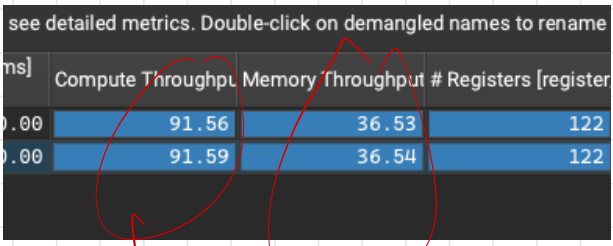


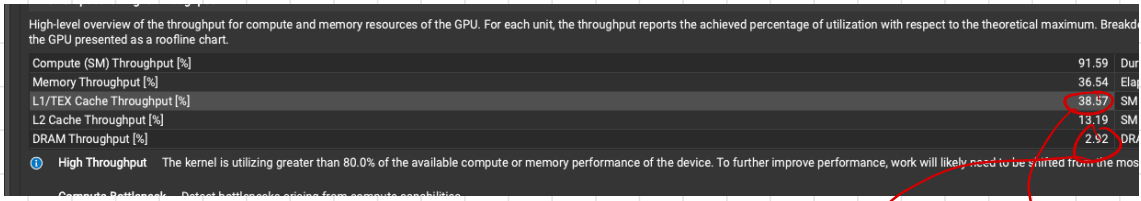
↳ much less memory pres.

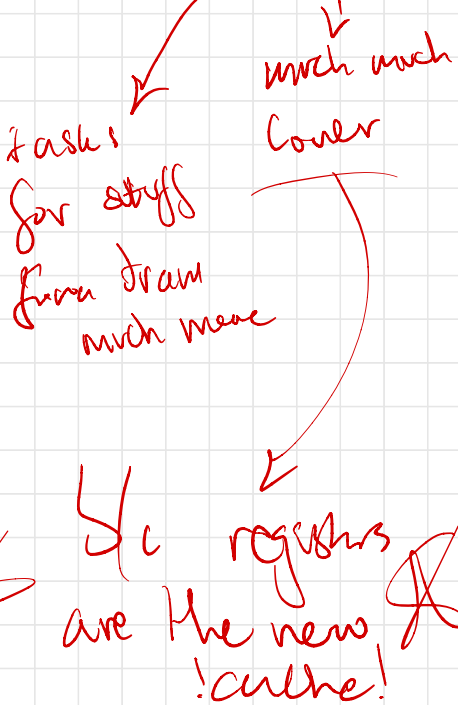
holy: from 63 → 70 → 92

w/ cublas

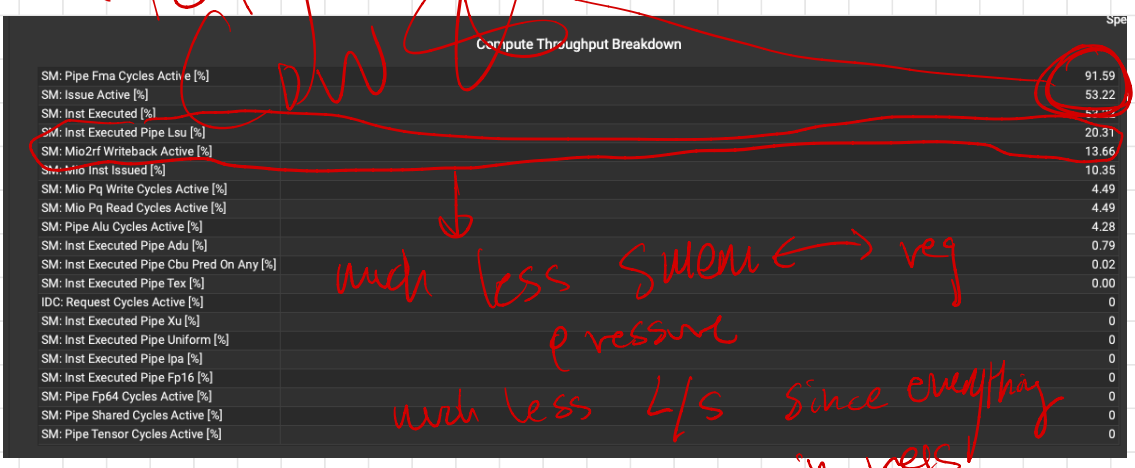
likely due to

- register tiling (massive AI ↑)
  (uses 122 regs)
- async
- double buffering

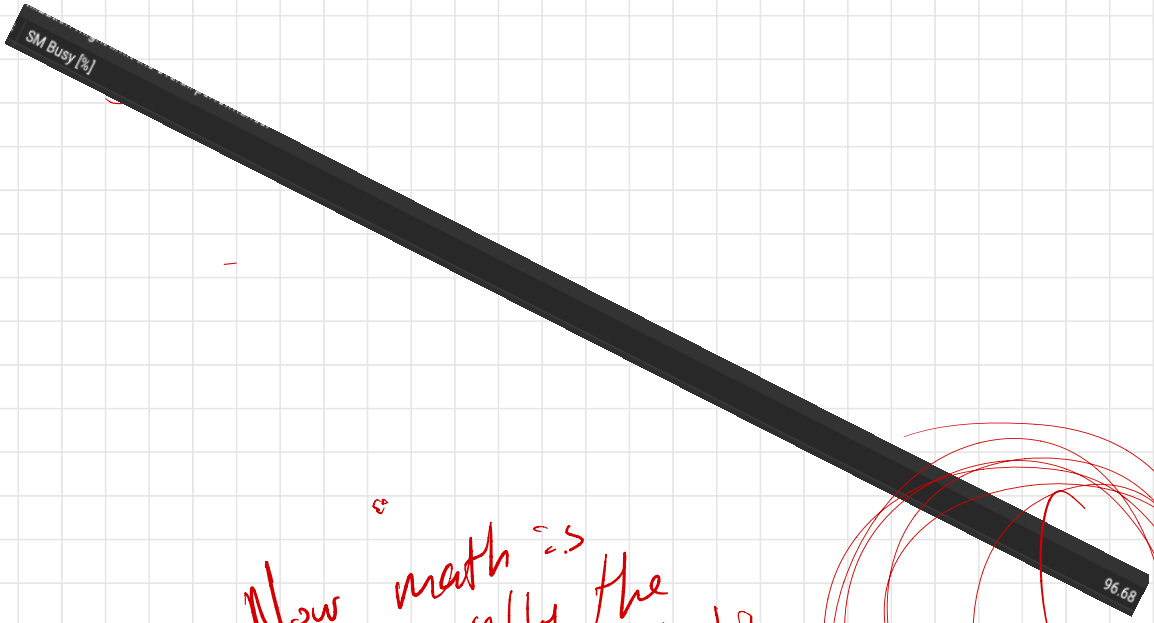

High-level overview of the throughput for compute and memory resources of the GPU. For each unit, the throughput reports the achieved percentage of utilization with respect to the theoretical maximum. Breakd
the GPU presented as a roofline chart.
Compute (SM) Throughput [%] 91.59 Dur
Memory Throughput [%] 36.54 Ela
L1/TEX Cache Throughput [%] 38.57 SM
L2 Cache Throughput [%] 13.19 SM
DRAM Throughput [%] 2.92 DRA
High Throughput The kernel is utilizing greater than 80.0% of the available compute or memory performance of the device. To further improve performance, work will likely need to be shifted from the mos


much much
Cover
tasks
for stuff
from dram
much more
Yo registers
are the new
!cache!

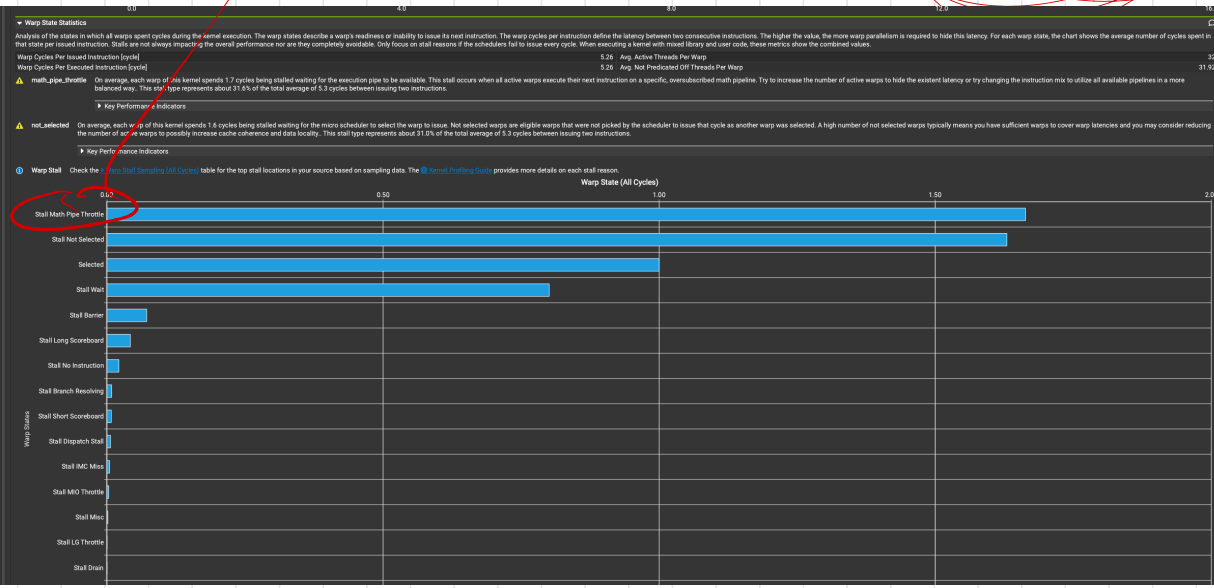

Holy


Compute Throughput Breakdown
SM: Pipe Fma Cycles Active [%] 91.59
SM: Issue Active [%] 53.22
SM: Inst Executed [%]
SM: Inst Executed Pipe Lsu [%] 20.31
SM: Mio2rf Writeback Active [%] 13.66
SM: Mio Inst Issued [%] 10.35
SM: Mio Pq Write Cycles Active [%] 4.49
SM: Mio Pq Read Cycles Active [%] 4.49
SM: Pipe Alu Cycles Active [%] 4.28
SM: Inst Executed Pipe Adu [%] 0.79
SM: Inst Executed Pipe Cbu Pred On Any [%] 0.02
SM: Inst Executed Pipe Tex [%] 0.00
IDC: Request Cycles Active [%] 0
SM: Inst Executed Pipe Xu [%] 0
SM: Inst Executed Pipe Uniform [%] 0
SM: Inst Executed Pipe Ipa [%] 0
SM: Inst Executed Pipe Fp16 [%] 0
SM: Pipe Fp64 Cycles Active [%] 0
SM: Pipe Shared Cycles Active [%] 0
SM: Pipe Tensor Cycles Active [%] 0

much less smem ⟷ reg
pressure
much less L/S since everything
in regs!



Now math is actually the bottleneck !?



18-1 occupancy compared to my 44-1.

Naive bottleneck: Uncoalesced loads

Tiled bottleneck: Shared mem bank conflicts

Cublas bottleneck?: Maybe 122 regs/thread → only 18-1 occupancy → much more ILP