměti SM a zařízení (scheduling problem) – závislé na konfiguraci zařízení (viz omezení CC) i aplikaci
- vlákna bloku jsou prováděna po svazcích (warps) o 32 vláknech
  - vlákna svazku běží synchronně v režimu SIMT (single-instruction multiple-threads)
  - vlákna polosvazku (half-warp) mohou využívat sdružený přístup do paměti (memory coalescing)
- verze CC (níže) omezuje velikost bloku (max. 512), mřížky (mez pro jednu dimenzi 65535) a svazku (32)
- zařízení v danou chvíli vykonává pouze jeden kernel (Fermi více)
- kernel nemůže obsahovat cokoliv (limitovaný počet mikroinstrukcí, limitované použití ukazatelů, nelze rekurze)

Compute capability (CC, podle CUDA Programming Guide, Appendix A)
- 1.0 … max. rozměry bloku: 512-512-64, celkem však max. 512 vláken/blok
  - max. rozměry mřížky: 65535-65535-1
  - velikost warpu: 32 vláken
  - paměť SM: registry 32 KB, sdílených 16 KB, pro konstanty 64 KB, max. lokálních 16 KB/vlákno
  - sloty SM: max. 8 aktivních bloků, max. 24 aktivních svazků (tj. 768 aktivních vláken)
  - max. velikost kernelu: 2 mil. instrukcí
- 1.1 … 32bitové atomické funkce
- 1.2 … 64bitové atomické funkce, paměť: registry 64 KB/SM, sloty SM: max. 32 svazků (tj. 1024 vláken)
- 1.3 … double precision (omezená podpora IEEE 754)
- 2.0 .. 32 procesorů/SM, 16 SM (512 procesorů), 8x větší výkon v DP, 64 sdílených KB/SM, 1536 vláken/SM
  - rychlost, cache, ECC, plná podpora IEEE 754 (SP, DP), L1 a L2 (768 KB) cache, ECC
  - souběžné provádění více kernelů

My:
- GeForce GTX 260 (4 tis. Kč): 24x8 = 192 (27x8=216??) procesorů, 896 MB DDR3, CC 1.3
  - max.: 715 GFLOPS, 180 W; pro srovnání, naše CPU i7-920: 43 GFLOPS, 130 W
- GeForce 9500 GT (1 tis. Kč): 4x8 = 32 procesorů, 512 MB DDR2, CC 1.1
- GeForce 8500GT, 8400GS: 2x8 = 16 procesorů, 256-512 MB DDR2, CC 1.1

Top (2009):

GeForce GTX 295	2 GPU x 30 SM x 8	1788 GFLOPS
Quadro Plex 2200	totéž	
Tesla S1070	4 GPU x 30 SM x 8	4140 GFLOPS (SP), 345 GFLOPS (DP)

Překladače pro NVIDIA GPU
CUDA C (NVIDIA), OpenCL (Khronos), Brook (Stanford University) … na bázi C
Microsoft DirectCompute … součást DirectX
PGI (Portland Group) … CUDA Fortran a akcelerátor (direktivy) pro PGI Fortran a C
Jacket … nadstavba k Matlabu

CUDA (Compute Unified Device Architecture):
– balík zahrnující překladač C, rozhraní API pro GPU hardware, driver, knihovny
(CUDA driver, CUDA toolkit, CUDA SDK; CUDA runtime; CUDA libraries – CUBLAS, CUFFT)
– poslední verze 2.3 a 3.0
– CUDA programování zahrnuje:
kernel functions … kód pro GPU v samostatných funkcích
local memory management … užívání lokální sdílené paměti SM
global memory management … alokace paměti a přesuny dat mezi hostitelem (host) a GPU (device)
volání kernel functions … rozložení vláken na multiprocesorech pomocí bloků a mřížky