Configuring Block/Grid Dimensions

Goal: Maximize throughput and utilization of SMs on GPU

General Strategies

- Do adequate work per thread to amortize overhead
- The grid should be sufficiently large to get multiple (3 or 4) Blocks (CUDA) or Work Groups (OpenCL) per SM.

If the number of Blocks/WGs is fewer than number of SMs, the GPU may be able to execute another kernel at the same time if it is on another stream (CUDA) or queue (OpenCL).

If there are enough Blocks/WGs, whether a SM can actually have more than one Block/WG resident and available to execute at a time depends on (i) shared memory per Block/WG, (ii) registers per Block/WG, and (iii) threads per Block/WG.

Numerical Targets

- Occupancy of a SM (see book, pp. 416-417)
  \[
  \text{occupancy} = \frac{\text{numberWarpsOnSM}}{\text{maxWarpsPerSM}}
  \]
  Denominator is just maxThreadsPerSM/warpSize
  Numerator is influenced by shared memory and register usage
- Threads per block/WG:
  \[
  \begin{align*}
  &\quad \text{should be a multiple of warp size.} \\
  &\quad \text{should be sufficiently large to exploit multiple warp schedulers on newer SMs.} \\
  &\quad \text{(Multiple warp schedulers allow multiple warps to run simultaneously on an SM.)}
  \end{align*}
  \]
- Tradeoff: Registers per thread versus shared memory usage

Caveats

- The formula given in the book (see below) and its implementation assume some data whose values are difficult to obtain.
- There are CUDA – OpenCL differences in terms of how and what data can be queried as well.
- Hence these formulas and code examples represent rough initial guidelines.
One Implementation of the numerical targets for Block (Work Group) size

\[
\text{threadsPerBlock} = \min \left( \frac{\text{numWarpSchedulers} \times \text{warpSize}}{\text{maxRegsPerBlock} / \text{regsPerThread}}, \frac{\text{maxSharedMemPerBlock} / \text{sharedMemPerThread}}{\text{max Threads Per SM}} \right)
\]

- Then round \( \text{threadsPerBlock} \) down, if necessary, to multiple of warp size
- Then compute total number of blocks (or global work size) using \( \text{ceil}(\text{totalNumThreads} / \text{threadsPerBlock}) \).
- See/run \text{executionConfHeur.cu}. 