Beyond Programmable Shading Course
ACM SIGGRAPH 2012

David Luebke, NVIDIA Research

GPU COMPUTING

History & Tech Transfer

Timeline – Hardware

(Ignoring prehistory: Ikonas, Pixel Machine, Pixel Planes, …)

GPU computing eras (NVIDIA):

			multipass rendering…	
2001	NV20	DX8	register combiners	µcode
2003	NV30	DX9	pixel & vertex shaders	Cg
2006	G80	DX10	unified shader, scatter	CUDA
2007	G92		atomics	
2008	GT200		double precision	
2010	Fermi	DX11	full-speed double precision, ECC, L1$	
2012	Kepler		dynamic parallelism	

(A Few) People

Greg James

Mark Harris

Ian Buck

Aaron Lefohn

Mike Houston

Dominick Gödeke

Robert Strzodka

Bill Mark

Jeff Bolz

Cliff Woolley

Kayvon Fatahalian

Naga Govindaraju

Scott Larsen

John Nickolls

Tim Purcell

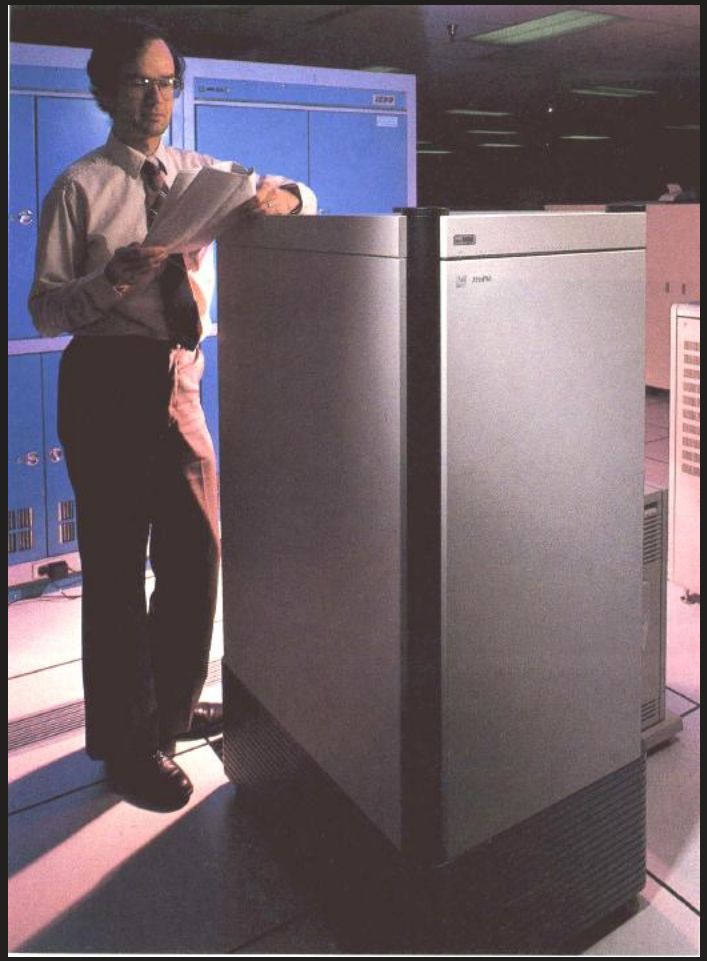
Nolan Goodnight

Tim Foley

many, many more…

(A Few) Pictures

John Nickolls
1950-2011

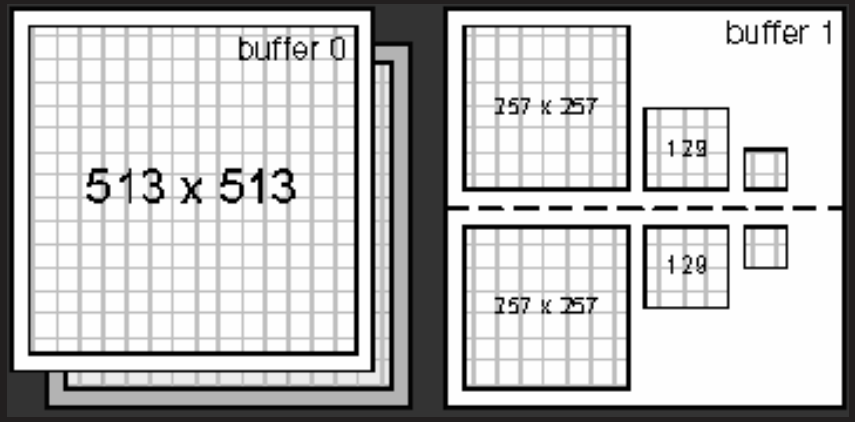


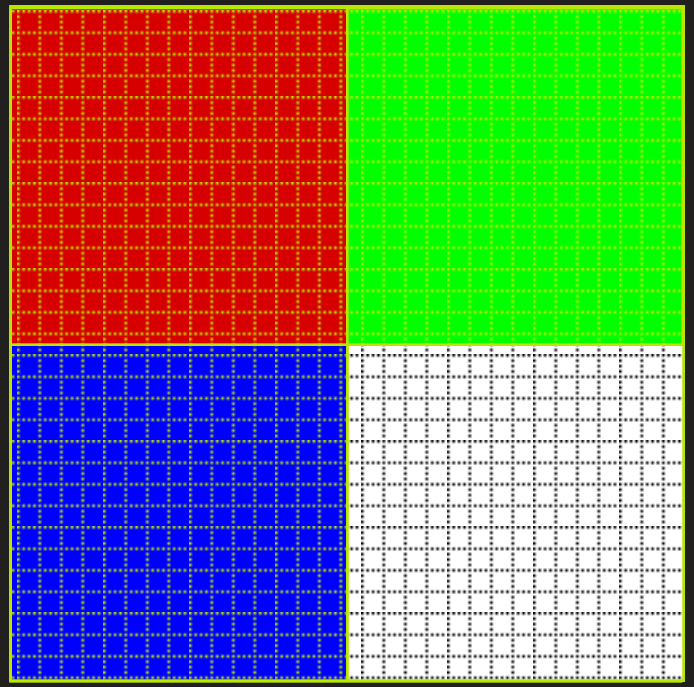
Jim Fischer with MasPar MP1 (front), Goddard MPP (back)

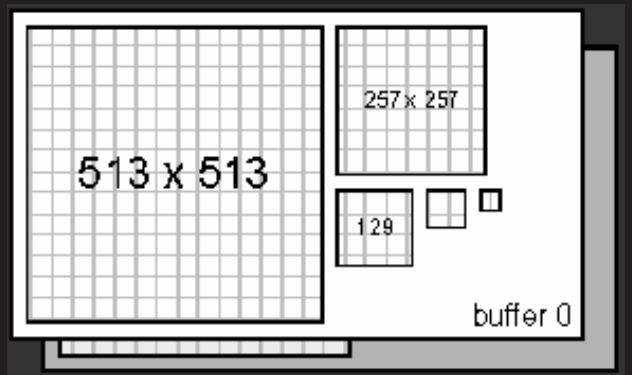
(A Few) Pictures

Smoothing of a noisy image by nonlinear diffusion computed in DirectX7 graphics hardware [Strzodka & Rumph, 2001]

(A Few) Pictures







A Multigrid Solver for Boundary Value Problems Using Programmable Graphics Hardware [Goodnight et al., 2003]

Lessons learned

2004 “road show”: John N, Mark H visit many US companies
[Oil & Gas, Finance, LANL, Mathworks…]

Stanford team working with many domain scientists

- Nobody wants to learn / port to a new language
- Need support for standard libraries: BLAS, FFT, etc
- Memory model (textures) a big problem: want ptrs, arrays, etc.

Consensus

GPU computing definitely inspired by academic work

Still needed brilliant, persuasive champions & leaders

Started small, focused on results: “We gave him a dime.”

A tricky balance: Much GPGPU quickly became pointless

Key GPGPU Researcher Challenges

- Foster high-level libraries, languages, platforms
  - Domain-specific tools & packages
  - “Horizontal” programming layers & patterns
- Motivating use cases: *computational graphics*
- Rethink algorithms, numerics, approaches
  - Computation is cheap, data movement is costly
  - The world is parallel

Final Thoughts – Education

We should teach parallel computing in CS 1 or CS 2

•Computers don’t get faster, just wider

Insertion Sort Heap Sort Merge Sort

Which goes faster on large data?

ALL Students need to understand this! **Early!**