

The background of the slide is a dark blue, abstract digital scene. It features a grid of glowing white lines and a central, bright, glowing diamond-shaped pattern that resembles a data visualization or a network structure. The overall aesthetic is futuristic and high-tech.

Innovative software
for manycore paradigms

Computing with GPUs: Status

Romain Dolbeau, <romain.dolbeau@caps-entreprise.com>



Outline

- History of GPUs as compute devices
- An overview of GPU architectures
- Evolution of GPUs
- GPU compute today

The History of GPGPU: The Dark Ages

- The 80s / early 90s: the advent of discrete 3D pipeline
 - Sun GT “graphics tower”: external box hooked to the computer via an interface card
- The 90s / early 21st century: the advent of consumer-grade 3D systems
 - From the 3Dfx Voodoo to the NVIDIA GeForce FX and ATI Radeon R200
- Finally, DirectX 9.0c and OpenGL 2.0 brings 32 bits floating point (FP) to GPUs: the GeForce 6 and Radeon R500

The History of GPGPU: The Dawn

- Early 2003, GPU overtakes CPU for single precision FP, with expectations for more (Khailany 2003)
- Programmable shaders appear at the same time
- First GPGPU applications (for instance, Bolz 2003)

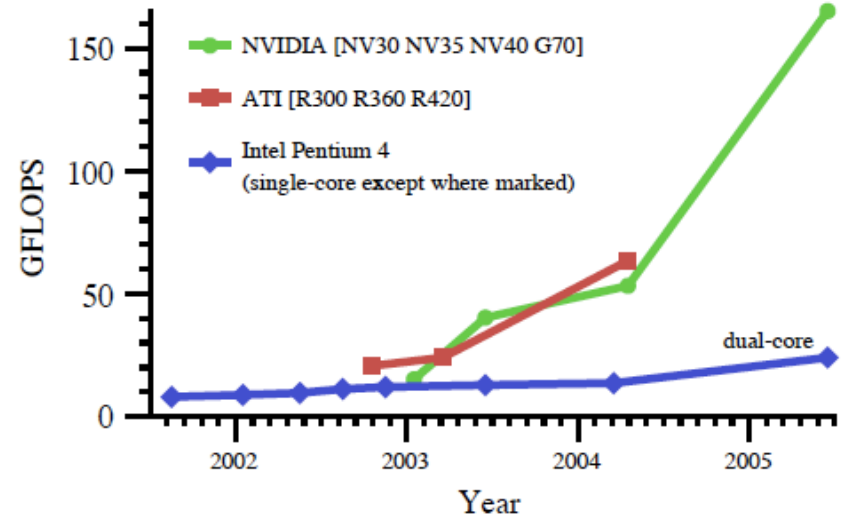
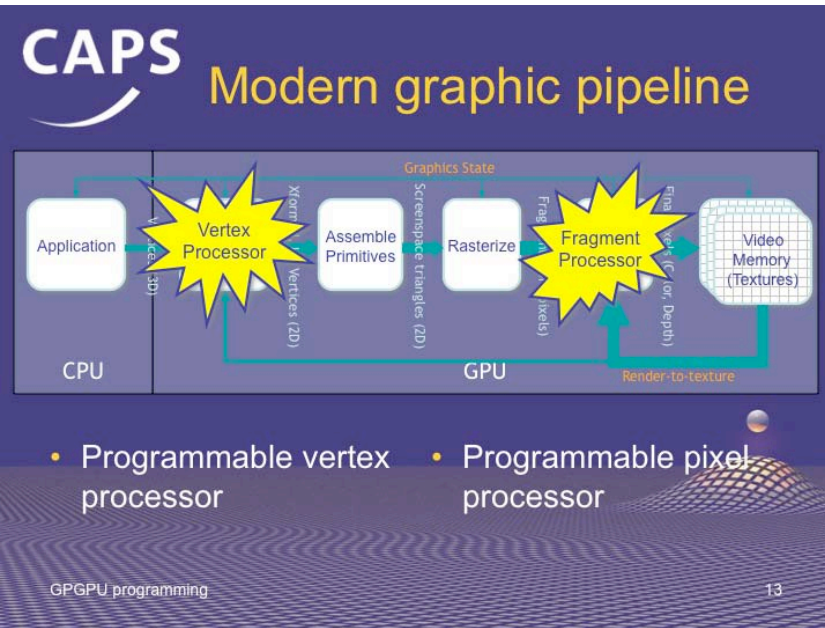


Figure 1: *The programmable floating-point performance of GPUs (measured on the multiply-add instruction as 2 floating-point operations per MAD) has increased dramatically over the last four years when compared to CPUs. Figure courtesy Ian Buck, Stanford University.*

The History of GPGPU: the OpenGL Era

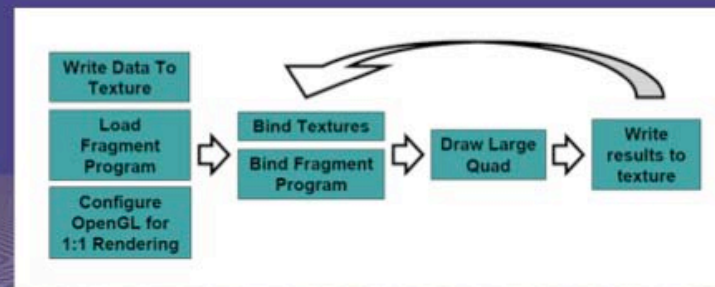
- From 2003 to late 2006/ early 2007
 - GPU Programming using OpenGL 3D library
 - Needs understanding graphic pipeline and work around limitations



APS GPU programming model

GPU as a stream processor

- Streams => textures
- Kernels => fragment program
- Foreach execution => draw single large quad



The History of GPGPU: the Mainstream Era

- Brook first beta was in late 2004 but was OpenGL-based
- Specialized vendors such as PeakStream released libraries and tools, again using OpenGL at first
- CAPS to demo a C-to-OpenGL generator at SC'06

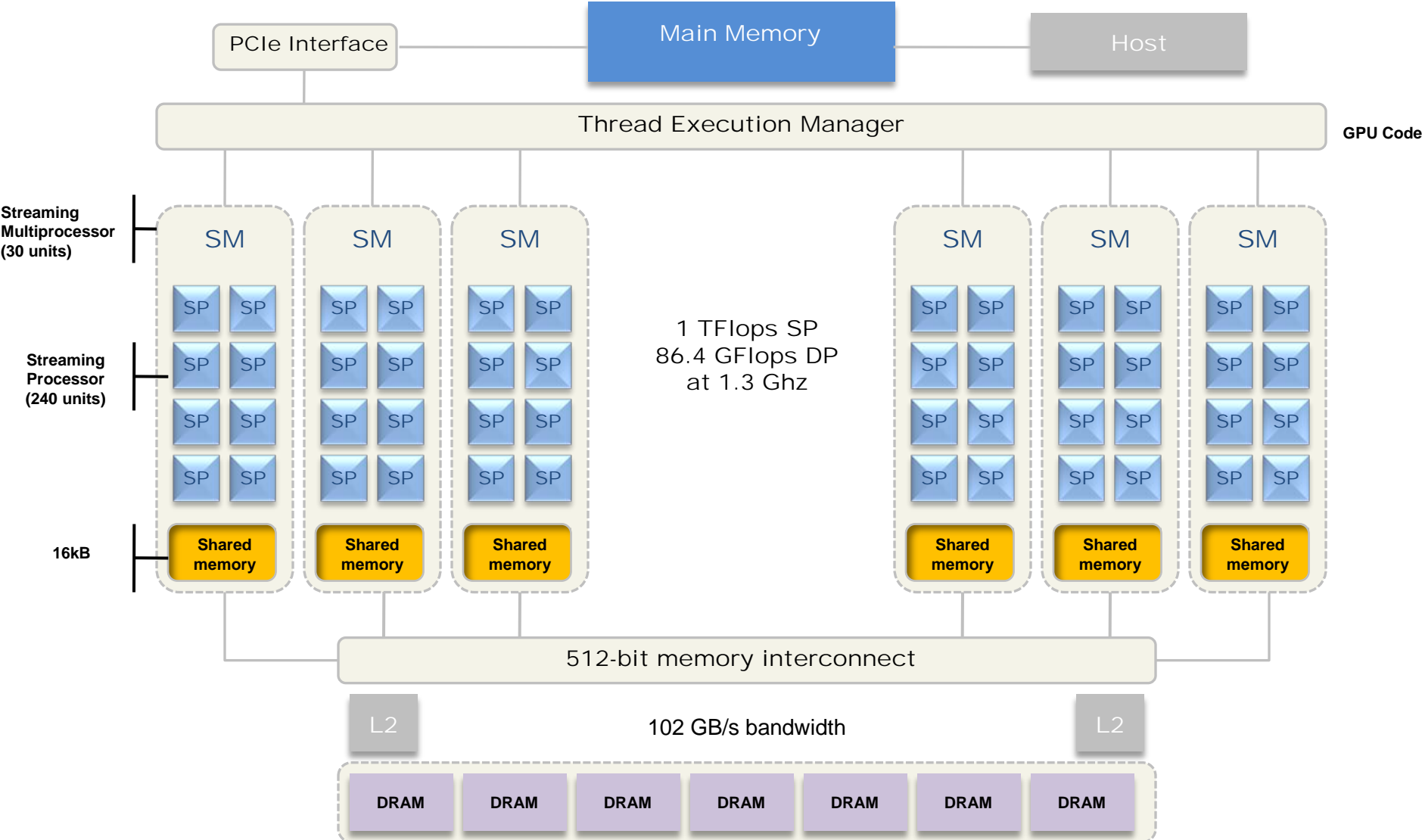
- Feb. 2007: the first release of NVIDIA CUDA
- Dec. 2007: the first release of the ATI Stream SDK, replacing CTM (Close To Metal)

- OpenGL is no longer needed, GPGPU becomes exploitable for more than research
- Late 2008, Double Precision FP is available !
- OpenCL as a new standard?

GPU Architecture Overview

- Fine grain massive data parallelism
 - Many streaming processors
- Exposed memory hierarchy
 - Large device memory
 - Small partially *shared* memory, ...
- Heavily pipelined memory
 - Small data caches
- Fixed amount of hardware resources
 - No virtual memory
 - Maximum number of threads
 - Maximum number of registers usage per group of threads
 - ...

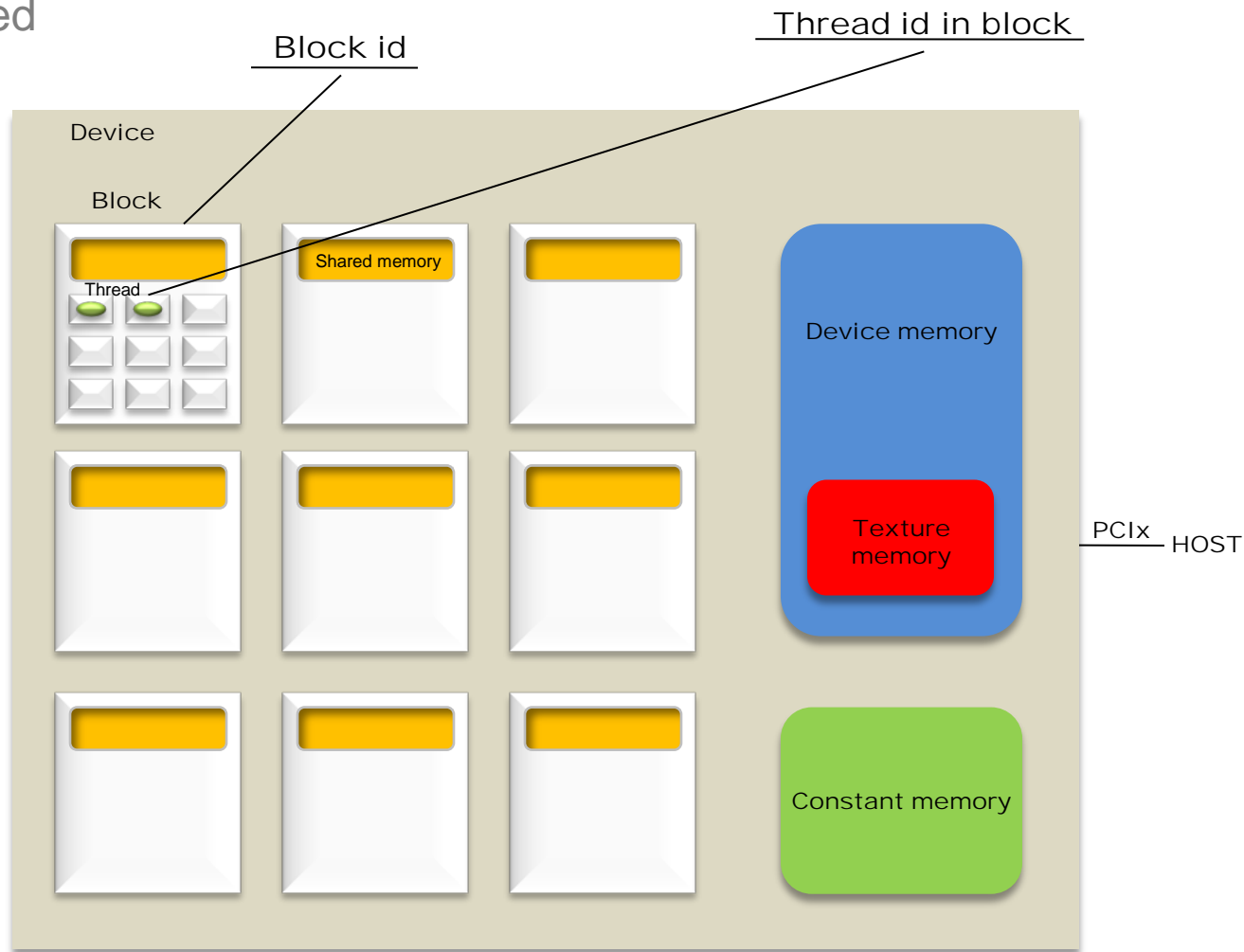
Example: NVIDIA T10



CUDA Grids and Blocks

- GPUs need 1000s of threads to be efficient
 - Highly pipelined
 - Highly parallel

- Blocks of thread are executed on the streaming multiprocessors



Consequences of GPU Architectures

- **Host-Device model**
 - High overhead when moving data between CPU and GPU
 - Not always good to move a computation to GPU
- **Large number of registers**
 - Loop unrolling is an important transformation
- **Small streaming multiprocessor shared memory**
 - Help reducing pressure on device memory
- **Penalty for over-spending a type of resource is huge**
 - e.g. too many threads by block
 - e.g. too many registers per thread, spilling is very expensive
- **Memory bound device**
 - High arithmetic computing density needed
- **Spatial locality between threads is crucial**
 - Exploit memory access coalescing capabilities
- **Expensive global synchronization**
 - No global hardware thread communication
- **Performance is very problem size dependent**
 - The number of threads and complexity of threads depend on the number of computations to perform

Evolution (1)

- GPUs are still multiplying compute resources (“cores” is an ambiguous term in GPUs, and not directly comparable across vendors)
 - NVIDIA
 - 8800GTX: 128 “cores”
 - GTX280: 240 “cores”
 - Upcoming “Fermi”: up to 512 “cores”
 - ATI
 - Radeon HD3870: 320 “cores”
 - Radeon HD4870: 800 “cores”
 - Radeon HD5870: 1600 “cores”
- CPUs are doing the same, but at a slower pace
 - Xeon 55xx is still only 4 cores, like the Xeon 54xx
 - Opteron has moved to 6 cores, as did the Xeon 74xx

Evolution (2)

- **Memory bandwidth is still a bottleneck**
 - GPUs can offer 100+ GB/s of bandwidth, but with strict requirements on access patterns
 - Xeon 55xx have more than thrice the bandwidth of the Xeon 54xx. And bandwidth is doubled by going from one to two sockets, as is the case of the Opteron.
- **GPU FP accuracy is now mostly in line with CPU, offering near-complete IEEE754 compliance**
- **GPGPU is still limited by PCIe connectivity**
 - No direct access from the CPU to the GPU memory

Evolution (3)

- CPU vendors try to close the gap
 - On-chip GPU
 - AMD Fusion (Llano), Intel Clarkdale / Arrandale, NVIDIA Tegra
 - Mostly embedded/desktop for now, high-perf to come?
- **Is GPU tending to become General Purpose Unit?**
 - Upcoming NVIDIA Fermi is more flexible than the current GT200 architecture
 - Intel Larrabee presentations insist on the versatility of this upcoming x86-based “GPU”: CPU-GPU convergence
 - GPGPU users push towards an easier-to-program architecture
 - ... but do graphics users agree?
 - Strength of GPU was specialization

GPGPU Today

- Supported on consumer and professional hardwares
 - Tesla S1070: external box hooked to the computer via an interface card
- Programmed with dedicated tools
 - NVIDIA CUDA, ATI Stream SDK
 - OpenCL
 - CAPS HMPP compiler, PGI compiler
- Not as “easy” as traditional homogeneous computing, but getting better
- Only 3 years after wide availability, already some success stories and major deployment
- OpenCL from Khronos consortium
 - C based, low level, great for code generation tool
- Upcoming NVIDIA Fermi architecture
 - Enhanced DP architecture
 - Unified memory space, ECC memory, concurrent kernels, ...

Some References

- Exploring the VLSI Scalability of Stream Processors (2003)
 - by Brucek Khailany , William J. Dally , Scott Rixner , Ujval J. Kapasi , John D. Owens , Brian Towles In International Conference on High Performance Computer Architecture
- Sparse matrix solvers on the GPU: conjugate gradients and multigrid (2003)
 - by Jeff Bolz , Ian Farmer , Eitan Grinspun , Peter Schröder ACM Trans. Graph
- <http://www.gpgpu.org/>
- http://www.nvidia.com/object/cuda_home.html
- <http://www.amd.com/stream/>
- <http://www.intel.com/technology/visual/microarch.htm>
- <http://www.khronos.org/>



CAPS

The logo features the word "CAPS" in a white, italicized, sans-serif font. To the right of the text are three white, 3D rectangular bars of varying heights and positions, creating a stylized graphic element.

Innovative Software for Manycore Paradigms