Scalable Parallel Programming with CUDA

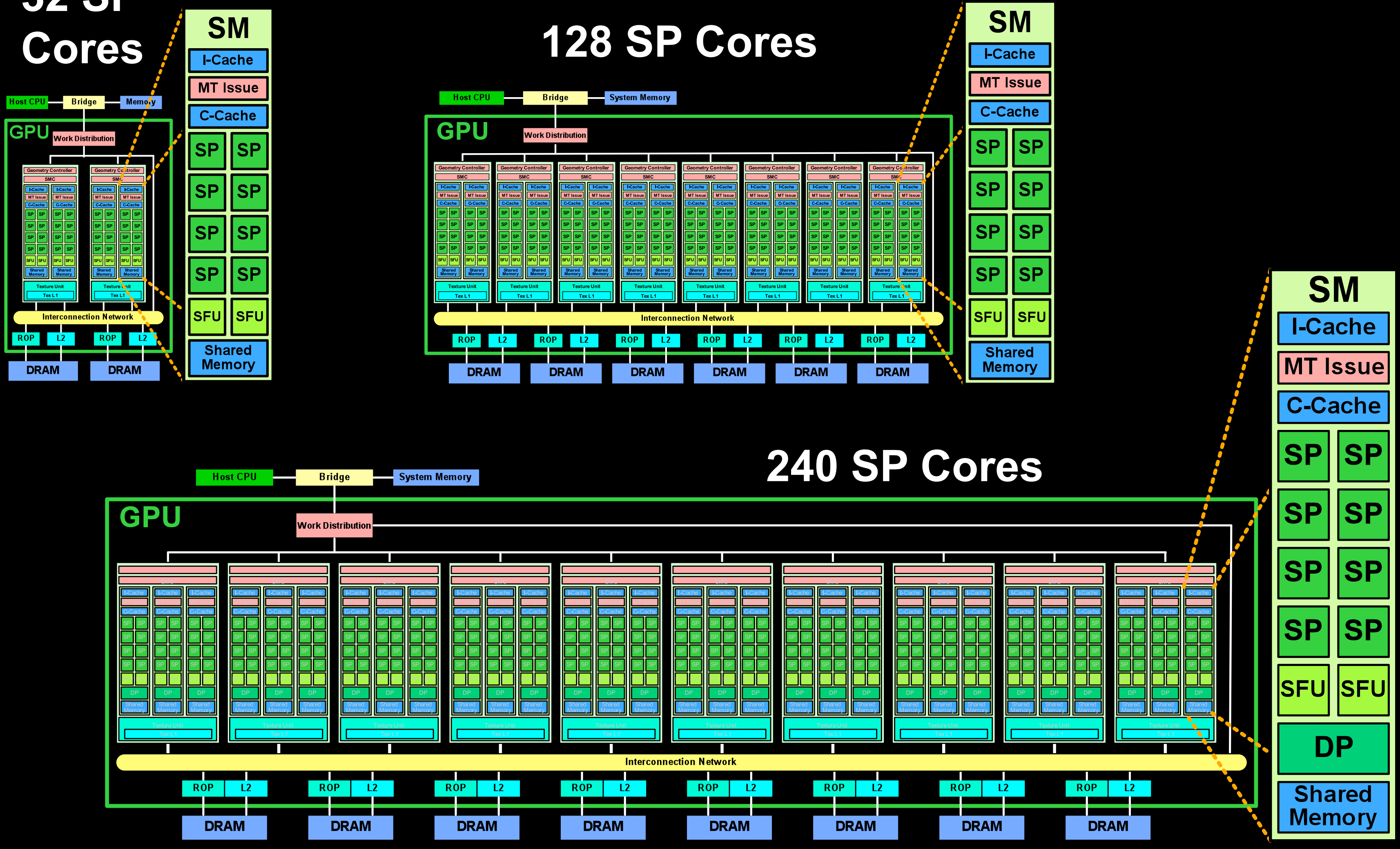
Introduction

John Nickolls

Parallelism is Scaling Rapidly

- CPUs and GPUs are parallel processors
  - CPUs now have 2, 4, 8, … processors
  - GPUs now have 32, 64, 128, 240, … processors
- Parallelism is increasing rapidly with Moore's Law
  - Processor count is doubling every 18 – 24 months
  - Individual processor cores no longer getting faster
- Challenge: Develop parallel application software
  - Scale software parallelism to use more and more processors
  - Same source for parallel GPUs and CPUs

GPU Sizes Require CUDA Scalability



32 SP Cores

128 SP Cores

240 SP Cores

CUDA is C for Parallel Processors

- CUDA is industry-standard C
  - Write a program for one thread
  - Instantiate it on many parallel threads
  - Familiar programming model and language

- CUDA is a scalable parallel programming model
  - Program runs on any number of processors without recompiling

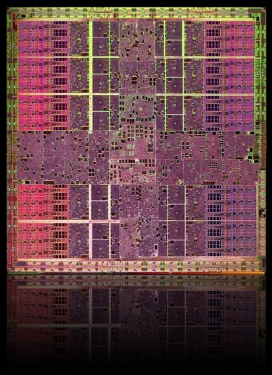
- CUDA parallelism applies to both CPUs and GPUs
  - Compile the same program source to run on different platforms with widely different parallelism
  - Map to CUDA threads to GPU threads or to CPU vectors

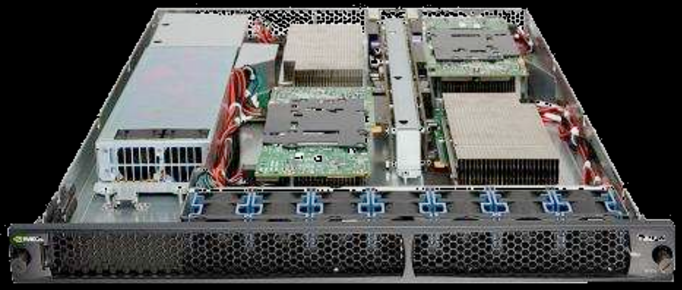
CUDA Uses Extensive Multithreading

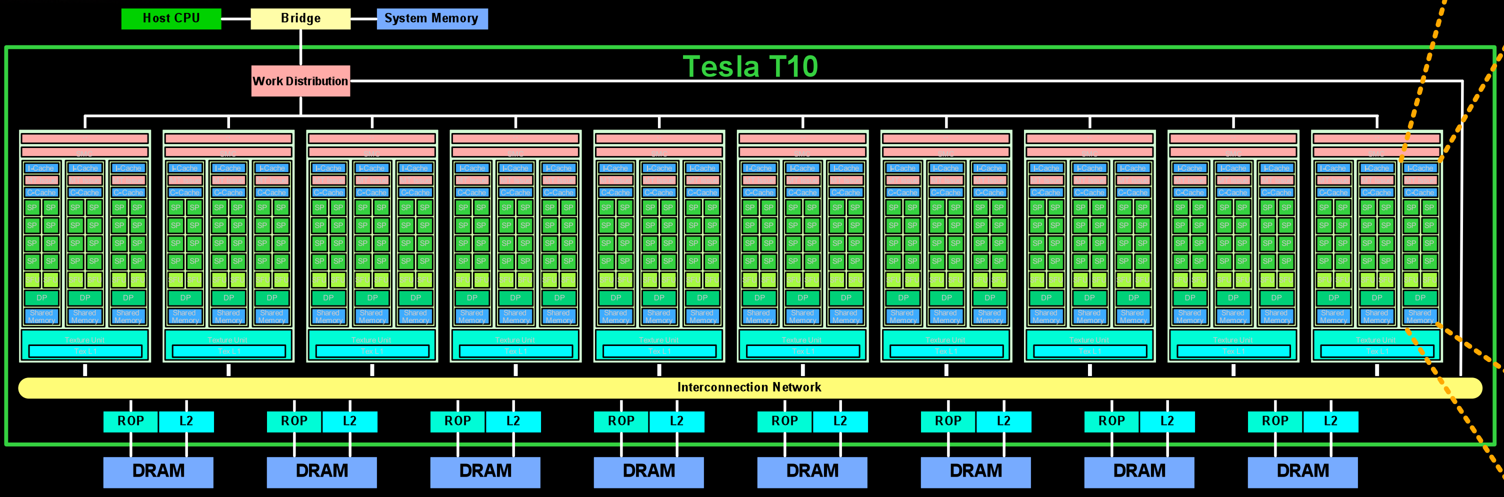
- CUDA threads express fine-grained data parallelism
  - Map threads to GPU threads or CPU vector elements
  - Virtualize the processors
  - You must rethink your algorithms to be aggressively parallel
- CUDA thread blocks express coarse-grained parallelism
  - Map blocks to GPU thread arrays or CPU threads
  - Scale transparently to any number of processors
- GPUs execute thousands of lightweight threads
  - One DX10 graphics thread computes one pixel fragment
  - One CUDA thread computes one result (or several results)
  - Provide hardware multithreading & zero-overhead scheduling

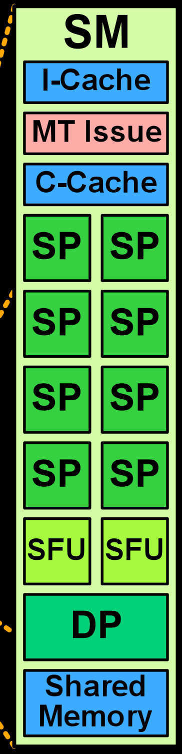
CUDA Computing with Tesla T10

- 240 SP processors at 1.5 GHz: 1 TFLOPS peak
- 128 threads per processor: 30,720 threads total









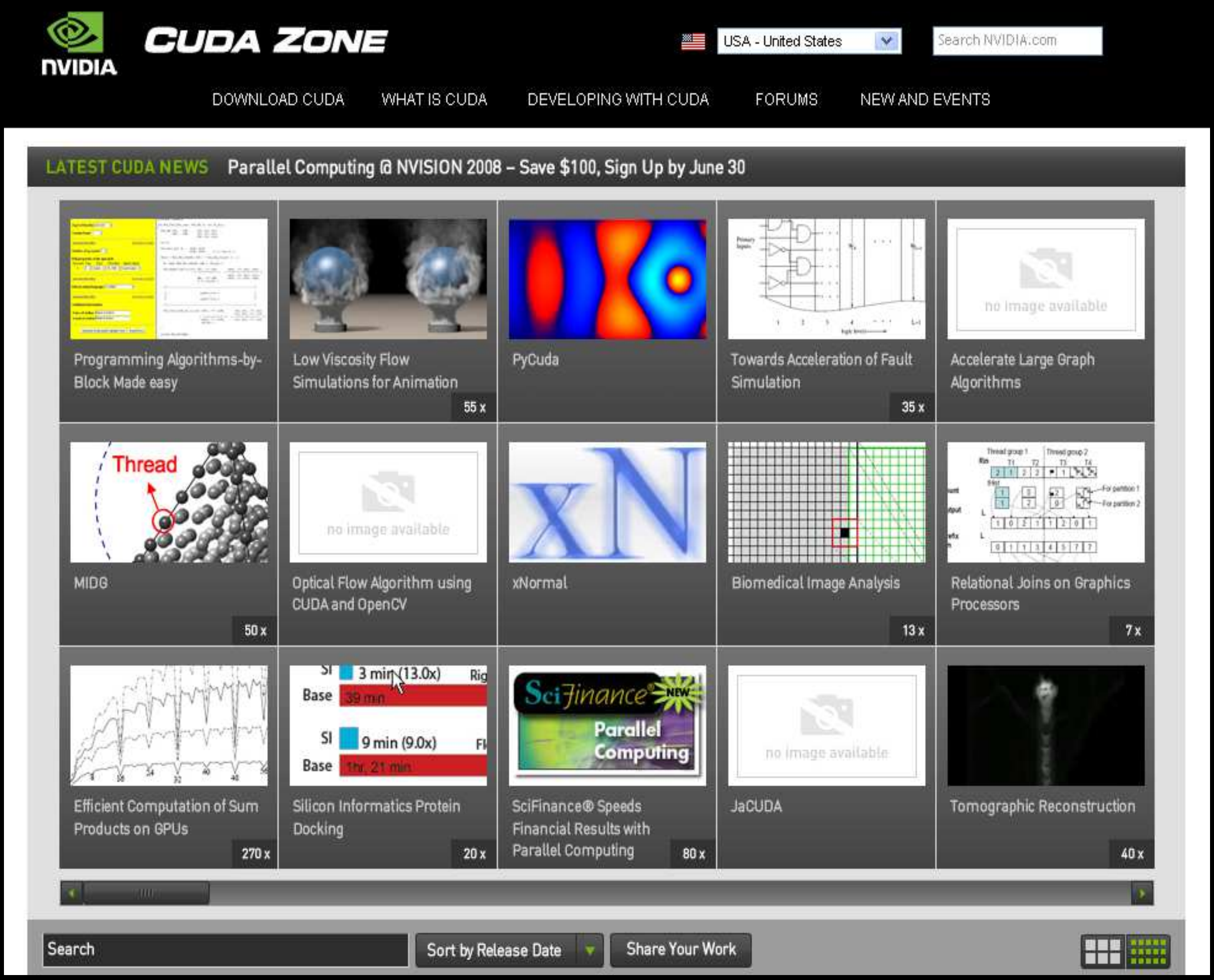
CUDA Computing Sweet Spots

- **Parallel Applications:**
  - **High arithmetic intensity:**
    Dense linear algebra, PDEs, *n*-body, finite difference, …
  - **High bandwidth:**
    Sequencing (virus scanning, genomics), sorting, database, …
  - **Visual computing:**
    Graphics, image processing, tomography, machine vision, …
  - **Computational modeling, science, engineering, finance, …**

Pervasive CUDA Parallel Computing

- CUDA brings data-parallel computing to the masses
  - Over 85 M CUDA-capable GPUs deployed since Nov 2006
- Wide developer acceptance
  - Download CUDA from www.nvidia.com/CUDA
  - Over 50K CUDA developer downloads
  - A GPU “developer kit” costs ~$200 for 500 GFLOPS
- Data-parallel supercomputers are everywhere!
  - CUDA makes this power readily accessible
  - Enables rapid innovations in data-parallel computing
- Parallel computing rides the commodity technology wave

CUDA Zone: www.nvidia.com/CUDA



LATEST CUDA NEWS Parallel Computing @ NVISION 2008 – Save $100, Sign Up by June 30

Programming Algorithms-by-Block Made easy

Low Viscosity Flow Simulations for Animation 55 x

PyCuda

Towards Acceleration of Fault Simulation 35 x

Accelerate Large Graph Algorithms

MIDG 50 x

Optical Flow Algorithm using CUDA and OpenCV

xNormal

Biomedical Image Analysis 13 x

Relational Joins on Graphics Processors 7 x

Efficient Computation of Sum Products on GPUs 270 x

Silicon Informatics Protein Docking 20 x

SciFinance® Speeds Financial Results with Parallel Computing 80 x

JaCUDA

Tomographic Reconstruction 40 x

- **Resources, examples, and pointers for CUDA developers**