Thread Batching

- Kernel launches a grid of thread blocks
  - Threads within a block can cooperate via shared memory
  - Threads within a block can sync
  - Threads in different blocks cannot cooperate
- Allows programs to transparently scale to different GPUs

Kernel Memory Access

- Per-thread:

  Register, Very fast on-chip memory

  Off-chip, uncached
- Per-block: Shared memory, on-chip small and fast
- Per-device: Off-chip large, persistent across kernel launches, Kernel I/O

CUDA Variable Speeds

- Memory on register, shared and constant are very fast. Local and global are much slower. Try to avoid the latter two. Important to use the right variables in the right place.

CUDA Performance

- GPUs only suitable for statistics when calculations are highly parallelizable and take significant time.
- Only for very large problems.
-
  - numerical integration
  - MCMC with very slow iterations (within-iteration parallelism)
  - "Simple" bootstraps
  - particle filtering (sequential monte carlo)
  - extremely difficult brute force optimization
  - Large matrix calculation
  - Single use applications, code is not very portable, hard for others to use.
- Not good for:
  - Fast iteration MCMC
  - "difficult" bootstrap
  - optimization problems
  - methodological work
  - any problem not worth the effort

RCUDA

- Provides CUDA API for R
- Calls functions within CUDA API inside of R
- Hides some of the memory management stuff
- Kernel still needs to be written in CUDA C (For homework going to have to write C):)
- Kernals are compiled to `ptx` code using `nvcc --ptx`
- Kernels are loaded via modules into R

Homework

Write kernel to generate truncated random normals

Call from R to do tests, timings, etc

ProbitMCMC

$$y_i|z_i = I_{\{z_i>0\}}$$

$$z_i|\beta \sim N(X_i^T\beta, 1)$$

EM: Find

$$\text{argmax}_\beta P(y|\beta)[= \int p(y|z)p(z|\beta)dz]$$

MCMC: Prior for β: $\beta \sim N(\beta_0, \Sigma_0)$. Sample from $p(\beta|y)$ using Gibbs.

$$\mathbf{z} = (z_1, \ldots, z_n) \quad p(\beta|z, y) \leftarrow \text{Normal}$$

$$P(\mathbf{z}|\beta, y) \leftarrow \text{truncated normals}$$