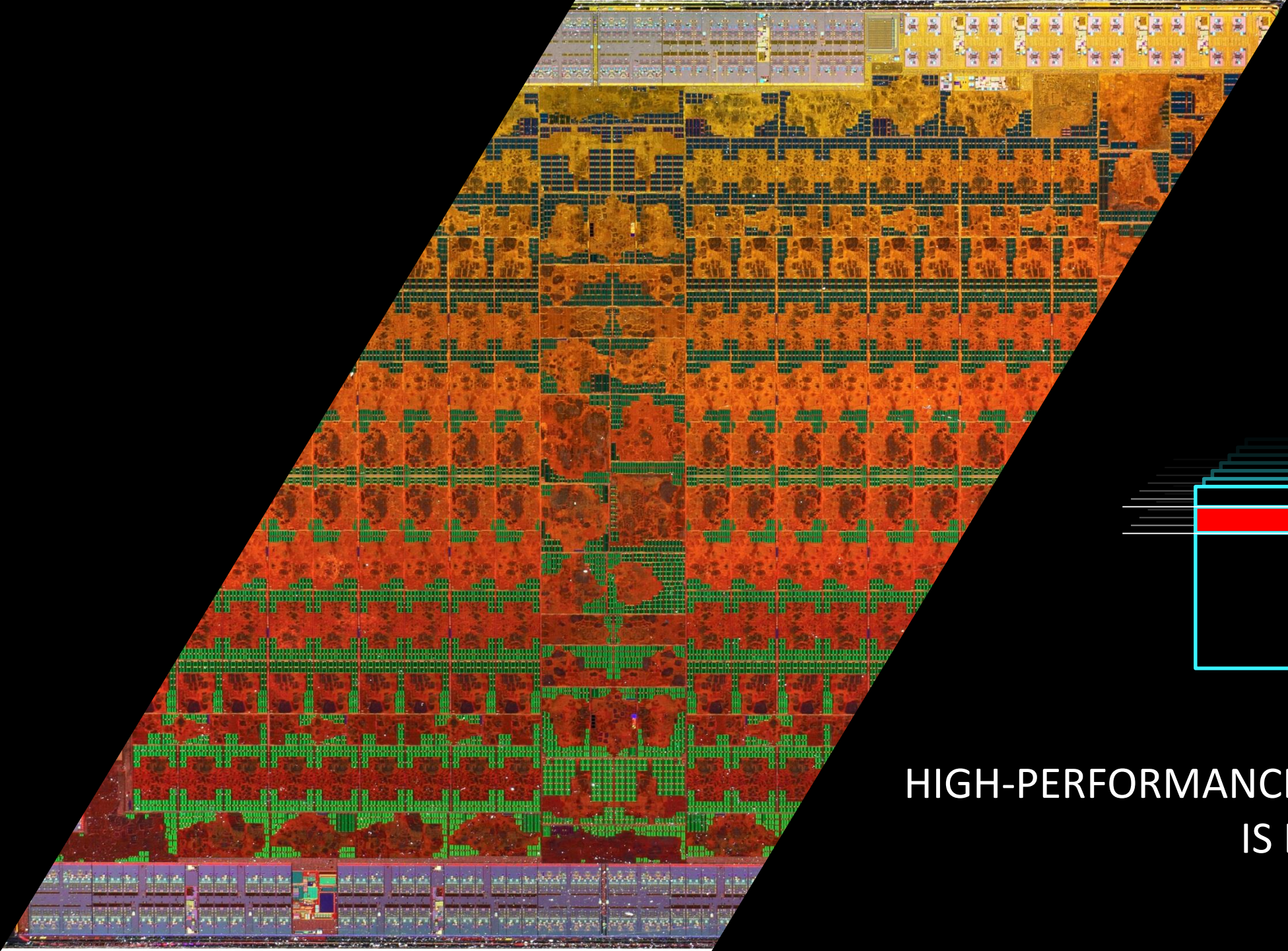
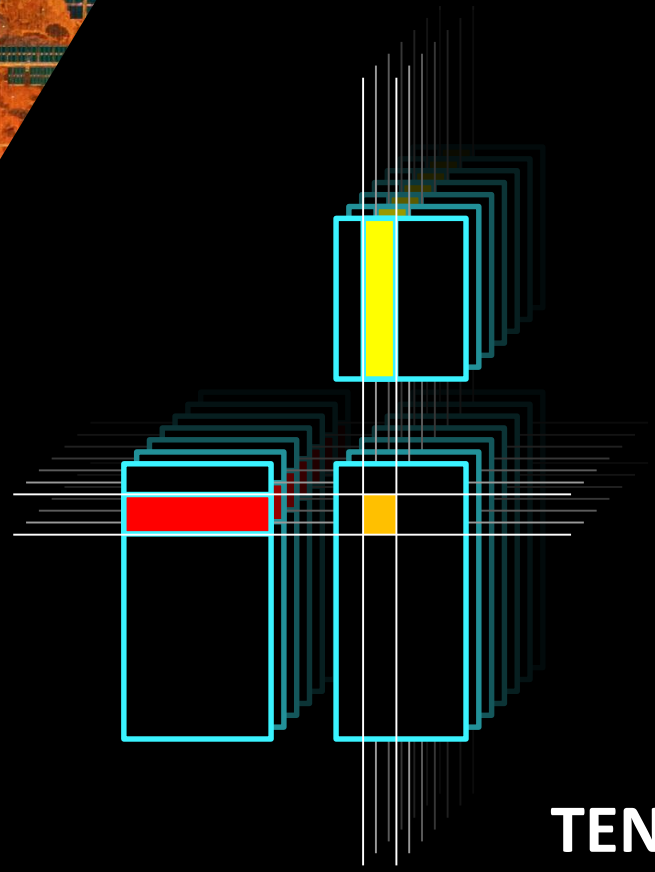


AMD



TENSILE

HIGH-PERFORMANCE GEMM ON GPUS IS LIKE A 3-YEAR OLD

David E Tanner

MOTIVATION

- GEMM API encapsulates trillions of problems, all of which behave differently on GPUs.
  - Precisions, Transpose A, Transpose B, M, N, K, Strides
- Many problem types behave similar to GEMM.
  - $C_{ij} = \sum_k A_{ik} * B_{kj}$ (gemm NN)
  - $C_{ijk} = \sum_l A_{ilk} * B_{jlk}$ (batched gemm NT)
  - $C_{ijk} = \sum_{lmn} A_{ilmnk} * B_{jlmnk}$ (batched gemm w/ 3D summation)
  - $C_{ijk} = \sum_{lmn} A_{inlkm} * B_{mjlkn}$ (batched gemm w/ 3D summation different data layout)
- Goal: auto-generate kernels with peak performance
  - For all problem types.
  - For all problem sizes.
  - On all GPUs.
  - Multiple languages: OpenCL, HIP, Assembly.

GPU PERFORMANCE PARAMETERS

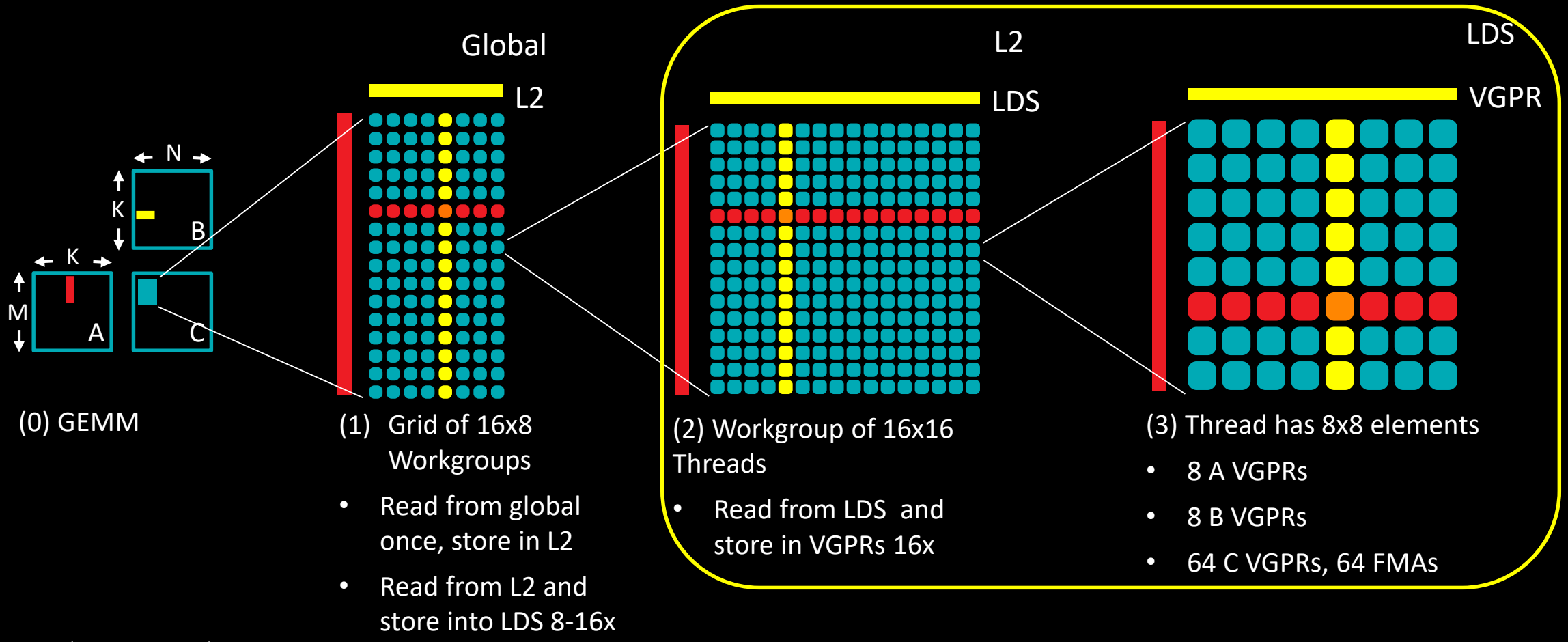
VEGA10 FRONTIER EDITION

- Compute Throughput
  - 13.1 TFlops = 2 (flops/cycle)*64(CUs)*64(lanes/CU)*1600MHz
- LDS Bandwidth
  - TB/s
- Caches
  - L2 shared by all CUs
  - L1 dedicated to CU
- Global Memory Bandwidth
  - 480 GB/s
  - coalescing
- Global Memory Latency
  - hundreds of cycles
  - hide using CU-occupancy or ILP
- CU Occupancy
  - limited resources (VGPRs, LDS) per CU
- Whole-GPU Occupancy
  - hundreds of work-groups
- LDS Latency
  - tens of cycles
  - hide using CU-occupancy or ILP
- Instruction Divergence
  - all threads within workgroup do same instruction else need to compute and apply execution masks
- Instruction Throughput
  - gemm requires 2*M*N*K instructions, all extras hurt efficiency
  - minimize instructions which don't count
  - maximize dual-issuing of instructions which don't count with instructions that do; must be from different wavefronts
    - VALU, SALU, LDS, global memory, branch
- Power
  - VALU, LDS, memory, caches

GENERAL KERNEL-LEVEL STRATEGY

TILING

- Tiling at all memory levels to read from lower-bandwidth memory less and from higher-bandwidth memory more to prevent bandwidth from being bottleneck

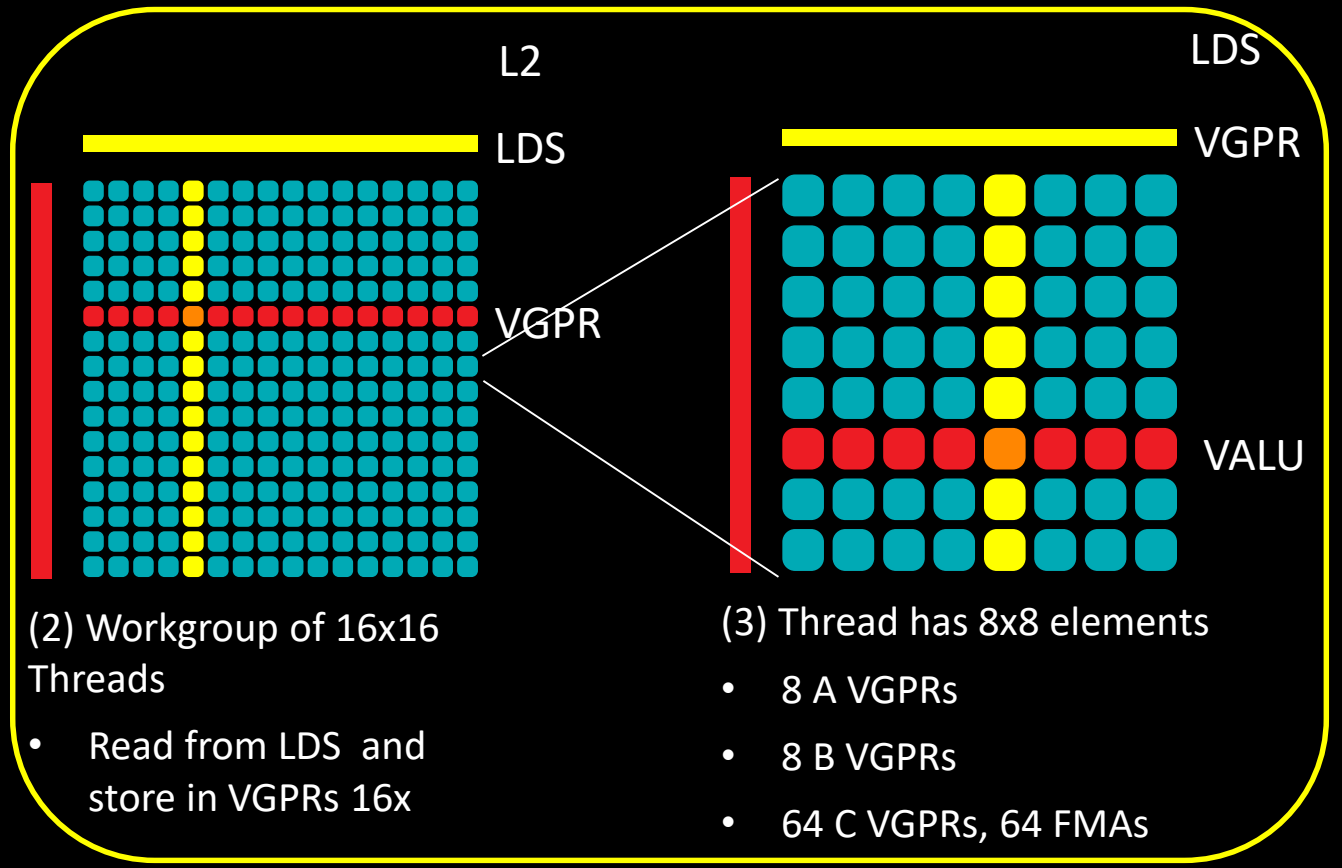


THREAD-TILE SUB-ITERATION

WHERE WE WANT TO GET TO

```
// read 8 A elements
ds_read_b128
ds_read_b128

// read 8 B elements
ds_read_b128
ds_read_b128

s_waitcnt // wait for loads

// 64 MACs
v_mac_f32
v_mac_f32
v_mac_f32
v_mac_f32
+ 60 more
```

16 threads reading exact same address of LDS; coalesced.



SUMMATION LOOP

no prefetching

```
calculate addresses

BEGIN LOOP
read global
wait global
barrier
write lds
barrier
read lds 0
wait lds 0
MACs 0
read lds 1
wait lds 1
MACs 1
…
read lds N-2
wait lds N-2
MACs N-2
read lds N-1
wait lds N-1
MACs N-1
END LOOP

write vgprs to C
```

subiter 0

subiter 1

…

subiter N-2

subiter N-1

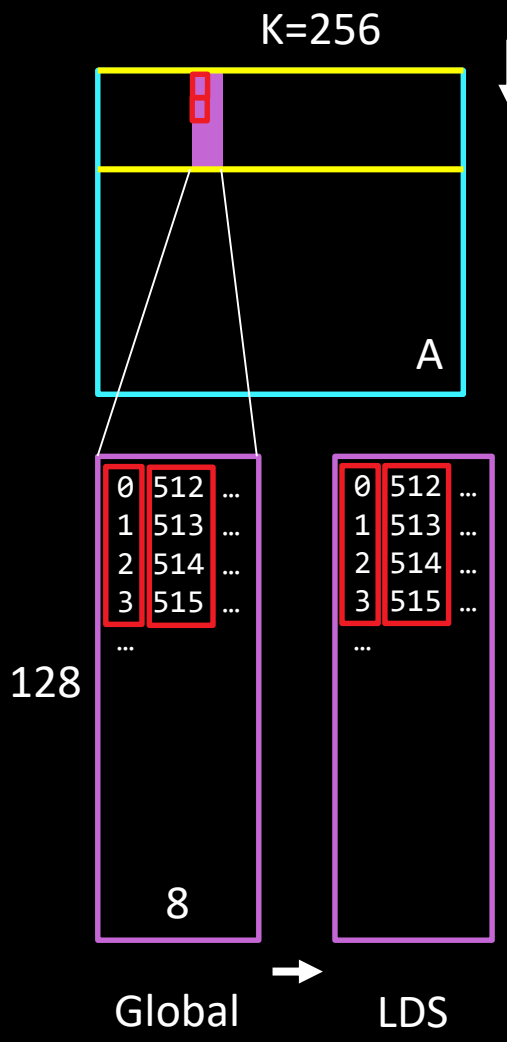
```
calculate addresses
prefetch iter 0

BEGIN LOOP
read global iter 1
read lds 1
wait lds read 0
MACs 0
read lds 2
wait lds read 1
MACs 1
…
read lds N-1
wait global read
write lds
swap lds red / black ptrs
wait read lds N-2
MACs N-2
wait write lds & N-1
barrier
read lds iter 1 subiter 0
MACs N-1
END LOOP

write vgprs to C
```

prefetch global

prefetch local

GLOBAL TO LDS

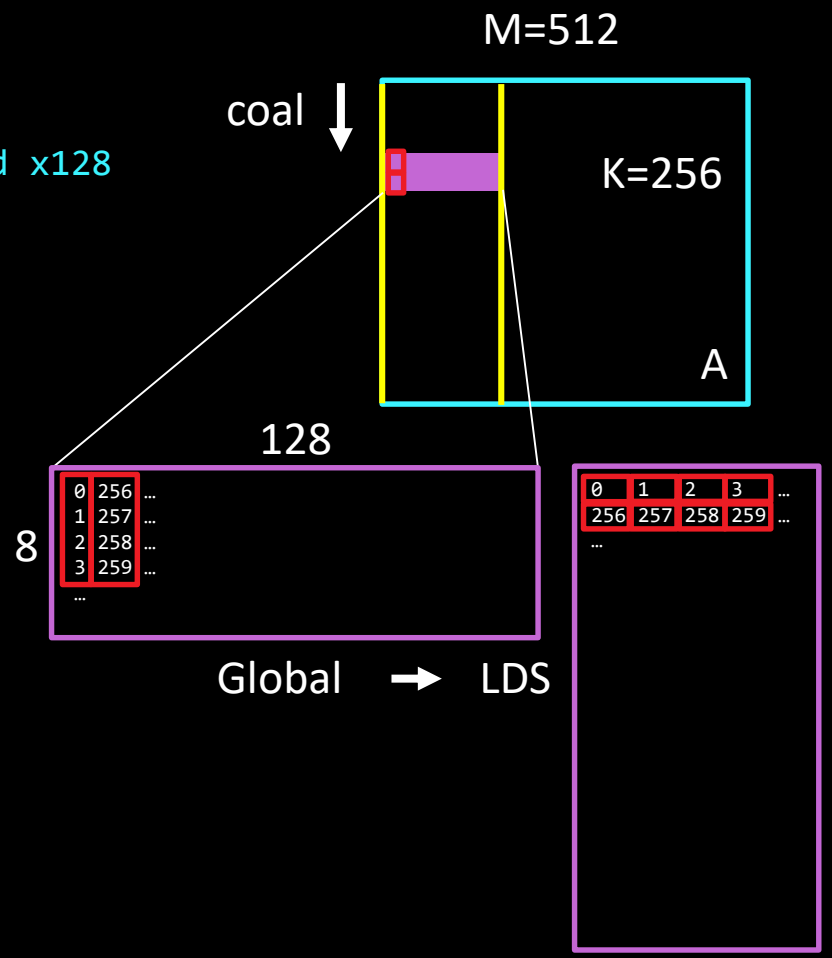
Don't Transpose Data



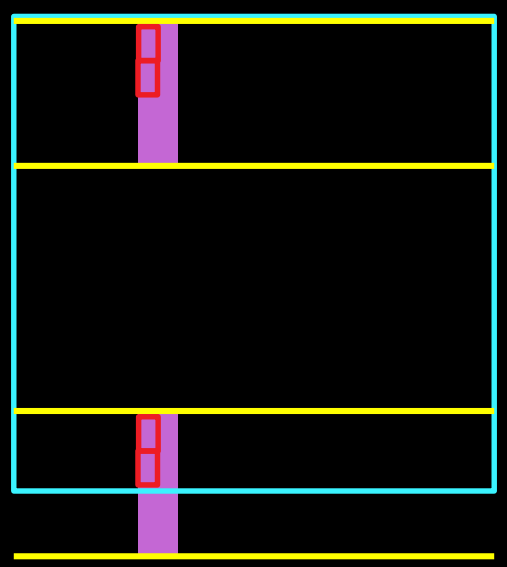

```
// read 4 A elements
// 512 bytes coalesced x8
flat_load_dwordx4

// read 4 B elements
flat_load_dwordx4

// write A to LDS
ds_write_b128

// write B to LDS
ds_write_b128
```

Do Transpose Data

```
// read 4 A elements
// 32 bytes coalesced x128
flat_load_dwordx4

// read 4 B elements
flat_load_dwordx4

// write A to LDS
ds_write_b32
ds_write_b32
ds_write_b32
ds_write_b32

// write B to LDS
ds_write_b32
ds_write_b32
ds_write_b32
ds_write_b32
```

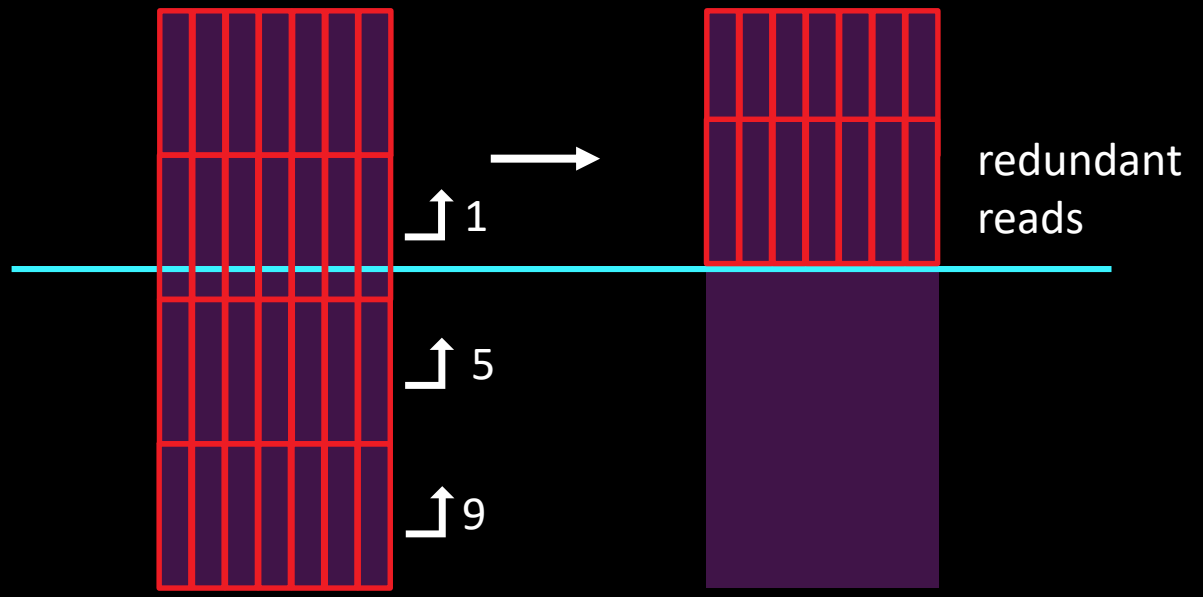


EDGES WITHOUT DIVERGENCE



coalesced

Protect against reading out of bounds M, N

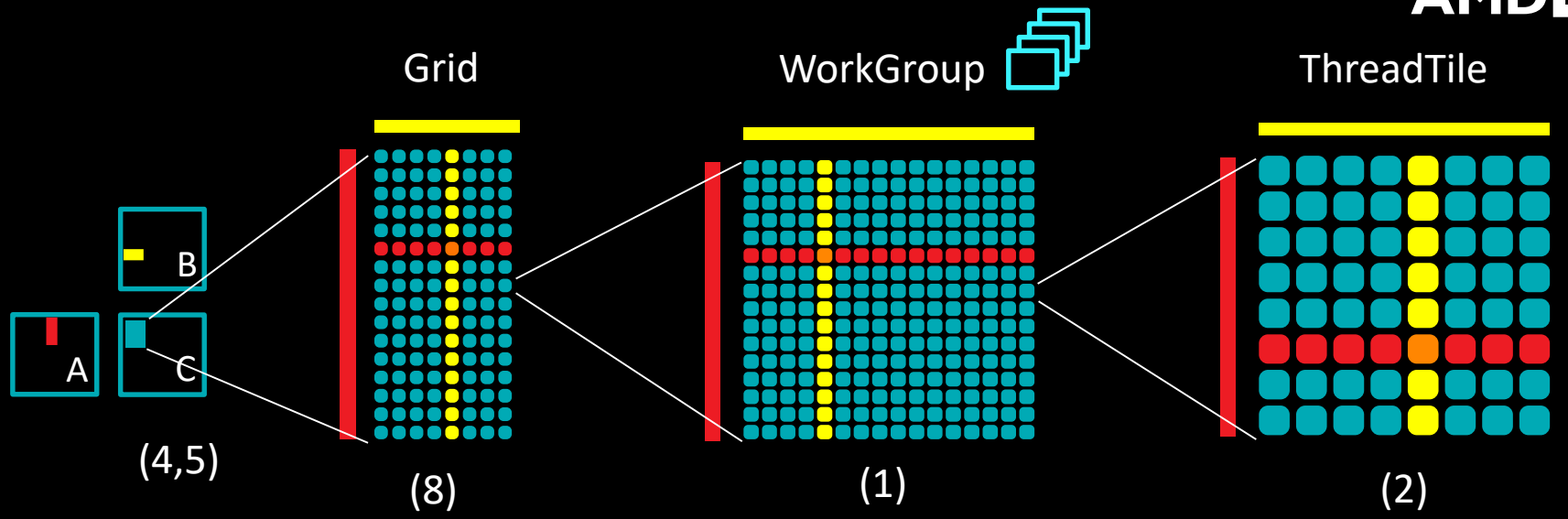


- Protect against reading out of bounds K

  Tail loop to handle K % UNROLL

KERNEL PARAMETERS

1) WorkGroup, LocalSplitU
2) ThreadTile
3) VectorWidth
4) GlobalSplitU
5) GlobalSplitUWGM
6) PrefetchGlobalRead
7) PrefetchLocalRead
8) WorkGroupMapping
9) LoopUnroll
10) NumLoadsCoalesced
11) GlobalReadCoalesceGroup
12) GlobalReadCoalesceVector
13) KernelLanguage
14) NonTemporal



(3,4,10,11,12)

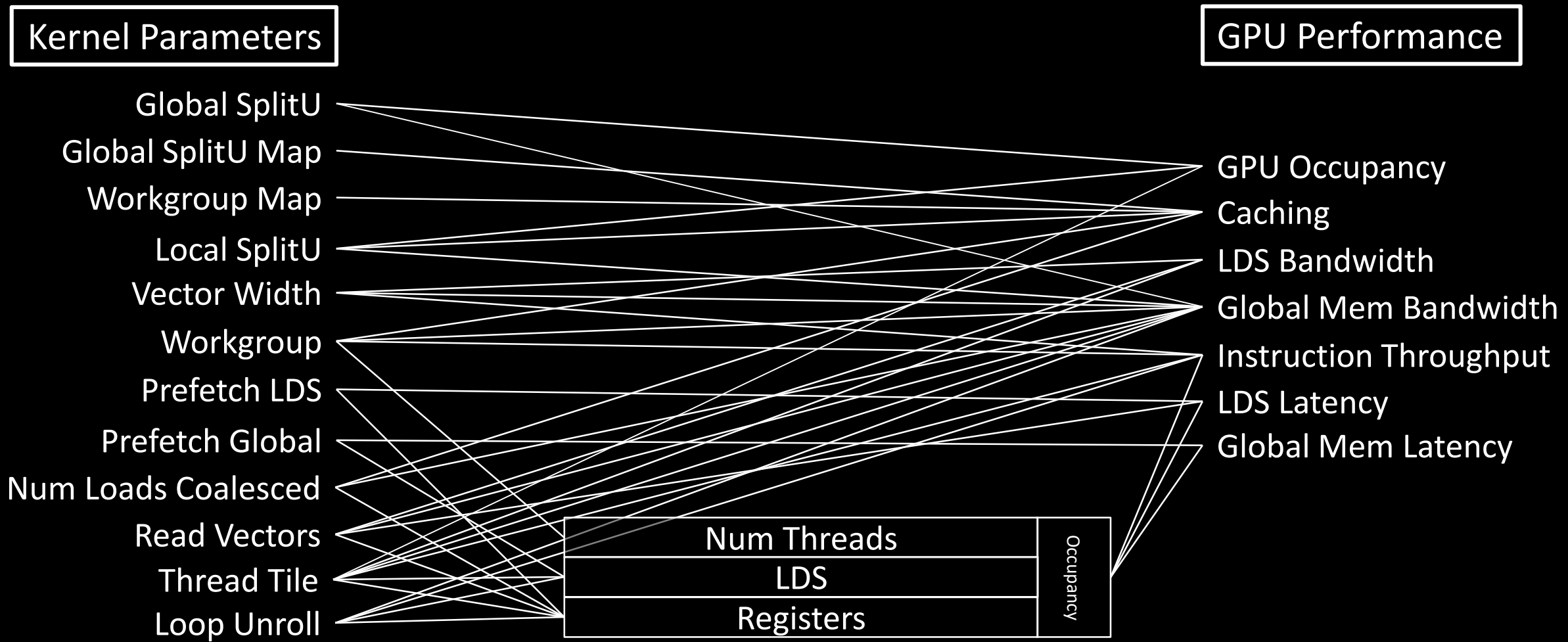
(3,6)

(7)

(9)

```
calculate addresses          read lds N-1
prefetch iter 0              wait global read
                             write lds
BEGIN LOOP                   swap lds red / black ptrs
load global iter 1           wait read lds N-2
read lds 1                   MACs N-2
wait lds read 0              wait write lds & N-1
MACs 0                       barrier
read lds 2                   read lds iter 1 subiter 0
wait lds read 1              MACs N-1
MACs 1                       END LOOP
…              (13)
                             write vgprs to C      (14)
```

KERNEL PARAMETERS AFFECT GPU PERFORMANCE



PERFORMANCE EXPERIMENTS

AMD

- Set of 100 kernels
  - ThreadTile = 8x8, 8x4, 8x2, 4x8, 4x4, 4x2, 2x8, 2x4, 2x2
  - WorkGroup = 16x16x1, 16x8x2, 8x16x2, 8x8x4
  - GlobalSplitU = 1, 2, 4, 6, 8
  - DepthU = 8
  - VectorWidth = max
  - PrefetchGlobalRead = True
  - PrefetchLocalRead = True
  - WorkGroupMapping = 8
- Benchmark kernels against 2D range of sizes for sgemm NT
  - M = 16 – 5632; N = 16 – 5632; K = 3104 (moderate)
  - M = 128 – 100,000; N = 16 – 256; K = 3104 (skinny N)
  - M = 64 – 512; N = 64 – 512; K = 1,048,576 (small MxN) GSU=16, 32, 64, 128
- Analyze performance and kernel properties for each data point.

PERFORMANCE

AMD

N

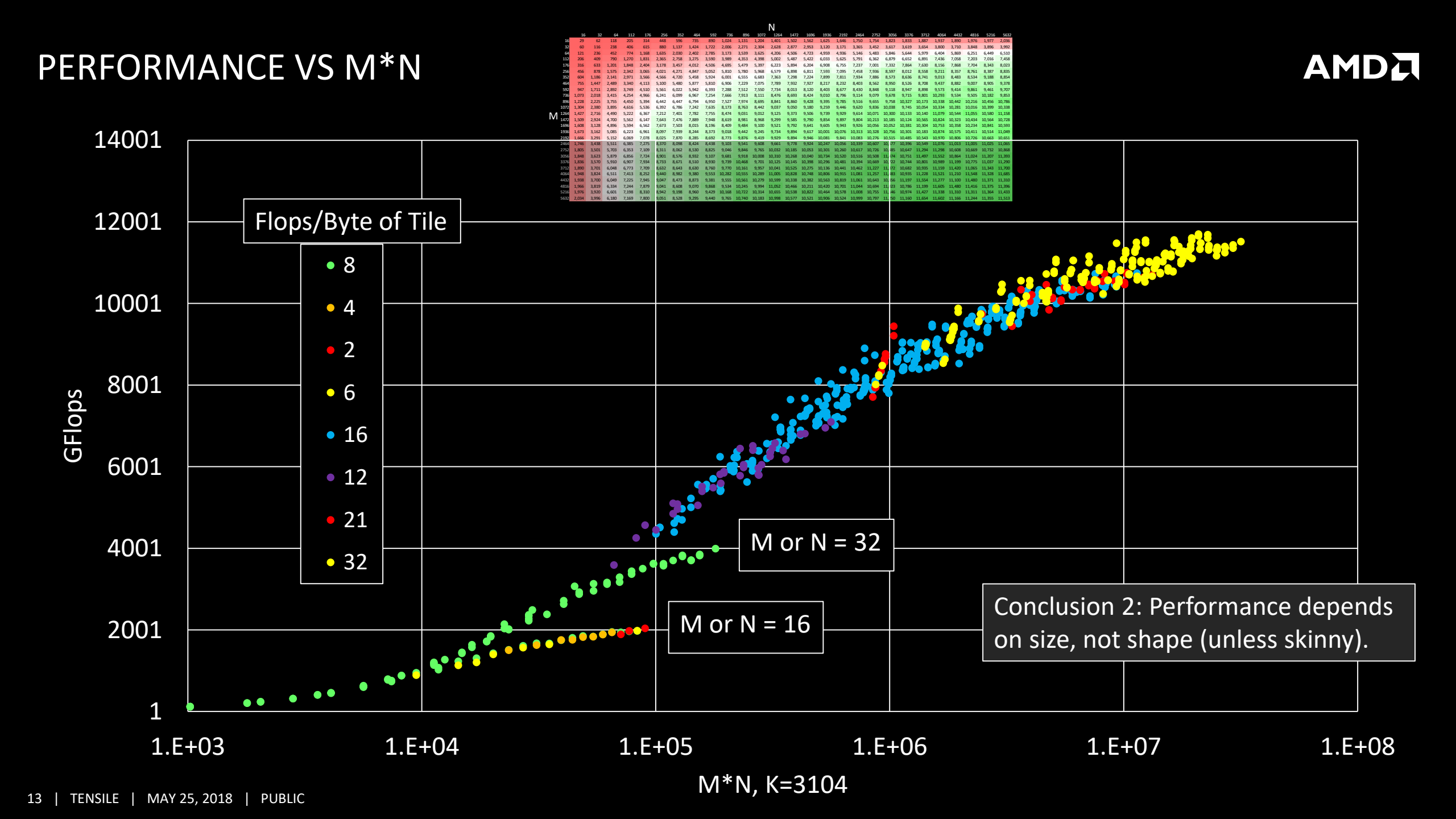
M	16	32	64	112	176	256	896	1072	1264	1472	1696	1936	2192	2464	2752	3056	3376	3712	4064	4432	4816	5216	5632
16	29	62	118	205	314	448	1,131	1,204	1,401	1,502	1,562	1,625	1,646	1,750	1,754	1,823	1,833	1,887	1,937	1,890	1,976	1,977	2,036
32	60	116	238	406	615	880	2,271	2,304	2,628	2,877	2,953	3,120	3,171	3,365	3,452	3,617	3,619	3,654	3,800	3,710	3,848	3,896	3,992
64	121	236	452	774	1,168	1,635	2,030	2,402	2,785	3,173	3,539	3,625	4,206	4,506	4,723	4,959	4,936	5,146	5,483	5,846	5,644	5,979	6,404	5,869	6,251	6,449	6,510
112	206	409	790	1,270	1,831	2,365	2,758	3,275	3,590	3,989	4,353	4,398	5,002	5,487	5,422	6,033	5,625	5,791	6,362	6,879	6,652	6,891	7,436	7,058	7,203	7,016	7,458
176	316	633	1,201	1,848	2,404	3,178	3,457	4,012	4,506	4,695	5,479	5,397	6,223	5,894	6,204	6,908	6,755	7,237	7,001	7,332	7,864	7,630	8,156	7,868	7,704	8,343	8,023
256	456	878	1,575	2,342	3,065	4,021	4,271	4,847	5,052	5,810	5,780	5,968	6,579	6,898	6,811	7,593	7,095	7,458	7,936	8,597	8,012	8,558	9,211	8,357	8,761	8,387	8,835
352	604	1,186	2,141	2,971	3,566	4,566	4,720	5,458	5,924	6,001	6,555	6,683	7,363	7,298	7,224	7,899	7,811	7,934	7,886	8,573	8,636	8,741	9,013	8,483	8,534	9,188	8,854
464	755	1,447	2,489	3,340	4,113	5,100	5,480	5,877	5,810	6,906	7,229	7,075	7,789	7,932	7,927	8,217	8,232	8,403	8,562	8,950	8,526	8,708	9,437	8,882	9,007	8,905	9,378
592	947	1,711	2,892	3,749	4,510	5,561	6,022	5,942	6,393	7,288	7,512	7,550	7,734	8,013	8,120	8,403	8,677	8,430	8,848	9,118	8,947	8,898	9,573	9,414	9,861	9,461	9,707
736	1,073	2,018	3,415	4,254	4,966	6,241	6,099	6,967	7,254	7,666	7,913	8,111	8,476	8,693	8,424	9,010	8,796	9,114	9,079	9,678	9,715	9,801	10,293	9,534	9,505	10,182	9,853
896	1,228	2,225	3,755	4,450	5,394	6,442	6,447	6,794	6,950	7,527	7,974	8,695	8,841	8,860	9,428	9,395	9,785	9,516	9,655	9,758	10,327	10,173	10,338	10,442	10,216	10,456	10,786
1072	1,304	2,380	3,895	4,616	5,536	6,392	6,786	7,242	7,635	8,173	8,763	8,442	9,037	9,050	9,180	9,259	9,446	9,620	9,836	10,038	9,745	10,054	10,334	10,281	10,016	10,399	10,338
1264	1,427	2,716	4,490	5,222	6,367	7,212	7,401	7,782	7,755	8,474	9,031	9,012	9,125	9,373	9,506	9,739	9,929	9,614	10,071	10,300	10,133	10,140	11,079	10,544	11,055	10,580	11,158
1472	1,509	2,924	4,700	5,562	6,147	7,643	7,476	7,889	7,948	8,619	8,981	8,968	9,299	9,585	9,790	9,854	9,897	9,804	10,213	10,185	10,124	10,565	10,824	10,323	10,434	10,564	10,728
1696	1,608	3,128	4,896	5,594	6,562	7,673	7,503	8,015	8,196	8,409	9,484	9,100	9,521	9,792	9,641	9,605	9,943	9,926	10,056	10,052	10,381	10,304	10,753	10,358	10,234	10,841	10,593
1936	1,673	3,162	5,085	6,223	6,961	8,097	7,939	8,244	8,373	9,018	9,442	9,245	9,734	9,894	9,617	10,001	10,076	10,313	10,328	10,756	10,301	10,183	10,874	10,575	10,411	10,514	11,049
2192	1,666	3,291	5,152	6,069	7,078	8,025	7,870	8,285	8,692	8,773	9,876	9,419	9,929	9,894	9,946	10,081	9,841	10,083	10,276	10,515	10,485	10,543	10,970	10,806	10,726	10,663	10,651
2464	1,746	3,438	5,511	6,385	7,275	8,370	8,098	8,424	8,438	9,103	9,541	9,608	9,661	9,778	9,924	10,247	10,056	10,339	10,607	10,577	10,396	10,549	11,076	11,013	11,005	11,025	11,065
2752	1,805	3,501	5,703	6,353	7,109	8,311	8,062	8,530	8,825	9,046	9,846	9,765	10,032	10,185	10,053	10,301	10,260	10,617	10,726	10,485	10,647	11,294	11,298	10,608	10,669	10,732	10,868
3056	1,848	3,623	5,879	6,856	7,724	8,901	8,576	8,932	9,107	9,681	9,918	10,008	10,310	10,268	10,040	10,734	10,520	10,516	10,508	11,474	10,751	11,497	11,552	10,864	11,024	11,207	11,393
3376	1,836	3,570	5,910	6,907	7,934	8,733	8,671	8,510	8,930	9,739	10,468	9,701	10,125	10,145	10,398	10,296	10,481	10,394	10,669	10,722	10,744	10,801	10,989	11,199	10,775	11,037	11,290
3712	1,890	3,701	6,048	6,773	7,709	8,632	8,643	8,630	8,760	9,770	10,161	9,957	10,041	10,525	10,275	10,136							11,159	11,420	11,065	11,343	11,700
4064	1,948	3,824	6,511	7,413	8,252	9,440	8,982	9,380	9,553	10,282	10,555	10,289	11,005	10,828	10,748	10,806							11,521	11,210	11,548	11,328	11,685
4432	1,938	3,700	6,049	7,225	7,945	9,047	8,473	8,873	9,381	9,555	10,561	10,279	10,599	10,338	10,382	10,563							11,277	11,100	11,480	11,371	11,310
4816	1,966	3,819	6,334	7,244	7,879	9,041	8,608	9,070	9,868	9,534	10,245	9,994	11,052	10,466	10,211	10,420							11,605	11,480	11,416	11,375	11,396
5216	1,976	3,920	6,601	7,198	8,310	8,942	9,198	8,960	9,429	10,168	10,722	10,314	10,655	10,538	10,822	10,464							11,338	11,310	11,311	11,364	11,433
5632	2,034	3,996	6,180	7,169	7,800	9,051	8,528	9,295	9,440	9,765	10,740	10,183	10,998	10,577	10,521	10,906							11,602	11,166	11,244	11,355	11,513

K = 3104

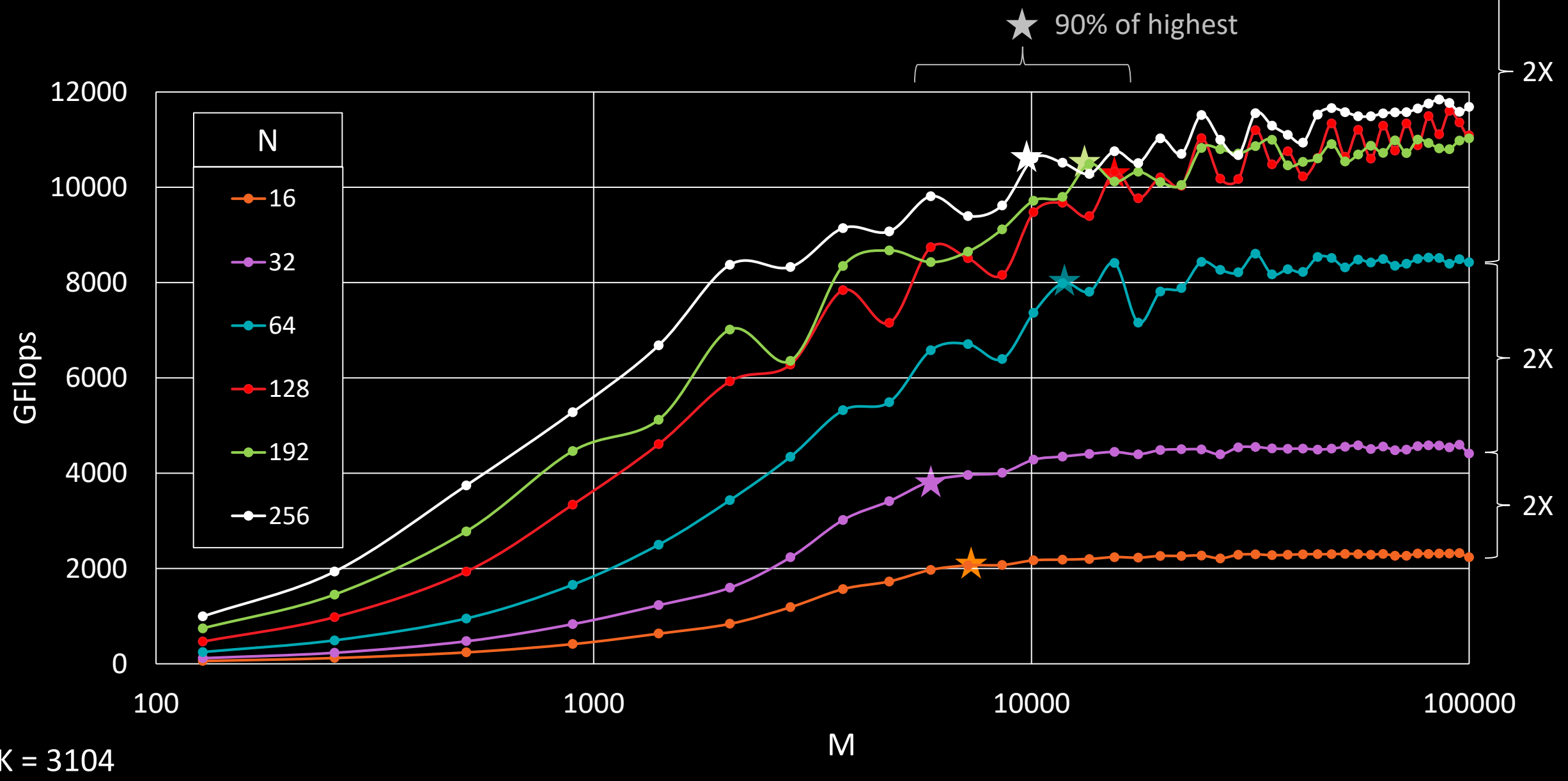
Peak: 13 TFlops

GFlops

PERFORMANCE VS M*N



LARGE & SKINNY

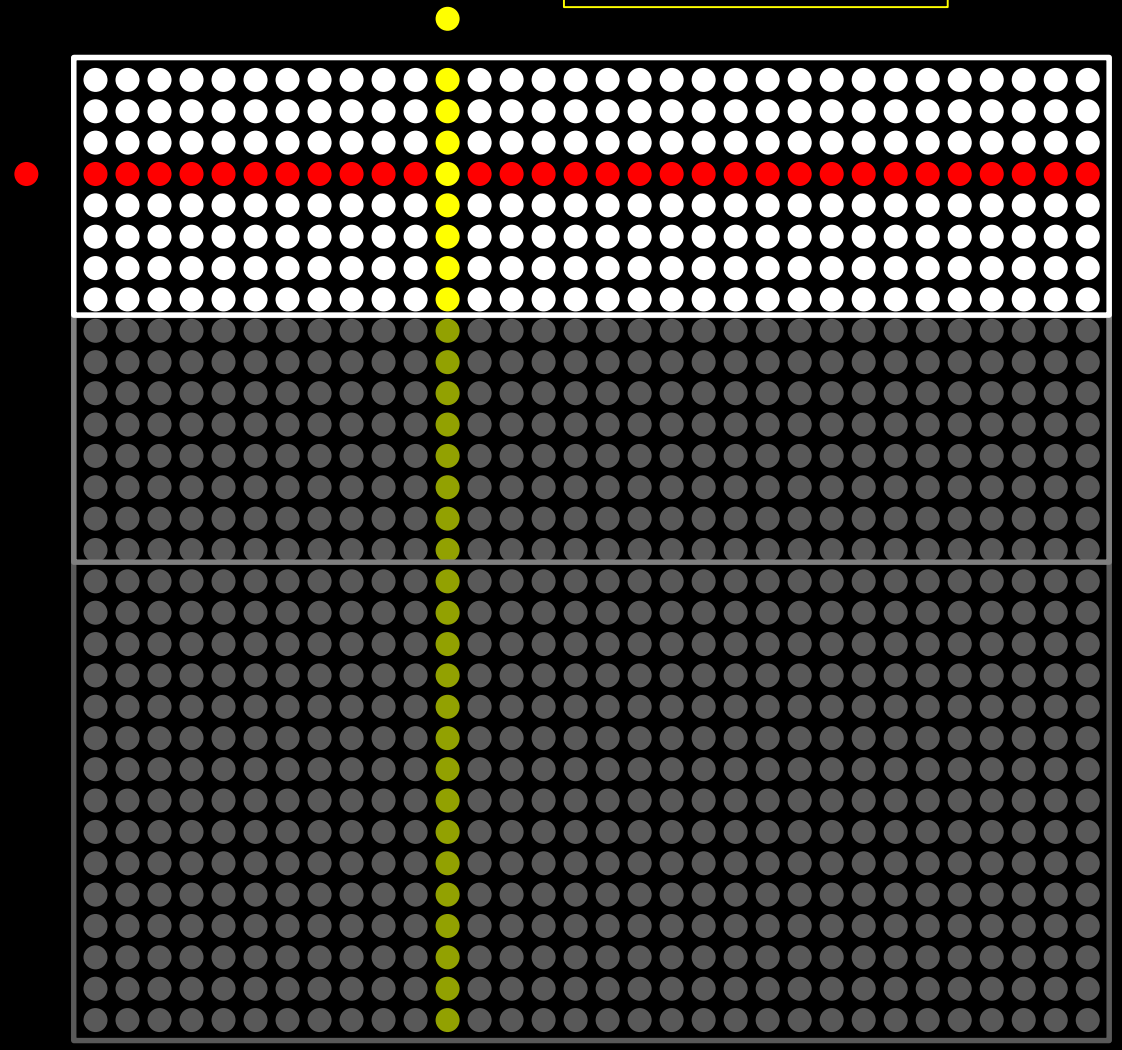


K = 3104

SKINNY DIMENSIONS HURT PERFORMANCE

Matrix B
- Loaded once
- Used in 8 MACs

Matrix A
- Loaded once
- Used in 32 MACs



8x32

32x32

SMALL MXN LARGE K

AMD

M \ N	64	128	256	384	448	512
64	4483	6171	7411	7831	7693	7805
128	6126	9009	10818	9139	9941	11316
256	7398	10930	11310	11177	10142	11595
384	7799	11516	11409	11237	10132	11568
448	7782	11611	10867	11788	10341	10897
512	8235	11500	11599	11675	10190	11630

Conclusion: For peak, MxN has to be at least 128x256, and M*N*K has to be sufficiently large (~3000^3).

K = 1,048,576

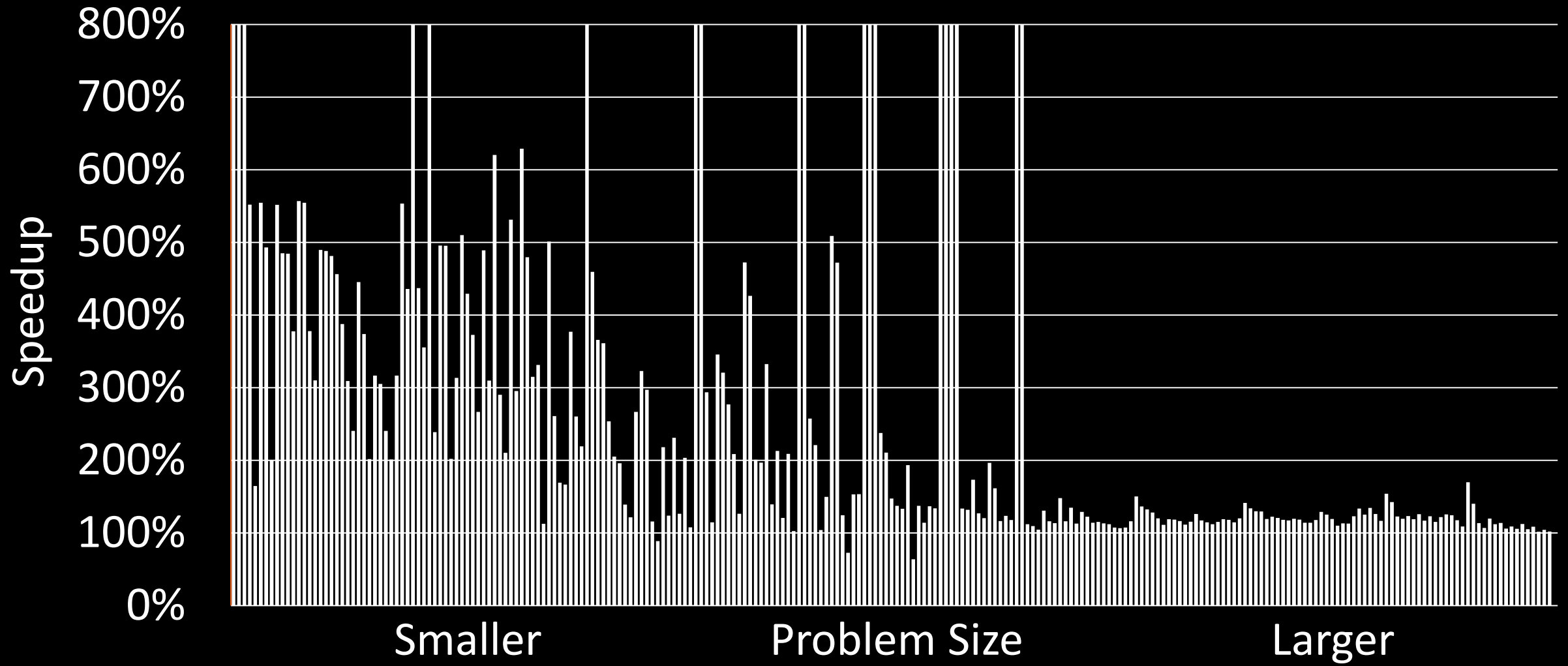
EXPERIMENT 2 - DEEPBENCH

AMD

- Set of thousands of kernels.
- Benchmark kernels against ~250 problem sizes of DeepBench.
- Analyze speedup of kernel tuned for exact problem size vs “fastest” 128x128 tile kernel.

M	N	K
35	8457	1760
512	1	500000
512	8	500000
1024	1	512
1024	32	512
1760	128	1760
8448	48000	2816

SPEEDUP OF DEEPBENCH SIZES



100% means equal performance

MANY PARAMETERS

- ThreadTile: 2x2, 4x2, 4x4, 6x3, 3x3, 8x4, 4x8, 6x8, 8x2, 8x6, 8x8
- WorkGroup: 8x8x2, 8x8x4, 16x4x4, 8x4x8, 8x4x8, 8x8x2, 16x8x2, 12x16x1, 8x12x2, 16x16x1, 32x2x4, 16x2x8, 32x4x2
- GlobalSplitU: 1, 2, 4, 8, 16, 32
- DepthU: 8, 16, 24, 32, 64

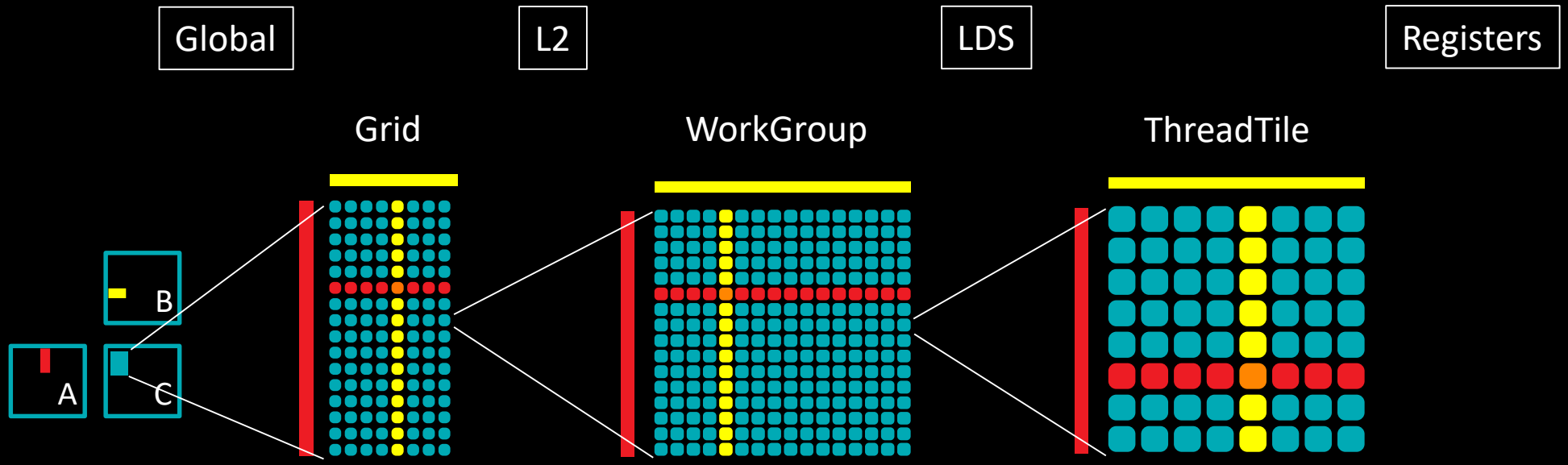
- 68 winning kernels
- 4,290 permutations amongst winning parameters (even more explored)

HGEMM - TFLOPS

											896	1072	1264	1472	1696	1936	2192	2464	2752	3056	3376	3712	4064	4432	4816	5216	5632	6064	6512	6976	7456	7952	8464	8992	9536	10096	10672	11264	11872	12496	13136	13792	14464	15152	15856	16576	17312	18064	18832	19616	20416	21232	22064
16	0.0	0.0	0.0	0.0	0.1	0.1	0.2	0.2	0.3	0.3	0.4	0.5	0.5	0.6	0.7	0.8	0.9	1.1	1.2	1.3	1.5	1.6	1.7	1.9	2.1	2.2	2.4	2.6	2.8	3.0	3.2	3.4	2.4	2.6	2.7	2.9	3.0	3.2	3.4	3.5	3.7	3.9	4.1	4.1	4.2	3.4	3.6	3.7	3.8	4.0	4.1	4.2	4.2
32	0.0	0.0	0.1	0.1	0.2	0.2	0.3	0.4	0.5	0.6	0.8	0.9	1.1	1.3	1.5	1.7	1.9	2.1	2.4	2.6	2.9	3.2	3.5	3.8	4.1	4.5	4.8	5.1	5.5	5.9	6.3	6.7	4.8	5.1	5.5	5.8	6.1	6.4	6.8	7.1	7.4	7.8	8.1	8.3	8.4	6.9	6.4	7.4	7.6	7.9	8.2	8.4	8.4
64	0.0	0.1	0.1	0.2	0.3	0.4	0.6	0.8	1.0	1.3	1.6	1.8	2.2	2.5	2.9	3.4	3.8	4.2	4.8	5.3	5.8	6.4	7.0	7.6	8.3	9.0	9.6	10.4	11.1	11.8	12.5	13.5	9.6	10.1	10.8	11.3	12.0	12.8	13.4	14.1	14.7	15.4	16.3	16.4	16.8	13.6	14.1	14.7	15.3	15.9	16.2	16.6	16.9
112	0.0	0.1	0.2	0.3	0.5	0.8	1.1	1.4	1.8	2.3	2.7	3.3	3.8	4.4	5.2	5.9	6.6	7.5	8.4	9.2	10.1	11.2	12.2	8.8	9.6	10.4	11.2	12.1	12.9	13.9	14.8	15.8	12.3	13.0	13.7	14.6	15.4	16.2	17.1	14.1	14.8	15.6	16.3	17.1	17.9	15.4	16.1	16.7	17.4	18.1	18.9	16.7	17.3
176	0.1	0.2	0.3	0.5	0.8	1.2	1.7	2.2	2.8	3.6	4.3	5.1	6.0	7.0	8.1	9.2	10.3	11.6	8.7	9.6	10.5	11.6	12.7	13.8	15.0	16.2	12.8	13.7	14.7	15.7	16.9	18.0	15.0	16.0	16.9	17.9	18.9	16.4	17.3	18.1	19.1	17.0	17.9	18.7	19.6	17.8	18.6	19.3	20.2	18.6	19.3	20.1	18.7
256	0.1	0.2	0.4	0.8	1.2	1.8	2.4	3.2	4.2	5.1	6.2	7.5	8.7	10.3	11.6	13.3	10.0	11.2	12.5	13.8	15.2	17.0	18.3	14.6	15.8	17.1	18.5	19.9	16.7	18.0	19.2	20.5	17.9	19.0	20.2	21.3	19.2	20.2	21.3	19.5	20.5	21.5	20.0	20.9	21.8	20.5	21.3	22.3	21.0	21.9	22.8	21.7	22.5
352	0.2	0.3	0.6	1.1	1.7	2.4	3.4	4.5	5.7	7.0	8.5	10.2	12.1	9.3	10.6	12.1	13.6	15.4	12.5	13.9	15.3	16.8	18.4	15.8	17.1	18.6	16.4	17.7	18.9	17.2	18.4	19.6	18.2	19.1	20.5	19.1	20.2	19.1	20.2	19.2	20.2	19.4	20.3	19.6	20.5	19.9	20.8	20.2	21.0	20.5	21.3	20.8	20.4
464	0.2	0.4	0.8	1.4	2.2	3.2	4.5	5.9	7.4	9.2	11.3	8.9	10.4	12.1	14.0	15.9	13.1	14.7	16.4	18.2	15.8	17.3	19.0	17.0	18.4	17.0	18.3	19.7	18.4	19.7	18.6	19.8	19.0	20.1	19.3	20.5	19.8	20.8	20.2	19.8	20.8	20.3	19.9	20.9	20.5	20.2	21.1	20.9	20.6	20.5	21.3	21.1	21.0
592	0.3	0.5	1.0	1.8	2.8	4.2	5.7	7.4	9.5	11.7	9.4	11.2	13.2	15.4	12.9	14.7	16.7	14.8	16.4	18.2	16.6	18.2	19.9	18.4	20.0	18.8	20.3	19.3	20.7	19.9	19.3	20.6	20.0	19.6	20.7	20.4	20.1	21.1	20.9	20.6	20.5	21.5	21.3	21.2	21.1	21.1	21.0	21.0	21.0	21.9	21.9	21.9	21.9
736	0.3	0.6	1.3	2.2	3.5	5.1	7.0	9.2	11.8	9.6	11.6	14.0	16.5	13.9	16.0	18.3	16.3	18.3	16.8	18.6	20.5	19.2	20.9	19.8	19.1	20.6	20.0	21.5	20.9	20.5	21.9	21.5	21.2	21.0	22.3	22.1	21.9	21.8	21.8	21.7	21.7	21.8	21.8	21.9	22.0	22.1	22.2	22.3	22.4	22.6	22.7	22.2	22.4
896	0.4	0.8	1.6	2.7	4.3	6.2	8.6	11.2	9.5	11.7	14.3	17.1	14.5	17.0	19.5	17.5	19.8	18.2	20.4	19.2	21.2	20.3	22.2	21.4	20.8	22.5	22.1	21.7	21.4	23.0	22.7	22.6	22.5	22.4	22.5	22.5	22.6	22.7	22.8	22.9	23.1	23.3	23.4	23.6	23.0	23.2	23.5	23.7	23.2	23.5	23.8	23.4	23.7
1072	0.5	0.9	1.9	3.2	5.1	7.4	10.1	8.8	11.2	14.0	16.9	14.8	17.4	15.9	18.3	17.2	19.4	18.5	20.7	20.0	19.5	21.4	21.0	20.8	20.6	22.3	22.2	22.1	22.1	22.2	22.3	22.4	22.6	22.7	23.0	23.2	22.5	22.7	23.0	22.5	22.8	23.1	22.7	23.1	23.4	23.1	22.9	23.2	23.0	23.4	23.3	23.1	23.5
1264	0.5	1.1	2.2	3.9	6.0	8.7	12.1	10.4	13.2	16.4	14.5	17.4	16.1	18.8	17.7	20.2	19.4	19.0	21.2	20.8	20.6	20.5	22.4	22.3	22.4	22.4	22.6	22.7	23.0	22.0	22.3	22.6	22.9	22.3	22.7	23.1	22.6	23.1	22.7	23.2	22.9	23.3	23.2	23.0	23.5	23.3	23.2	23.2	23.2	23.6	23.6	23.6	23.7
1472	0.6	1.3	2.6	4.4	7.1	10.2	9.2	12.1	15.4	14.0	17.0	15.9	18.8	17.9	17.6	20.0	19.7	19.5	21.7	21.7	21.7	21.8	22.0	22.2	22.5	22.8	23.1	22.2	22.6	23.1	22.5	22.9	22.5	23.0	23.5	23.2	23.0	23.5	23.4	23.3	23.2	23.7	23.7	23.7	23.7	23.7	23.8	23.4	23.5	23.6	23.7	23.9	23.6
1696	0.7	1.5	2.9	5.1	8.1	11.7	10.7	13.9	12.9	16.1	19.5	18.3	17.7	17.5	20.2	20.0	20.0	20.2	20.4	20.7	21.0	21.4	21.8	22.3	21.5	22.0	22.6	22.0	22.6	22.3	22.8	22.6	22.4	23.1	23.0	22.9	22.9	22.9	23.0	23.0	23.1	23.2	23.4	23.1	23.2	23.4	23.2	23.4	23.3	23.5	23.4	23.3	23.5
1936	0.8	1.7	3.4	5.9	9.2	13.2	12.1	15.9	14.7	18.3	17.5	17.2	20.2	19.9	20.0	20.2	20.5	20.8	21.2	21.7	20.7	21.3	21.9	22.6	22.0	22.7	22.4	22.1	22.8	22.7	22.6	22.6	22.6	22.7	22.8	22.9	23.0	23.2	22.9	23.1	23.3	23.1	23.3	23.2	23.1	23.4	23.3	23.3	23.3	23.3	23.3	23.3	23.4
2192	0.9	1.9	3.8	6.6	10.4	9.9	13.7	13.0	16.6	16.2	19.7	19.4	19.4	19.6	20.0	20.4	21.0	21.5	20.6	21.2	22.0	22.7	22.2	21.9	22.6	22.4	22.3	22.3	23.1	22.4	22.5	22.7	22.9	23.1	22.7	23.0	23.2	23.0	23.4	23.2	23.2	23.1	23.5	23.1	23.5	23.2	23.3	23.3	23.4	23.3	23.4	23.5	23.4
2464	1.1	2.1	4.3	7.5	11.6	11.2	15.3	14.6	14.7	18.3	18.2	18.5	19.0	19.5	20.1	20.8	21.6	20.7	21.5	22.3	21.9	22.7	22.5	22.4	22.3	23.2	23.3	22.6	22.8	23.0	23.2	22.9	23.2	23.0	23.3	23.2	23.1	23.5	23.5	23.5	23.6	23.6	23.7	23.5	23.6	23.5	23.6	23.5	23.5	23.7	23.6	23.6	23.6
2752	1.2	2.4	4.8	8.4	8.7	12.5	12.5	16.4	16.4	16.8	20.4	20.7	21.2	21.8	20.4	21.3	20.5	21.5	22.5	22.1	22.0	23.0	22.9	23.0	23.1	23.2	23.5	23.0	23.3	23.6	23.4	23.2	23.1	23.6	23.5	23.6	23.6	23.7	23.8	23.9	23.7	23.9	24.0	24.0	23.9	23.8	23.8	23.8	23.9	23.9	24.0	24.0	23.9
3056	1.3	2.6	5.3	9.3	9.6	13.8	13.9	18.2	18.3	18.5	19.2	19.9	20.8	21.6	20.7	21.7	21.3	22.4	22.1	23.2	23.2	23.2	23.4	22.7	23.0	23.4	23.7	23.5	23.3	23.7	23.7	23.6	23.7	23.7	23.8	23.6	23.8	23.9	23.8	23.7	23.7	23.9	23.9	24.0	24.0	23.8	23.9	23.8	23.9	23.9	24.1	24.0	23.9
3376	1.5	2.9	5.8	10.2	10.5	15.2	15.3	15.8	16.6	20.5	21.2	19.5	20.6	21.7	21.0	20.7	21.9	21.9	22.0	23.2	22.4	22.7	23.0	23.4	23.1	23.6	23.4	23.3	23.4	23.4	23.5	23.2	23.4	23.6	23.5	23.7	23.6	23.9	23.6	23.7	23.7	23.8	23.9	23.8	24.0	23.9	23.8	23.8	23.9	23.9	23.9	24.0	23.9
3712	1.6	3.2	6.4	11.2	11.6	16.9	16.8	17.4	18.2	19.2	20.3	21.4	20.5	21.8	21.4	21.3	22.7	22.7	23.0	23.2	22.7	23.1	23.6	23.4	23.3	23.2	23.8	23.4	23.5	23.7	23.5	23.7	23.6	23.9	23.9	23.9	23.9	24.0	23.9	24.0	23.9	24.0	24.0	24.0	24.0	24.0	24.1	24.0	24.1	24.1	24.1	24.1	24.1
4064	1.8	3.5	7.0	12.1	12.6	18.4	18.5	19.0	19.9	20.9	22.1	21.0	22.4	22.0	21.8	21.9	22.2	22.5	22.9	23.4	23.0	23.6	23.4	23.4	23.5	23.6	23.7	23.5	23.7	23.6	23.5	23.5	23.6	23.6	23.8	23.9	23.8	24.0	23.9	23.9	23.9	24.0	24.0	23.9	24.1	23.9	24.0	23.8	24.0	23.9	24.0	23.9	23.8
4432	1.9	3.8	7.5	8.8	13.7	14.5	15.8	17.0	18.4	19.8	21.4	20.8	22.3	22.2	22.3	22.5	21.8	22.4	23.0	22.7	23.4	23.4	23.4	23.0	23.2	23.5	23.3	23.7	23.6	23.6	23.7	23.5	23.6	23.5	23.7	23.7	23.7	23.7	23.8	23.7	23.8	23.7	23.9	23.8	23.9	23.9	23.8	23.8	23.7	23.8	23.9	23.8	23.7
4816	2.1	4.1	8.3	9.6	14.9	15.8	17.1	18.4	20.0	19.1	20.8	20.6	22.4	22.5	21.5	22.0	22.6	22.3	23.1	23.0	23.1	23.3	23.5	23.2	23.1	23.5	23.4	23.5	23.6	23.7	23.6	23.5	23.7	23.7	23.8	23.8	23.7	23.8	23.8	23.7	23.7	23.8	23.8	23.9	23.9	23.9	23.9	23.7	23.9	23.8	23.8	23.7	23.7
5216	2.2	4.5	9.0	10.4	16.1	17.1	18.5	17.0	18.8	20.7	22.5	22.3	22.4	22.8	22.0	22.7	22.4	23.2	23.2	23.3	23.6	23.2	23.6	23.5	23.4	23.5	23.6	23.8	23.6	23.9	23.8	23.5	23.6	23.7	23.9	23.8	24.0	24.0	23.8	23.8	23.9	23.9	24.0	24.0	24.0	24.0	23.8	23.8	23.8	23.7	23.8	23.8	23.7
5632	2.4	4.8	9.7	11.2	12.8	18.5	16.4	18.3	20.3	20.0	22.0	22.2	22.6	23.1	22.6	22.4	22.4	23.3	23.5	23.7	23.4	23.8	23.7	23.3	23.4	23.6	23.9	23.8	23.7	24.1	23.8	24.0	23.9	24.0	24.0	24.0	24.1	24.2	24.1	24.1	24.0	24.1	24.1	24.1	24.0	24.0	24.0	24.0	24.0	24.0	24.0	23.8	23.8
6064	2.6	5.2	10.4	12.1	13.7	19.9	17.6	19.7	19.3	21.5	21.7	22.1	22.7	22.2	22.1	22.1	22.3	22.6	23.0	23.5	23.4	23.4	23.5	23.7	23.5	23.8	23.8	23.5	23.9	23.7	23.7	23.6	23.6	23.7	23.8	23.9	23.9	24.1	23.9	24.0	24.0	23.9	24.0	23.8	23.8	23.8	23.8	23.7	23.7	23.7	23.8	23.7	23.7
6512	2.8	5.5	11.1	12.9	14.7	16.8	18.9	18.4	20.7	20.9	21.4	22.1	23.0	22.6	22.6	22.8	23.1	22.8	23.3	23.3	23.4	23.5	23.7	23.6	23.6	23.6	23.7	23.9	23.8	24.0	23.8	23.9	24.0	23.9	24.1	24.1	23.9	24.0	24.0	23.9	23.9	23.8	23.8	23.9	23.9	24.0	23.8	23.8	23.9	23.7	23.9	23.8	23.7
6976	3.0	6.0	11.8	13.9	15.8	18.0	17.2	19.7	19.9	20.5	23.0	22.2	21.9	23.1	22.3	22.6	22.4	23.0	23.6	23.7	23.4	23.7	23.6	23.6	23.7	23.9	24.1	23.7	24.0	23.8	23.7	23.7	23.7	23.9	24.0	24.0	23.8	24.1	24.0	24.0	23.8	23.9	23.9	23.8	23.8	23.8	23.7	23.7	23.6	23.6	23.7	23.7	23.6
7456	3.1	6.3	12.6	14.8	16.9	19.3	18.4	18.6	19.3	21.9	22.7	22.3	22.3	22.5	22.8	22.6	22.5	23.2	23.4	23.6	23.5	23.5	23.5	23.7	23.5	23.8	23.8	23.6	23.8	23.7	23.7	23.5	23.7	23.8	23.8	23.9	23.8	24.0	23.8	23.7	23.7	23.8	23.8	23.6	23.7	23.7	23.6	23.5	23.5	23.7	23.6	23.6	23.5
7952	3.4	6.7	13.3	15.8	18.0	20.5	19.6	19.8	20.5	21.5	22.6	22.4	22.6	23.0	22.6	22.6	22.7	22.9	23.2	23.7	23.2	23.7	23.5	23.5	23.5	23.6	24.0	23.6	23.9	23.7	23.6	23.5	23.7	23.7	23.8	23.9	23.7	23.9	23.7	23.6	23.6	23.7	23.7	23.6	23.6	23.7	23.6	23.5	23.5	23.5	23.5	23.6	23.5
8464	2.4	4.8	9.5	12.2	15.1	17.9	18.1	18.9	20.0	21.2	22.5	22.6	22.9	22.6	22.5	22.6	22.9	23.2	23.2	23.7	23.4	23.6	23.6	23.6	23.7	23.6	23.9	23.7	24.0	23.7	23.7	23.7	23.6	23.7	23.9	23.8	23.8	23.8	23.7	23.7	23.6	23.6	23.7	23.5	23.7	23.5	23.5	23.4	23.5	23.5	23.5	23.6	23.4
8992	2.6	5.1	10.2	13.0	16.0	19.0	19.3	20.1	19.6	21.0	22.5	22.8	22.3	23.0	23.1	22.6	23.0	23.0	23.6	23.7	23.6	23.9	23.6	23.5	23.7	23.7	24.0	23.7	23.9	23.9	23.8	23.7	23.7	23.8	23.8	23.8	23.7	24.0	23.8	23.7	23.7	23.8	23.7	23.5	23.7	23.5	23.5	23.5	23.4	23.5	23.6	23.5	23.4
9536	2.7	5.5	10.8	13.7	16.9	20.1	20.4	19.3	20.7	22.2	22.5	22.9	22.7	23.5	22.9	22.7	22.7	23.3	23.5	23.8	23.4	23.9	23.8	23.7	23.7	23.8	24.0	23.8	24.0	24.0	23.8	23.8	23.9	23.7	23.8	23.9	23.7	23.9	23.8	23.7	23.7	23.7	23.7	23.6	23.8	23.6	23.6	23.5	23.6	23.5	23.6	23.6	23.5
10096	2.9	5.7	11.3	14.5	17.9	21.3	19.1	20.4	20.3	22.0	22.5	23.2	23.1	23.2	22.9	22.9	23.0	23.2	23.6	23.6	23.7	23.9	23.9	23.7	23.8	23.8	24.0	23.9	24.1	24.1	23.9	23.9	23.8	23.9	24.0	23.8	23.9	23.9	23.7	23.8	23.7	23.9	23.8	23.7	23.8	23.7	23.8	23.7	23.6	23.6	23.7	23.7	23.5
10672	3.0	6.1	12.0	15.3	18.9	19.1	20.2	19.7	20.1	21.9	22.6	22.5	22.6	23.0	22.9	23.0	23.2	23.2	23.6	23.7	23.6	24.0	23.8	23.7	23.7	24.0	24.1	23.9	23.9	23.9	23.8	23.7	23.8	23.7	23.8	23.9	23.6	23.9	23.7	23.7	23.7	23.8	23.7	23.5	23.6	23.5	23.5	23.5	23.5	23.5	23.6	23.7	23.5
11264	3.2	6.4	12.8	16.1	16.4	20.2	19.1	20.8	21.1	21.8	22.7	22.7	23.0	23.6	22.9	23.1	23.0	23.5	23.7	23.9	23.9	24.0	23.9	23.7	23.8	24.0	24.1	24.1	24.0	24.1	23.8	23.8	23.8	23.8	23.9	23.8	23.8	23.8	23.9	23.6	23.8	23.7	23.8	23.6	23.6	23.6	23.6	23.5	23.6	23.6	23.6	23.6	23.5
11872	3.4	6.8	13.4	17.1	17.3	21.2	20.1	20.2	20.9	21.7	22.8	23.0	22.7	23.4	23.0	22.8	23.3	23.5	23.8	23.8	23.6	23.8	23.9	23.8	23.7	23.8	24.2	23.9	23.9	24.0	23.8	23.8	23.7	23.8	23.8	23.7	23.7	24.0	23.6	23.7	23.6	23.8	23.7	23.6	23.7	23.6	23.7	23.6	23.6	23.7	23.6	23.7	23.4
12496	3.6	7.1	14.1	14.1	18.1	19.5	19.2	19.7	20.6	21.7	22.9	22.5	23.2	23.3	23.0	23.0	23.2	23.5	23.9	23.7	23.7	24.0	23.9	23.7	23.7	23.9	24.1	23.9	23.9	24.0	23.7	23.7	23.7	23.7	23.7	23.8	23.7	23.8	23.7	23.6	23.8	23.8	23.7	23.6	23.8	23.7	23.5	23.5	23.5	23.6	23.6	23.6	23.5
13136	3.7	7.4	14.8	14.9	19.0	20.4	20.2	20.7	20.5	21.7	23.1	22.8	22.9	23.2	23.1	23.3	23.2	23.6	23.7	23.7	23.7	23.9	23.9	23.8	23.7	24.0	24.0	24.0	23.9	23.9	23.7	23.6	23.6	23.7	23.8	23.6	23.7	23.9	23.6	23.7	23.5	23.7	23.7	23.5	23.7	23.6	23.5	23.5	23.5	23.6	23.6	23.6	23.4
13792	3.9	7.8	15.5	15.6	17.0	21.5	19.4	20.3	21.5	21.8	23.2	23.1	23.3	23.7	23.2	23.1	23.1	23.6	23.9	23.9	23.8	24.0	24.0	23.7	23.7	23.9	24.0	23.9	23.8	24.0	23.8	23.7	23.6	23.9	23.7	23.9	23.8	23.8	23.8	23.8	23.8	23.7	23.7	23.6	23.7	23.7	23.7	23.6	23.6	23.7	23.7	23.6	23.6
14464	4.1	8.2	16.2	16.4	17.8	20.0	20.3	19.9	21.3	21.8	23.4	22.7	23.1	23.7	23.4	23.3	23.4	23.7	24.1	23.9	23.9	24.0	24.0	23.8	23.8	24.0	24.2	24.0	23.8	23.9	23.8	23.7	23.7	23.7	23.8	23.8	23.7	23.9	23.7	23.7	23.6	23.7	23.7	23.6	23.7	23.7	23.6	23.6	23.6	23.7	23.6	23.7	23.4
15152	4.1	8.2	16.6	17.1	18.7	20.9	19.6	20.9	21.2	21.9	23.6	23.1	23.0	23.7	23.1	23.2	23.1	23.5	24.0	23.9	23.8	24.0	23.9	23.8	23.9	24.0	24.1	23.8	23.9	23.8	23.7	23.5	23.5	23.5	23.6	23.7	23.5	23.8	23.5	23.6	23.5	23.6	23.6	23.4	23.6	23.6	23.5	23.4	23.4	23.5	23.6	23.5	23.3
15856	4.2	8.4	16.7	17.9	19.5	21.8	20.5	20.5	21.1	22.0	23.0	23.3	23.5	23.7	23.2	23.1	23.5	23.6	23.9	24.0	24.0	24.0	24.0	23.9	23.9	24.0	24.1	23.8	23.9	23.8	23.7	23.6	23.7	23.7	23.8	23.8	23.6	23.9	23.7	23.8	23.8	23.7	23.7	23.6	23.7	23.7	23.6	23.6	23.5	23.6	23.7	23.7	23.6
16576	3.4	6.8	13.6	15.4	17.8	20.5	19.9	20.2	21.0	22.1	23.2	23.1	23.3	23.7	23.4	23.4	23.2	23.5	23.9	23.8	23.9	24.0	23.9	23.8	23.9	23.9	24.1	23.8	24.0	23.9	23.8	23.7	23.5	23.6	23.6	23.7	23.6	23.7	23.7	23.7	23.6	23.7	23.7	23.6	23.7	23.5	23.5	23.5	23.4	23.5	23.5	23.6	23.5
17312	3.6	7.1	14.2	16.0	18.6	21.3	20.7	21.1	21.0	22.2	23.5	22.9	23.2	23.8	23.2	23.3	23.2	23.6	23.8	23.9	23.8	24.1	24.0	23.7	23.8	23.8	24.1	23.8	23.8	23.7	23.6	23.6	23.5	23.5	23.6	23.8	23.5	23.7	23.7	23.5	23.6	23.6	23.6	23.5	23.6	23.5	23.5	23.5	23.4	23.4	23.5	23.5	23.5
18064	3.7	7.4	14.8	16.7	19.4	22.3	20.2	20.9	21.0	22.3	23.7	23.2	23.2	23.4	23.4	23.3	23.3	23.6	23.9	23.8	23.8	24.0	23.9	23.8	23.7	23.8	24.0	23.7	23.8	23.7	23.5	23.6	23.4	23.6	23.5	23.7	23.5	23.7	23.5	23.6	23.5	23.6	23.6	23.4	23.5	23.5	23.5	23.4	23.4	23.5	23.5	23.5	23.4
18832	3.8	7.7	15.3	17.3	20.1	21.1	21.0	20.7	21.0	22.4	23.3	23.0	23.2	23.5	23.3	23.3	23.4	23.5	23.9	23.9	23.9	24.1	23.9	23.7	23.9	23.8	24.0	23.7	23.9	23.6	23.5	23.5	23.4	23.4	23.6	23.6	23.5	23.7	23.5	23.5	23.5	23.6	23.6	23.4	23.5	23.5	23.4	23.4	23.4	23.4	23.5	23.5	23.4
19616	4.0	7.9	15.8	18.0	18.6	21.9	20.5	20.5	21.9	22.6	23.5	23.3	23.6	23.6	23.5	23.3	23.3	23.7	23.9	23.9	23.9	24.1	23.9	23.7	23.8	23.7	24.1	23.7	23.7	23.6	23.6	23.5	23.5	23.5	23.6	23.6	23.5	23.8	23.7	23.6	23.6	23.6	23.7	23.5	23.6	23.5	23.4	23.4	23.5	23.4	23.5	23.5	23.4
20416	4.1	8.1	16.3	18.7	19.3	22.8	21.3	21.3	21.9	22.7	23.8	23.2	23.6	23.7	23.4	23.3	23.4	23.7	24.0	24.1	23.9	24.1	24.1	23.8	23.8	23.8	24.1	23.8	23.9	23.7	23.6	23.5	23.5	23.6	23.6	23.8	23.7	23.8	23.6	23.6	23.6	23.7	23.6	23.5	23.7	23.5	23.5	23.4	23.5	23.5	23.5	23.6	23.5
21232	4.2	8.4	16.6	16.6	20.1	21.7	20.8	21.1	21.9	22.2	23.4	23.1	23.6	23.9	23.3	23.3	23.6	23.6	24.1	24.0	24.0	24.1	23.9	23.8	23.7	23.8	23.8	23.7	23.8	23.8	23.6	23.6	23.6	23.6	23.6	23.8	23.7	23.7	23.7	23.6	23.6	23.6	23.7	23.5	23.7	23.6	23.5	23.5	23.5	23.5	23.5	23.6	23.5
22064	4.2	8.4	16.9	17.3	18.7	22.5	20.4	21.0	21.9	22.4	23.7	23.4	23.7	23.7	23.6	23.4	23.4	23.6	23.9	23.9	23.9	24.1	23.8	23.7	23.6	23.7	23.9	23.7	23.7	23.7	23.5	23.5	23.4	23.4	23.5	23.5	23.5	23.7	23.5	23.5	23.4	23.6	23.5	23.4	23.6	23.6	23.4	23.5	23.4	23.4	23.5	23.5	23.3

HOW DO WE MAKE SMALL SKINNY FASTER?

IF WE CANT USE LARGE TILE, PRECEDING MEMORY MUST BE FASTER



AMD

Thank you

SUPPORTED PROBLEM TYPES

AMD

- Example:

$$C_{ijk} = \alpha \sum_{lmn} A_{inlkm} * B_{mjlkn} + \beta C_{ijk}$$

- Tensor indices are ordered shortest to largest stride, i.e., the zeroth index/dimension has a stride of 1 (typically).
- C indices are labeled alphabetically starting with "i".
- Summation indices are labeled alphabetically picking up where C indices left off.
- A, B indices are then labeled according to which summation or C index they correspond to.
- Tensile kernel will employ:
  - Enough work-groups to cover all the dimensions of C
  - Enough nested loops to carry out the multi-dimensional summation.
- A kernel for a given problem type will give correct answer for any problem sizes.

LARGE & SKINNY - 90% OF HIGHEST

MINIMUM PROBLEM SIZE FOR HIGH PERFORMANCE

N	TFlops	90%	M	MT	Flops/Byte	# WG	WG/CU	CU Occ
16	2.3	2.0	7168	64x32	10.6	112	1.75	4
32	4.3	3.8	5888	32x32	8	184	1.3	8
64	8.7	8.0	11776	128x32	12.8	184	1.3	4
128	11.5	10.3	15488	64x128	21.3	242	3.8	3
192	11.0	10.5	13568	64x64	16	636	5.6	4
256	11.8	10.6	10112	128x64	21.3	316	4.9	3

"skinny"

Not "skinny"

Large Tile

Overfill CUs

Conclusion 3: To achieve peak performance, need sufficient work to fill GPU with large tiles.

Conclusion 4: Performance of small or skinny sizes bottle-necked by L2 bandwidth.

Vega10 @ 27.3 Flops / byte =

(13200 Gflop/sec) / (483 Gbyte/sec)

L2 is 2X faster