Lecture 20: Parallel performance

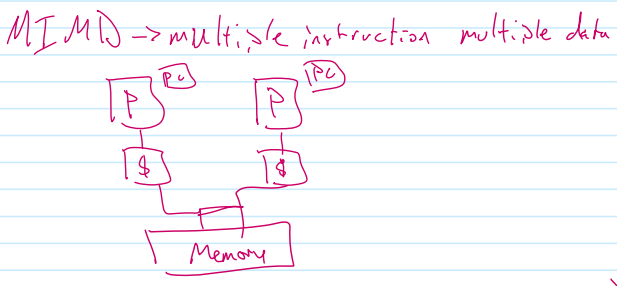
Thursday, March 15, 2018 10:05 AM

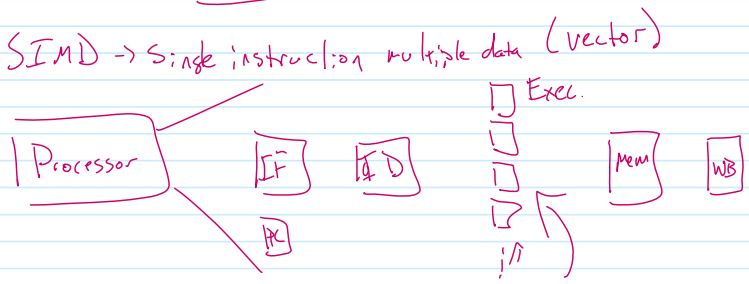
Final is in Wellman 126

Extra office hours Mon. 4:30-6

Outline

- Finish up parallel architectures
- Amdahl's Law
- Some reasons parallelism is hard
- Roofline model

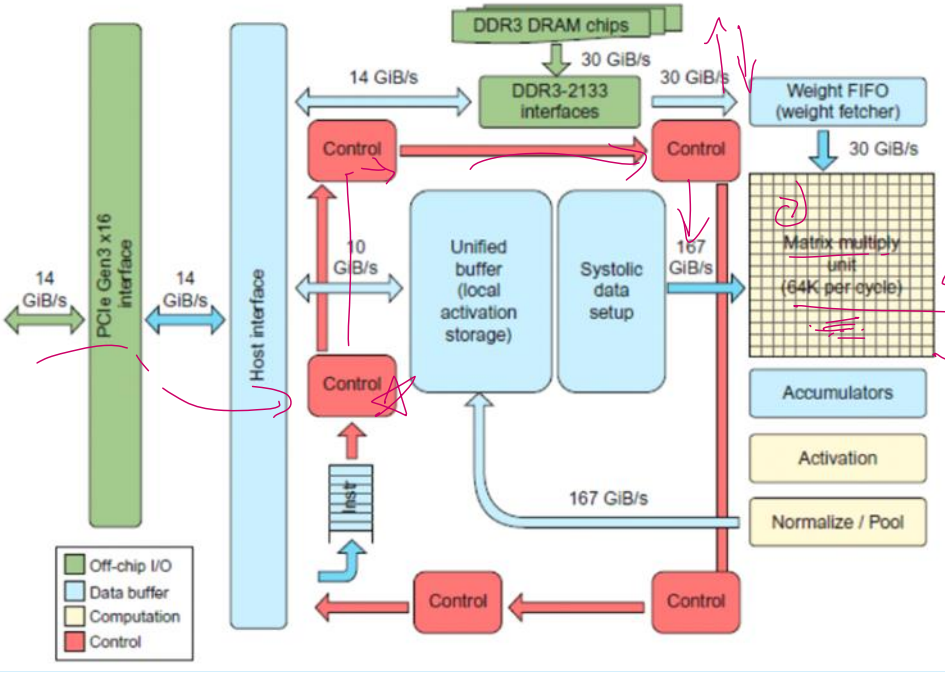


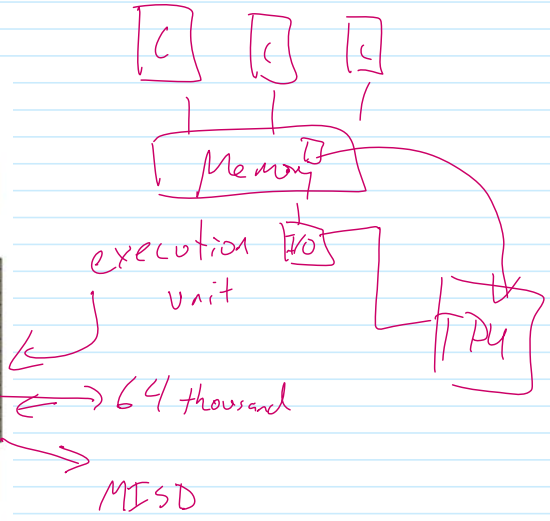


New parallel architectures

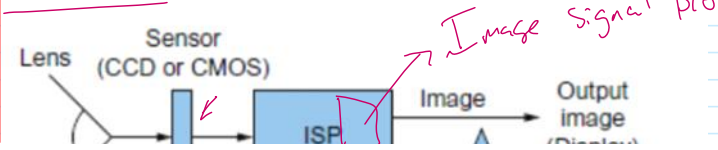
Google's TPU

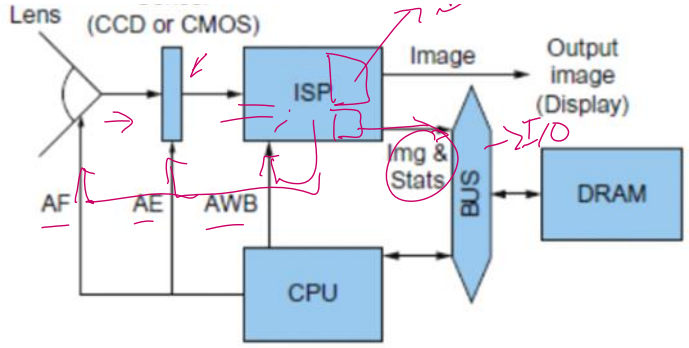
Tensor processing Unit

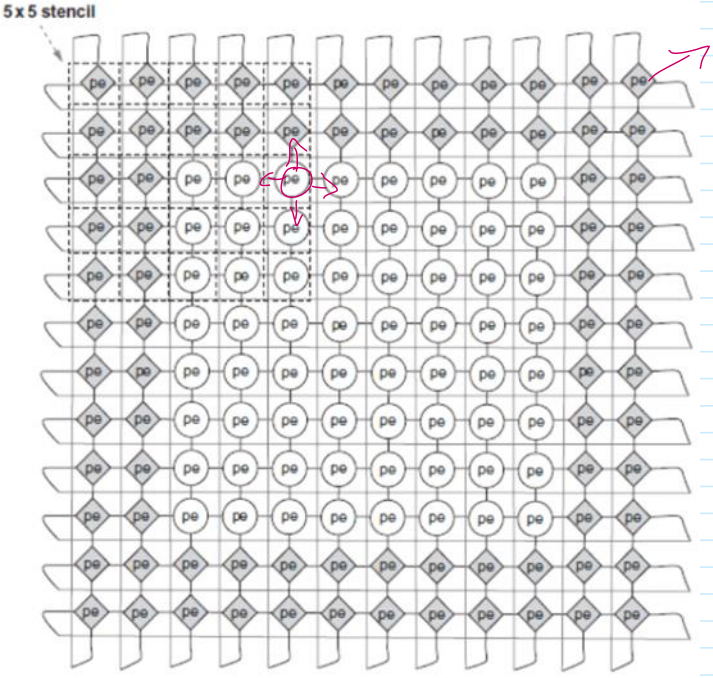




Pixel visual core



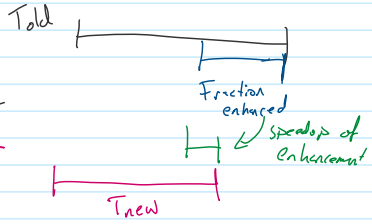




Amdahl's Law (again)

Performance

$$\text{Speedup} = \frac{T_{old}}{T_{new}} = \frac{T_{1\,proc.}}{T_{p\,proc}} = \frac{1}{1 - fe + \frac{fe}{se}}$$



Parallelism

$$\text{Speedup} = \frac{1}{1 - f + \frac{f}{p}}$$

p → # of processors in our system

f → fraction we can parallelize

50% parallelizable

Speedup w/ 2 cores?

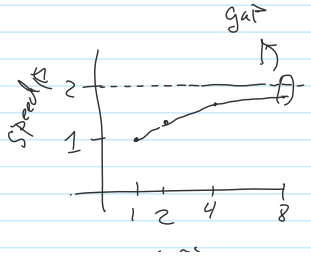
$$S = \frac{1}{1 - .5 + .5/2} = \frac{1}{.75} = \frac{1}{3/4} = \frac{4}{3} = 1.33\times$$

4 cores

$$S = \frac{1}{1 - .5 + .5/4} = \frac{1}{.625} = \frac{1}{5/8} = \frac{8}{5} = 1.6\times$$

8 cores

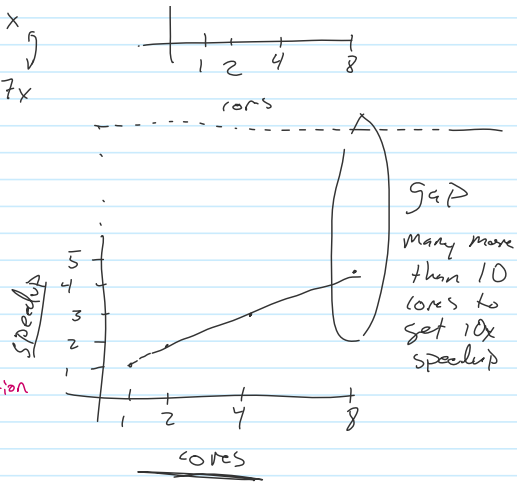
$$S = \frac{1}{1 - .5 + .5/8} = \frac{1}{.5625} = \frac{1}{9/16} = \frac{16}{9} = 1.77\times$$



$S = \frac{1}{1-.5+.5/4} = \frac{1}{.625} = \frac{5}{8} = \frac{8}{5} = 1.6\times$

8 cores $S = \frac{1}{1-.5+.5/8} = \frac{1}{.5625} = \frac{1}{\frac{9}{16}} = \frac{16}{9} = 1.77\times$

∞ cores $S = \frac{1}{1-.5+.5/\infty} = \frac{1}{.5} = 2\times$



90% parallelizable

2 cores $S = \frac{1}{1-.9+.9/2} = \frac{1}{.55} = 1.82\times$

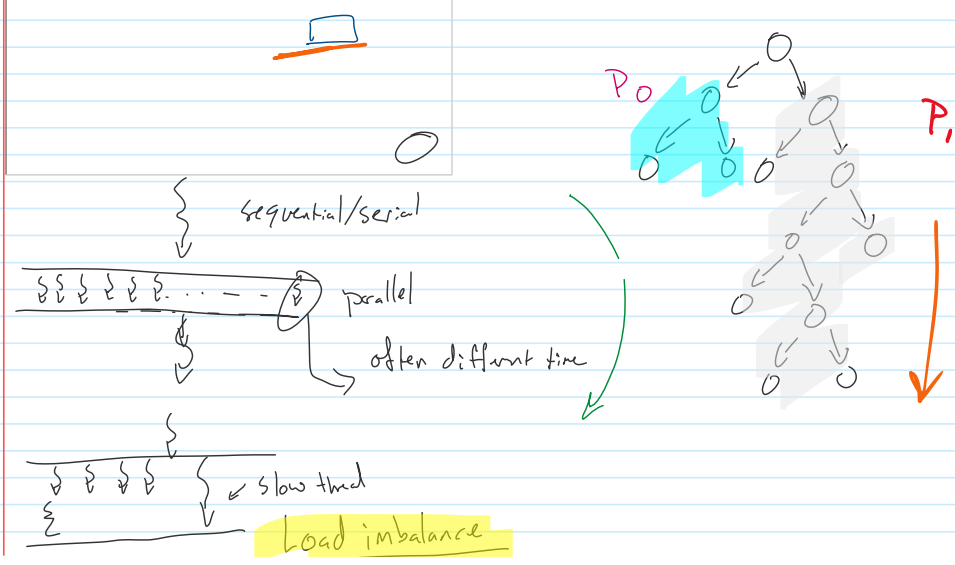
4 cores $S = \frac{1}{1-.9+.9/4} = 3.07$

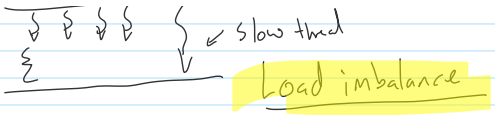
8 cores $S = \frac{1}{1-.9+.9/8} = 4.7$ ← serial portion

∞ cores peak possible $\Rightarrow S = \frac{1}{1-.9} = 10\times$

Look @ IPython notebook on github

1/64





Load imbalance

Solving imbalance
→ make sure all threads have equal partitions

Much worse w/ dynamic parallelism where you "create" work as you go
→ Work queue where all processors put new work & fetch work from here when idle
→ Work stealing (cilk)

Scaling

→ perform "same" for equal work per processor as # processors and amount of work — 1 user to 1,000 users → each user has same part
→ Scaling throughput
→ Weak scaling

Strong scaling
↳ for a constant problem size incr processors decr latency

Speed → operations per second
↳ floating point → FLOPS — GFlops billions; TFlops trillions

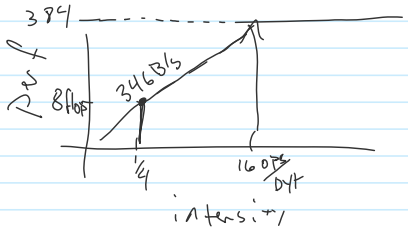
4 cores 3 SIMD SIMD 256-bit (8 floats) 4 GHz
$4 \cdot 3 \cdot 8 \cdot 4 \cdot 10^9 = 384$ GFLOPS ←

Memory bandwidth for getting data in/out → 34.1 GB/s

Arithmetic intensity — # flop / byte of data

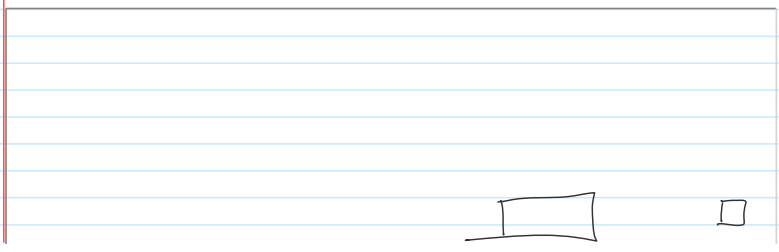
3 ops 3 ld/st 4 bytes/float

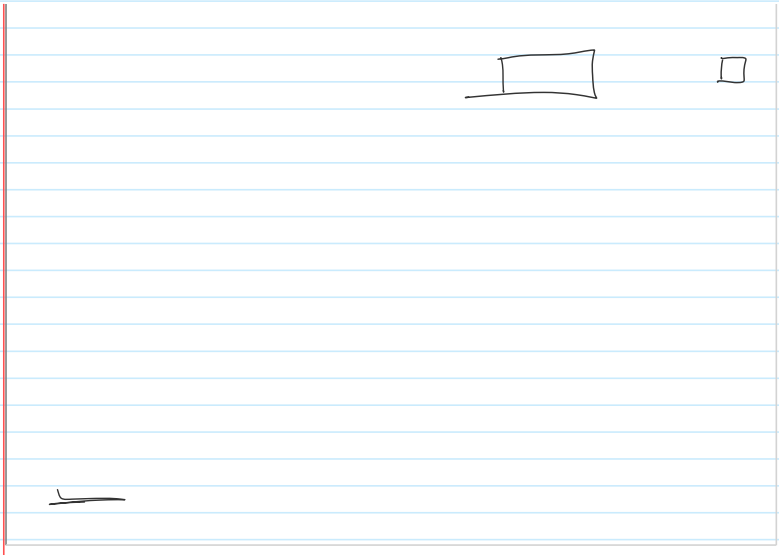
3 ops / 12 bytes — 1 op / 4 bytes →



roofline model

Top 500 supercomputers (https://www.top500.org/lists/2017/11/)





Roofline model of performance

Nvidia SPU
Titan V
↳ 10Tflops