GPGPU

General Purpose computation on Graphics Processing Units

Mark Harris, 2002

Perform demanding calculations on the GPU instead of the CPU!

At first, appeared to be a wild idea, but is now a very serious technology! Results were highly varied in the early years, but the GPU advantage has grown bigger and bigger.

Information Coding / Computer Graphics, ISY, LiTH

Key components of the GPGPU trend

High processing power in parallel

Programmability: Introduction of shader programs, much more flexible, programmable for any problem.

Floating-point buffers: Vital! Initially with poor precision. 32-bit floating-point decent... but not really impressive.

GPGPU solutions

- Using fixed pipeline graphics
- Shader programs
- CUDA
- OpenCL

Fixed pipeline GPGPU

Reformulate a problem to something that can be done by standard graphics operations.

Limited success 1999/2000. Not of any practical interest!

Shader-based GPGPU

Portable! Most GPUs can use shaders, no need for extra software, run using standard software/drivers.

All modern shader languages (GLSL, Cg, HLSL) are similar and easy to program in.

Requires a re-mapping of data to textures.

Very good results already in 2005: 8x speedups overall reported!

Information Coding / Computer Graphics, ISY, LiTH

GPGPU using shaders

Has less attention now, due to CUDA.

Still interesting:

- Apart from some reusable standard code, it is not very complicated.
- Portable to most GPUs with no extra software.
- Excellent performance.

CUDA-based GPGPU

Only works on NVidia hardware.

Requires extra software - which isn't very elegant.

Nice integration of CPU and GPU code in the same program.

Excellent results! 100x speedups are common - before optimizing! Even low-end GPUs give significant boosts.


Information Coding / Computer Graphics, ISY, LiTH

OpenCL-based GPGPU

Works on various hardware - not only GPUs.

Developed by Khronos Group, pushed by Apple.

Harder to get started, software looks pretty much like programming shaders.

Use the source, Luke!

Three trivial examples:

Hello World! for CUDA

Hello World! for OpenCL

Hello World for GLSL

Information Coding / Computer Graphics, ISY, LiTH

So what GPU should you get?

For CUDA, go for Fermi or Kepler boards!

Kepler has more cores but Fermi is still strong in double precision.

GTX560 good middle-range board, best Fermi price/performance and reasonable power consumption.

GTX660Ti best Kepler price/performance and even better power consumption.

Avoid overclocked boards!

Don't bother with "professional" Quadro boards.

AMD is DEFINITELY an option for shaders or OpenCL but not CUDA.

In the Southfork lab

GTX660Ti

Exactly one year old. Still one of the fastest.

1300 cores!

Close-to-high-end mid-range board. Great price/performance, lots of parallelism to play with, and pretty nice power consumption.

Information Coding / Computer Graphics, ISY, LiTH

That's all, folks!

Next friday: Introduction to CUDA