GPU, Multi/Many Core Computing I: Introduction to HPC

Exercise 3: Stochastic simulations, random numbers, MPI

Exercise 1

- Performance
  - focus of this course is HPC => performance affects grade
  - use native instructions whenever possible & try alternatives
  - proper analysis important
- Superscalar architecture
  - computations and transfers can be overlapped
  - T = max(C/Cd, M/Md)
- Report your results properly
  - especially when results look weird => comment on it
  - put relevant info in pdf => avoid “hidden” text-files

Corrections

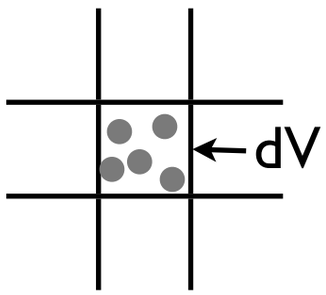
- Grading includes effort, correctness and quality
  - Extra-points possible for extra-nice additional work
  - Effort includes completed work and good report
  - Quality includes proper analysis and good performance
- Late hand-ins = -10% per day
  - i.e. 90 points achieved 2 days too late = 72 points

Random walk

- How can one simulate diffusion with a random walk?

$$\frac{\partial u}{\partial t} = \nu \Delta u \quad \text{—— ?? ⟶} \quad \mathbf{x}(t_{n+1}) = \mathbf{x}(t_n) + \boldsymbol{\xi}(t_n),$$
$$\xi_i \sim \mathcal{N}(0, 2\,\nu\,\delta t)$$

- u = concentration of some species = N / dV



- "walk" = diffusive motion of particles
  - model diffusive motion of N particles (per unit volume)
  - will converge for large N

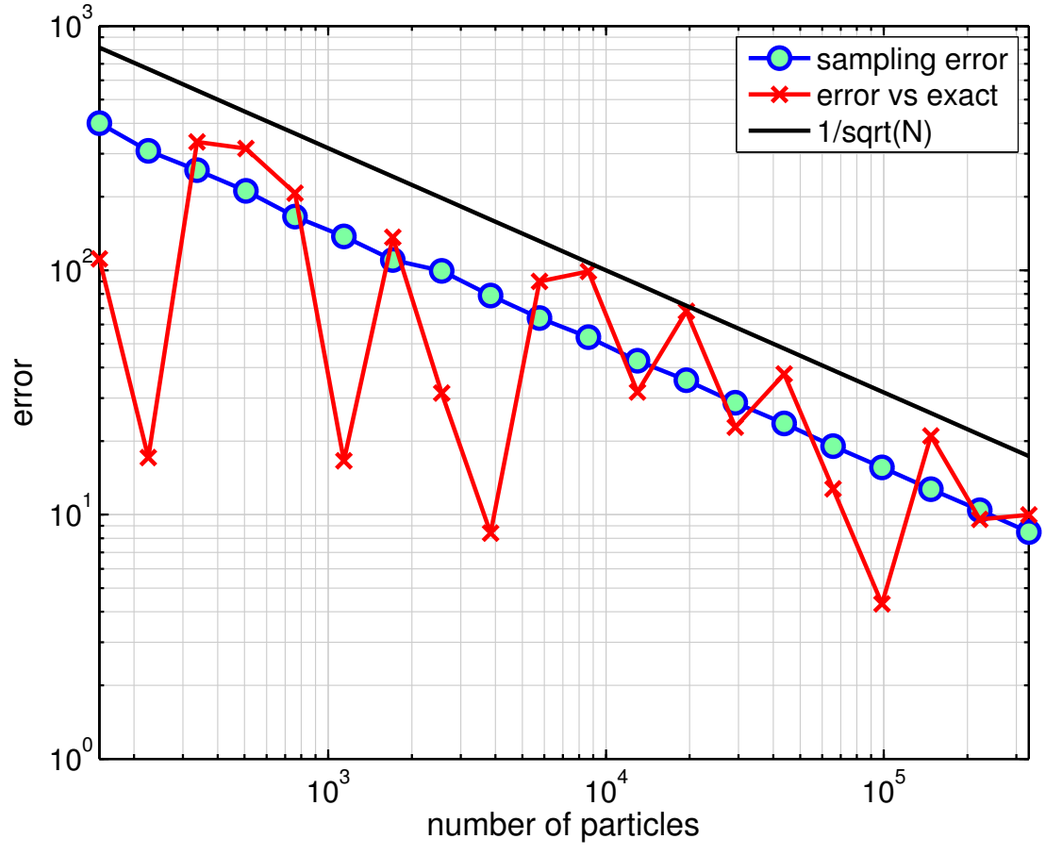
Reduction

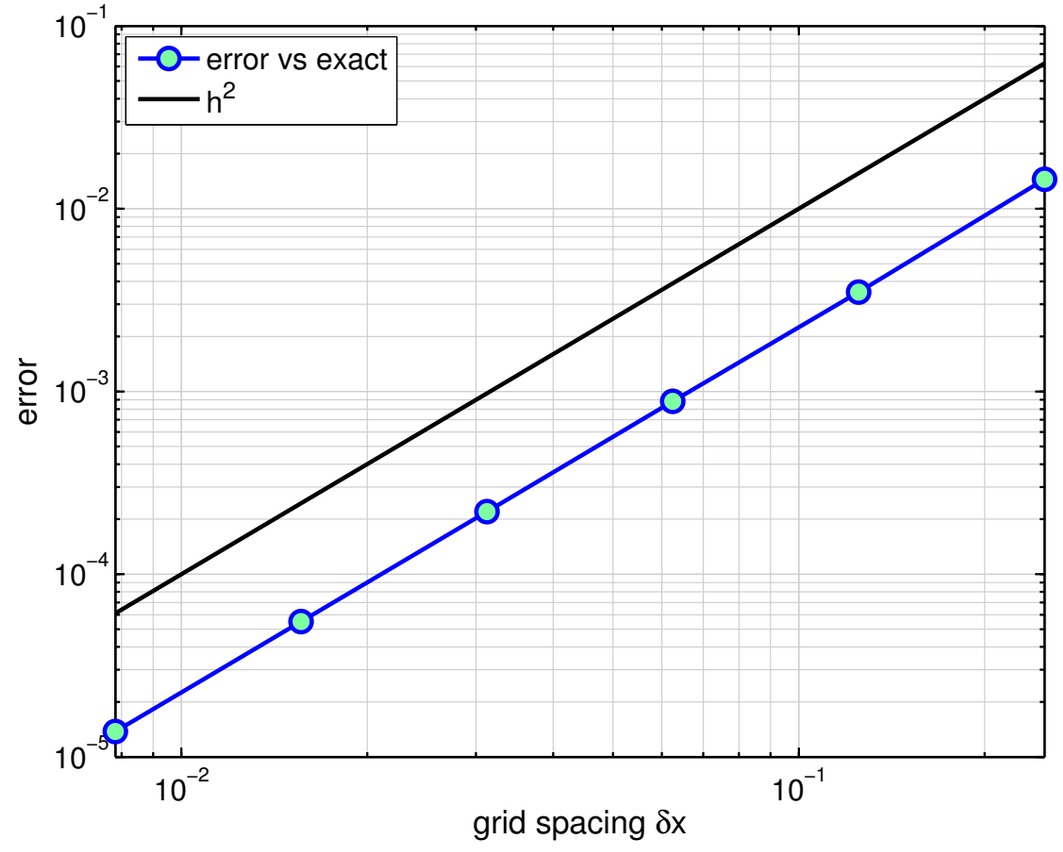
- Seen in lecture:
  - Class Oct-5-2011 => page 45ff (src: CUDA SDK)
    - see also http://developer.download.nvidia.com/compute/cuda/1_1/Website/projects/reduction/doc/reduction.pdf
  - AFDS 2011: Introduction to OpenCL => page 19ff
  - GTC 2010: An Introduction to CUDA C => page 35ff
- You might want to try using the CPU for last sums...

Convergence revisited

- Convergence = error to fixed exact solution vs ...
  - number of samples, particles or grid points
  - (for iterative methods it might also be number of iterations)

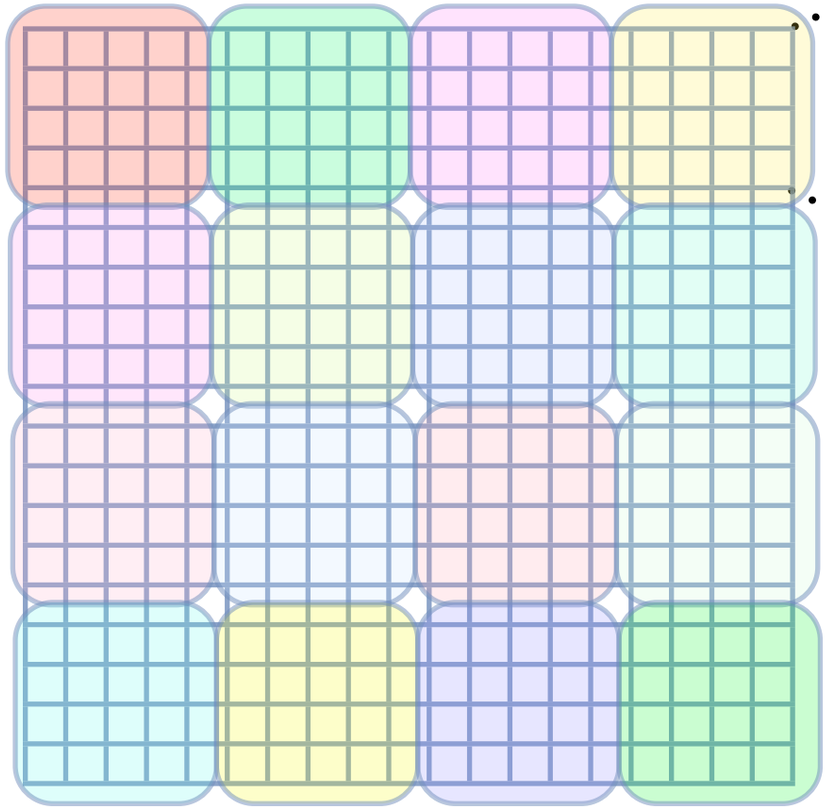
Examples

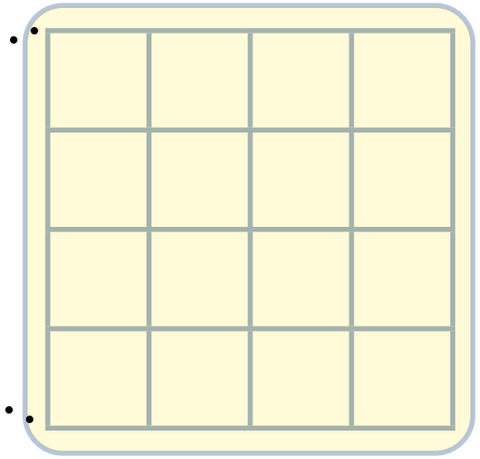




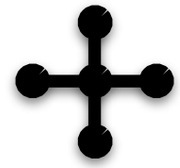
Ghost cells

Domain decomposition

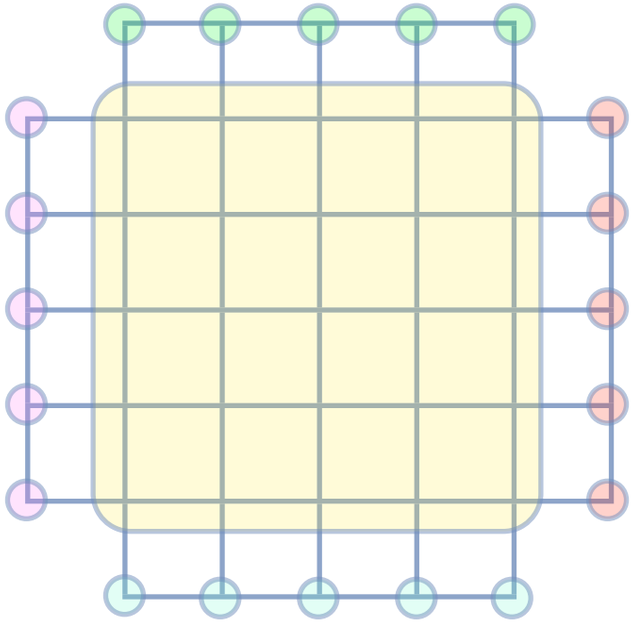




$$\frac{\partial u_{i,j}}{\partial t} = RHS(u_{i,j})$$



Need additional points
for 5-point stencil

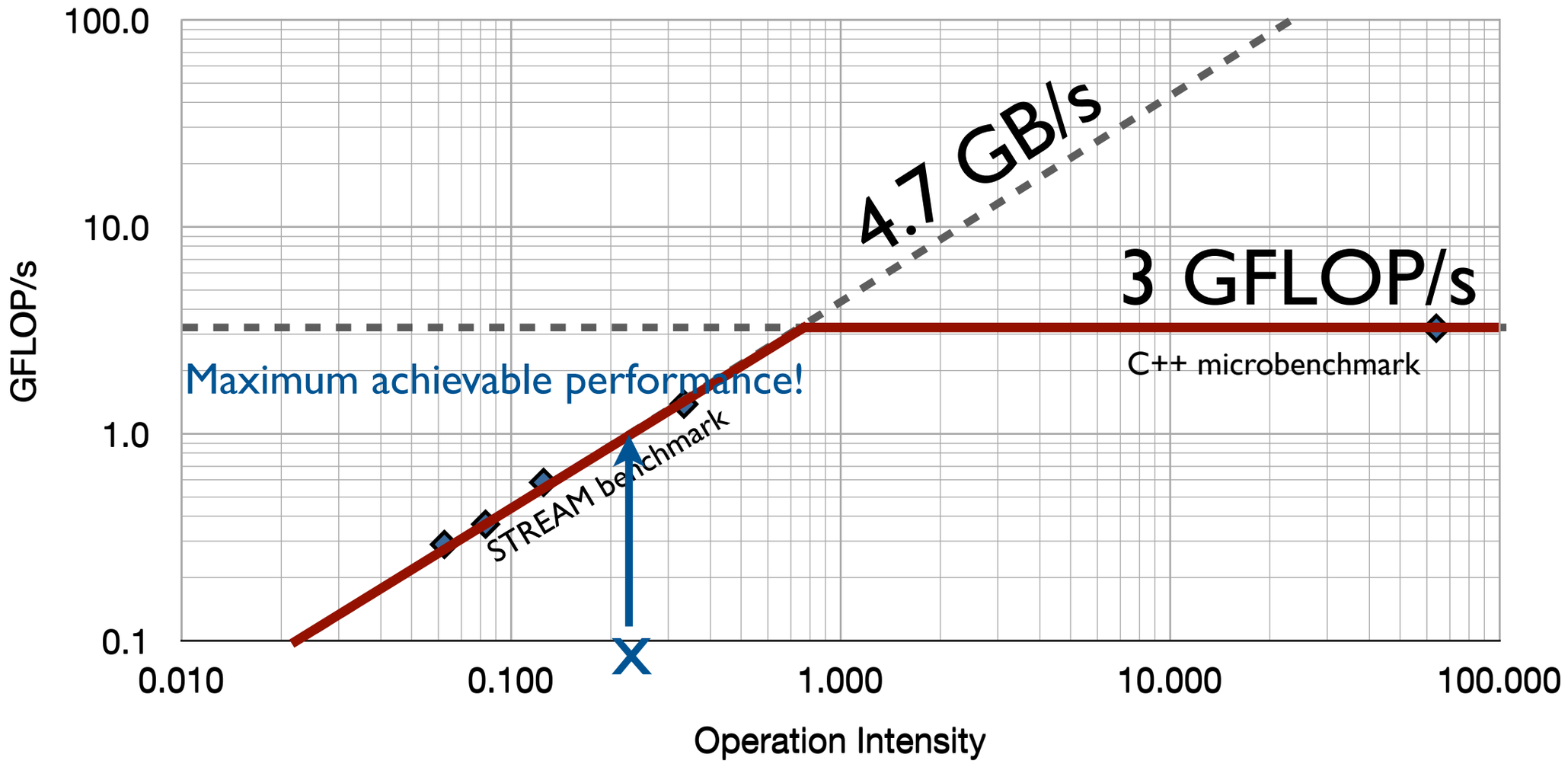


Evaluate RHS(u):

1.) Send/Receive ghosts

2.) Do finite diff. as usual

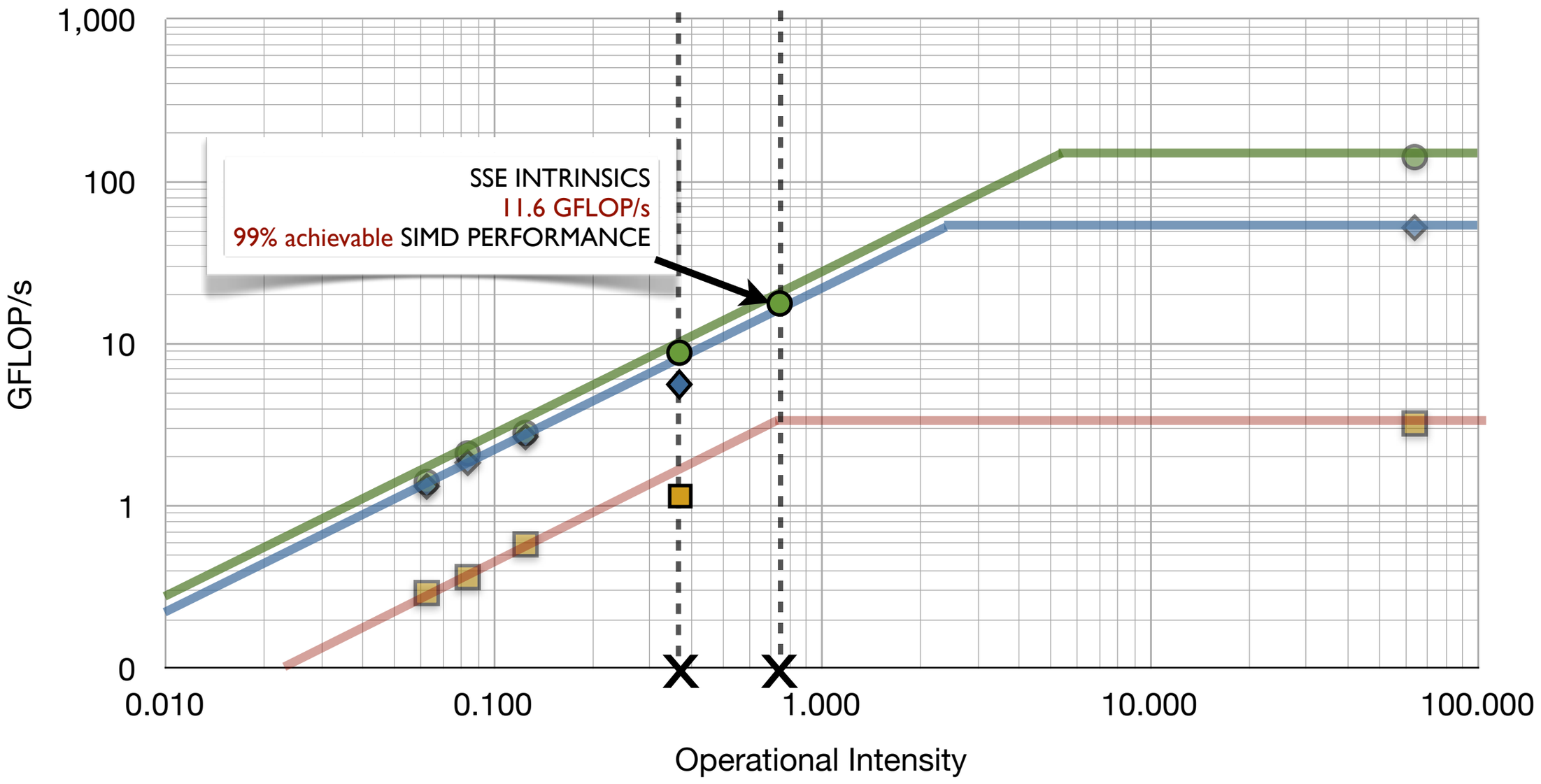
The roofline model



◆ 4x Quad-Core AMD Opteron 8380 @ 2.5GHz - 1 Thread - C++

→ P = min(R · Peak Bandwidth, Peak Compute Power) [GFLOP/s]

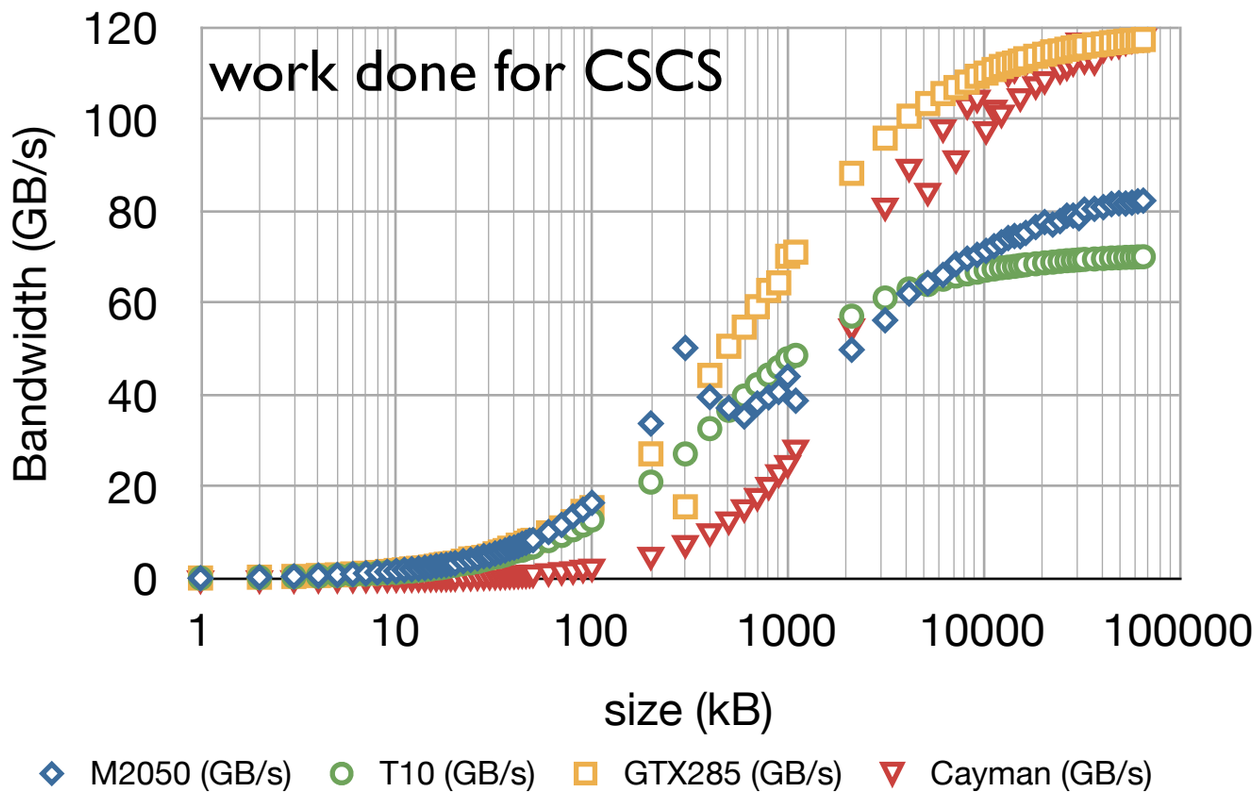
Roofline example



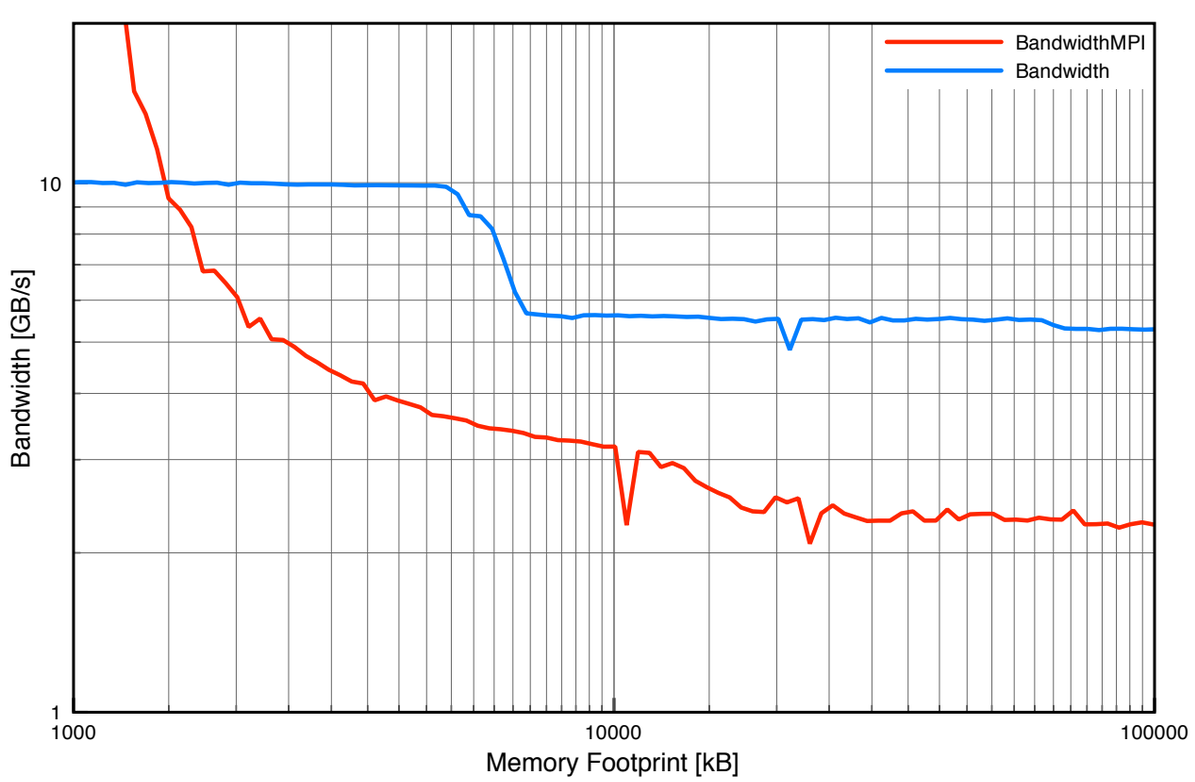
Bandwidth

- Tricky to measure, trickier to interpret
  - best-case assumption: all memory transfers are optimal

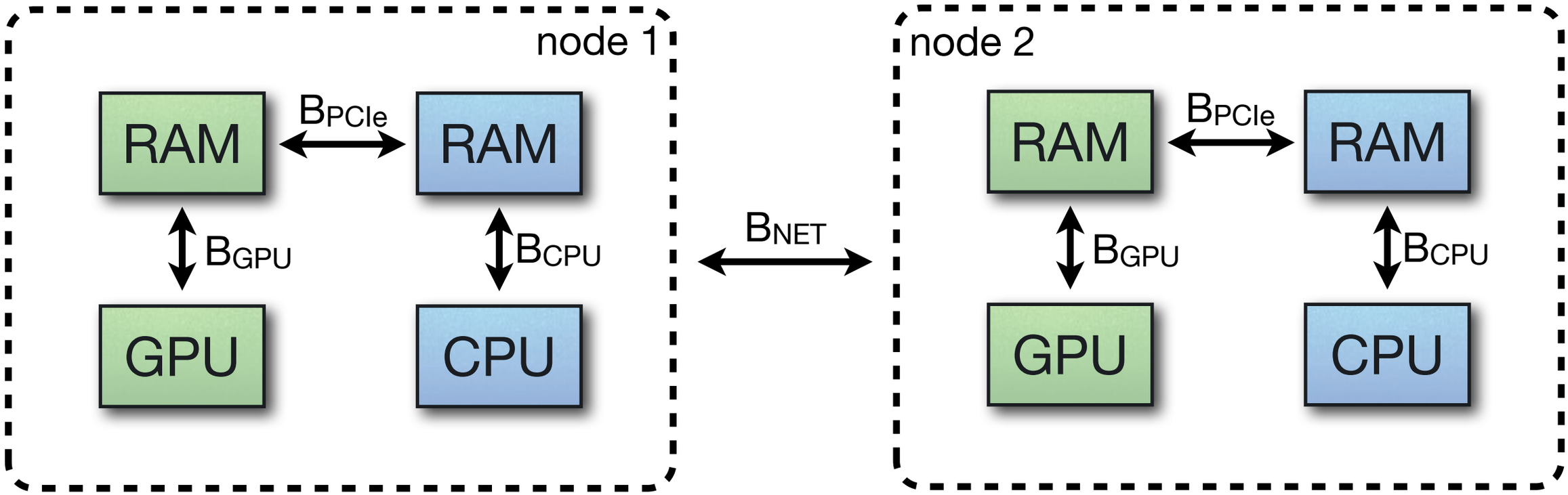
GPU bandwidth



CPU bandwidth (1 core vs MPI)



MPI transfers



- MPI needs copy on RAM before sending
- Send M bytes from CPU to CPU: (i.e. BandwidthMPI)
  - $T_{CPU} = M/B_{CPU} + M/B_{NET} + M/B_{CPU} + Latency$
- GPU to GPU: $T_{GPU} = M/B_{PCIe} + T_{CPU} + M/B_{PCIe}$

Monte Carlo

- Source: CUDA SDK
- Used in model for option pricing
  - Y independent Monte Carlo evaluations with N samples each to evaluate Y options
    - Optimizations also for small N and large Y
    - Single thread might do more than 1 M.C. sample
  - sum-reduce as in CUDA SDK reduction example