Rogers ISCA15

A Varaible Warp Size Architecture

Divergence

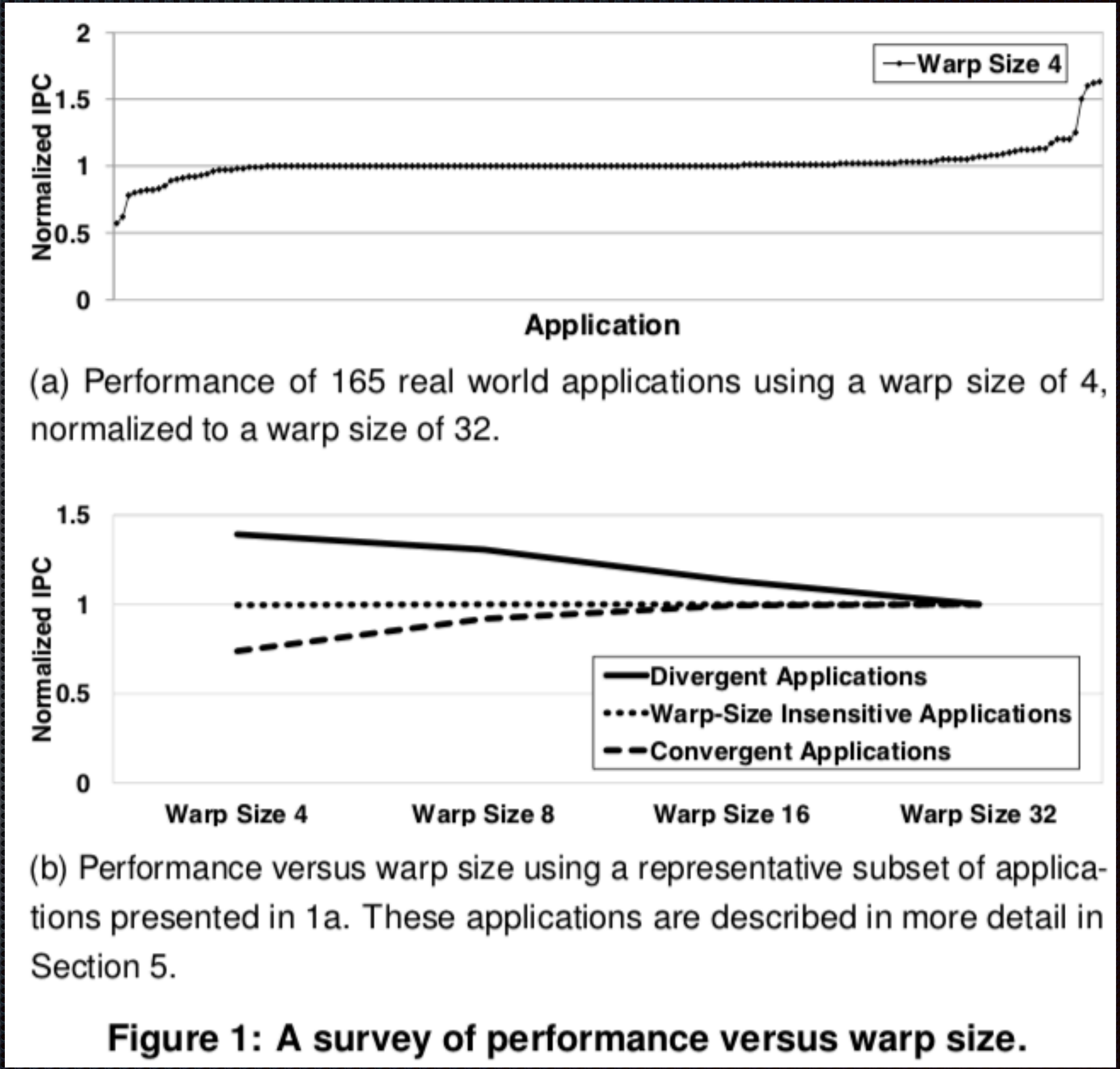
- Threads run well if they do the same thing everywhere
- Branches cause control flow to diverge — part of the warp goes idle
- Memory accesses that can’t be coalesced into dense cache-line fetches are another form of divergence — memory bandwidth is underutilized

Application Study

- The vast majority of applications work well at 32 threads per warp (may be biased by architecture)

- A small number will perform better with 4 threads

- A similar number will be worse (called convergent)



(a) Performance of 165 real world applications using a warp size of 4, normalized to a warp size of 32.

(b) Performance versus warp size using a representative subset of applications presented in 1a. These applications are described in more detail in Section 5.

Figure 1: A survey of performance versus warp size.

IPC/Lanes Active

- Convergent applications increase IPC by 0.4 for 4-threads
- Divergent application decrease IPC by 0.3 for 4-threads
- Lane utilization increases by 4 and decreases by 6, respectively

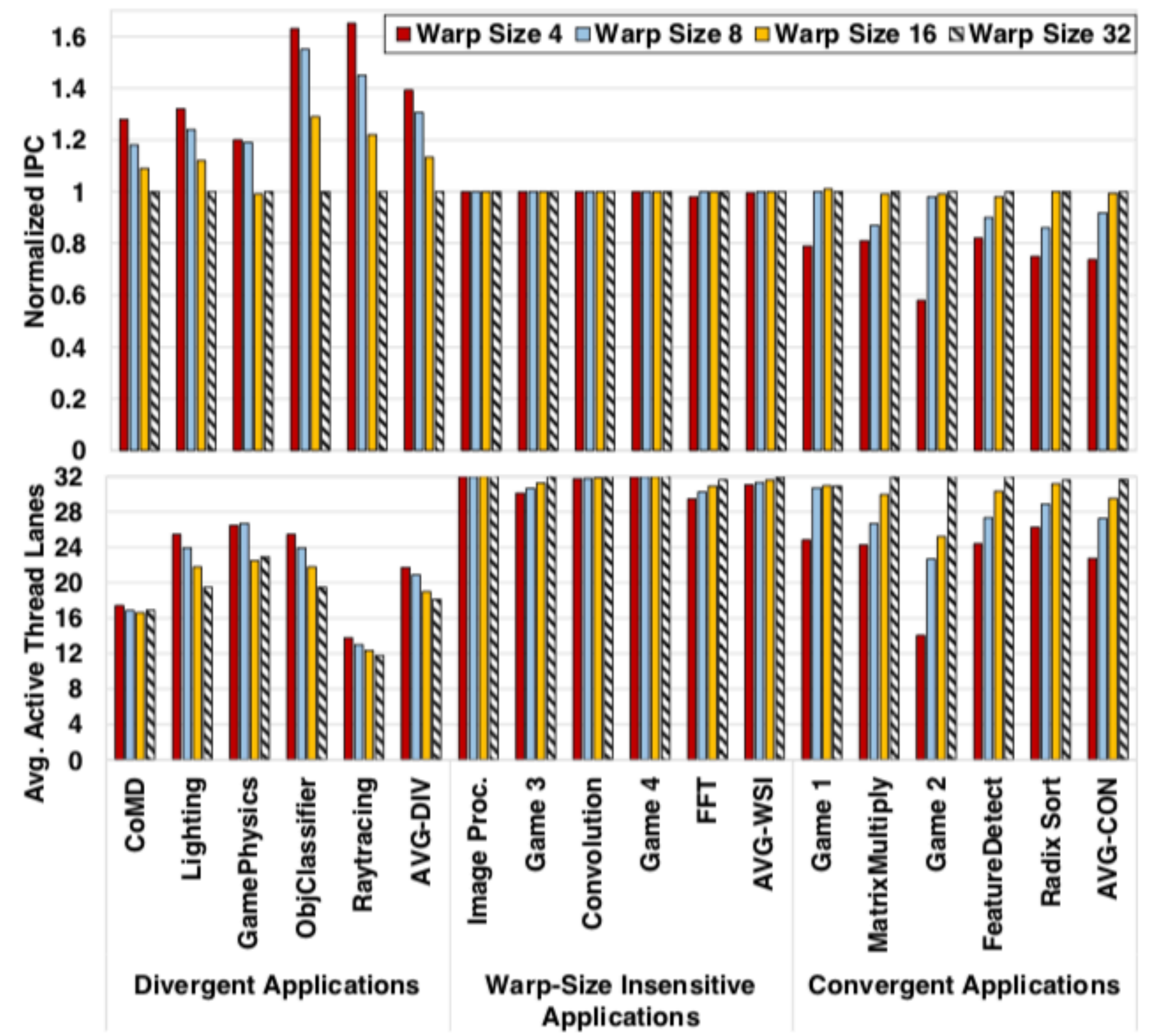


Figure 2: Normalized IPC (top) and the average number of active thread lanes on cycles when an instruction is issued (bottom). All configurations can issue 32 thread instructions per cycle.

Instruction Fetches

- Increase significantly for smaller warp sizes
- Puts more pressure on the L1 I-Cache
- So add L0 I-caches

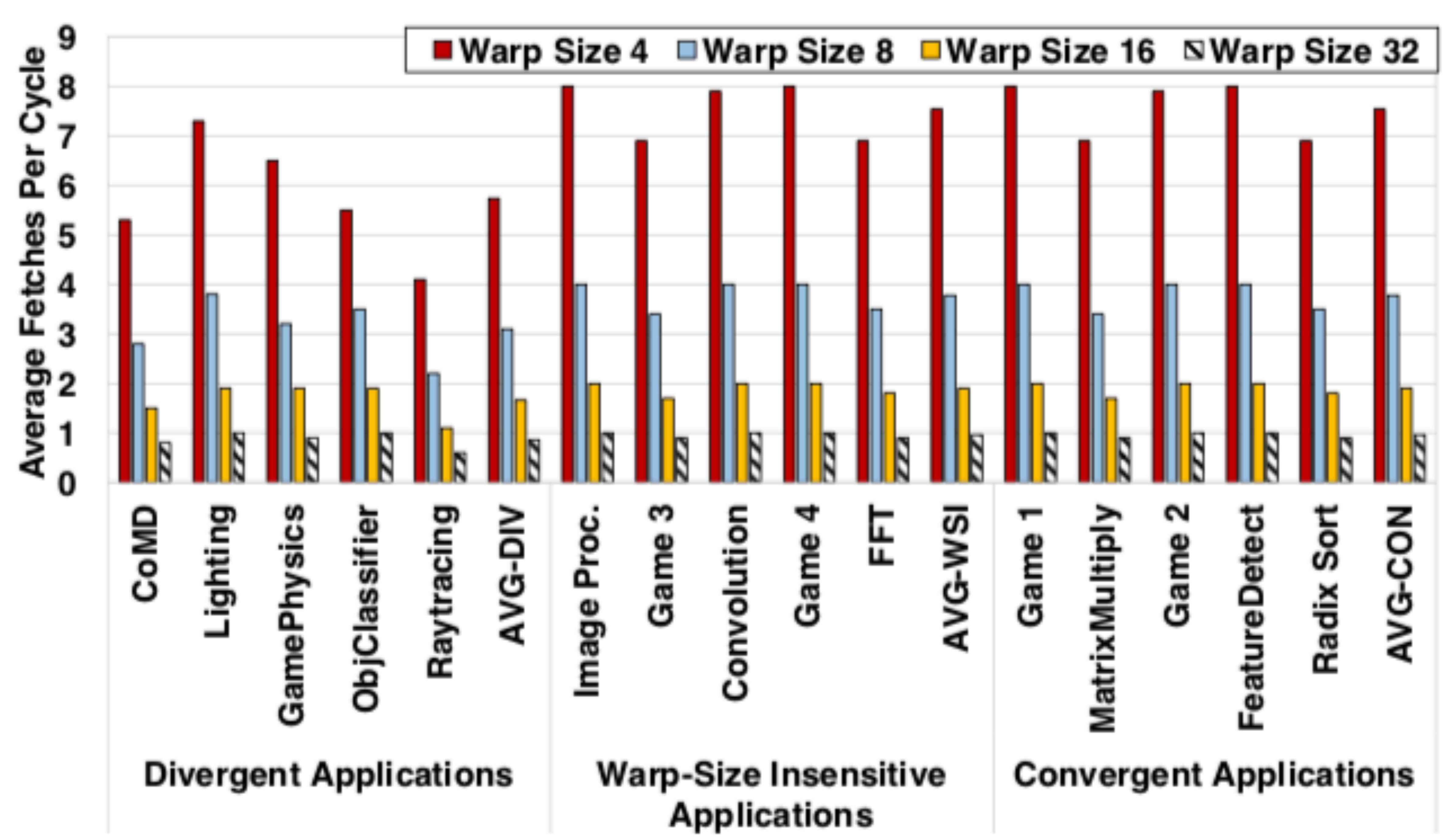


Figure 5: Average instructions fetched per cycle. Fetch bandwidth is scaled to match issue bandwidth for each warp size.

Variable Size Warps

- Use 4 threads for divergent apps, gang together for others

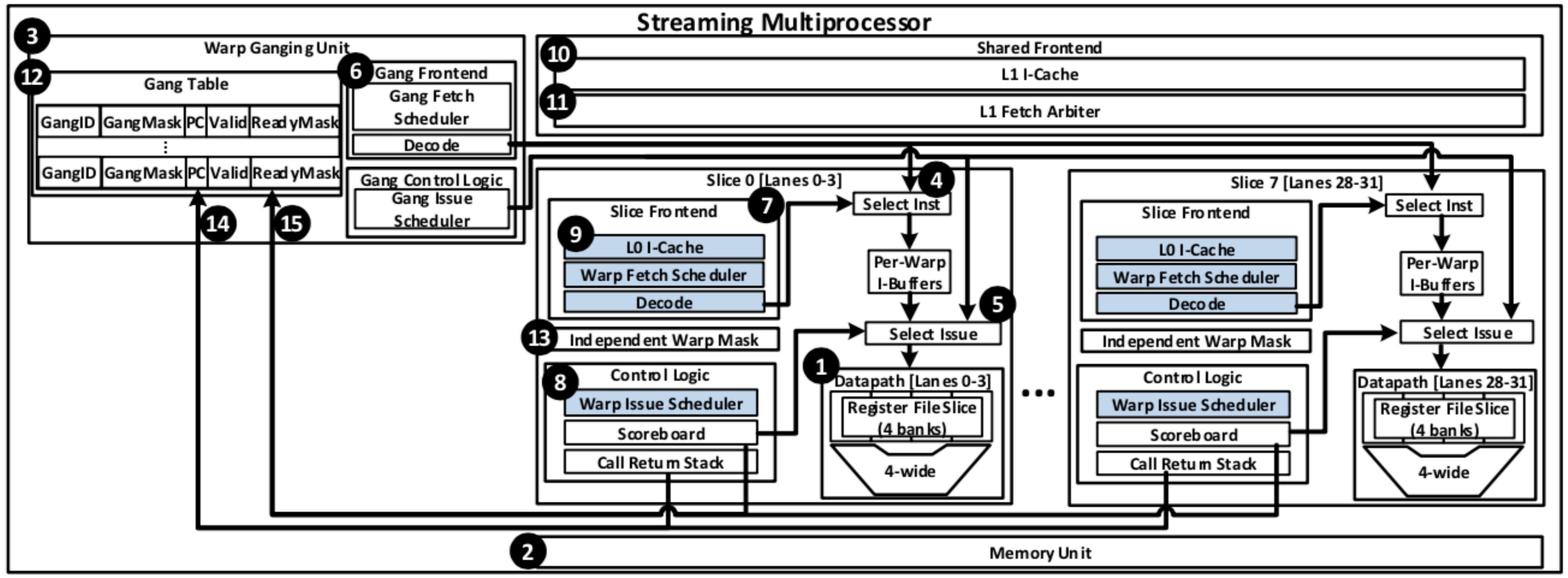


Figure 6: Variable Warp Sizing SM microarchitecture. Shaded units are disabled when operating in ganged mode to save energy.

Ganging

- Gangs are groups of 4 threads within a 32-thread warp
- A mask determines which slices are active
- Gangs can split at most 4 ways (groups of two 4-thread warps)
- Gangs can merge after splitting

Performance

- 32-warp, 4-warp
- Inelastic splitting (control divergence)
- Elastic splitting/merging
- Elastic with memory divergence

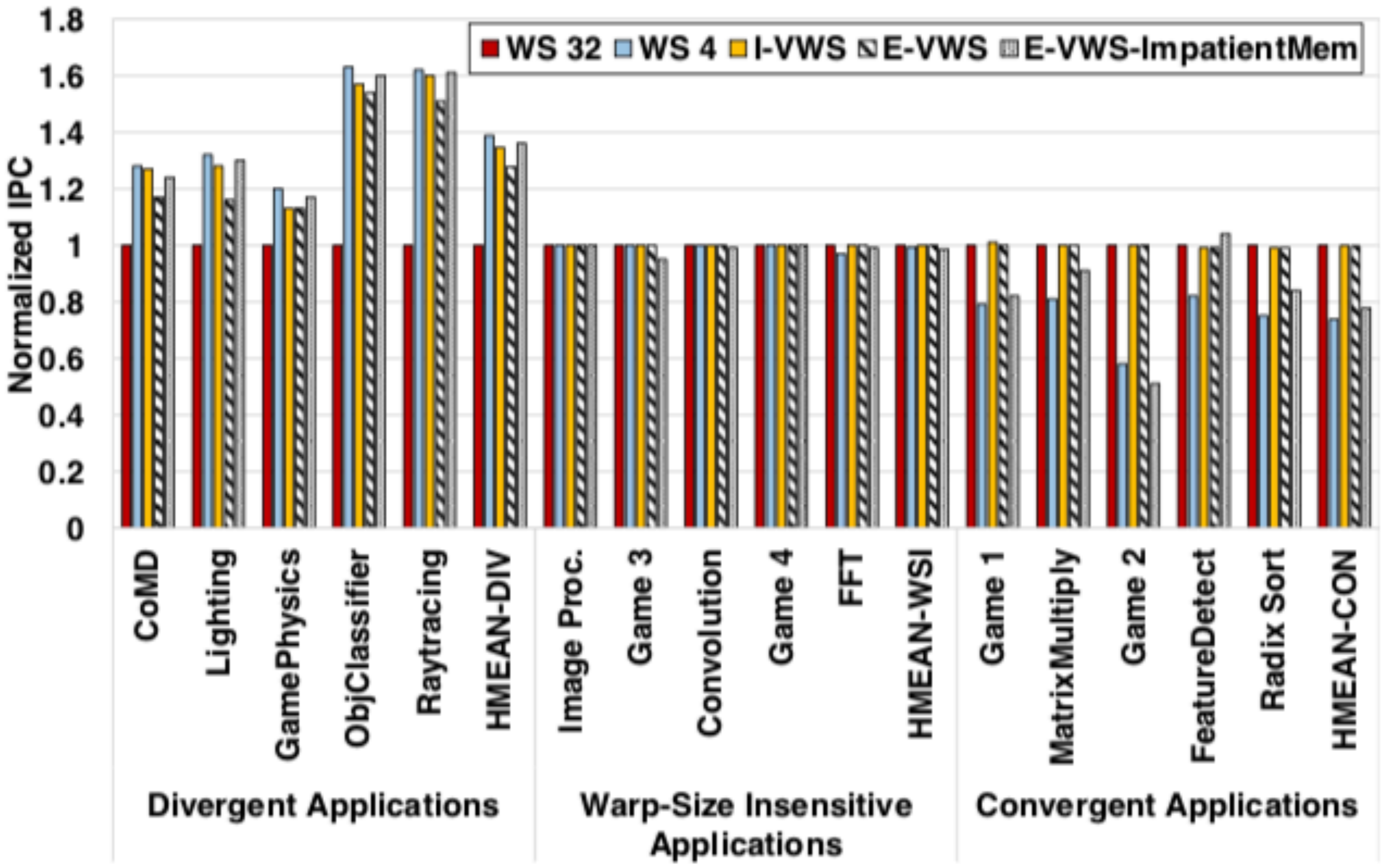


Figure 7: Performance (normalized to WS 32) of large warps, small warps, and different warp ganging techniques.

Performance

- Get most of divergent improvement, avoid most of convergent loss

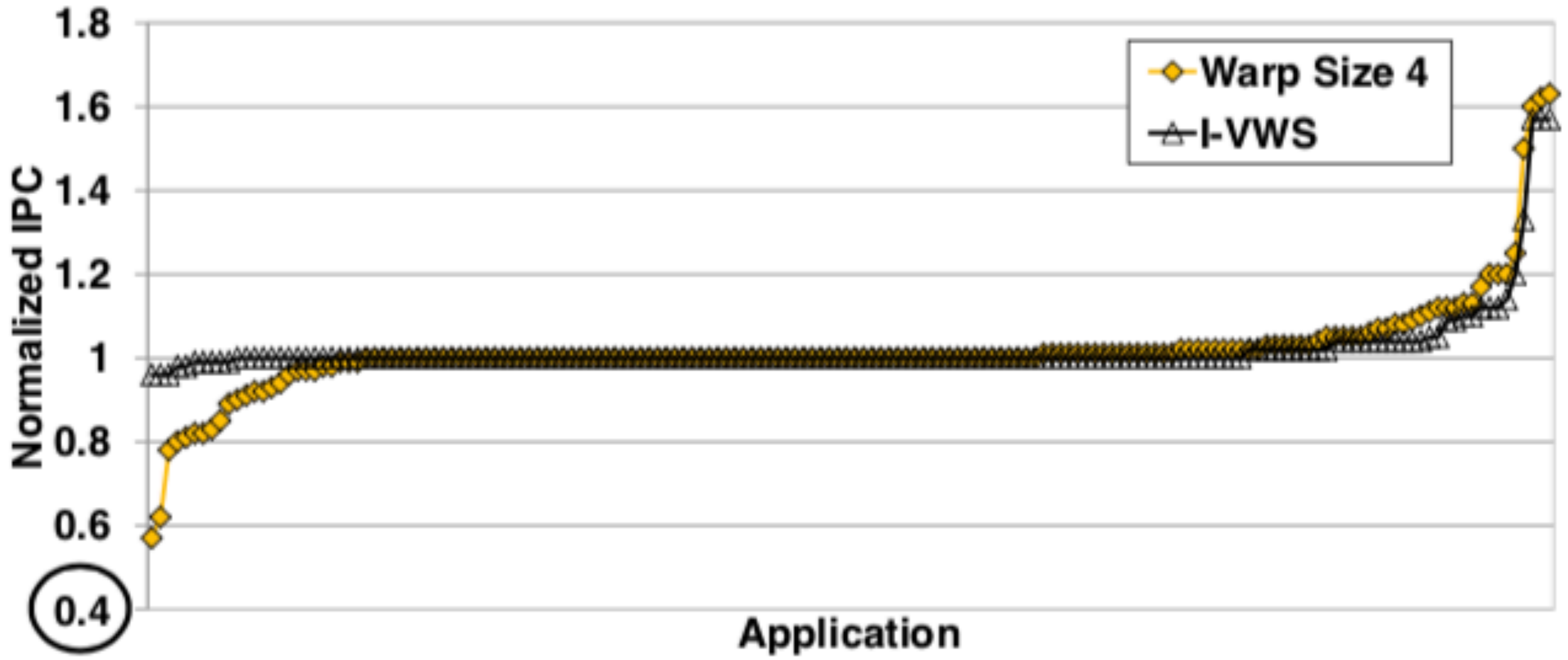


Figure 8: Performance (normalized to WS 32) of I-VWS and WS 4 on all the applications from in Figure 1a.

Explored many split/merge policies

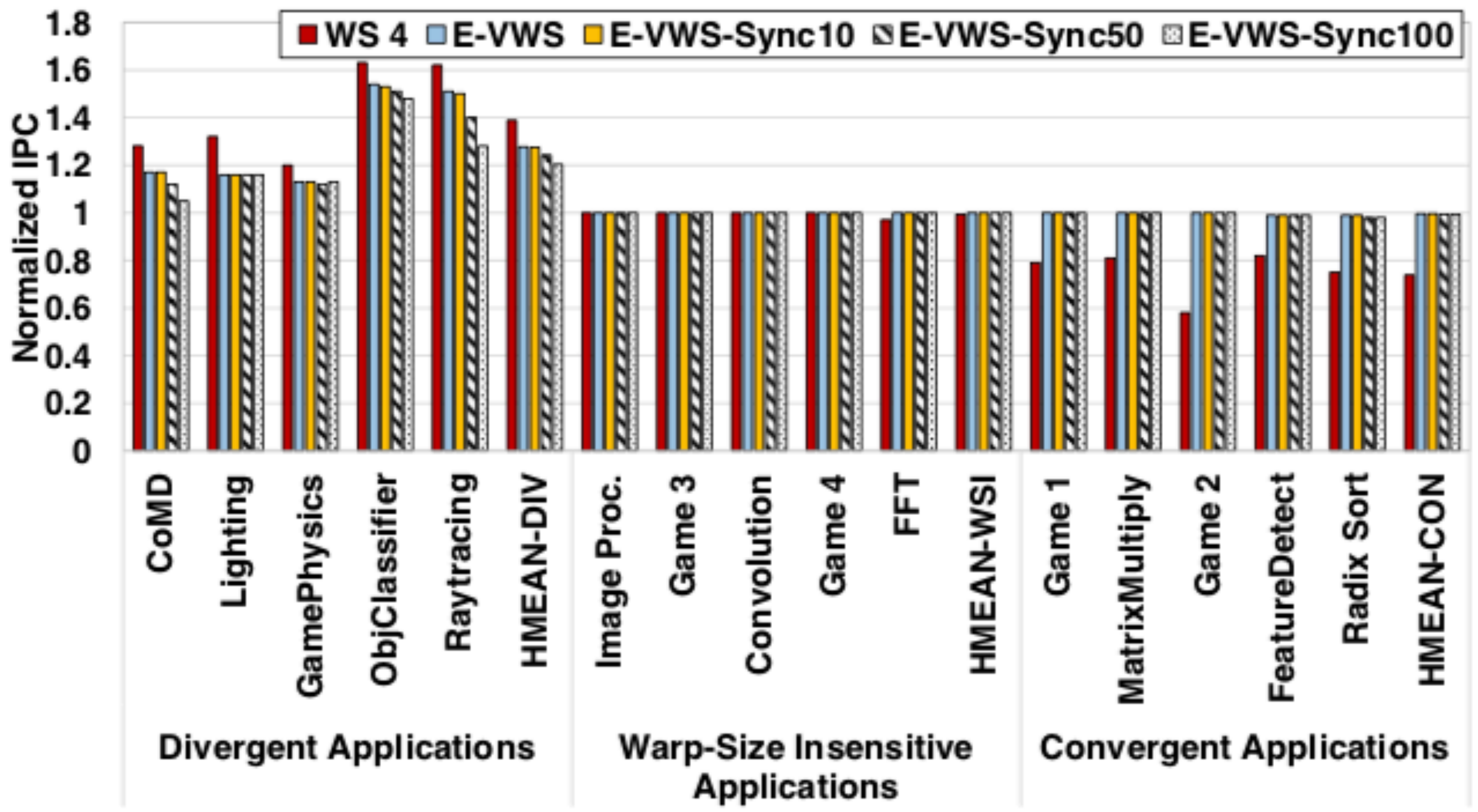


Figure 14: Performance (normalized to WS 32) of elastic gang reformation techniques.

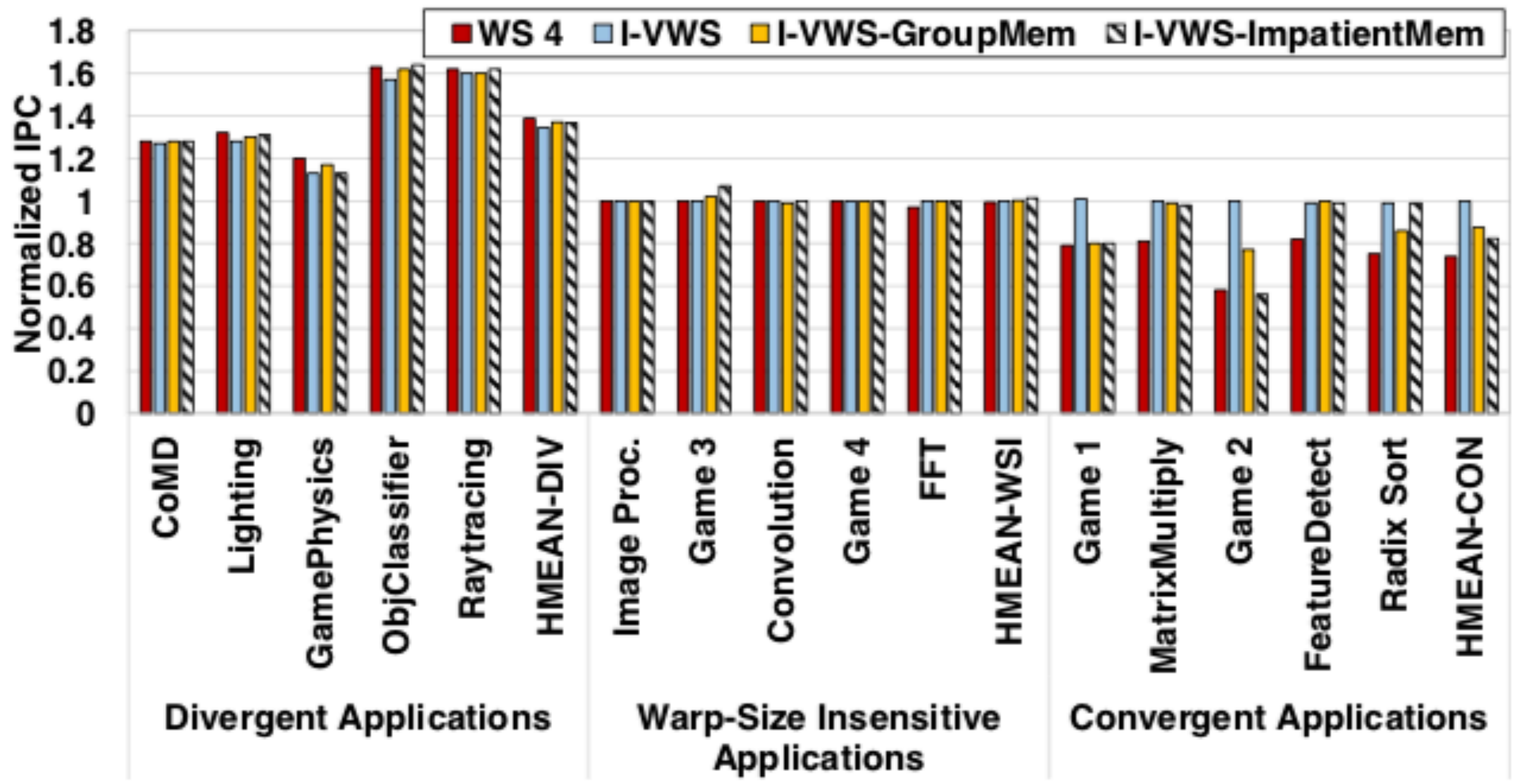


Figure 16: Performance (normalized to WS 32) of different gang splitting policies.

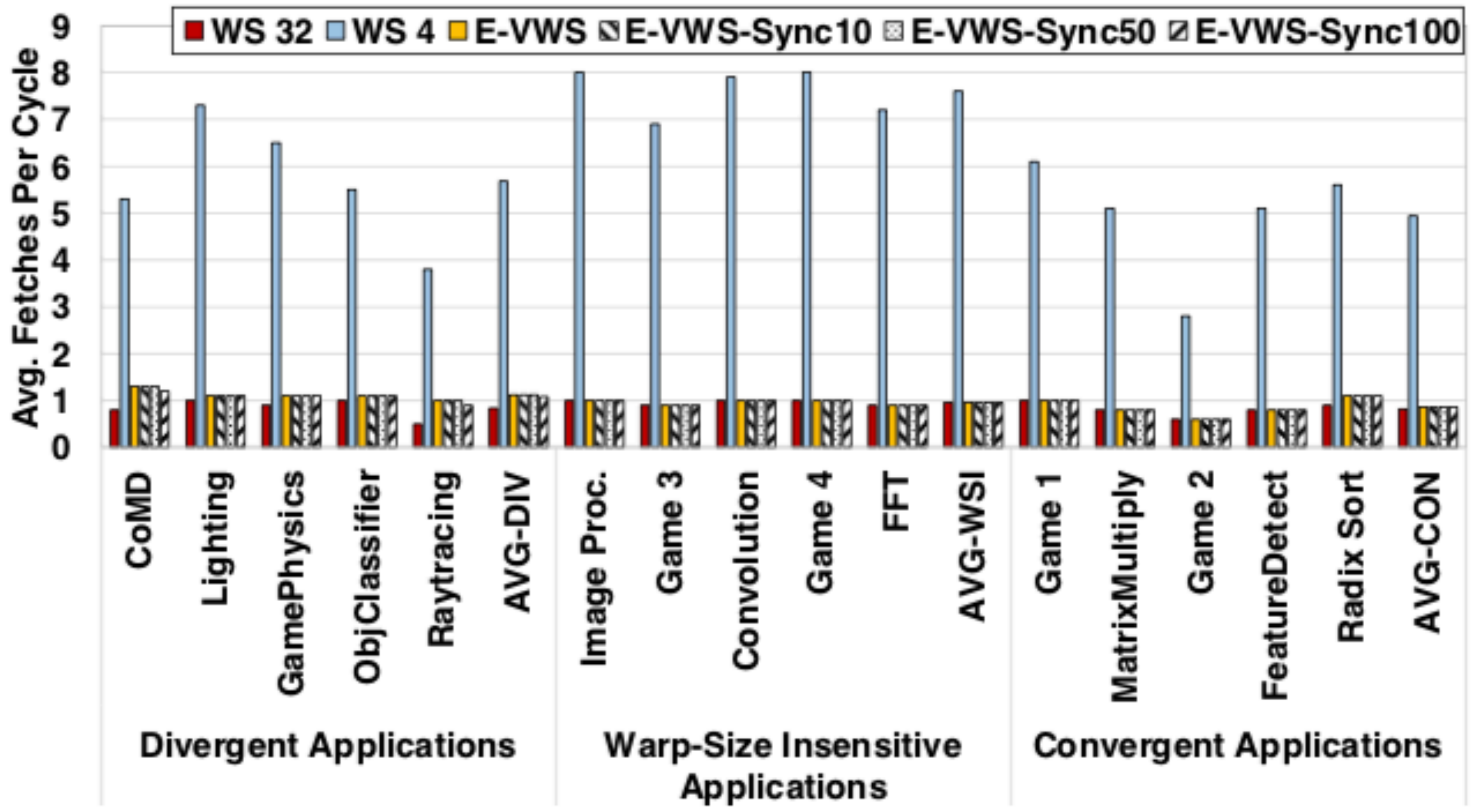


Figure 15: Average fetches per cycle with different gang reformation techniques.

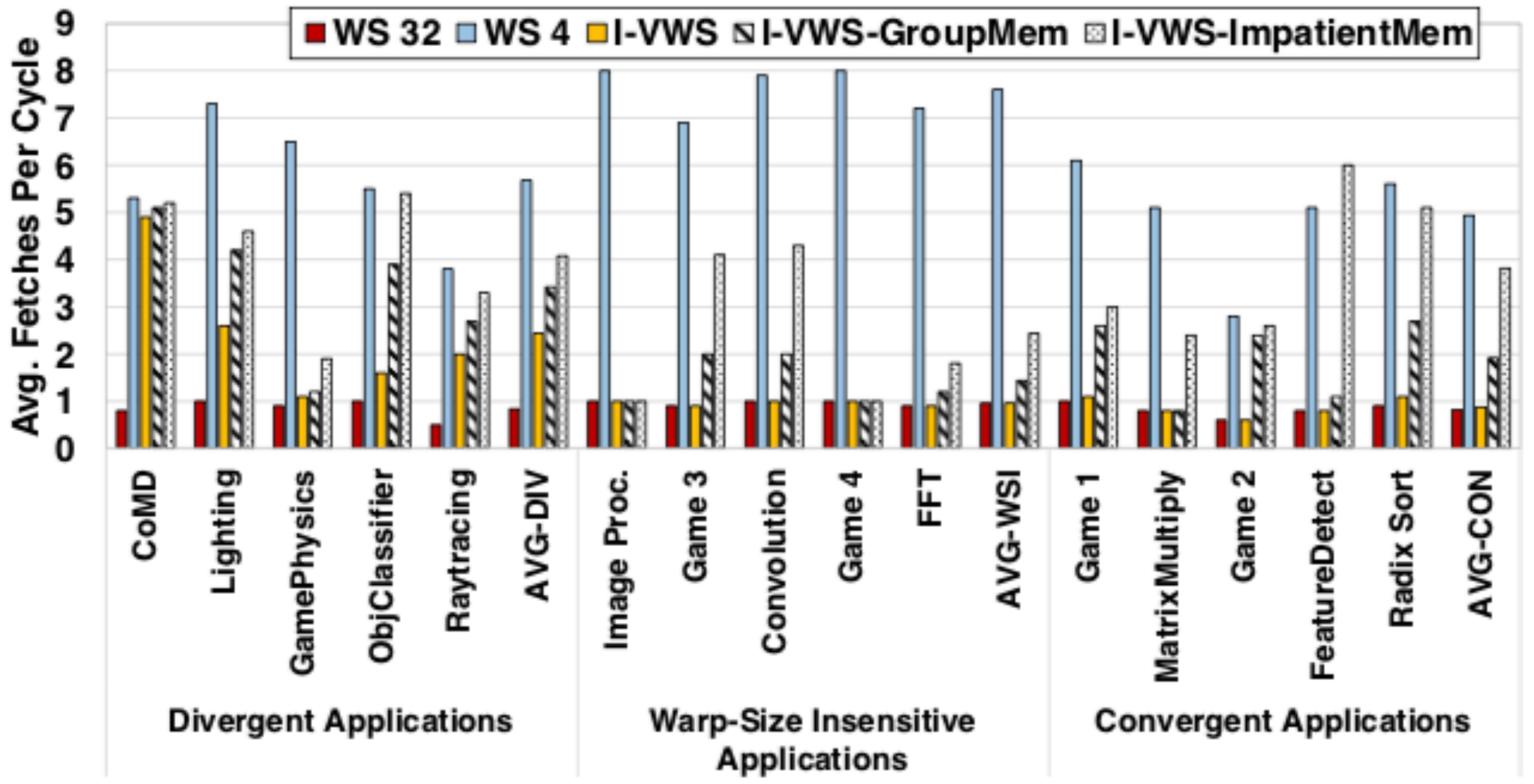


Figure 17: Average fetches per cycle using different gang splitting policies.

Related work

- Thread Block Compaction and Dynamic Warp Formation
- Best performance is on raytracing (note that NVIDIA added raytrace units in Turing)

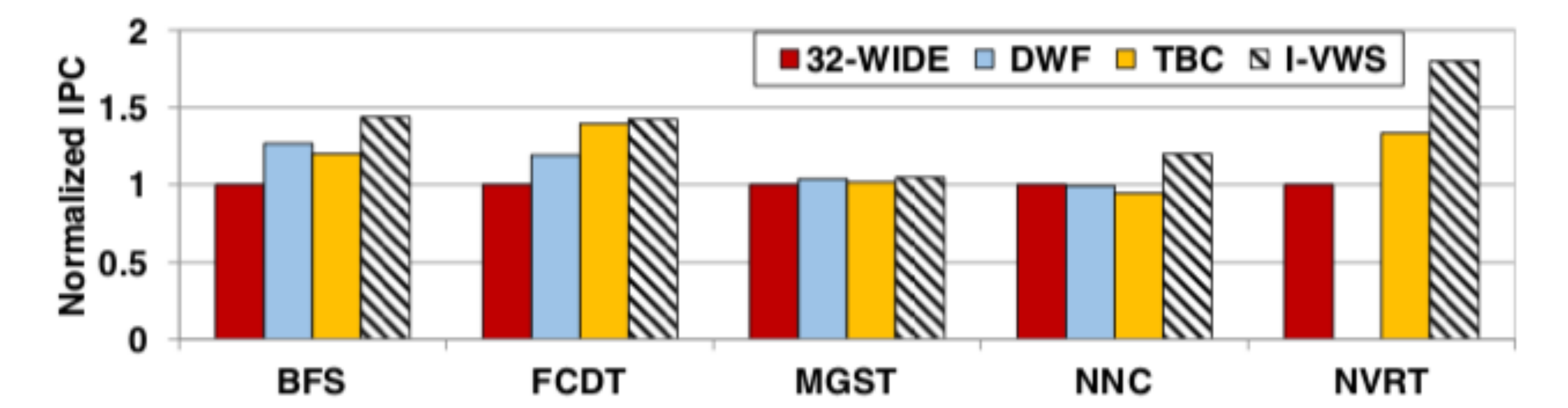


Figure 19: Performance (normalized to the 32-wide warp baseline) using the released TBC infrastructure [10].

Discussion

Joupi ISCA17

In-Datacenter Performance Analysis of a Tensor Processing Unit
(78 Authors)

Motivation

- Voice recognition for search engine input
- Inference only, limited set of NNs, models reduced to integer form
- Low latency
- Low power
- Build in a hurry and keep cost low (chip small)

Co-Processor

- Driven by CPU — does not have its own instruction fetch
- Large memory and large number of 8-bit integer MAC units
- Plug into PCI-e bus, like a SATA hard drive

Overview

4 Mb in Accumulators

24 Mb in Unified Buffer

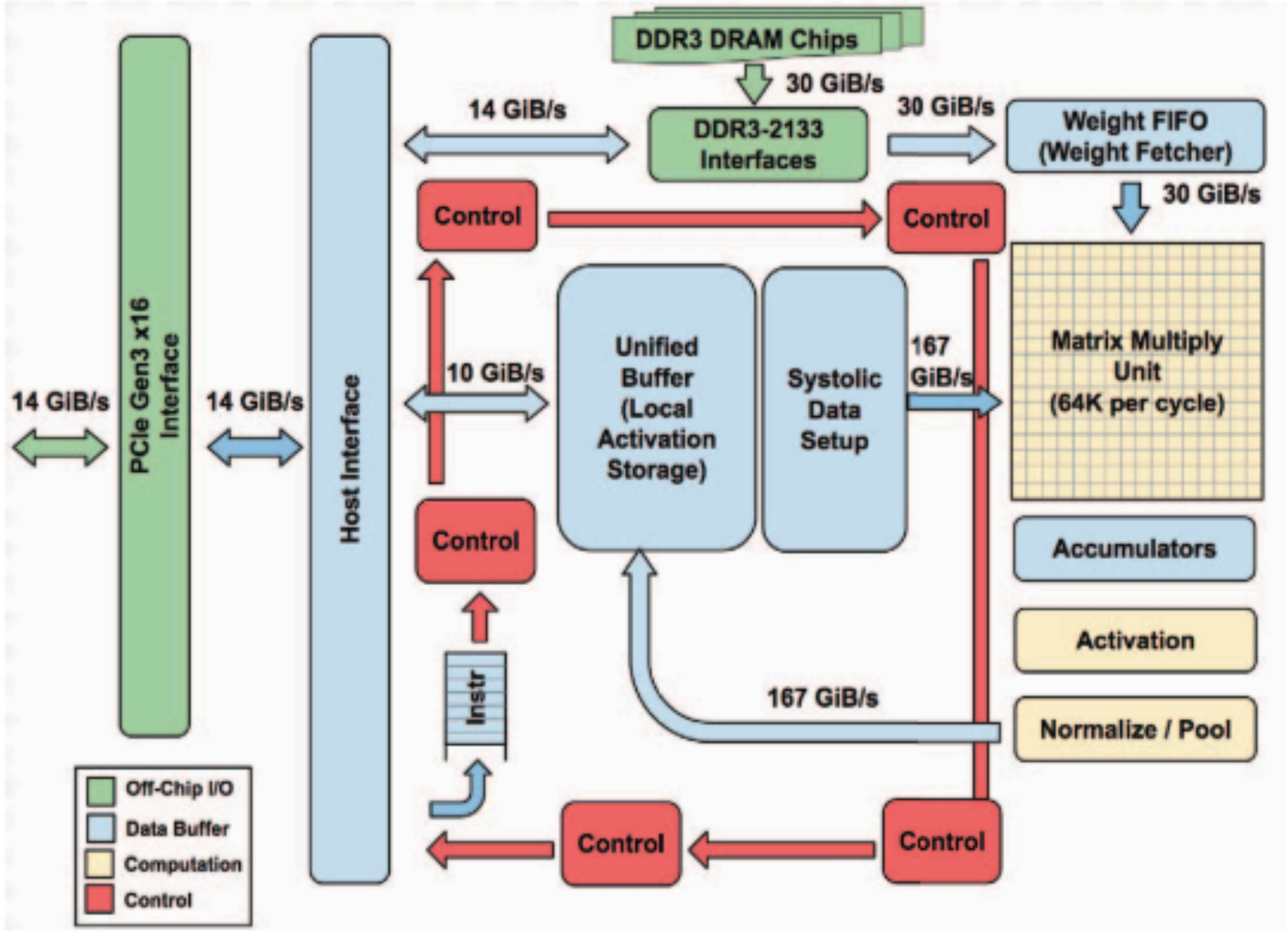


Figure 1. TPU Block Diagram. The main computation is the yellow Matrix Multiply unit. Its inputs are the blue Weight FIFO and the blue Unified Buffer and its output is the blue Accumulators. The yellow Activation Unit performs the nonlinear functions on the Accumulators, which go to the Unified Buffer.

Floorplan

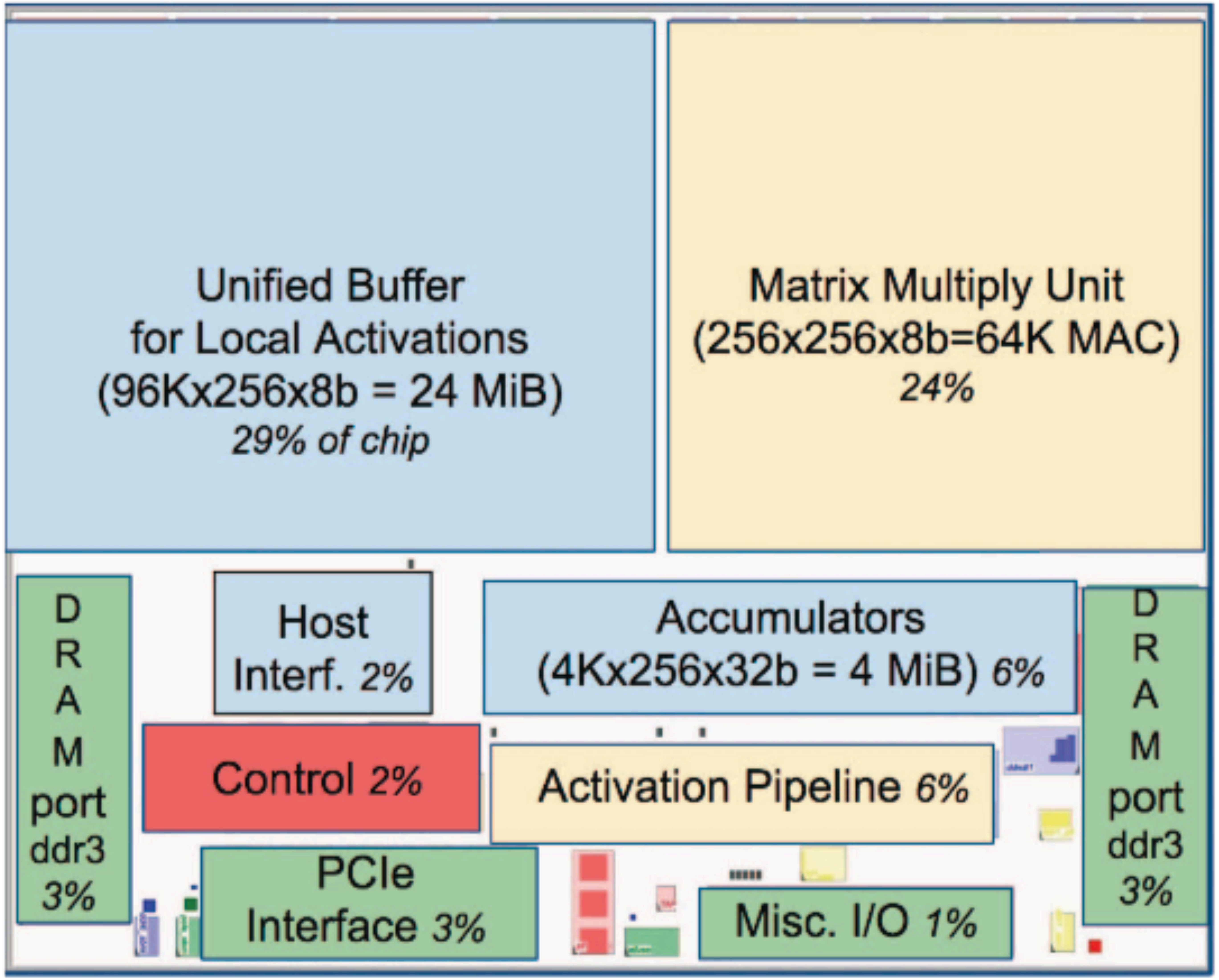


Figure 2. Floorplan of TPU die. The shading follows Figure 1. The light (blue) datapath is 67%, the medium (green) I/O is 10%, and the dark (red) control is just 2% of the die. Control is much larger (and much harder to design) in a CPU or GPU.

CISC Instructions

- PCIe latency slows issue to 10 to 20 cycles
- Instructions are large operations and include repeat field (about a dozen)
- Main 5 are: Read_Host_Memory, Read_Weights, MatrixMultiply/Convolve, Activate, Write_Host_Memory

Comparison Systems

Model	*Die*										*Benchmarked Servers*				
	mm^2	*nm*	*MHz*	*TDP*	*Measured*		*TOPS/s*		*GB/s*	*On-Chip Memory*	*Dies*	*DRAM Size*	*TDP*	*Measured*	
					Idle	*Busy*	8b	FP						*Idle*	*Busy*
Haswell E5-2699 v3	662	22	2300	145W	41W	145W	2.6	1.3	51	51 MiB	2	256 GiB	504W	159W	455W
NVIDIA K80 (2 dies/card)	561	28	560	150W	25W	98W	--	2.8	160	8 MiB	8	256 GiB (host) + 12 GiB x 8	1838W	357W	991W
TPU	<331*	28	700	75W	28W	40W	92	--	34	28 MiB	4	256 GiB (host) + 8 GiB x 4	861W	290W	384W

Table 2. Benchmarked servers use Haswell CPUs, K80 GPUs, and TPUs. Haswell has 18 cores, and the K80 has 13 SMX processors. Figure 10 shows measured power. The low-power TPU allows for better rack-level density than the high-power GPU. The 8 GiB DRAM per TPU is Weight Memory. GPU Boost mode is not used (see Section 8). SECDED and no Boost mode reduce K80 bandwidth from its advertised 240 to 160 GB/s. No Boost mode and single die vs. dual die performance reduces advertised K80 peak TOPS/s from 8.7 to 2.8. (*The TPU die is less than half the Haswell die size.)

Performance

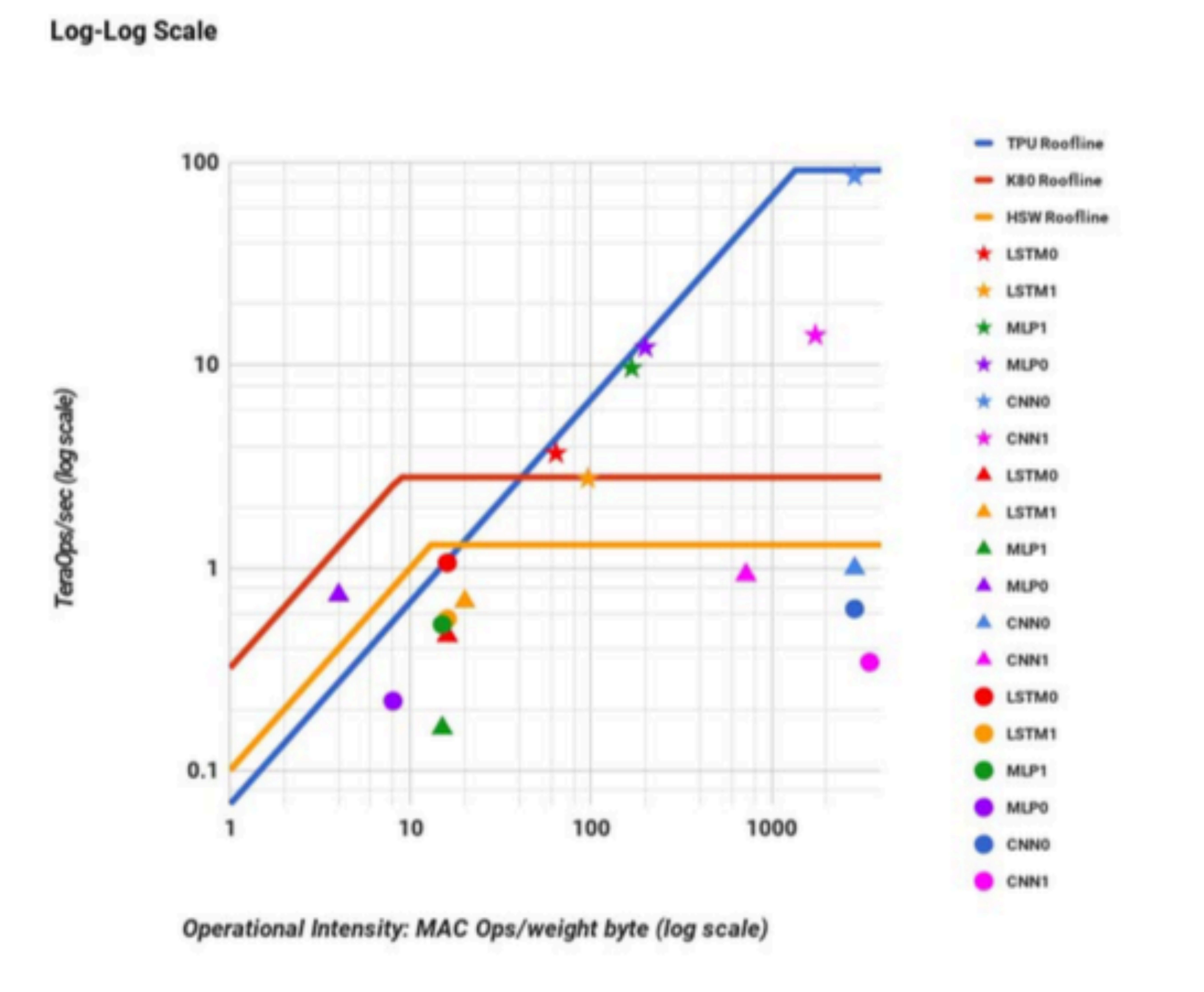


Figure 8. Figures 5-7 combined into a log-log graph. Stars are for the TPU, triangles are for the K80, and circles are for Haswell. All TPU stars are at or above the other two rooflines.

Energy

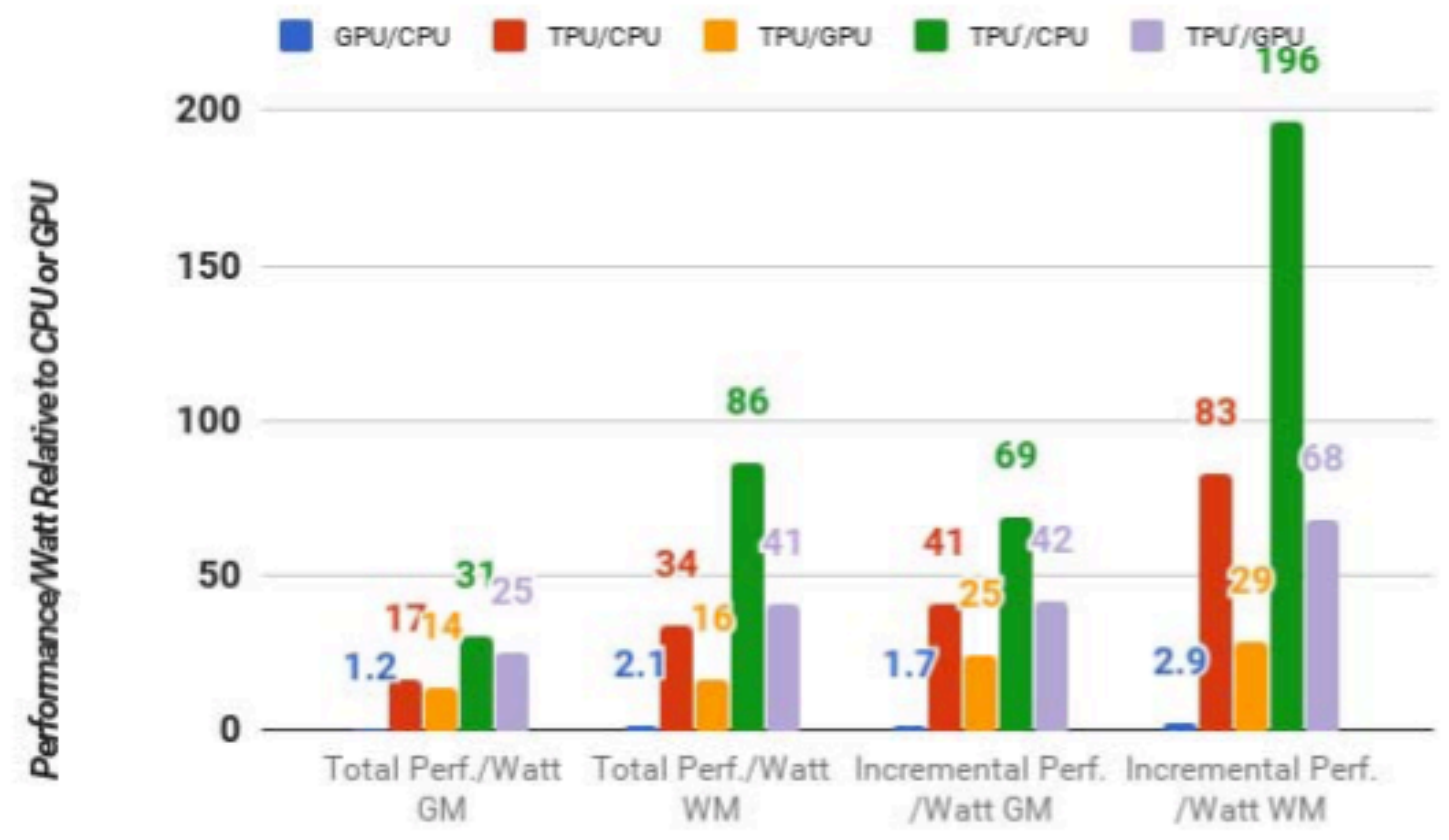


Figure 9. Relative performance/Watt (TDP) of GPU server (blue) and TPU server (red) to CPU server, and TPU server to GPU server (orange). TPU' is an improved TPU that uses GDDR5 memory (see Section 7). The green bar shows its ratio to the CPU server, and the lavender bar shows its relation to the GPU server. Total includes host server power, but incremental doesn't. GM and WM are the geometric and weighted means.

Discussion