ENERGY-EFFICIENT ARCHITECTURES FOR EXASCALE SYSTEMS

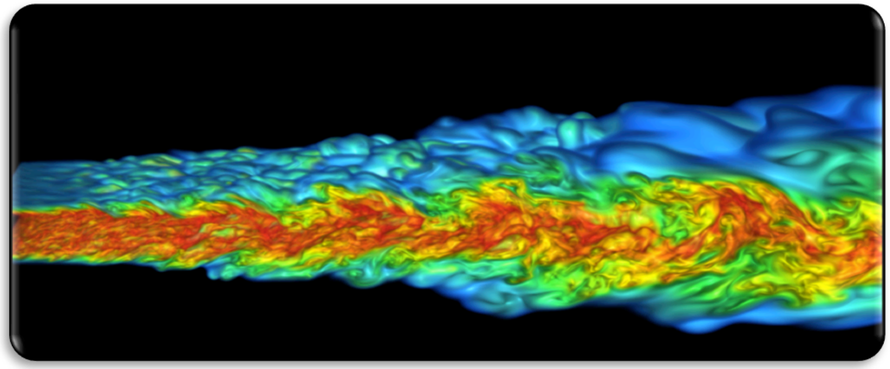
Dr. Stephen W. Keckler
Senior Director of Architecture Research, NVIDIA

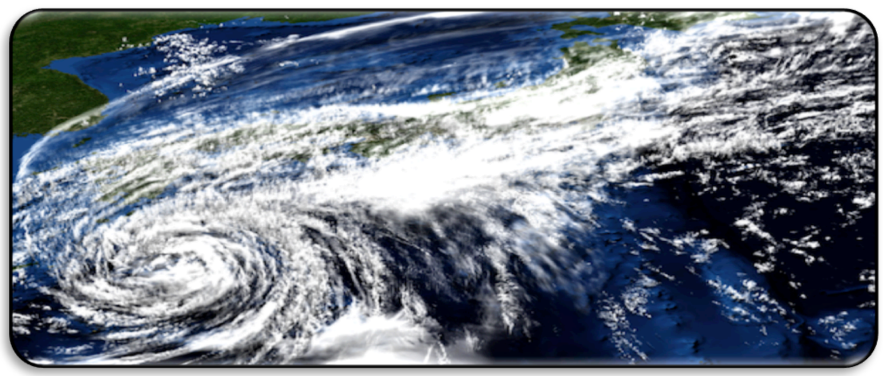
The Goal:

Sustained ExaFLOPS on Problems of Interest

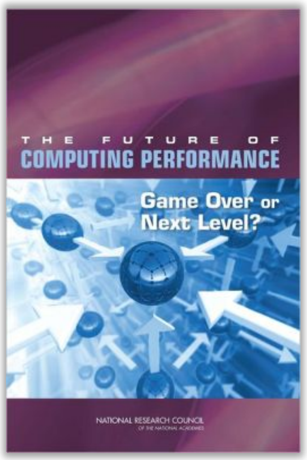
…

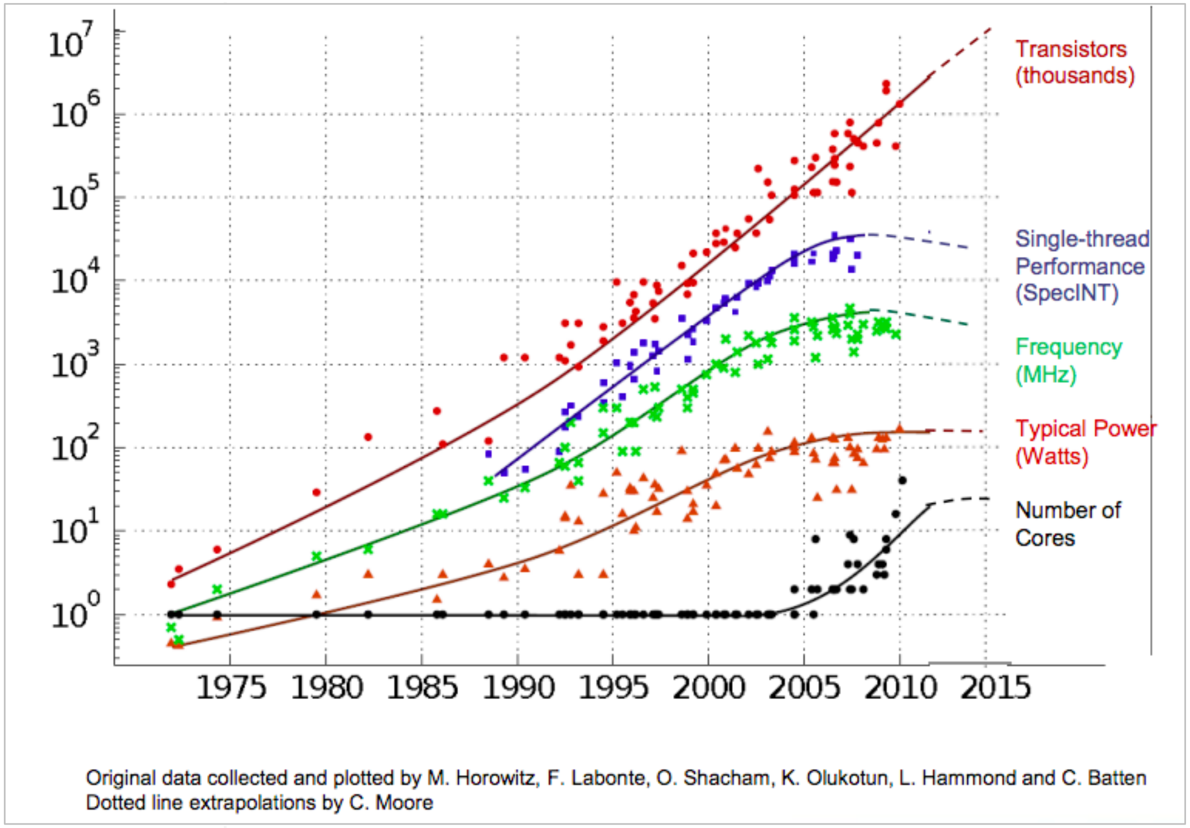
at reasonable cost



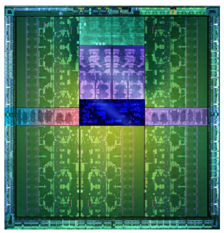


The End of Historic Scaling





Source: C Moore, Data Processing in ExaScale-ClassComputer Systems, Salishan, April 2011



CORAL
150-300PF (5-10x)
11MW (1.1x)
14-27 GFLOPs/W (7-14x)
Lots of Threads

2023

1,000PF (50x)
72,000HCNs (4x)
20MW (2x)
50 GFLOPs/W (25x)
~10^{10} Threads (1000x)

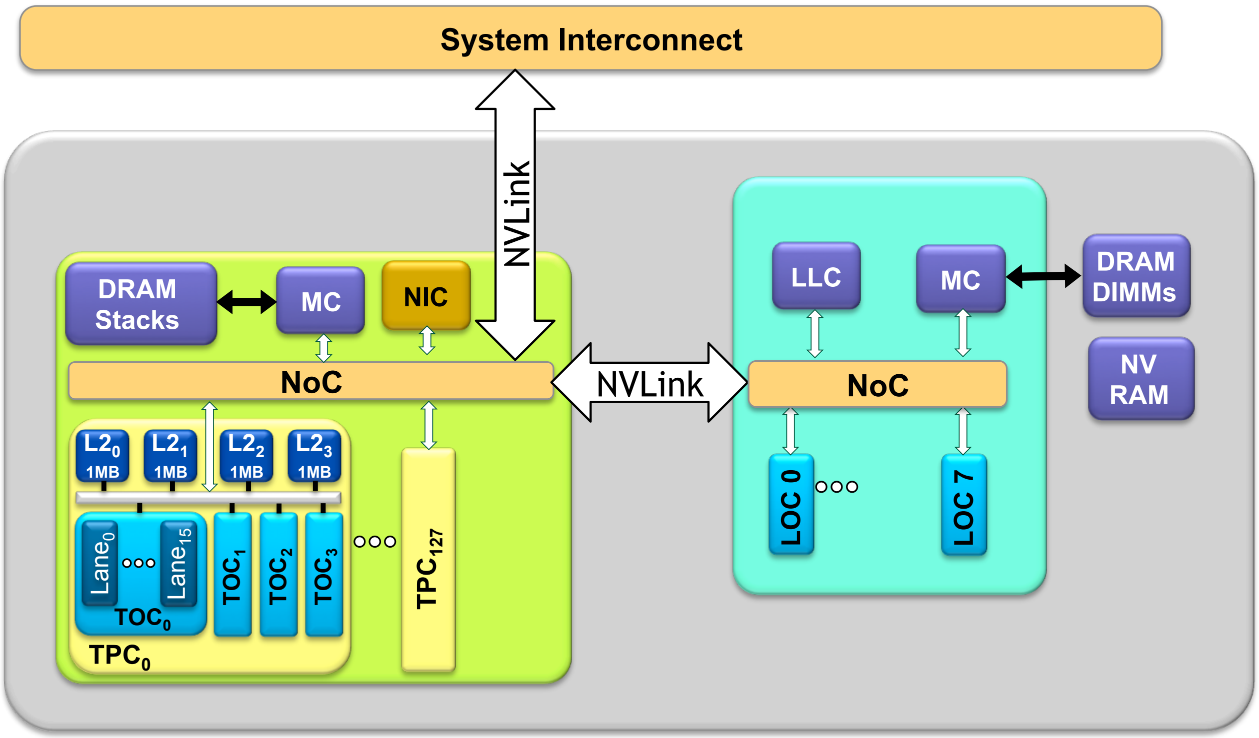
2013

2017

You Are Here

20PF
18,000 GPUs
10MW
2 GFLOPs/W
~10^{7} Threads

HETEROGENEOUS NODE



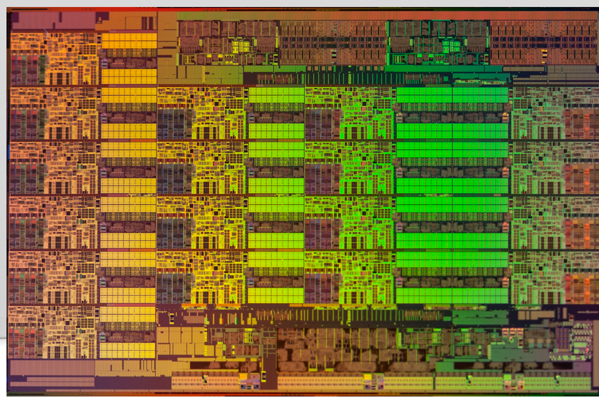
How do we get to 50GFlops/Watt?

Start with an energy-efficient architecture

CPU
130 pJ/flop (Vector SP)

Optimized for Latency

Deep Cache Hierarchy

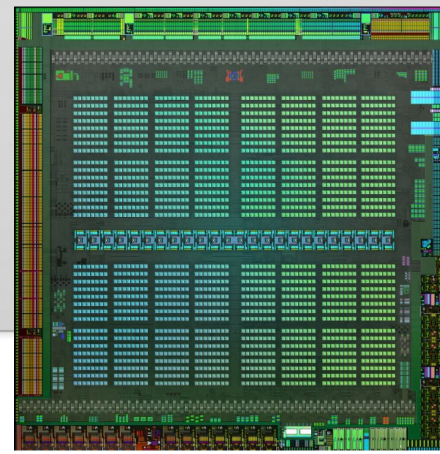


Haswell
22 nm

GPU
30 pJ/flop (SP)

Optimized for Throughput

Explicit Management
of On-chip Memory

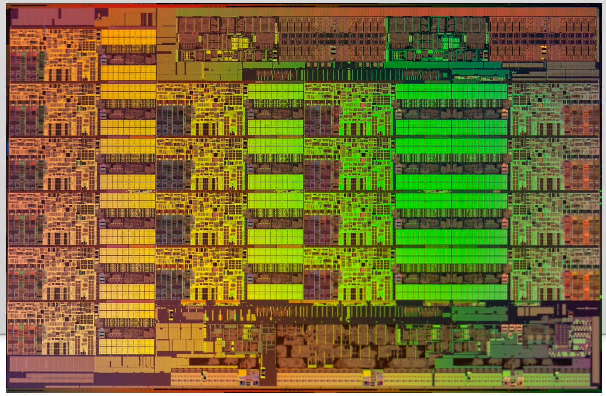


Maxwell
28 nm

CPU
2 nJ/flop (Scalar SP)

Optimized for Latency

Deep Cache Hierarchy

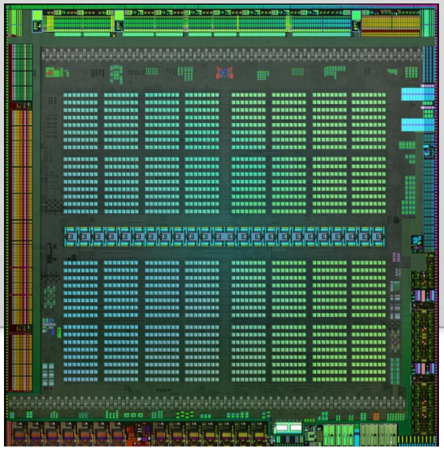


Haswell
22 nm

GPU
30 pJ/flop (SP)

Optimized for Throughput

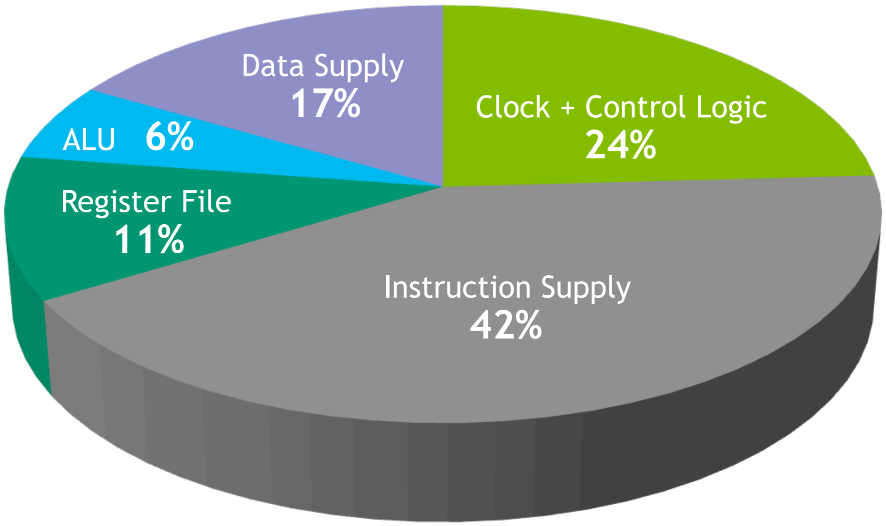
Explicit Management
of On-chip Memory



Maxwell
28 nm

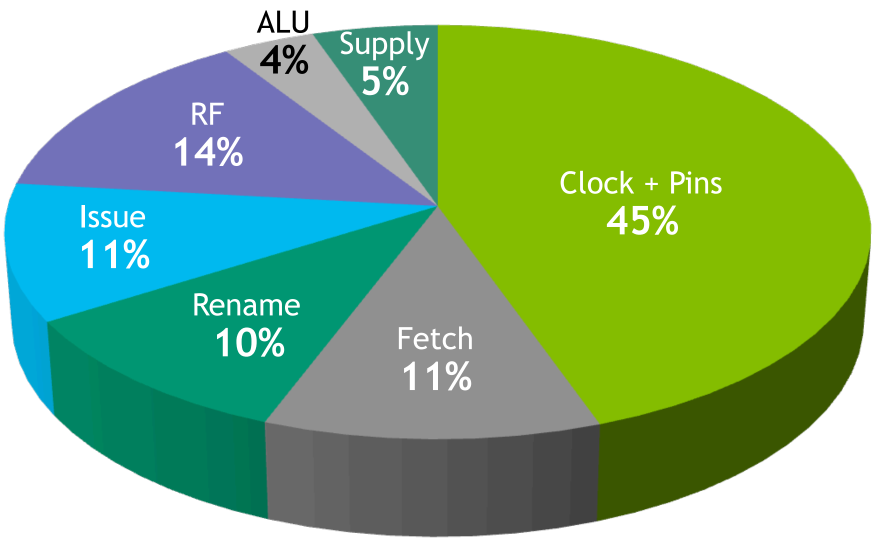
HOW IS POWER SPENT IN A CPU?

In-order Embedded



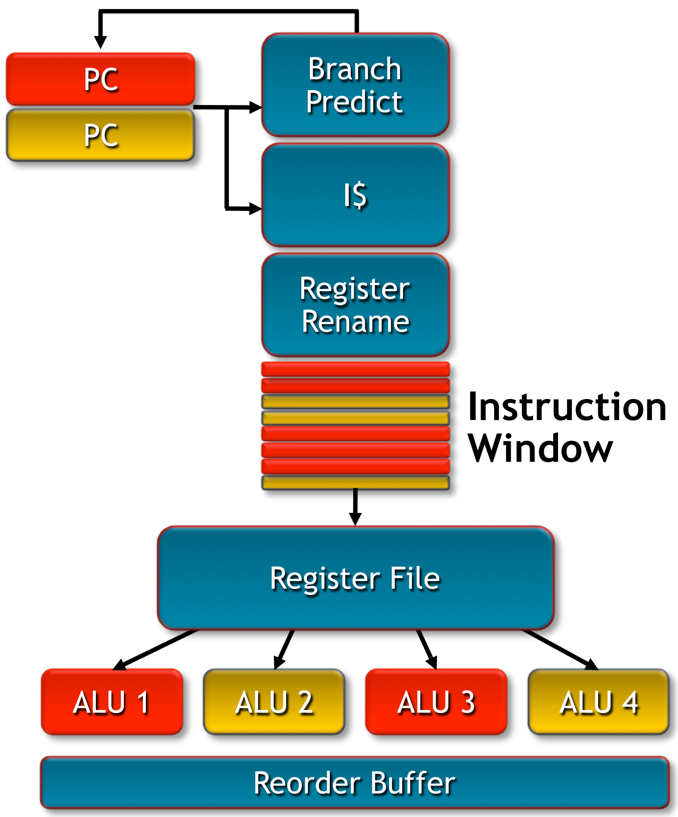
Dally [2008] (Embedded in-order CPU)

OOO Hi-perf

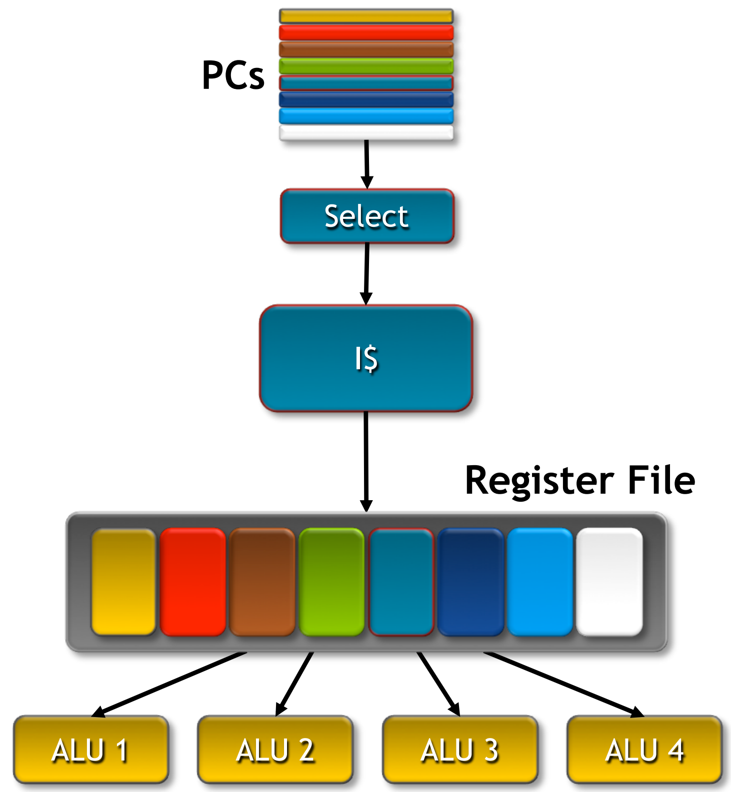


Natarajan [2003] (Alpha 21264)

Latency-Optimized Core (LOC)



Throughput-Optimized Core (TOC)



How do we continue to scale energy efficiency

...in a world where technology scaling is diminished?

Do Less Work

Eliminate waste and redundancy

Move fewer bits

Move data more efficiently

DO LESS WORK

Mixed Precision Arithmetic

double-precision

1	11	52

5x precision bits
60x range

single-precision

1	8	23

Only use as much precision as you need

Exploit mix of representations

Scaled arithmetic

4x throughput
4x bandwidth
4x capacity
< ¼ energy/op

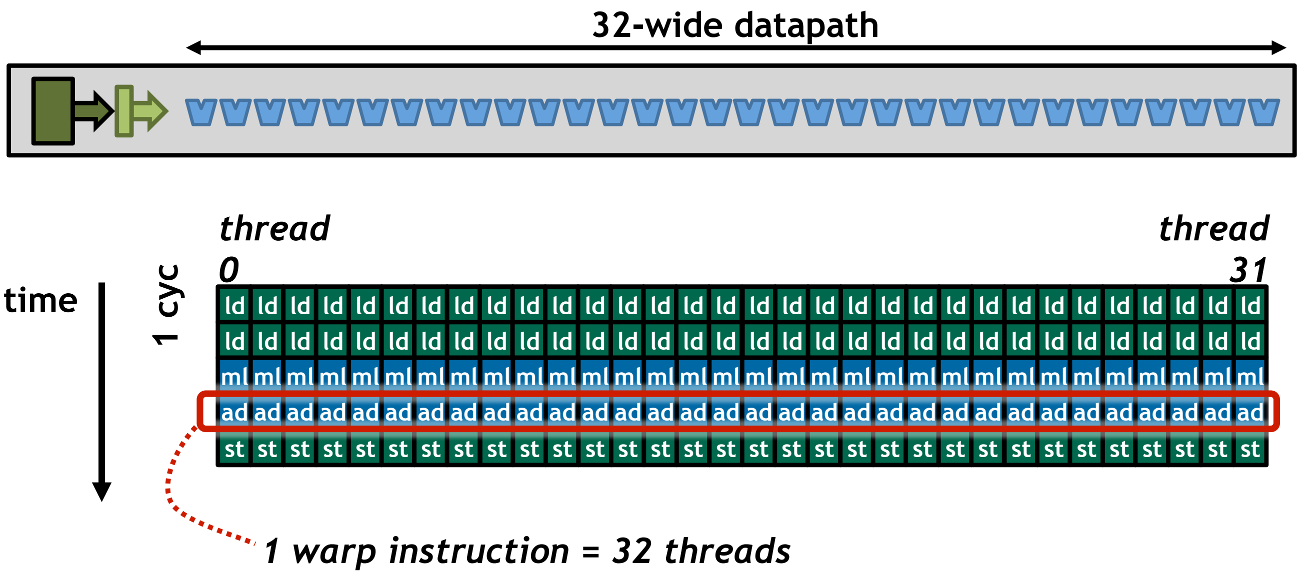
half-precision

1	5	10

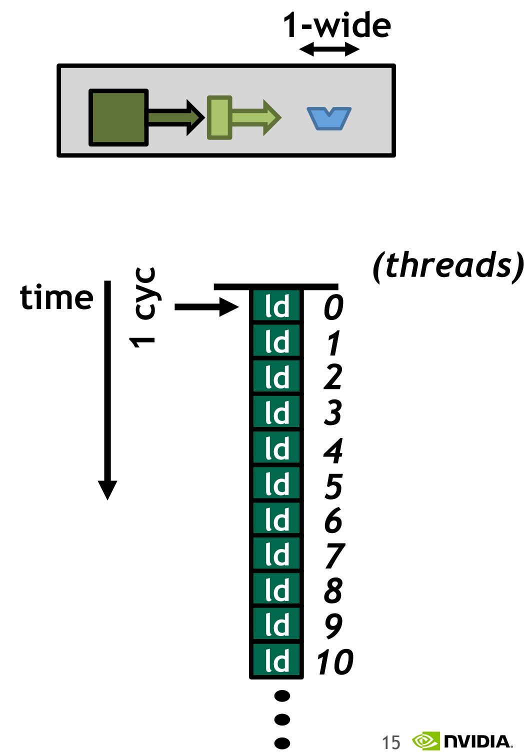
ELIMINATE WASTE

Temporal SIMT

Spatial SIMT (current GPUs)

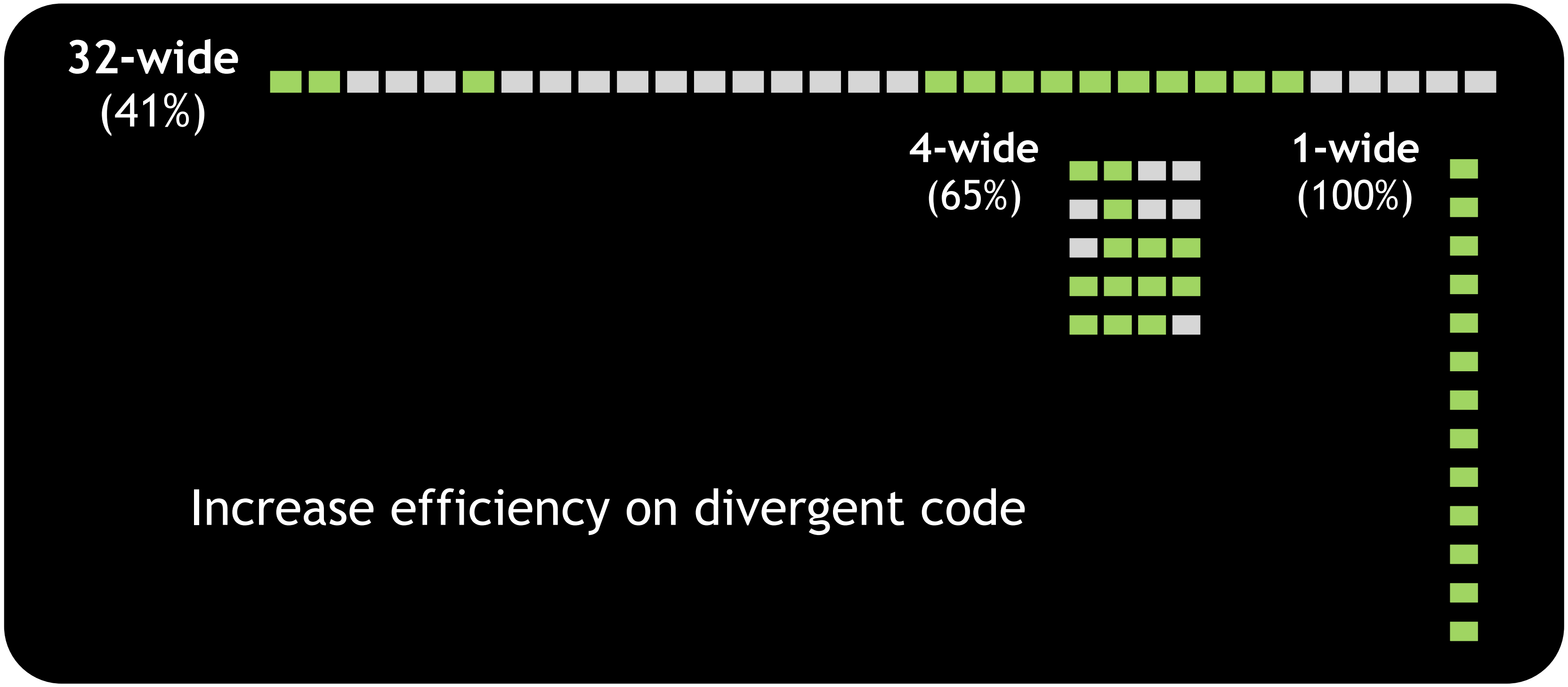


Pure Temporal SIMT



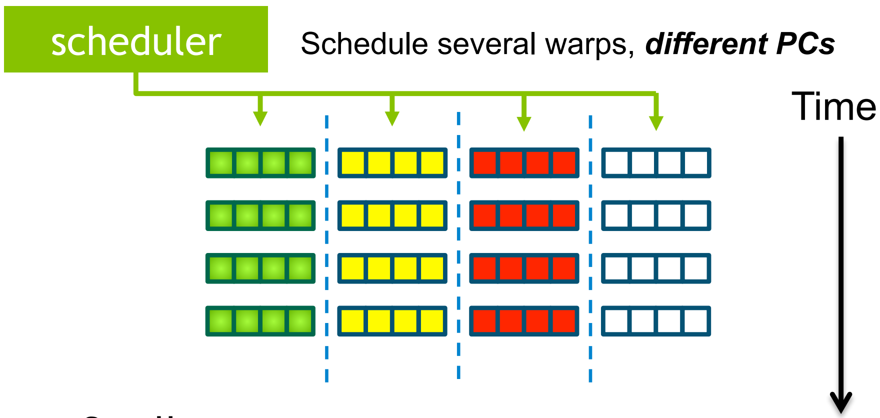
ELIMINATE WASTE

Temporal SIMT



ELIMINATE WASTE

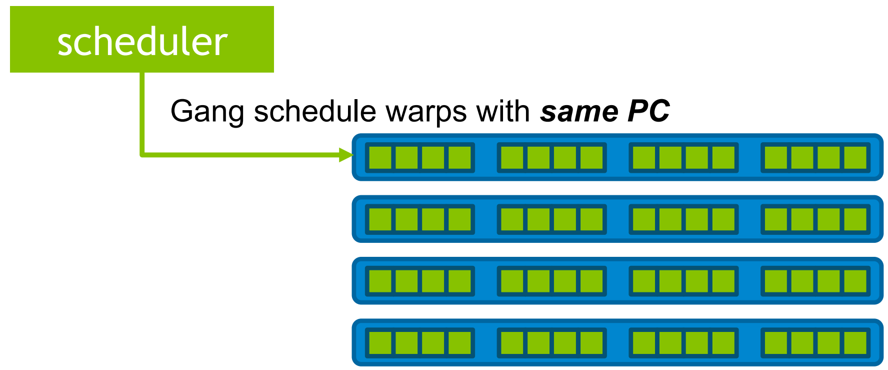
Variable Warp Sizing



Small warps

+ Improved perf for divergent code

+ Better SIMD utilization



Emulate wide warp HW

+ Wider converged execution

+ Memory locality/convergence

+ Reduced power (frontend)

Rogers [ISCA 2015]

ELIMINATE REDUNDANCY

Scalarization

SIMT Execution

scalar op	LD R2←<A>	LD R2←<A>	LD R2←<A>	LD R2←<A>
vector load	LD R3←R2, 1	LD R3←R2, 2	LD R3←R2,3	LD R3←R2, 4
vector op	ADD R4←R3, 2	ADD R4←R3, 2	ADD R4←R3, 2	ADD R4←R3, 2

Scalarized SIMT Execution

LD SR2←<A>				
VLD SR3←SR2, 1				
	ADD R4←SR3, 2	ADD R4←SR3, 2	ADD R4←SR3, 2	ADD R4←SR3, 2

Lee [CGO 2013]

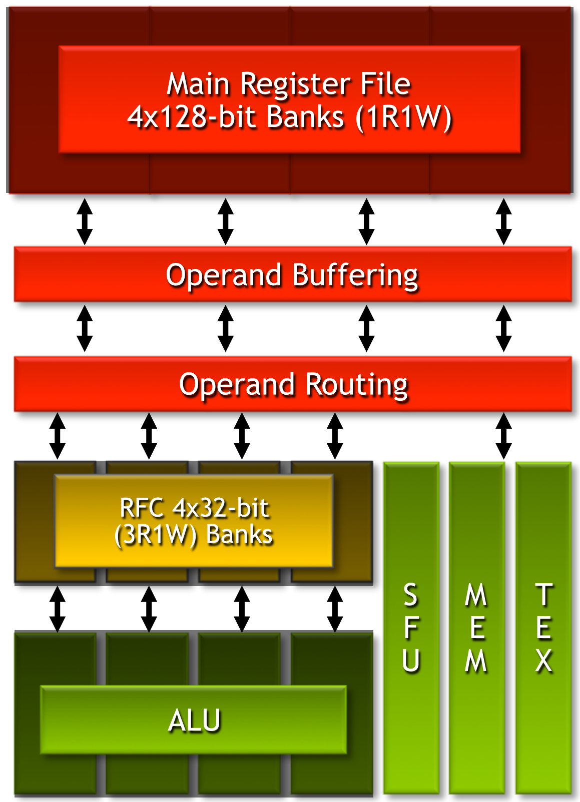
MOVE FEWER BITS

Register File Cache (RFC)

Small multi-ported register file

Capture locality of commonly used operands

Can reduce RF energy by 50%



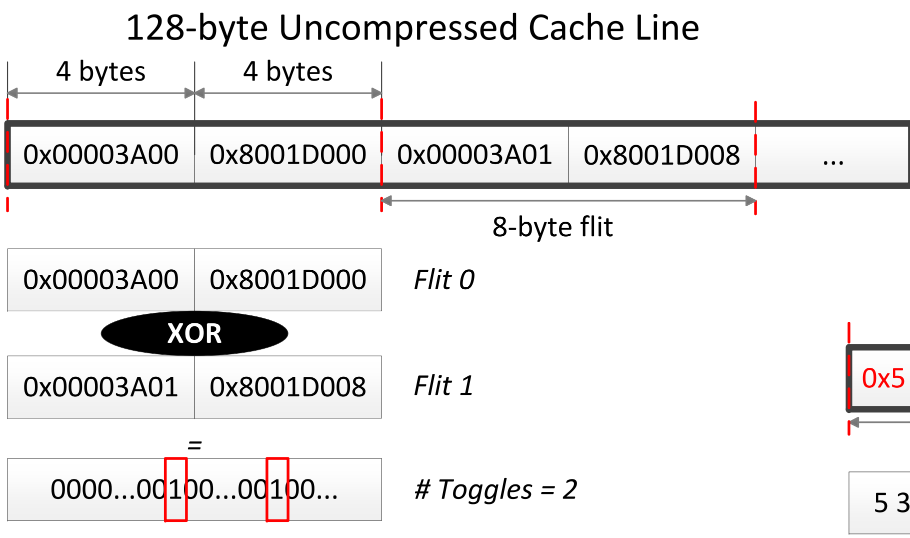
Gebhart [ISCA 2011]

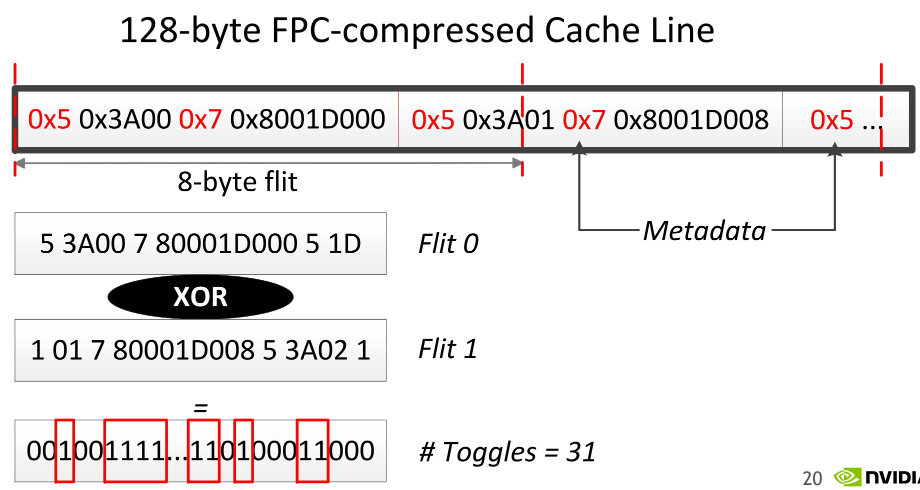
MOVE DATA MORE EFFICIENTLY

Toggle-aware Compression

Compression can increase power consumption

Goal: reduce bus toggling

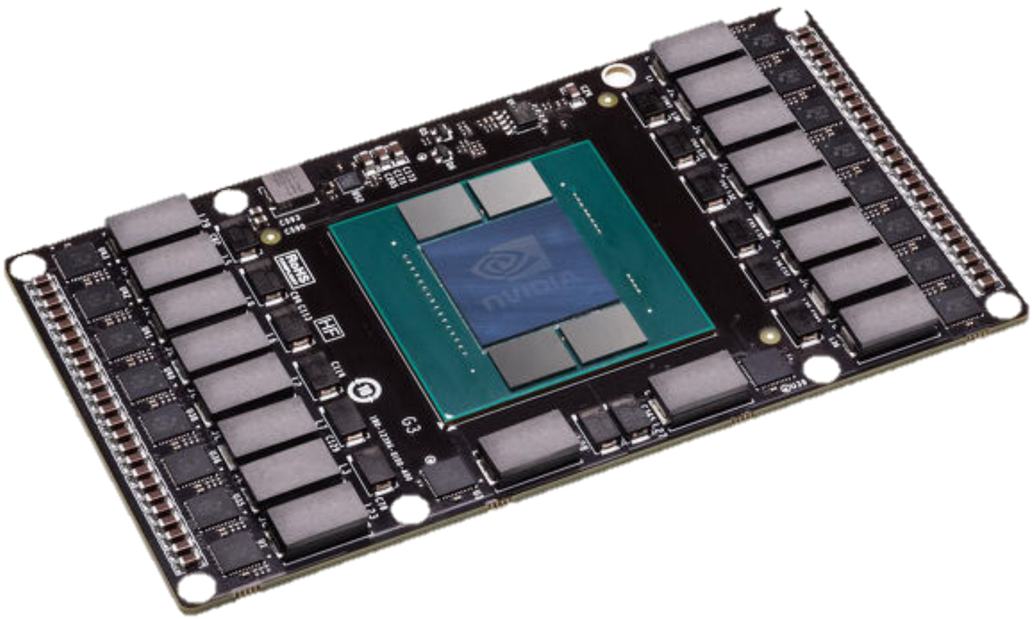




Pekhimenko [HPCA 2016]

MINIMIZE DATA MOVEMENT

Packaging



High-bandwidth on-package memory

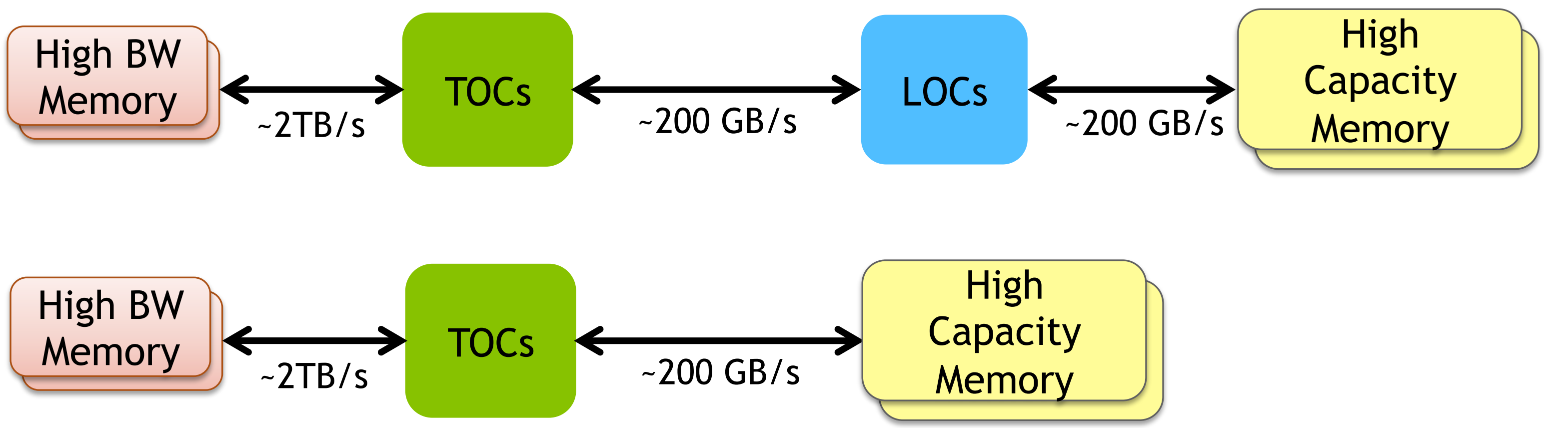
Reduces distance

Increases bandwidth

Offers opportunity to optimize signaling circuits

MINIMIZE DATA MOVEMENT

Heterogeneous DRAM Architectures



Challenges

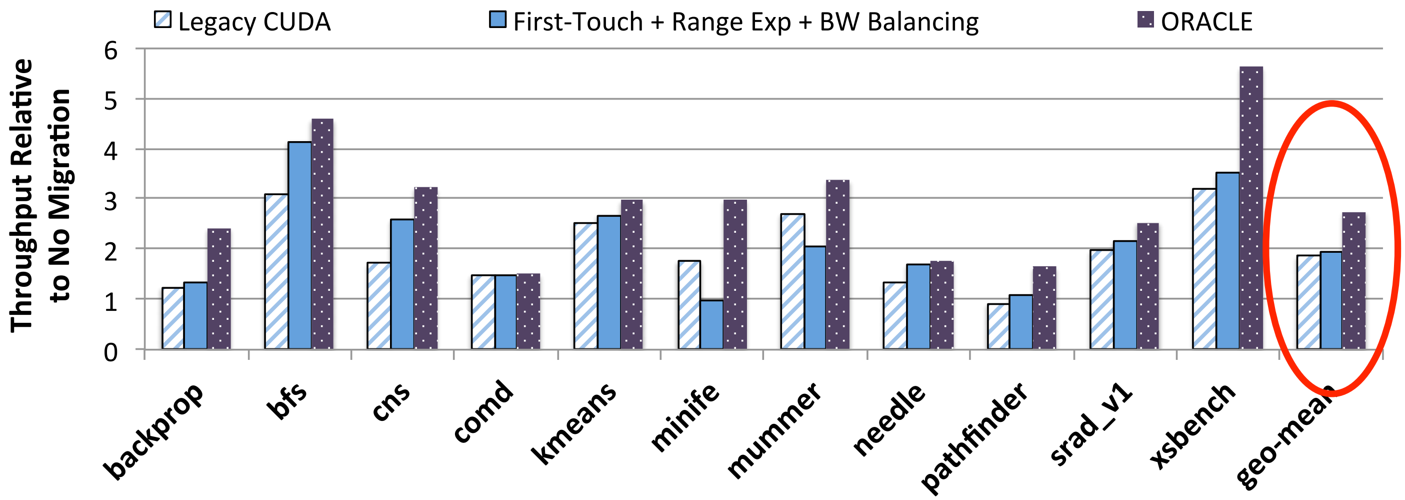
- Exploiting all available bandwidth
- Maximizing locality for frequently accessed data

MINIMIZE DATA MOVEMENT

Software-managed Caching with On-Package Memory

Strategies

- Aggressively migrate pages upon First-Touch to GDDR memory
- Pre-fetch neighbors of touched pages to reduce TLB shootdowns
- Throttle page migrations when nearing peak BW



Competitive with manual memory copy

Close to “perfect” prefetch

Agarwal [HPCA 2015]

MINIMIZE DATA MOVEMENT

Hardware Managed DRAM Cache

Tag overhead: hundreds of MB

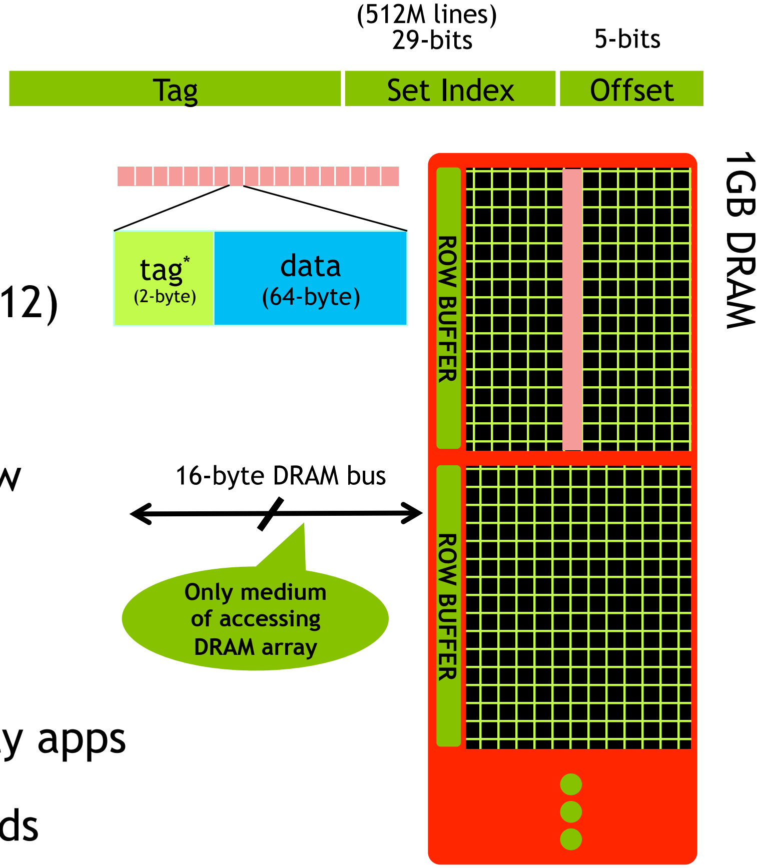
- Alloy tag and data in same DRAM row (Micro12)

Cache organization: optimize for bandwidth

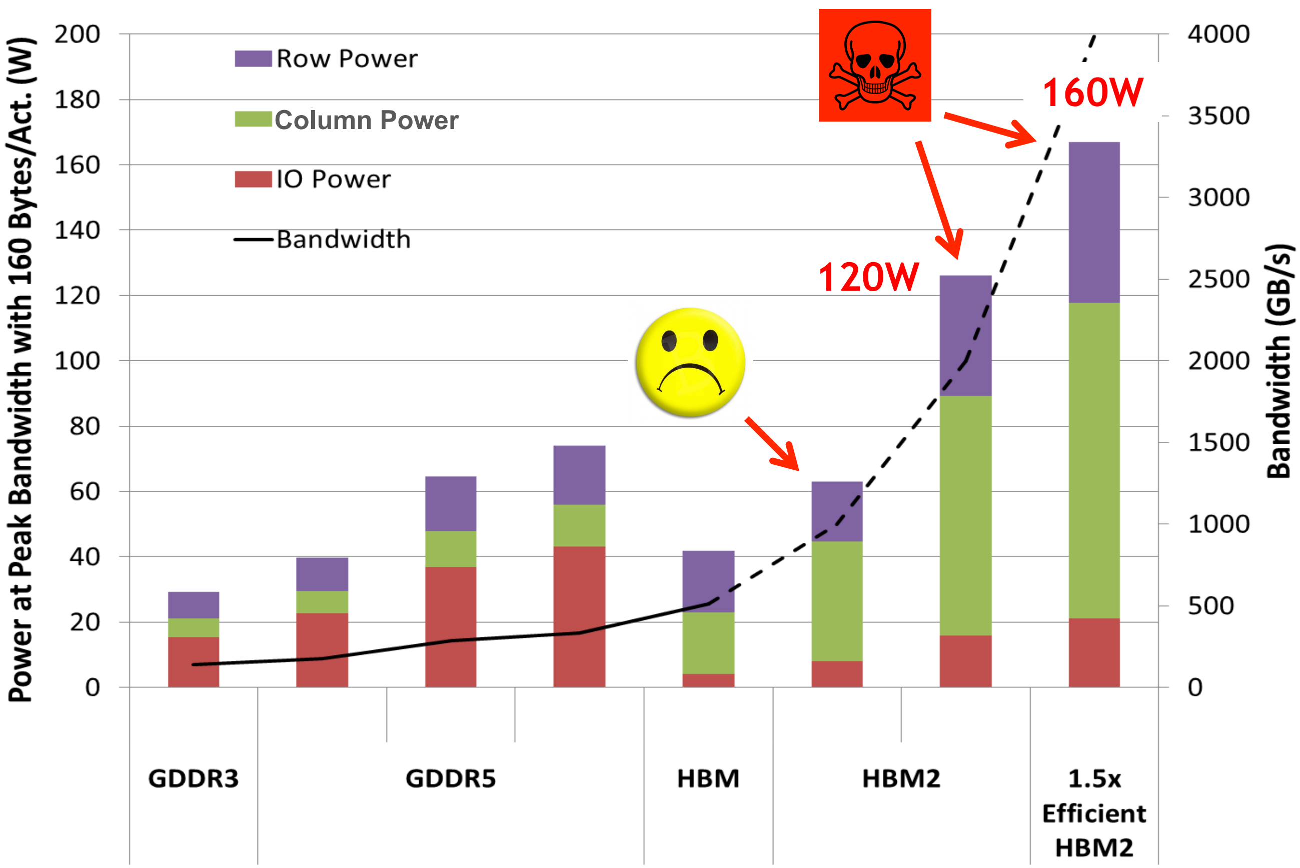
- Direct mapped, consecutive sets in same row

Results

- Fine-grained transfers good for lower locality apps
- Can eliminate some page migration overheads



LOOMING MEMORY POWER CRISIS



SUMMARY

Do Less Work

Eliminate waste and redundancy

Move fewer bits

Move data more efficiently


NVIDIA