# CSE 591: GPU Programming

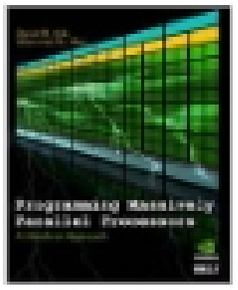
## Basics on Architecture and Programming

Klaus Mueller

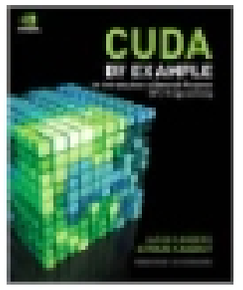
Computer Science Department

Stony Brook University

## Recommended Literature

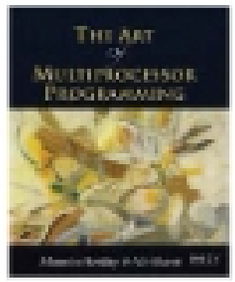


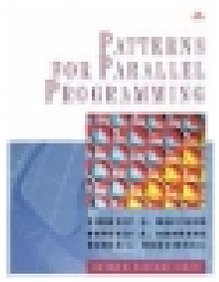
text book




reference book

programming guides
available from nvidia.com




more general books on
parallel programming

## Course Topic Tag Cloud

Architecture

Limits of parallel programming

Host

Performance tuning

Kernels

Debugging

Thread management

OpenCL

Algorithms

Memory

CUDA

Device control

Example applications

Parallel programming

## Course Topic Tag Cloud

Architecture

Limits of parallel programming

Host

Performance tuning

Kernels

Debugging

Thread management

OpenCL

Algorithms

Memory

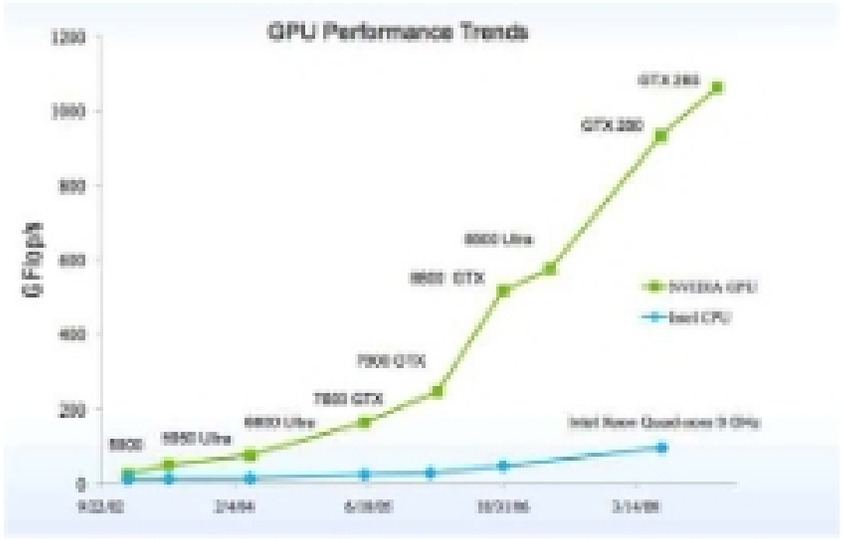
CUDA

Device control

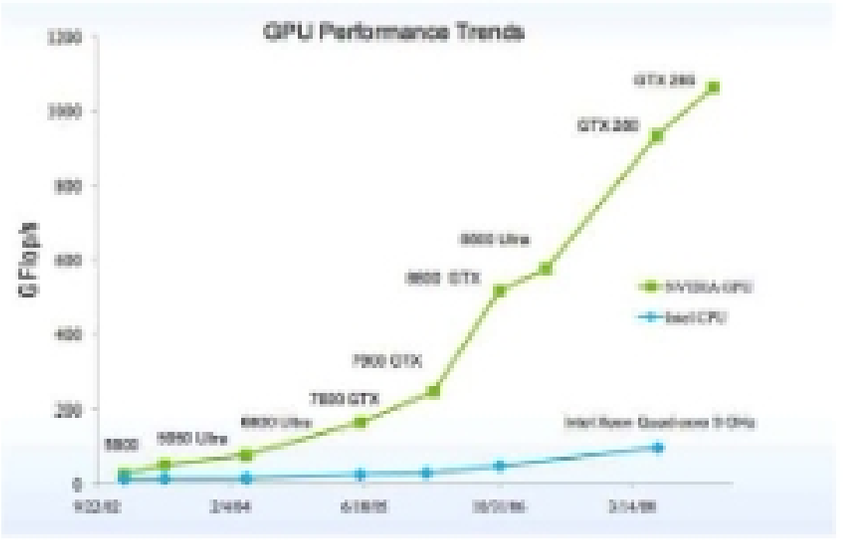
Example applications

Parallel programming

## Speedup Curves




## Speedup Curves




but wait, there is more to this…..

## Amdahl's Law

Governs theoretical speedup

$$S = \frac{1}{(1-P) + \frac{P}{S_{parallel}}} = \frac{1}{(1-P) + \frac{P}{N}}$$

P: parallelizable portion of the program
S: speedup
N: number of parallel processors

## Amdahl's Law

Governs theoretical speedup

$$S = \frac{1}{(1-P) + \frac{P}{S_{parallel}}} = \frac{1}{(1-P) + \frac{P}{N}}$$

P: parallelizable portion of the program
S: speedup
N: number of parallel processors

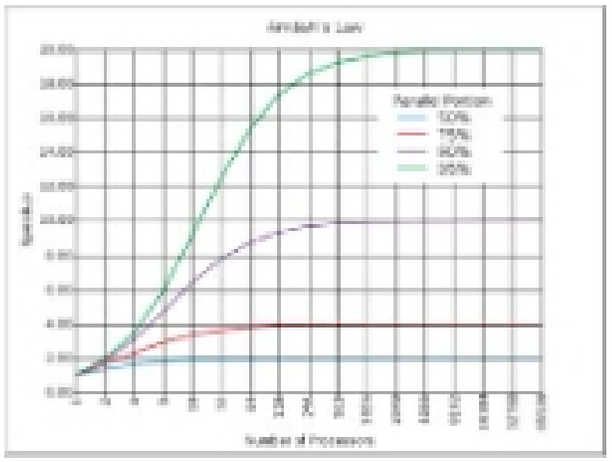
P determines theoretically achievable speedup

* example (assuming infinite N): P=90% → S=10
  P=99% → S=100

## Amdahl's Law

How many processors to use

- when P is small → a small number of processors will do
- when P is large (embarrassingly parallel) → high N is useful



## Focus Efforts on Most Beneficial

Optimize program portion with most 'bang for the buck'

- look at each program component
- don't be ambitious in the wrong place

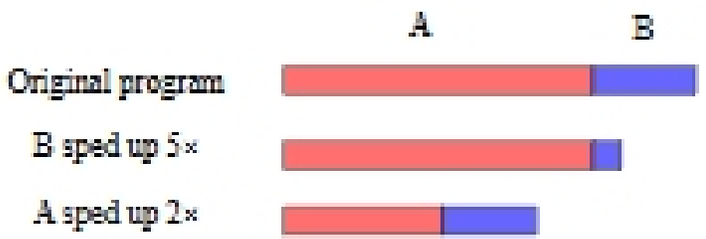
## Focus Efforts on Most Beneficial

Optimize program portion with most 'bang for the buck'

- look at each program component
- don't be ambitious in the wrong place

Example:

- program with 2 independent parts: A, B (execution time shown)




- sometimes one gains more with less

## Beyond Theory....

Limits from mismatch of parallel program and parallel platform

- man-made 'laws' subject to change with new architectures