2D reduction

Thursday, November 4, 2021 15:35

Input

problem size

$m \times n$

Hardware Sources

① #sm (how many blocks can run concurrently)

#maxT / #maxW → upper bound for threads | warps | block

② #minT / #minW → lower bound for threads | warps | block to cover FGMT

e.g. #sm=16 → #maxT = 1024 / #maxW = 32

#minT = 512 / #minW = 16

③ RF / shared mem per sm

– check when generating Algorithm

Concern

① efficiency or performance

e.g.1 16 × 128 → 16×1

– performance: 16 blocks on 16 sm

each block has 128 threads

e.g.2 8×16×128 → 8×16×1

– efficiency: for each 16×128

we use 2 blocks

each block has 1024 threads

② we assume each block has similar run time

so it's best to have k × #sm blocks

If n < 32

only consider efficiency scheduling

① $n = 2, 4, 8, 16$ → assign $\frac{32}{n}$ rows to 1 warp

Algo: one warp reduce and get results for $\frac{32}{n}$ rows

② $n \neq 2, 4, 8, 16$ → assign 32 rows to 1 warp

Algo: one warp first read n*32 elements to shared mem and each thread add one row

TW = Total number of warps needed

$\leq k \times (\underbrace{\#sm \times \#maxW}_{512})$

e.g. 768 ≤ 2 × 512

option1: one kernel launch → 32 blocks

each block has $\frac{768}{32}$ = 24 warps

option2: kernel launch → 16 blocks with 32 warps

kernel launch → 16 blocks with 16 warps

scheduling performance

If m < #sm

– we can assign one or more sm to one row

– one sm only need to compute one row

block formation

#bru (blocks per row upper bound) = $\lfloor \frac{\#sm}{m} \rfloor$

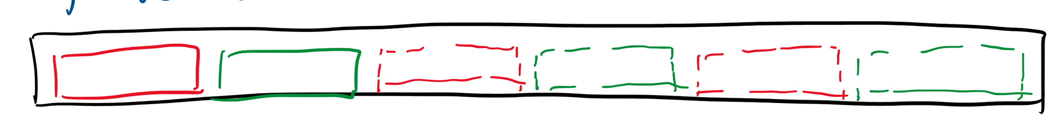
e.g. #sm = 16, m = 15, #bru = 1

#br = min{bru, max{1, $\lfloor \frac{n}{\#minT} \rfloor$}}

e.g. bm = 2, n = 200, #minT = 512 → #br = 1

#wb = min{#maxW, $\lfloor \frac{n}{32 \times \#br} \rfloor$}

Algo detail



– intra row → block-stride loop

– inter-blocks → atomic add to global mem

– inter-warp → atomic add to global mem
  atomic add to shared mem
  non-atomic write to shared mem

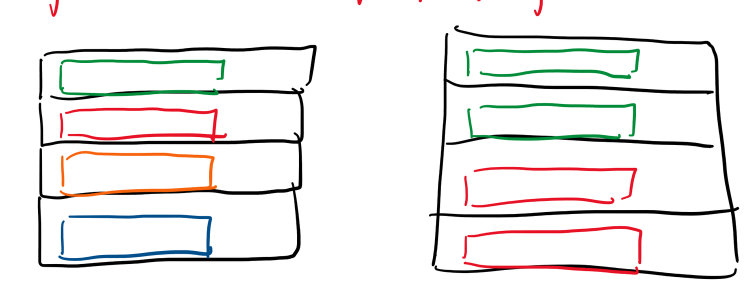
If m > #sm

wru (warps per row upper bound) = $\lfloor \frac{\#maxW \times \#sm}{m} \rfloor$

#wr = min{#wru, $\lfloor \frac{n}{32} \rfloor$}

If #wr > 1

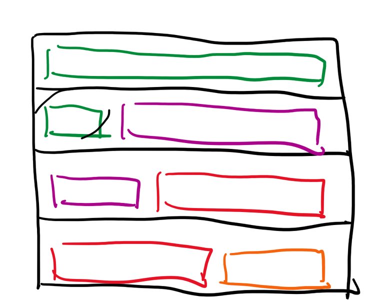
option 1: 1 or k rows per block



– no sync between blocks

– simple control flow

option2: allow 1 row belong to multiple blocks



– wr > #minT, $\frac{\#maxT}{2}$ → 1 row per block

– #minT < wr < $\frac{\#maxT}{2}$ → k rows per block (k ≥ 1)

– wr < #minT, $\frac{\#maxT}{2}$ → k rows per block (k > 1)

e.g.1 m = 25, wr = 20 → option 2

16 blocks, each blocks has 32 warps

e.g.2 m = 31, wr = 16 → option 1

32 blocks, each block has 16 warps

If wr < 1

#wru = $\lfloor \frac{k \times \#maxW \times \#sm}{m} \rfloor$

smallest k for #wru > 1

scheduling efficiency

wr = $\lfloor \frac{n}{32} \rfloor$

#maxW warps per block

$\frac{m \times wr}{\#maxW}$ blocks