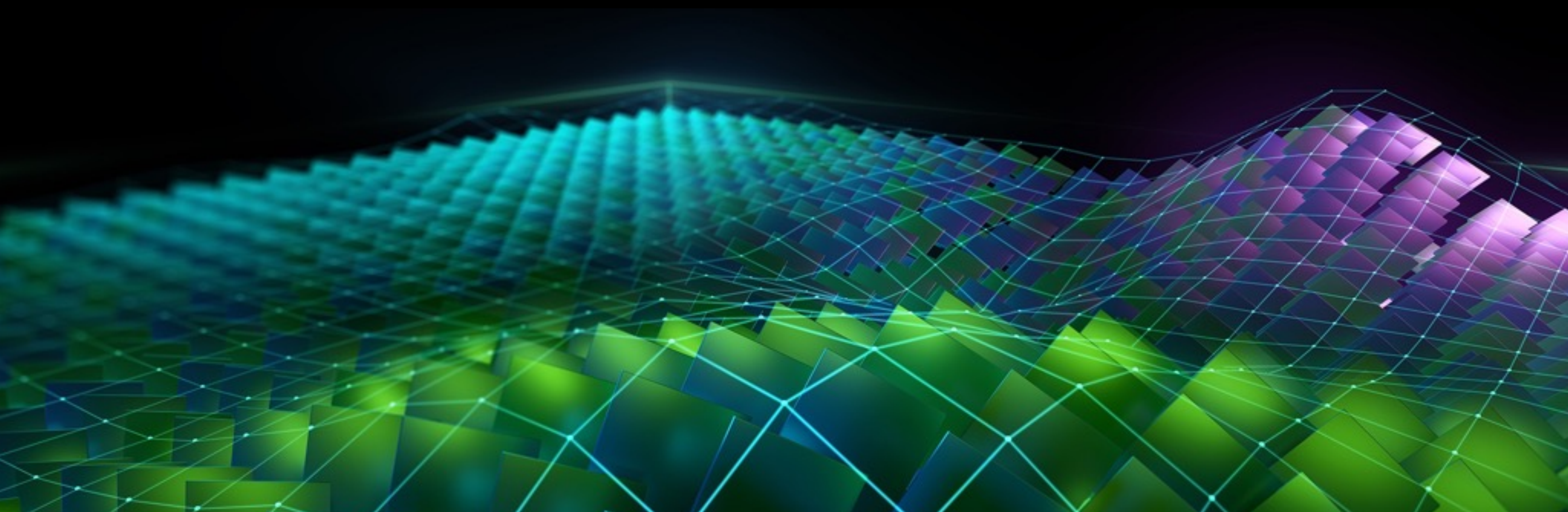
Exploiting Concurrency in GPUs

Cristiana Bentes

Professor of Systems Engineering and Computer Science
State University of Rio de Janeiro
Brazil



Agenda

- HPC Accelerated Era
- GPUs
- Concurrent Kernel Execution
  - improve concurrency opportunities
  - order of submission
  - kernel characterization
  - kernel interference
  - preemption

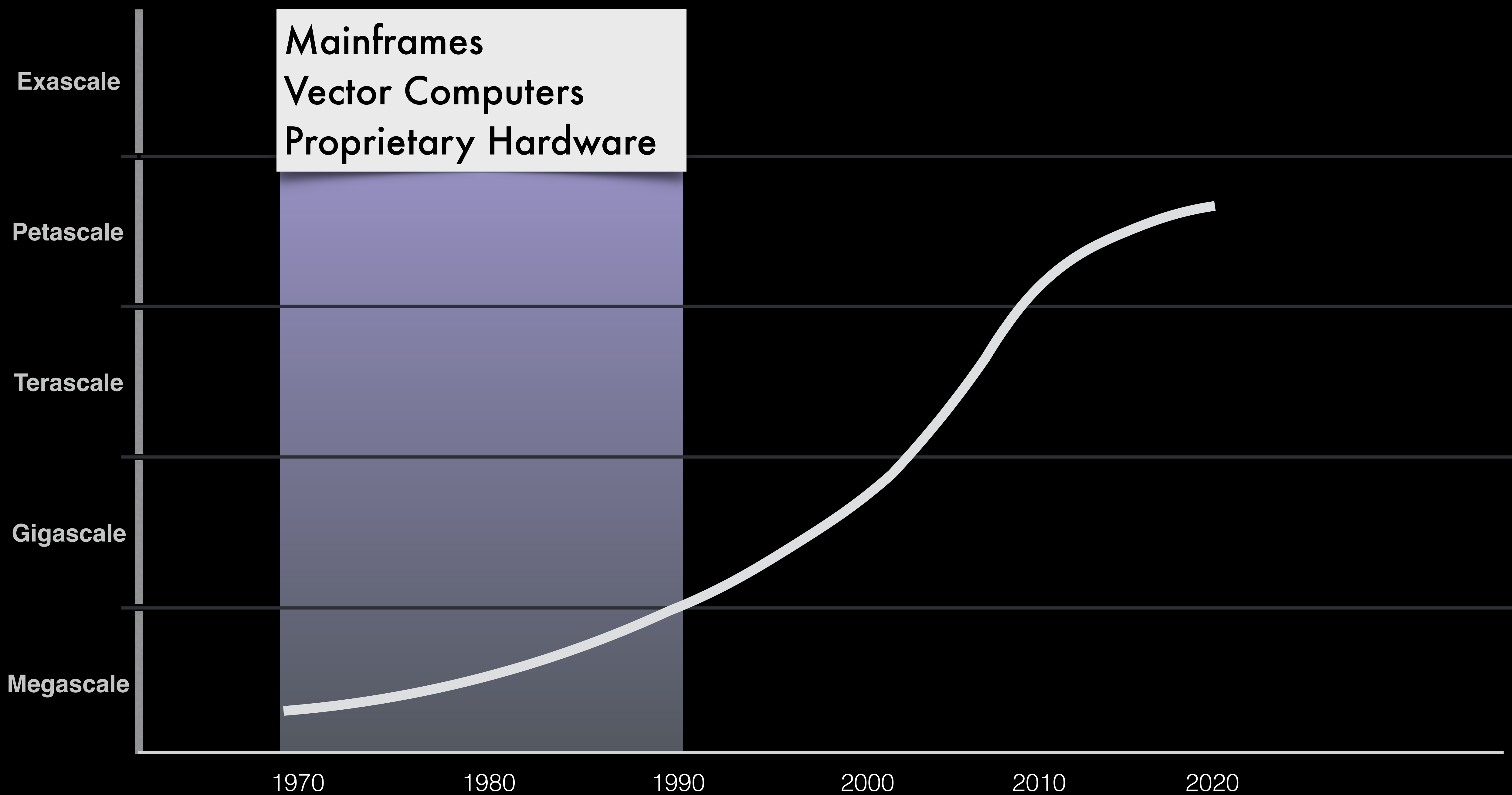
Would you have predicted that about 10 years ago many of our top HPC systems would be GPU

The Coming Age of Extreme Heterogeneity - Jeffrey S. Vetter

Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, **Fujitsu** RIKEN Center for Computational Science Japan	7,630,848	442,010.0	537,212.0	29,899
2	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, **IBM** DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148,600.0	200,794.9	10,096
3	**Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, **IBM / NVIDIA / Mellanox** DOE/NNSA/LLNL United States	1,572,480	94,640.0	125,712.0	7,438
4	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway, **NRCPC** National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
5	**Selene** - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband, **Nvidia** NVIDIA Corporation United States	555,520	63,460.0	79,215.0	2,646
6	**Tianhe-2A** - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000, **NUDT** National Super Computer Center in Guangzhou China	4,981,760	61,444.5	100,678.7	18,482
7	**JUWELS Booster Module** - Bull Sequana XH2000 , AMD EPYC 7402 24C 2.8GHz, NVIDIA A100, Mellanox HDR InfiniBand/ParTec ParaStation ClusterSuite, **Atos** Forschungszentrum Juelich (FZJ) Germany	449,280	44,120.0	70,980.0	1,764
8	**HPC5** - PowerEdge C4140, Xeon Gold 6252 24C 2.1GHz, NVIDIA Tesla V100, Mellanox HDR Infiniband, **Dell EMC** Eni S.p.A. Italy	669,760	35,450.0	51,720.8	2,252
9	**Frontera** - Dell C6420, Xeon Platinum 8280 28C 2.7GHz, Mellanox InfiniBand HDR, **Dell EMC** Texas Advanced Computing Center/Univ. of Texas United States	448,448	23,516.4	38,745.9	
10	**Dammam-7** - Cray CS-Storm, Xeon Gold 6248 20C 2.5GHz, NVIDIA Tesla V100 SXM2, InfiniBand HDR 100, **HPE** Saudi Aramco Saudi Arabia	672,520	22,400.0	55,423.6	

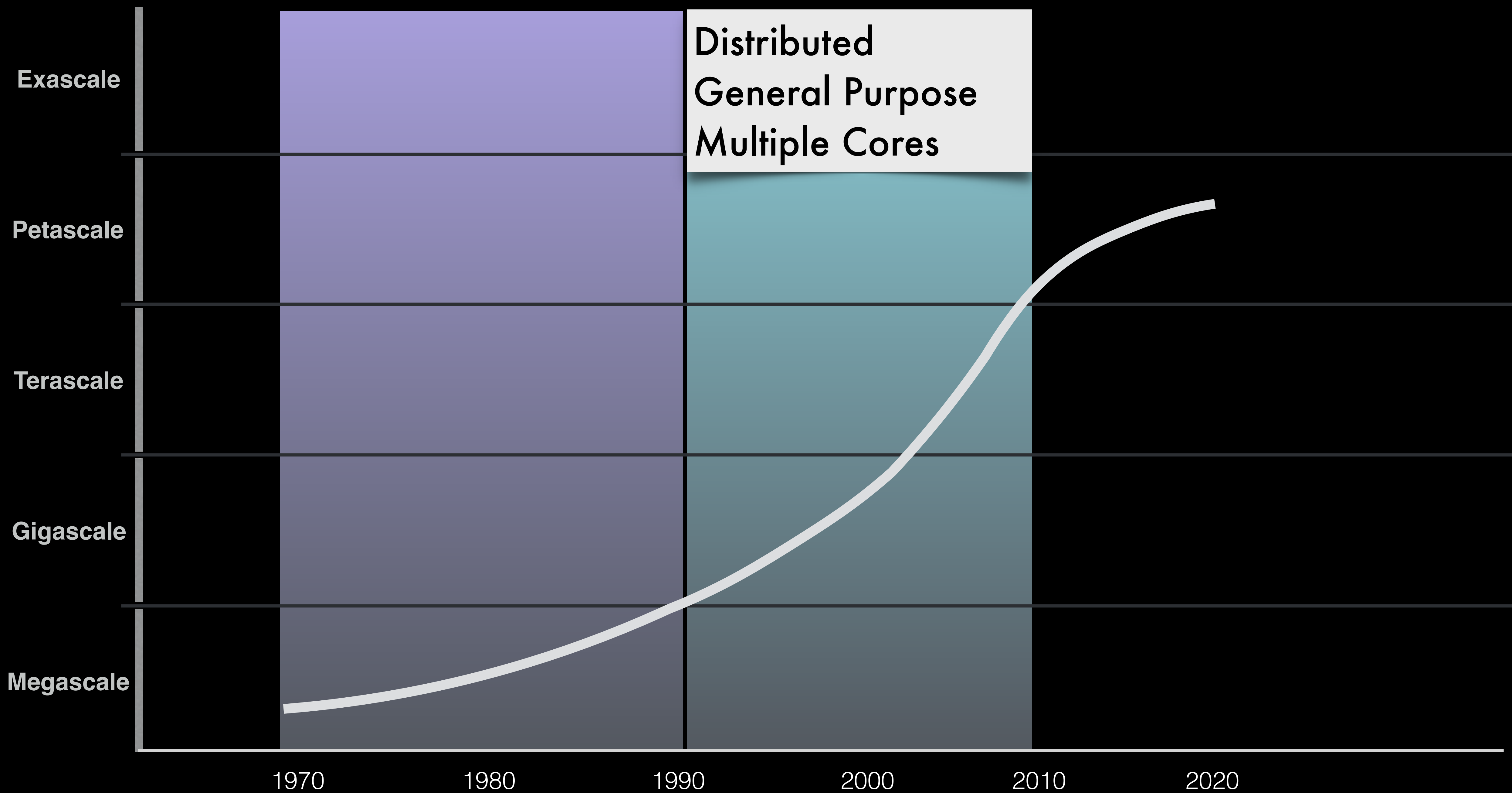
How did we get here?

HPC Jurrasic Era

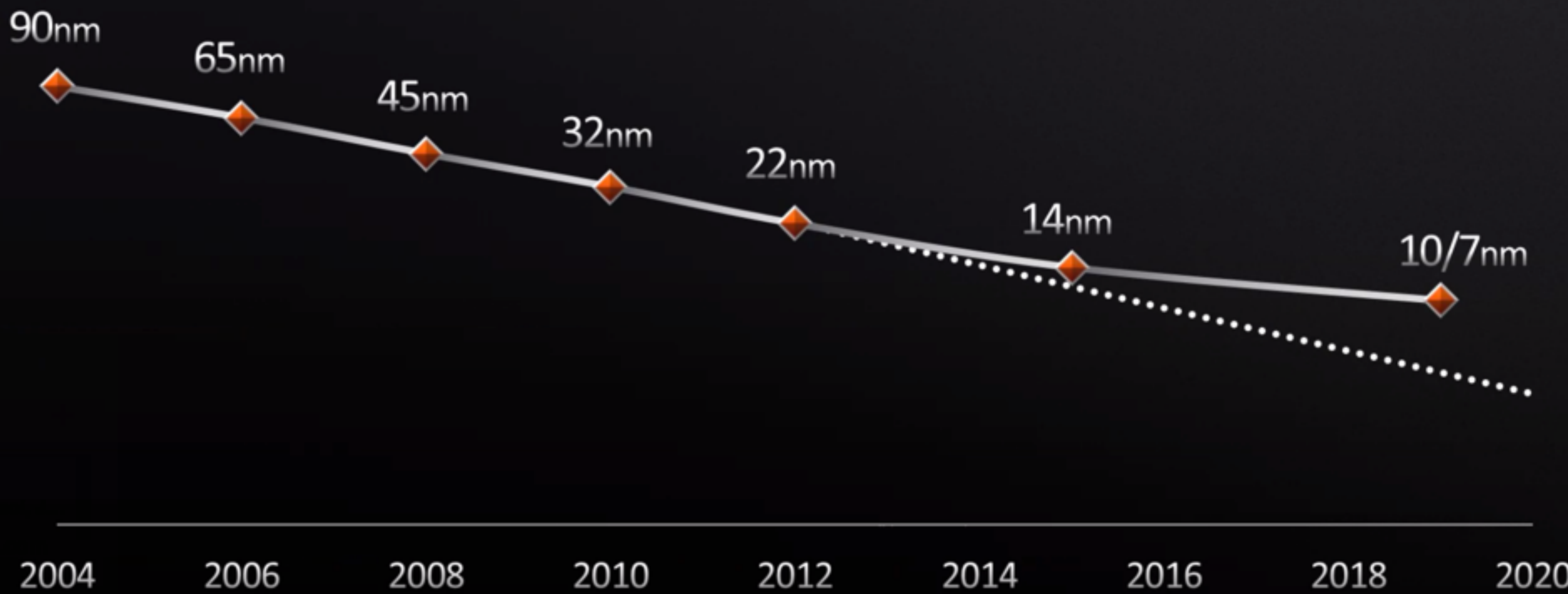


“Attack of the Killer Micros”

HPC Distributed Era

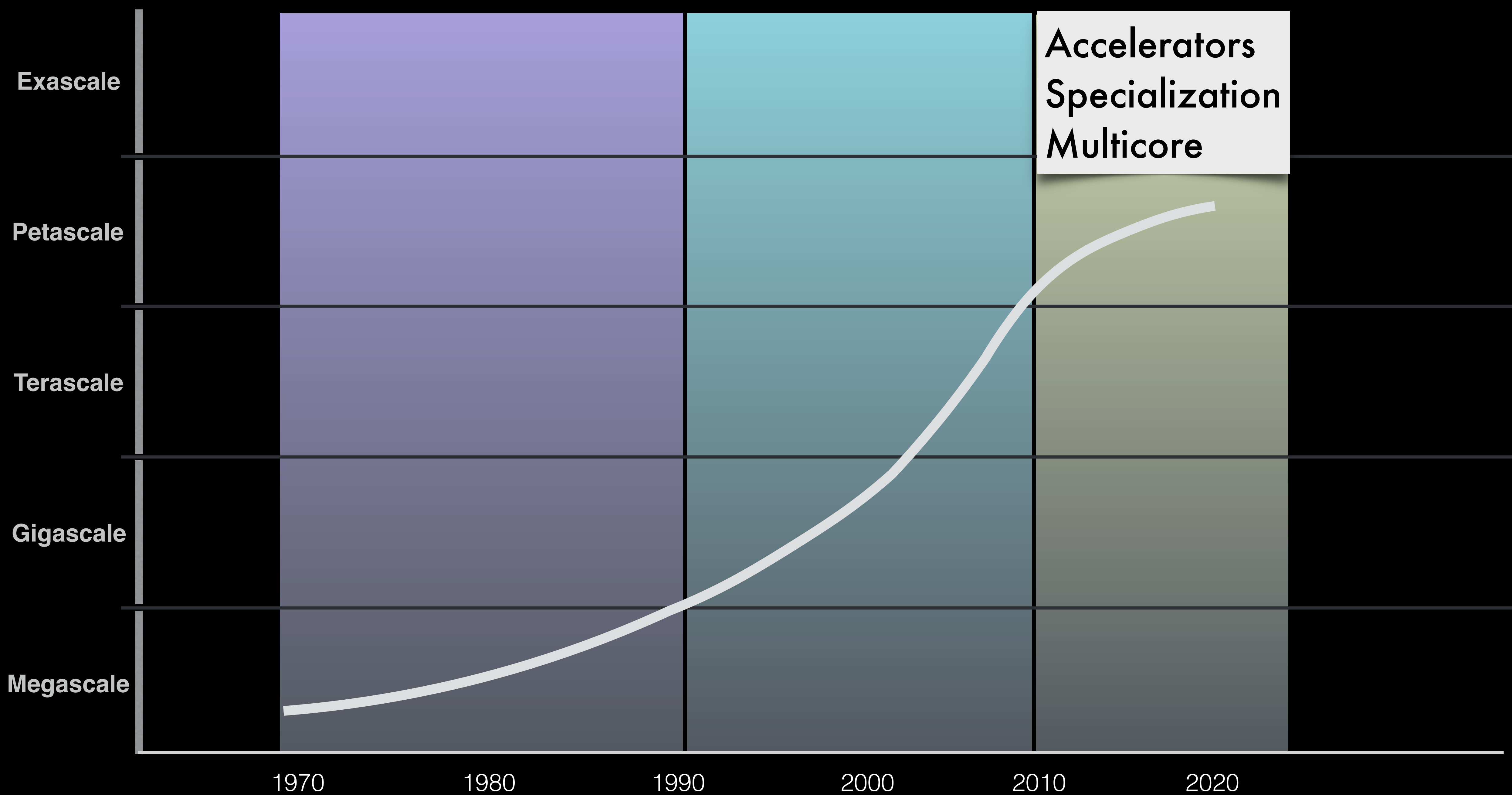


MOORE'S LAW KEEPS SLOWING



Challenges and Opportunities for Extreme-Scale Computing, Michael Schulte

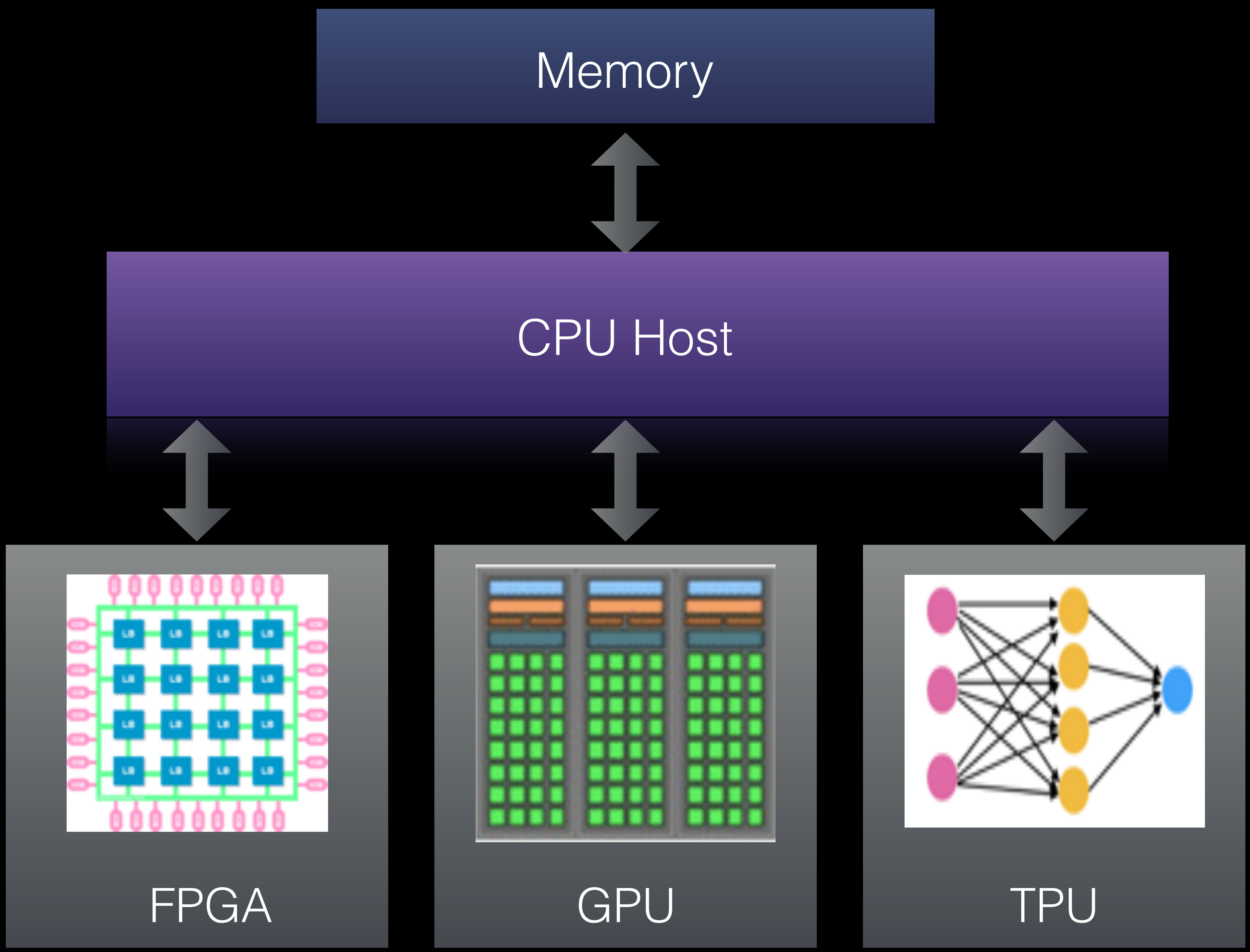
HPC Accelerated Era



HPC Acceletared Era

Specialization

Heterogeneous Systems



- HPC systems are becoming more heterogeneous
- Improve performance and power efficiency

Industry is investing...

HOME | NEWS | LISTS | STATISTICS | RESOURCES | ABOUT | MEDIA KIT | GREEN500

Home / News / Microsoft Goes All in for FPGAs to Build Out AI Cloud

Microsoft Goes All in for FPGAs to Build Out AI Cloud

Michael Feldman | September 27, 2016 08:42 CEST

E-mail | Tweet | Like | G+1 | Share

Software giant bets the (server) farm on reconfigurable computing

Microsoft has revealed that Altera FPGAs have been installed across every Azure cloud server calling "the world's first AI supercomputer." The deployment spans 15 countries and represen more than one exa-op. The announcement was made by Microsoft CEO Satya Nadella and eng opening keynote at the Ignite Conference in Atlanta.

The FPGA build-out was the culmination of more than five years of work at Microsoft to find a learning and other throughput-demanding applications and services in its Azure cloud. The ef when the company launched Project Catapult, the R&D initiative to design an acceleration fab applications. The rationale was that CPU evolution, a la Moore's Law, was woefully inadequate demands of these new hyperscale applications. Just as in traditional high performance compu keeping up with demand.

Doug Burger with Microsoft-designed FPGA card

Intel Xe Graphics: Release Date, Specs, Everything We Know

By Jarred Walton 24 days ago

Intel Xe Graphics is expected to join the dedicated graphics card ...ut can it possibly compete with AMD and

CADE METZ BUSINESS 05.18.16 03:57 PM

GOOGLE BUILT ITS VERY OWN CHIPS TO POWER ITS AI BOTS

GOOGLE

GOOGLE HAS DESIGNED its own computer chip for driving deep neural networks, an AI technology that is reinventing the way Internet services operate.

This morning at Google I/O, the centerpiece of the company's year, CEO Sundar Pichai said that Google has designed an ASIC, or application-specific integrated circuit,

TECHNOLOGY | News Wire | Nov 27, 2018

Amazon unveils its own server chip, challenging Intel on price

Ian King, Bloomberg News

The Amazon.com logo is displayed outside the company's fulfillment center in Kenosha, Wisconsin, U.S. , Photographer: Jim Young/Bloomberg

Amazon.com Inc. (AMZN.O) has taken a big step toward reducing reliance on Intel Corp. (INTC.O) for a critical component of its cloud-computing service.

The largest cloud company unveiled its own server processors late Monday and said the Graviton chips will support new versions of its main EC2 cloud-computing service. Until now, Amazon -- and other big cloud

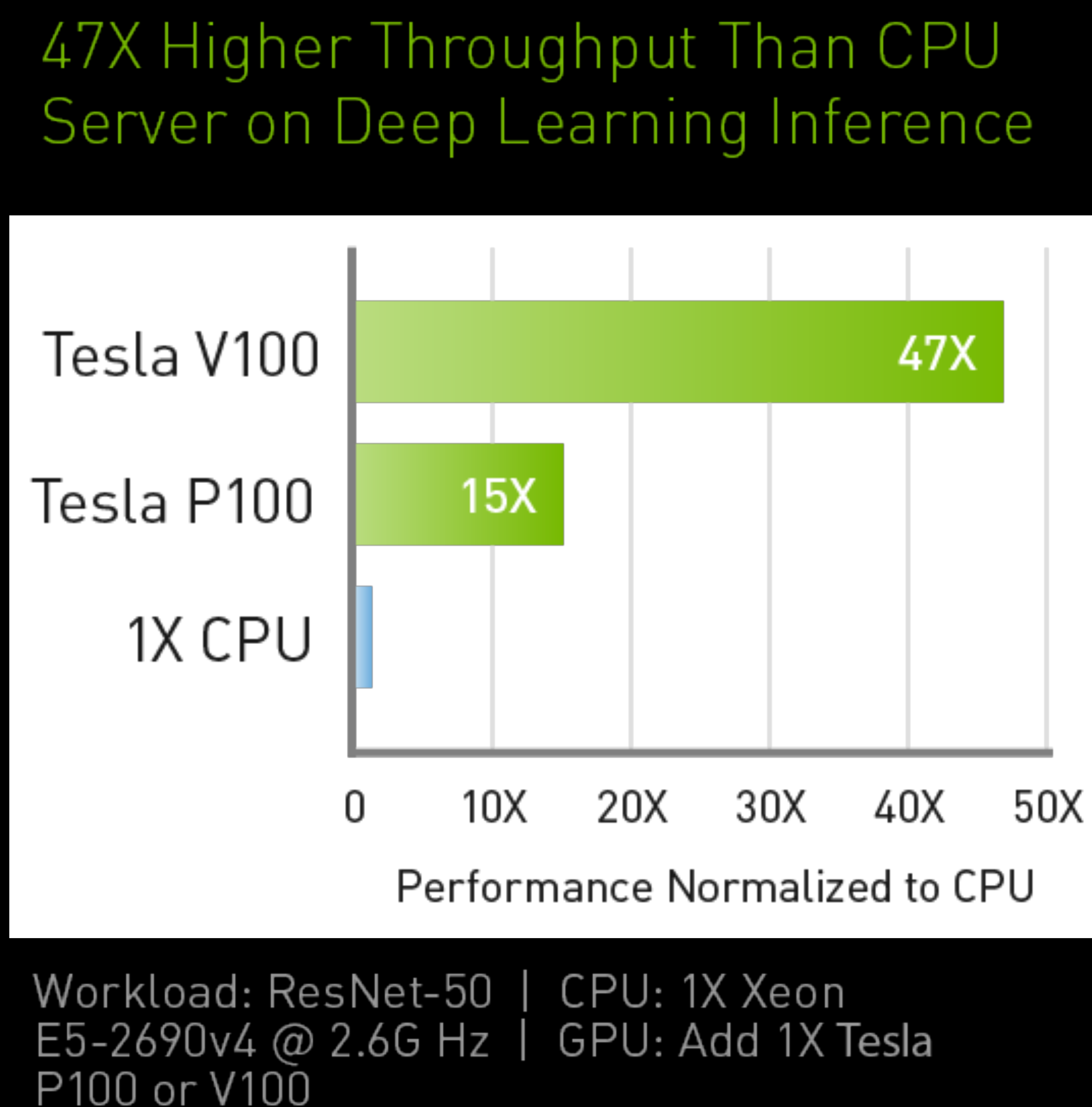
Amazon (AMZN:UW)
3,284.72 ▼ 12.65 (0.38%)
As of: 08/22/20 4:08:37 pm
(delayed at least 15 minutes)

Why GPUs?

Compared to CPUs

- higher parallelism
- higher memory bandwidth
- no operating system
- restricted execution model

Compared to CPUs

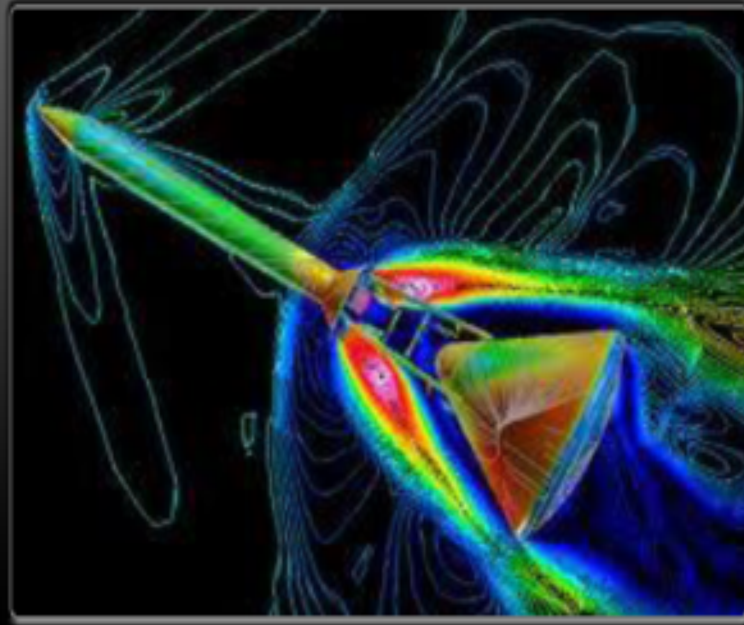


Compared to CPUs

	Throughput	Power	Throughput/Power
Intel Skylake	4.5 TFLOPS	205 Watts	21.9 GFLOPS/Watt
NVIDIA V100	14.9 TFLOPS	300 Watts	49.6 GFLOPS/Watt

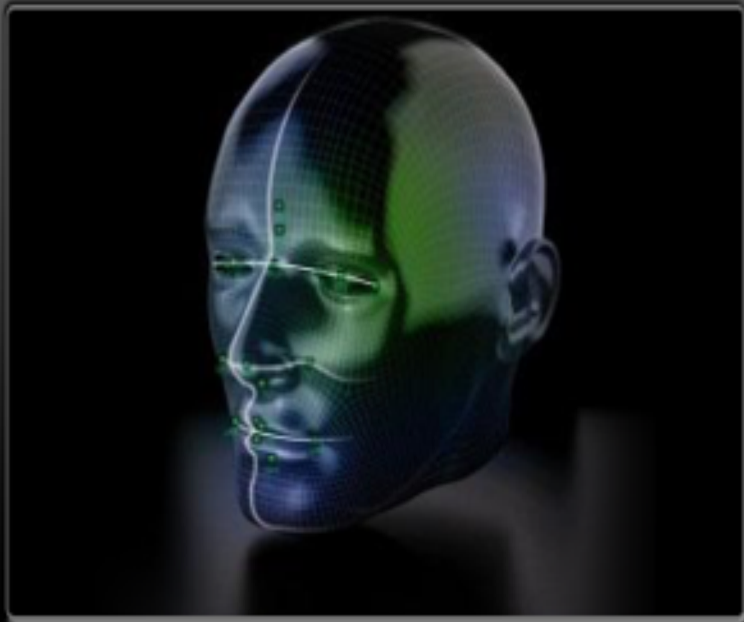
Satellite Imaging

Video Enhancement

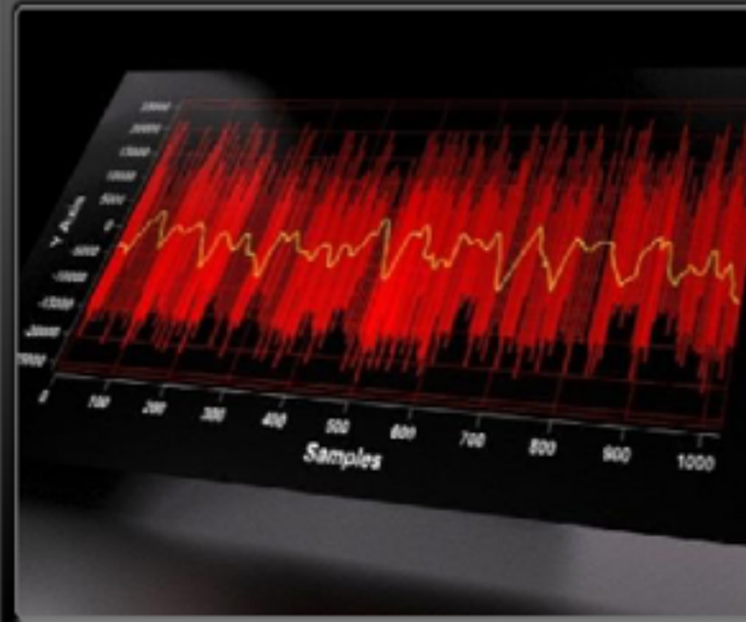


Aerodynamics/CFD

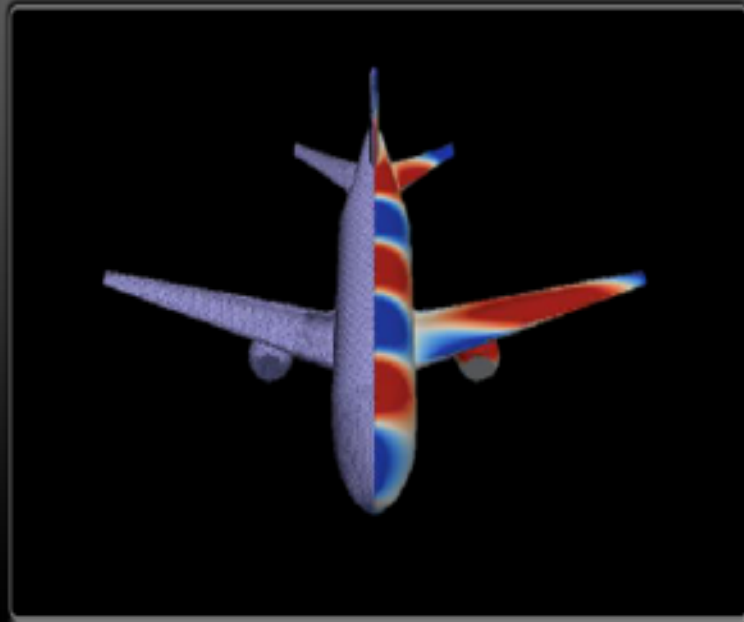
10x-100x Faster Thanks to GPUs



Computer Vision

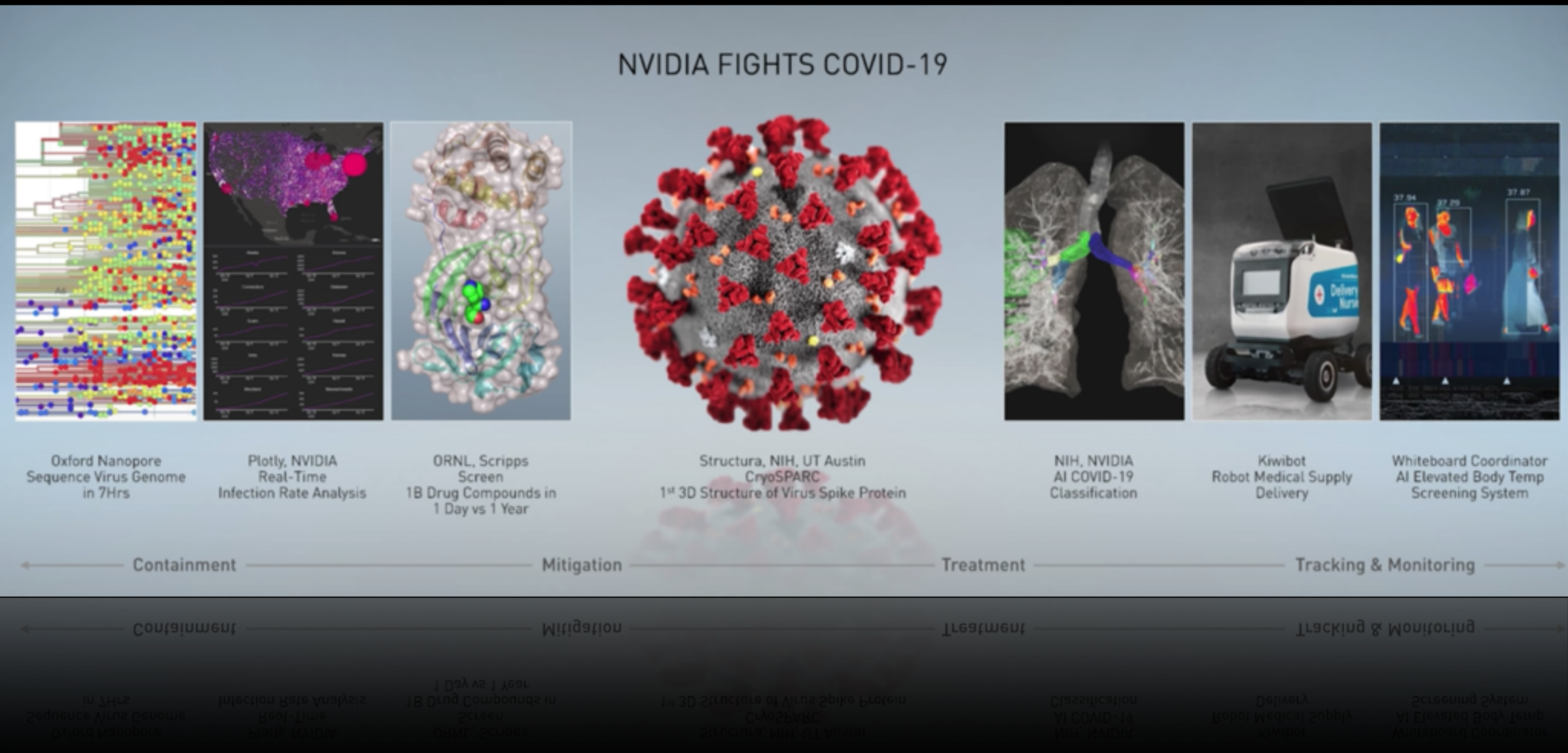


Signal Processing



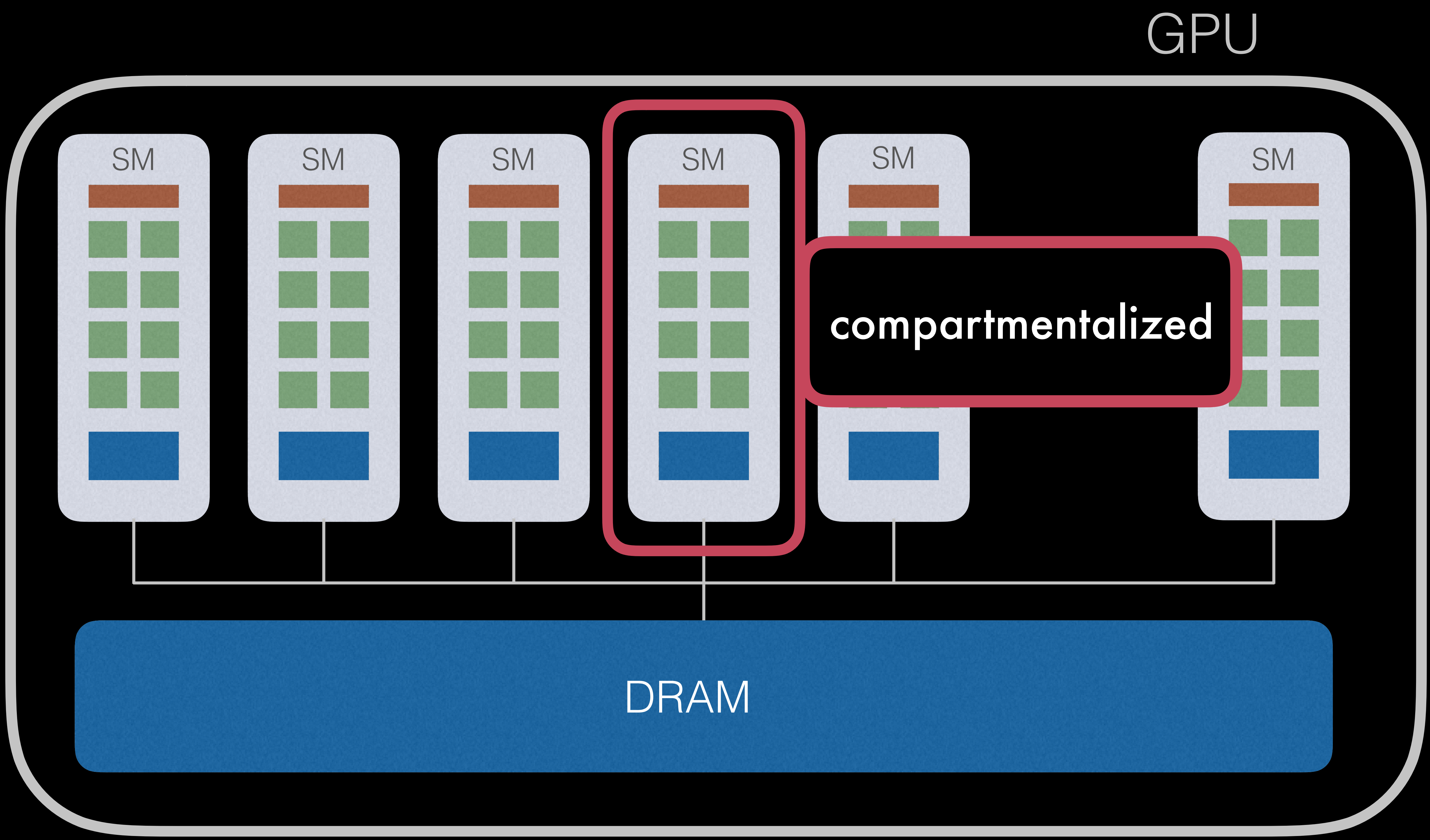
Stealth & Antenna

Covid-19 efforts

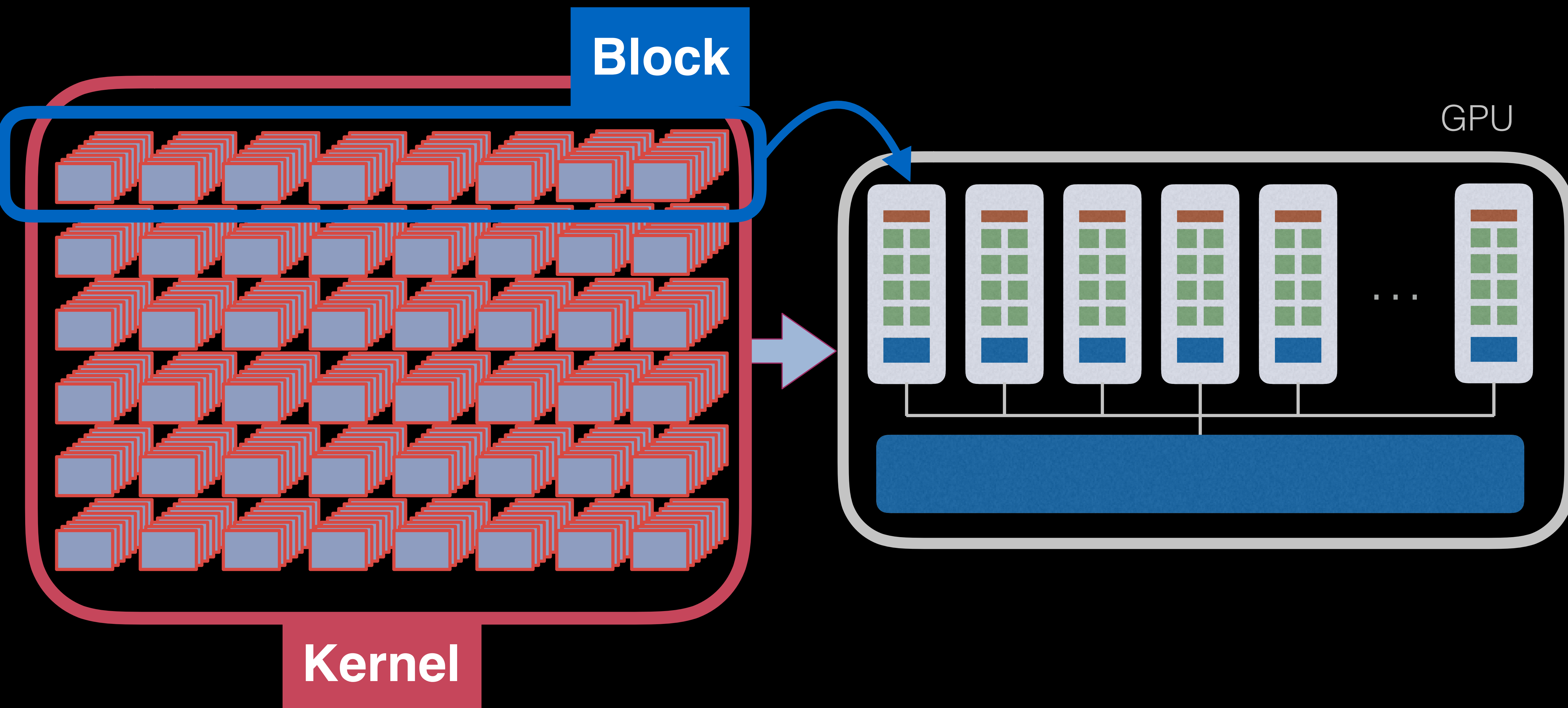


Inside a GPU

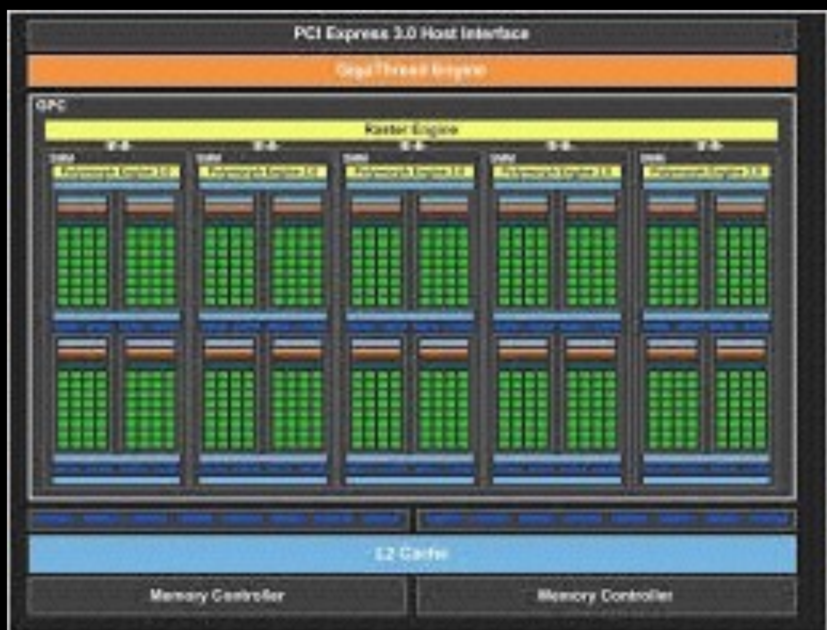
Architecture



Execution Model



Along the years...



Maxwell (2014)

Pascal (2016)

Volta (2017)

GPU hardware resources have grown considerably

Can one kernel fully utilize all these resources?

Resource Underutilization

Program	Kernel	TB	TPB	T%	R%	S%	B%
bfs	BFS_in_GPU	1	512	2	2	2	1
	BFS_multi_blk...	14	512	33	31	26	12
mri-q	ComputePhiMag...	4	512	10	5	0	4
	ComputeQ_GPU	1024	256	83	94	0	62
fft	GPU_FFT_Global	1024	128	67	62	0	100
stencil	block2D_hybrid...	512	256	67	94	17	50
cutcp	cuda_cutoff...	121	128	67	75	69	100
tpacf	gen_hists	201	256	50	70	81	38
histo	histo_final	42	512	100	94	0	38
	histo_intermediates	65	498	100	75	0	38
	histo_main	84	768	100	94	100	25
	histo_prescan	64	512	100	75	25	38
sad	larger_sad_calc_16	99	32	15	33	0	88
	larger_sad_calc_8	99	128	59	66	0	88
	mb_sad_calc	1584	61	33	50	38	100
mm	mysgemmNT	528	128	50	94	6	75
lbm	performStream...	13000	100	58	98	0	88
spmv	spmv_jds_texture	112	192	88	98	0	88
	Average			60	67	20	57

Pai, Sreepathi, Matthew J. Thazhuthaveetil, and Ramaswamy Govindarajan (2013). "Improving GPGPU concurrency with elastic kernels." ASPLOS 2013: 407-418.

Resource Underutilization

Use less resources → spend less energy?

Resource Underutilization

Use less resources → spend less energy?

GPUs are not energy proportional

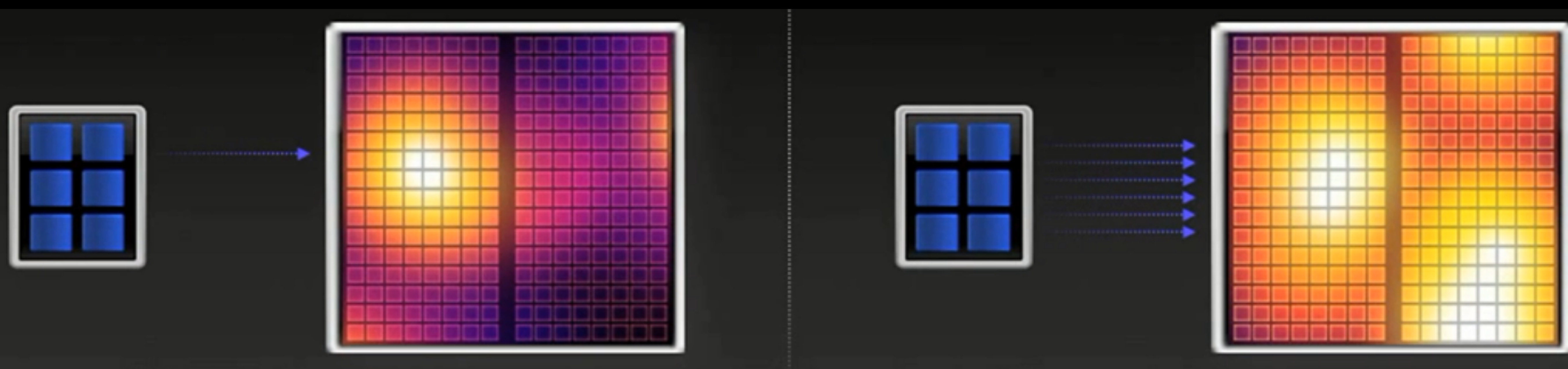
Power consumption does not reduce linearly with load reduction

Highly inefficient to underutilize the GPU

Concurrent Kernel Execution

Co-scheduling

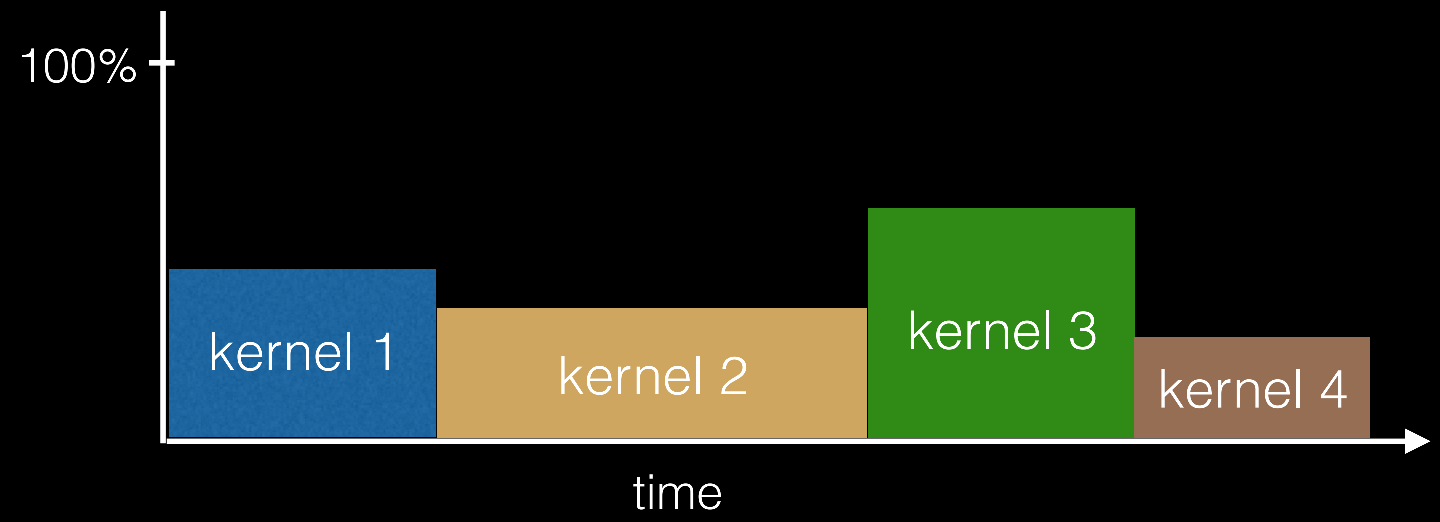
Single kernel | Hyper-Q technology



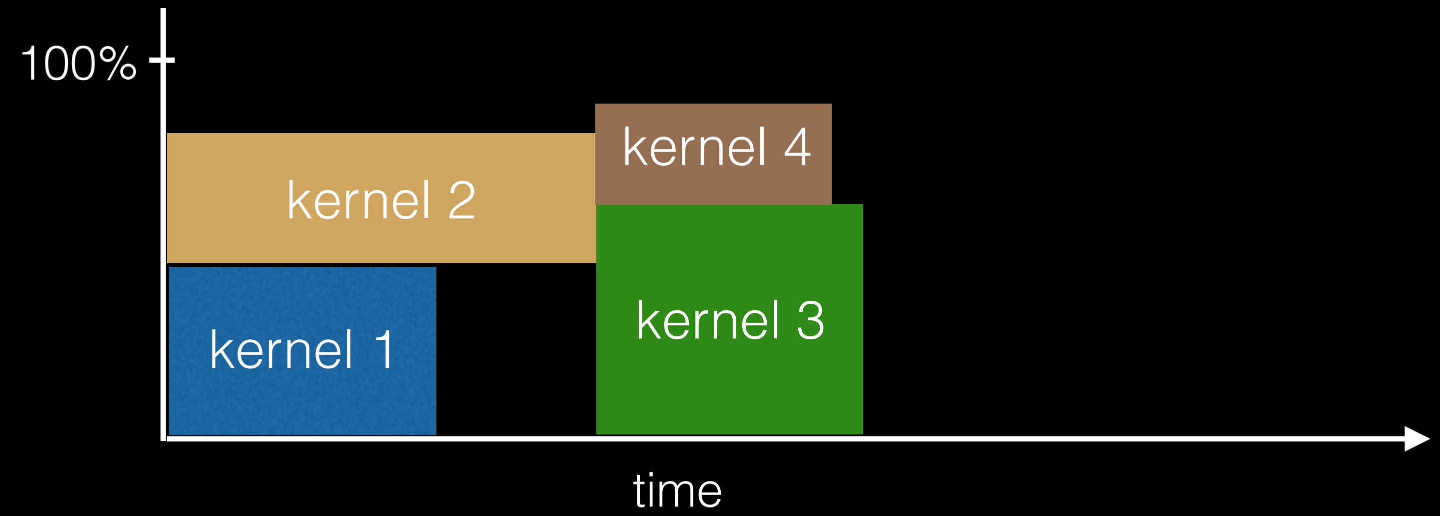
Improve resource utilization

Co-scheduling

Single kernel



Concurrent kernel

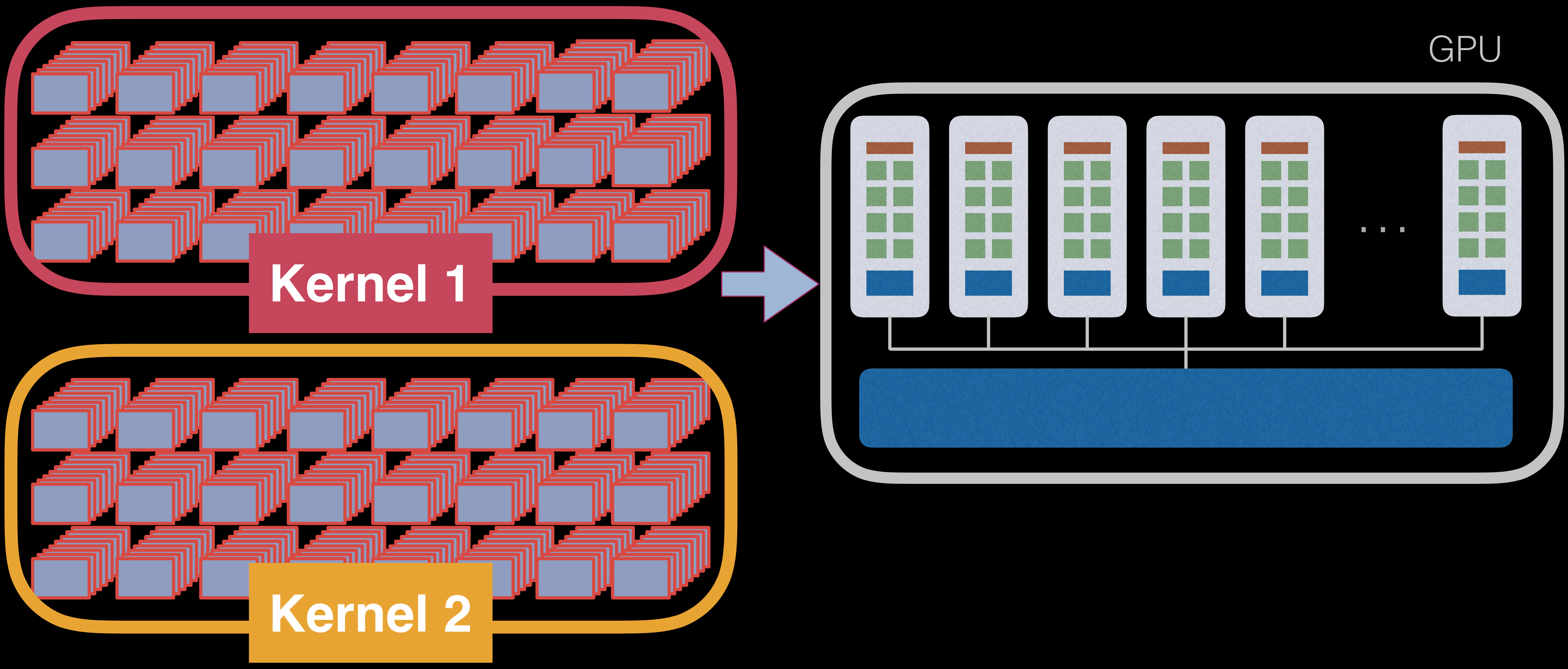


Improve throughput

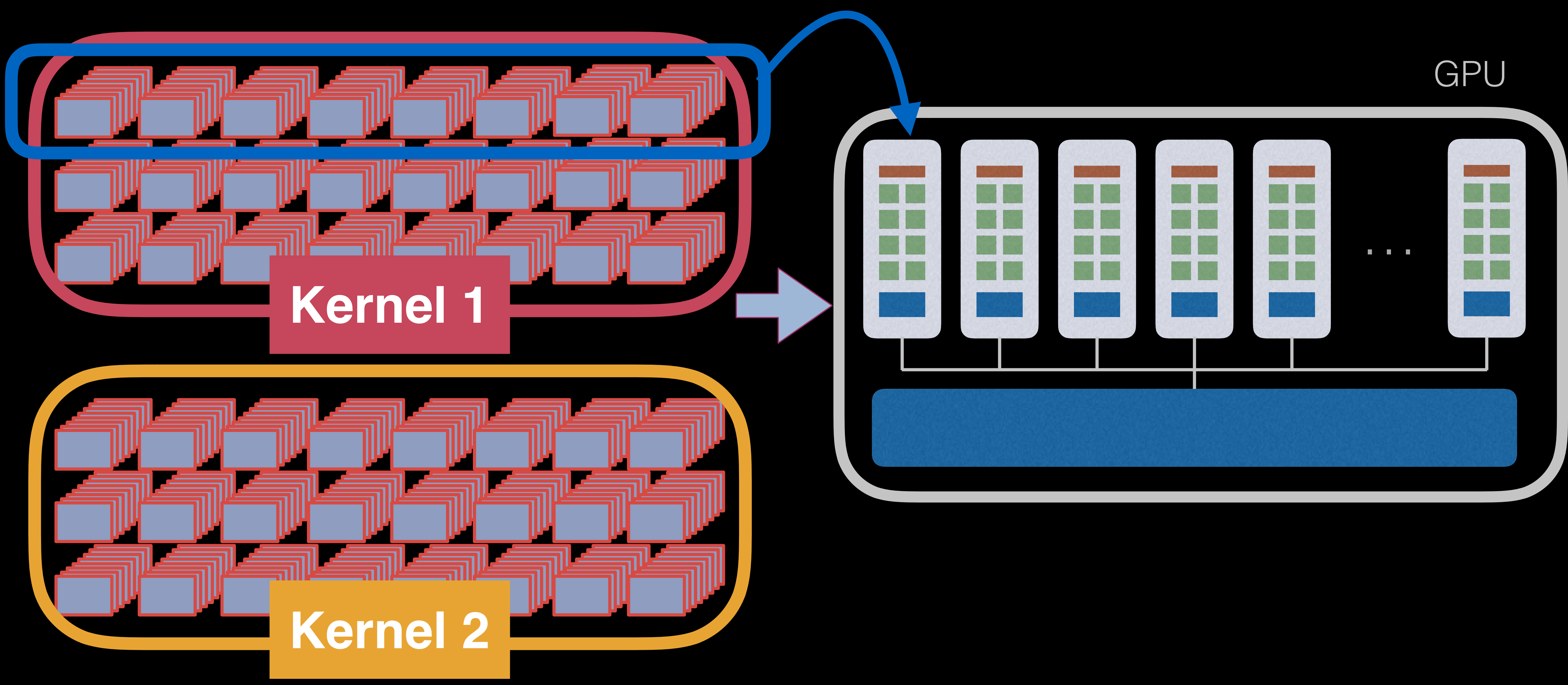
Co-scheduling

- GPUs do not have an operating system
- Scheduling is performed by the hardware
- Left-over scheduling policy

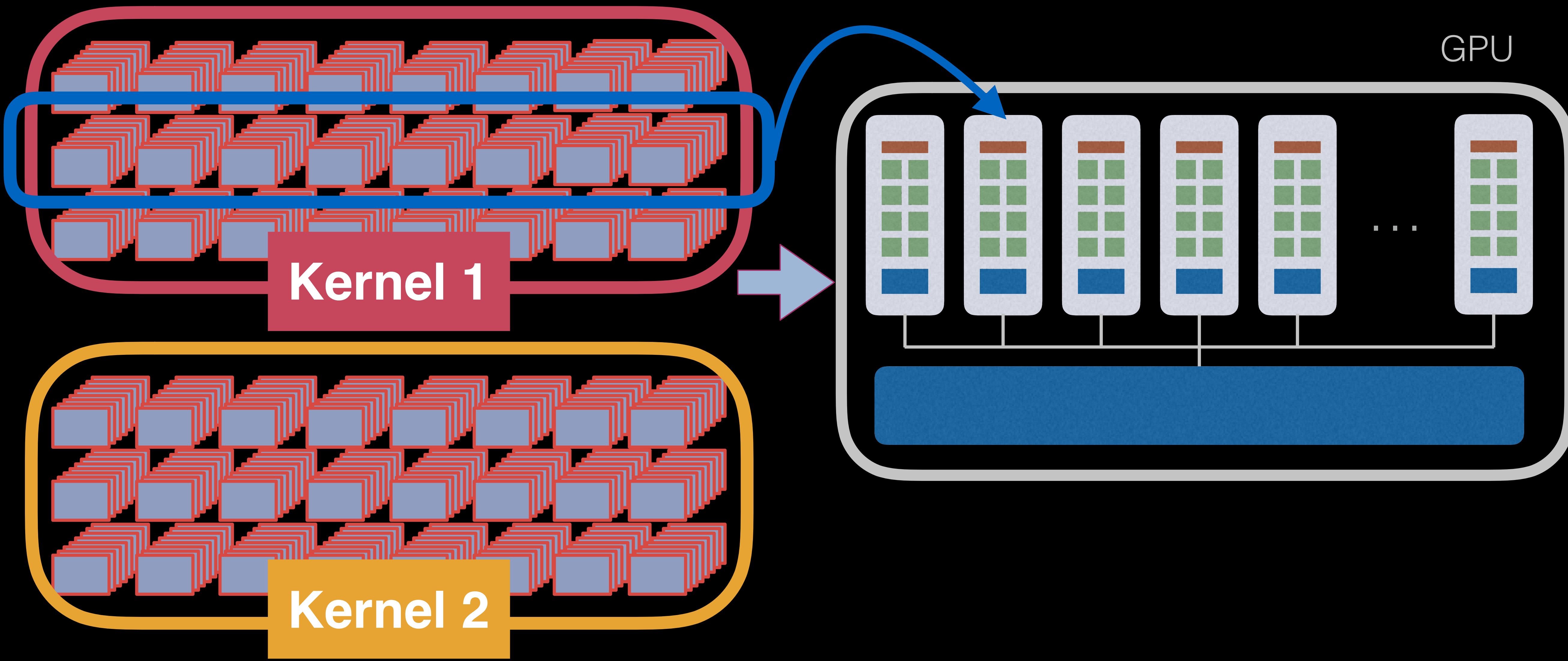
Letf-over Policy



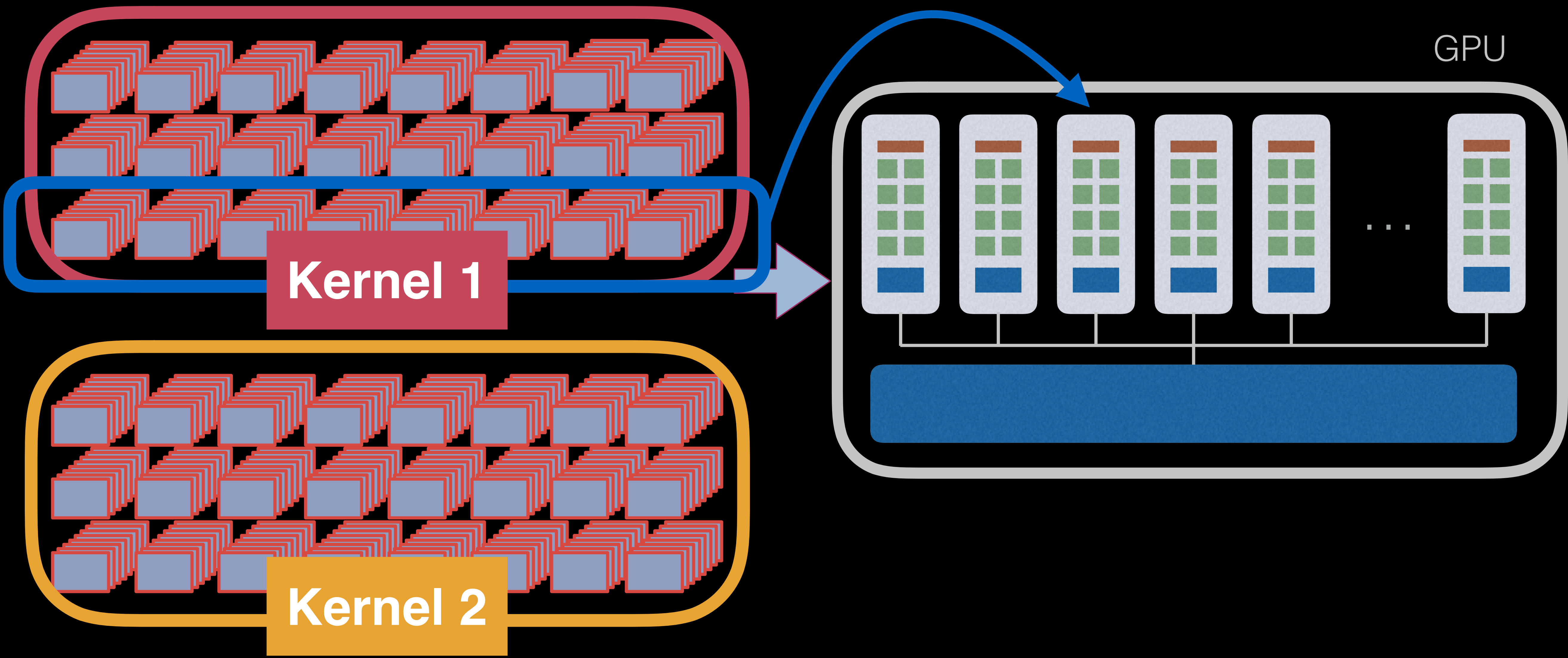
Letf-over Policy



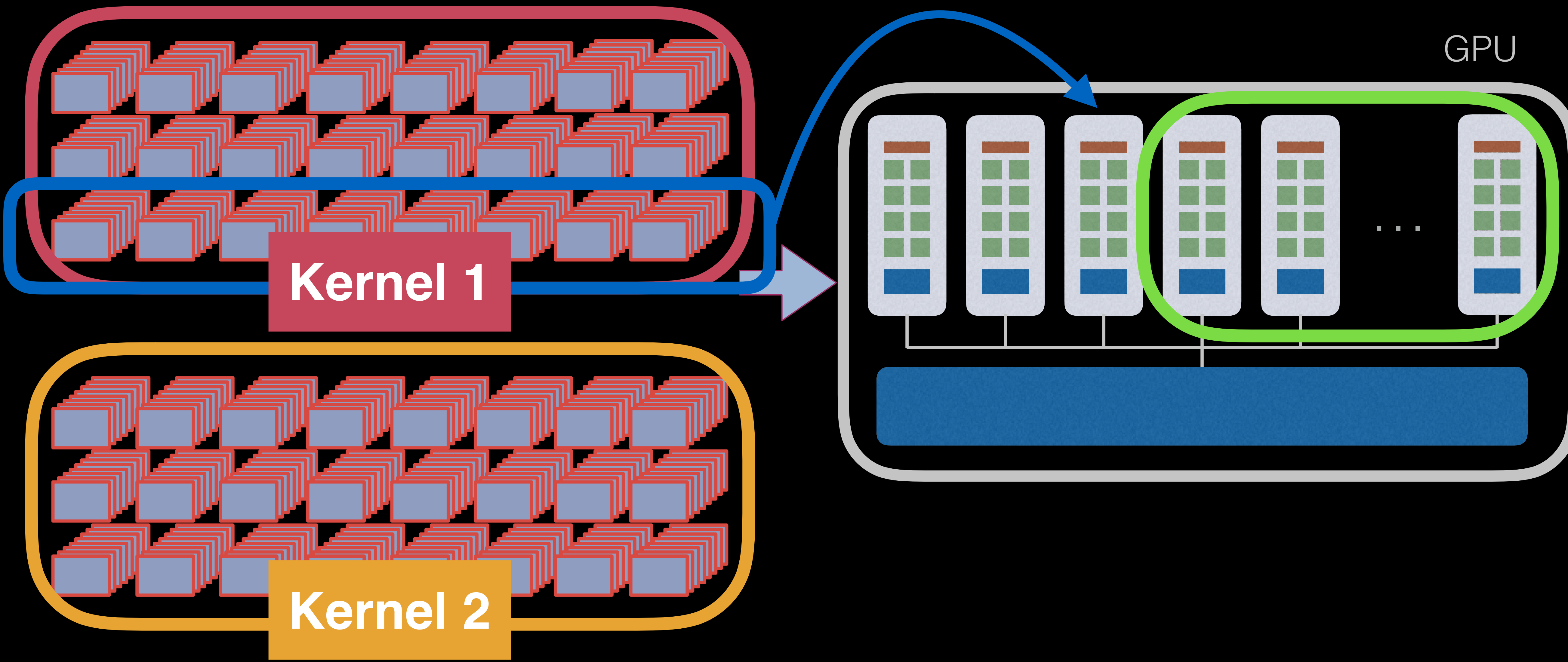
Letf-over Policy



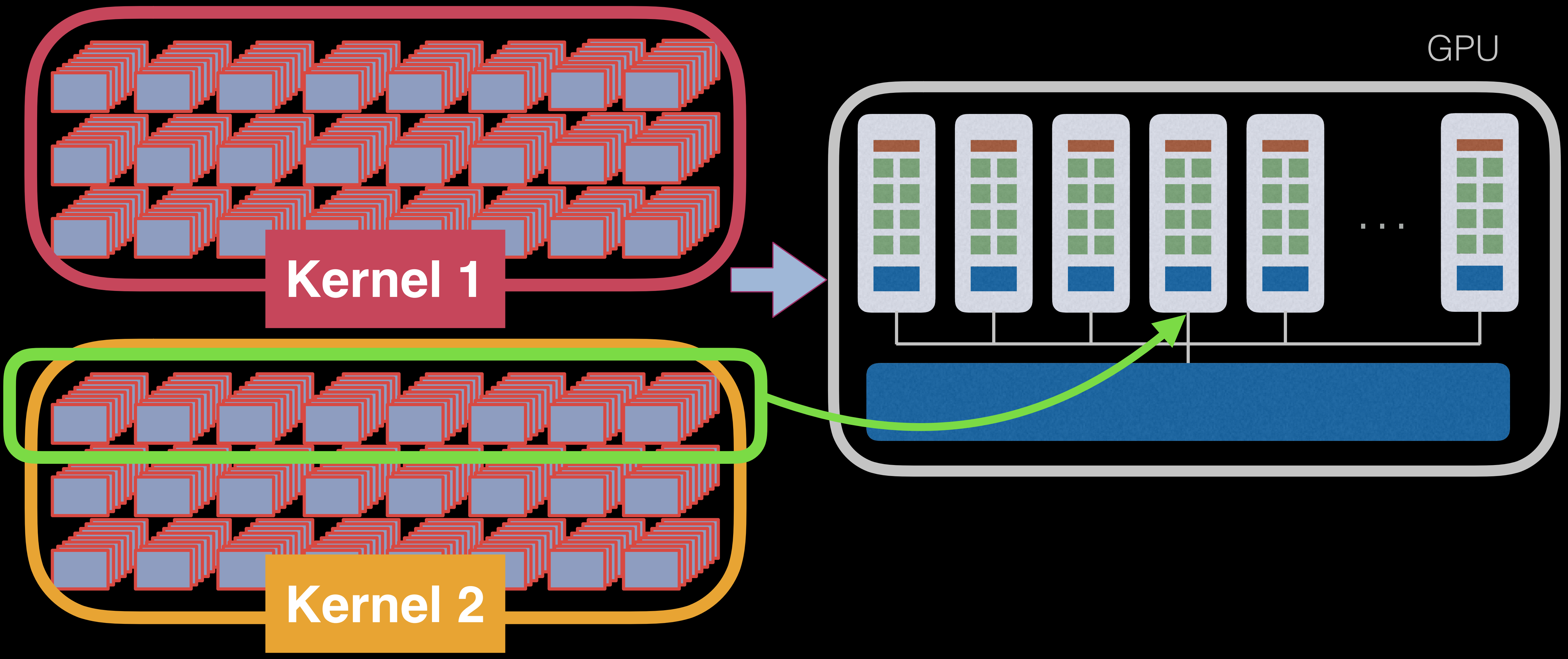
Letf-over Policy



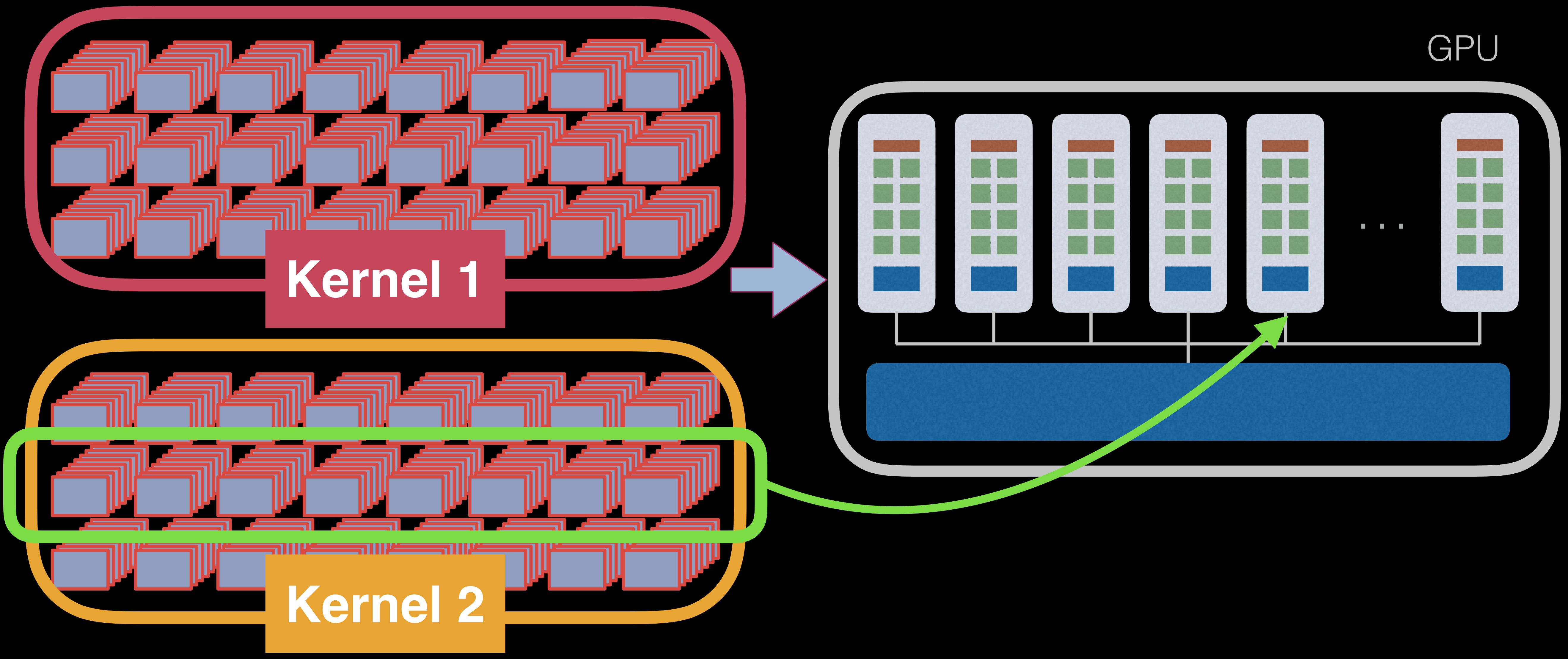
Letf-over Policy



Letf-over Policy



Letf-over Policy

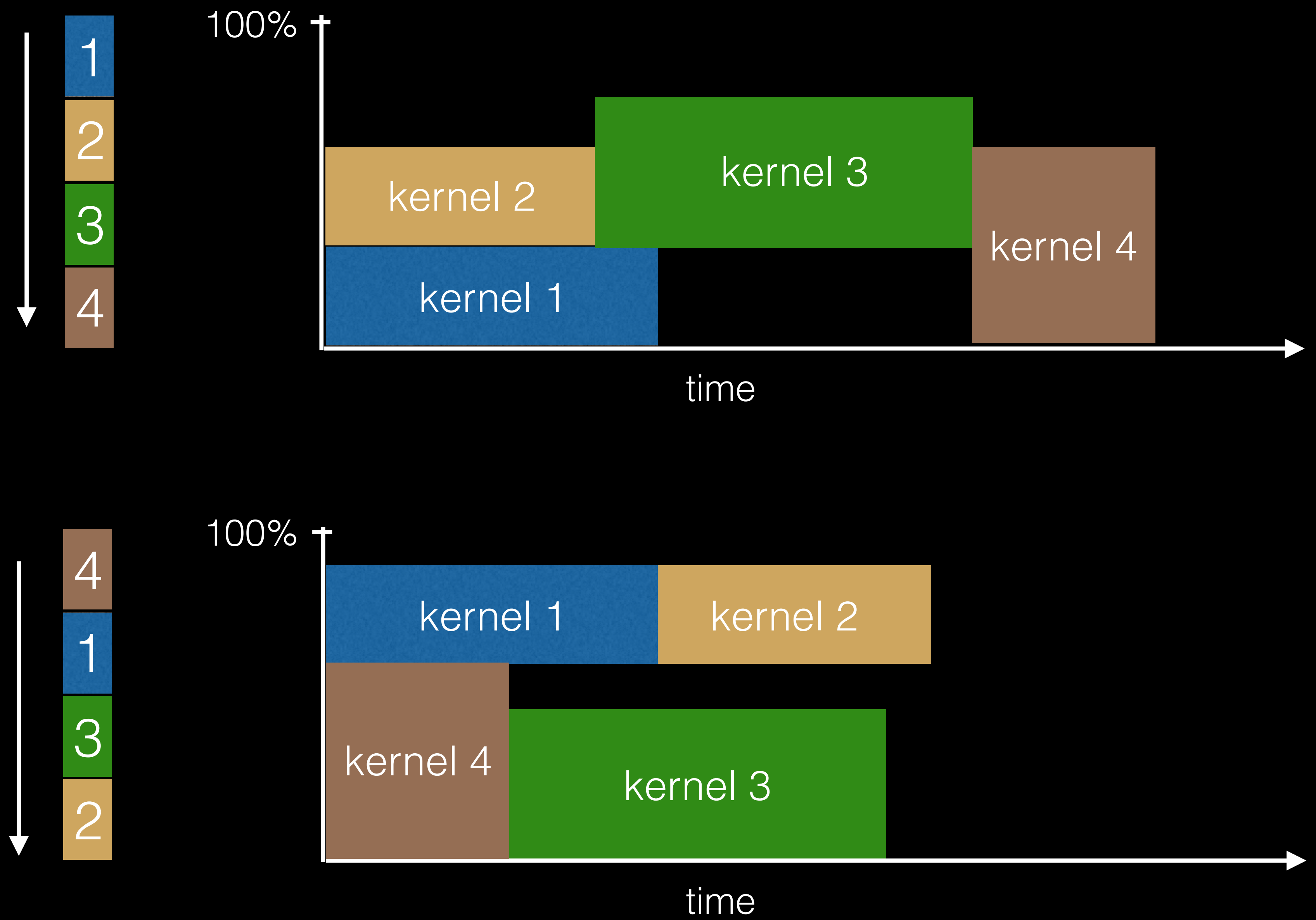


Problems

- Priority to the first kernel
- Order of submission matters

Problems

Submission order



Resource usage

kernel 1	30%
kernel 2	30%
kernel 3	50%
kernel 4	60%

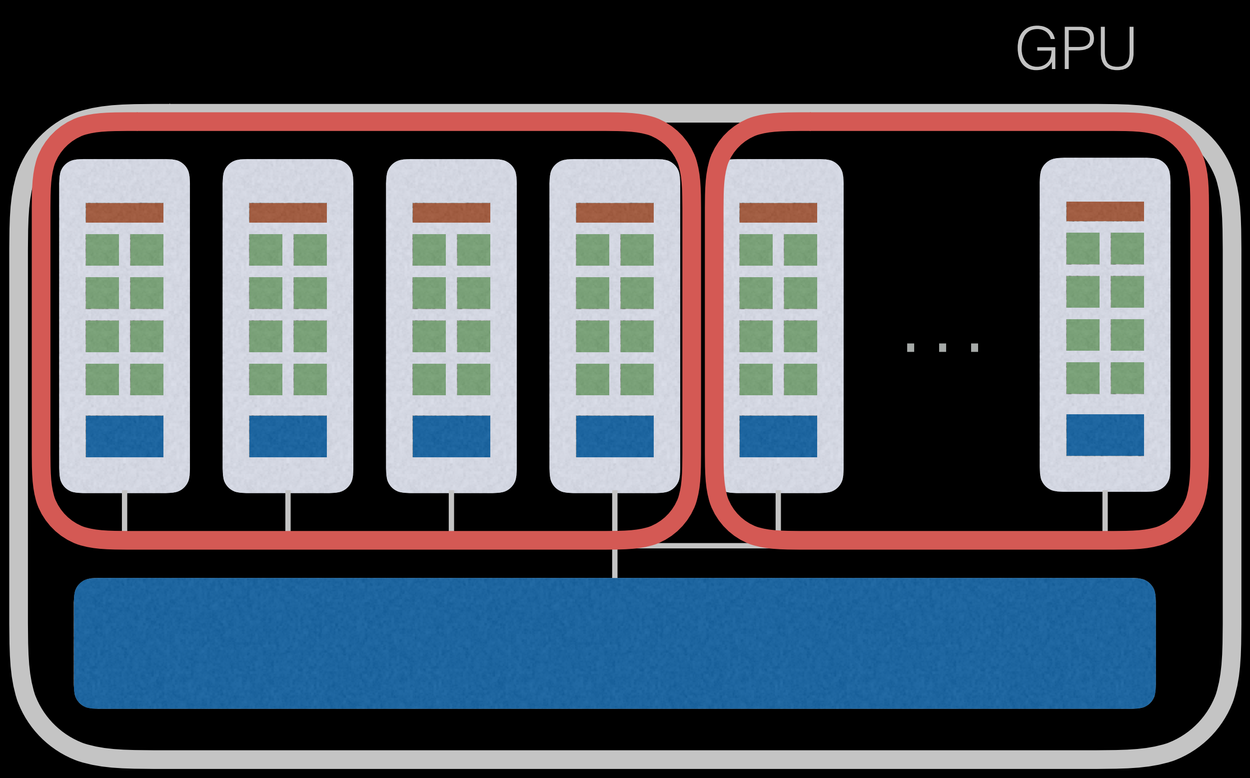
Therefore…

- Co-scheduling naively may result in almost no performance improvement
- Performance mainly depends on how the kernels require the resources

Main research topics on Concurrent Kernel Execution

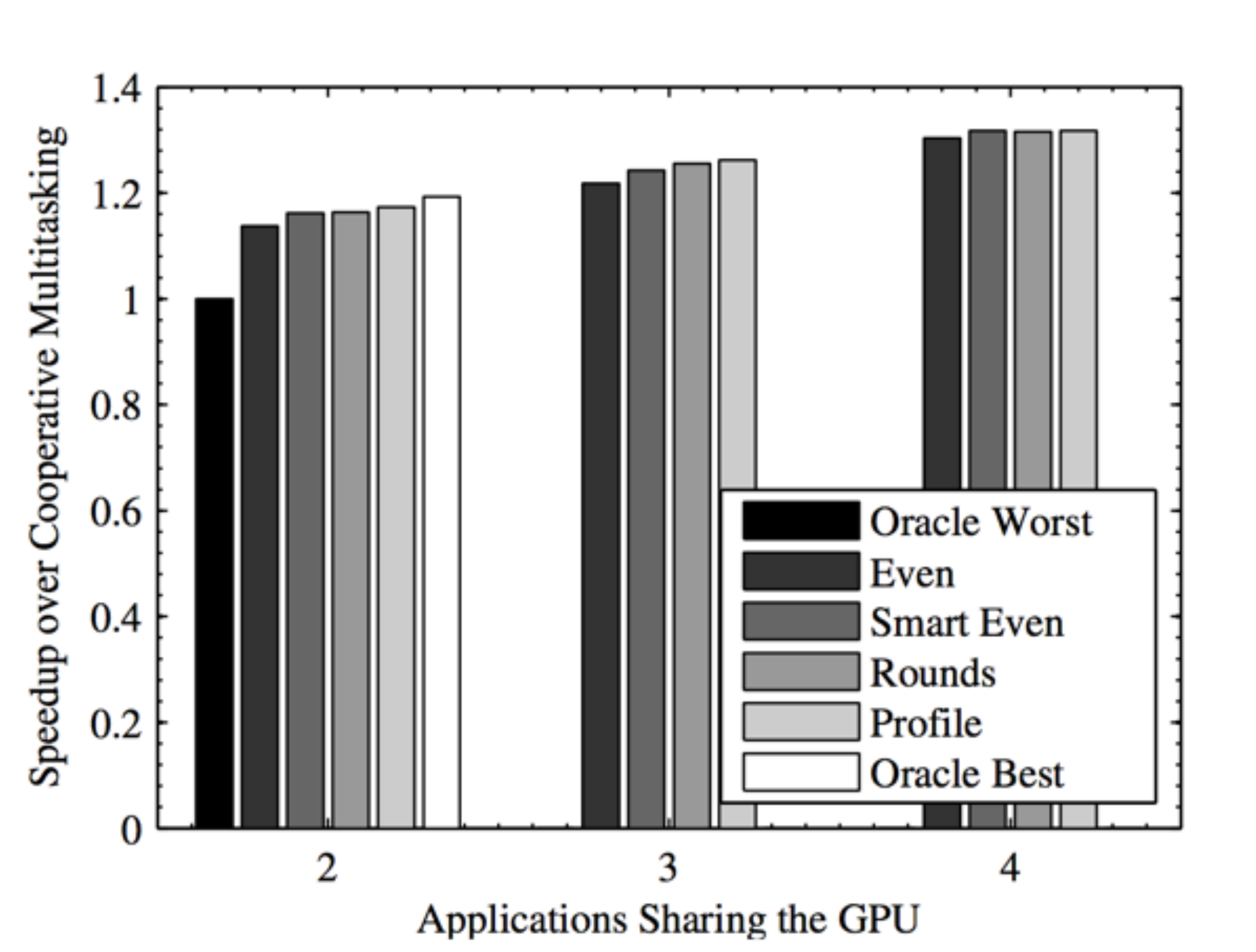
1. Improving concurrency opportunities

Spatial multitasking



- Avoid Left-over first kernel prioritization
- Split SMs into groups
- Each group of SMs runs a different kernel

Spatial Multitasking

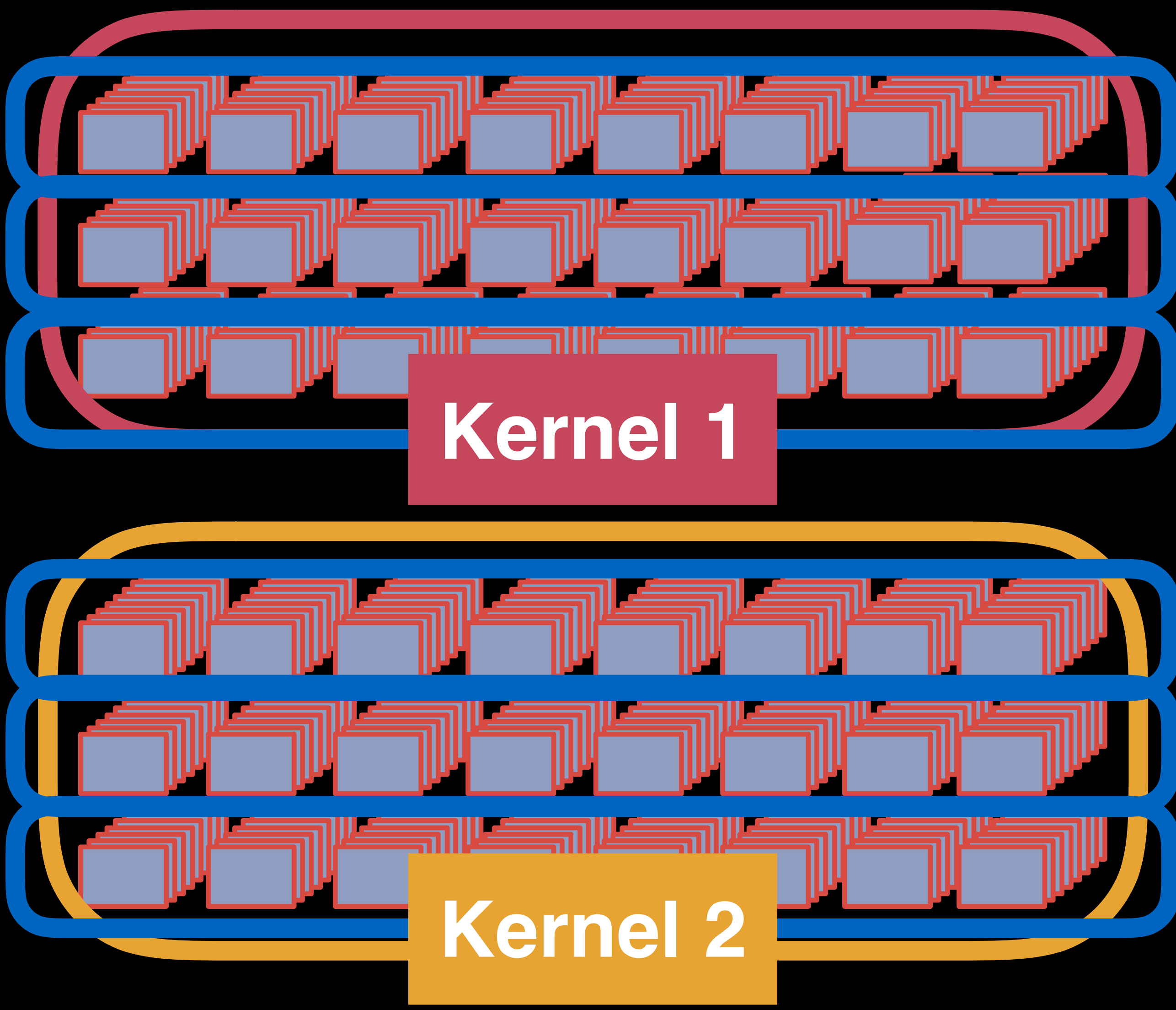


Average speedup of spatial multitasking over cooperative multitasking for several SM partitioning heuristics.

Adriaens, Jacob T., et al. "The case for GPGPU spatial multitasking." IEEE International Symposium on High-Performance Comp Architecture. IEEE, 2012.

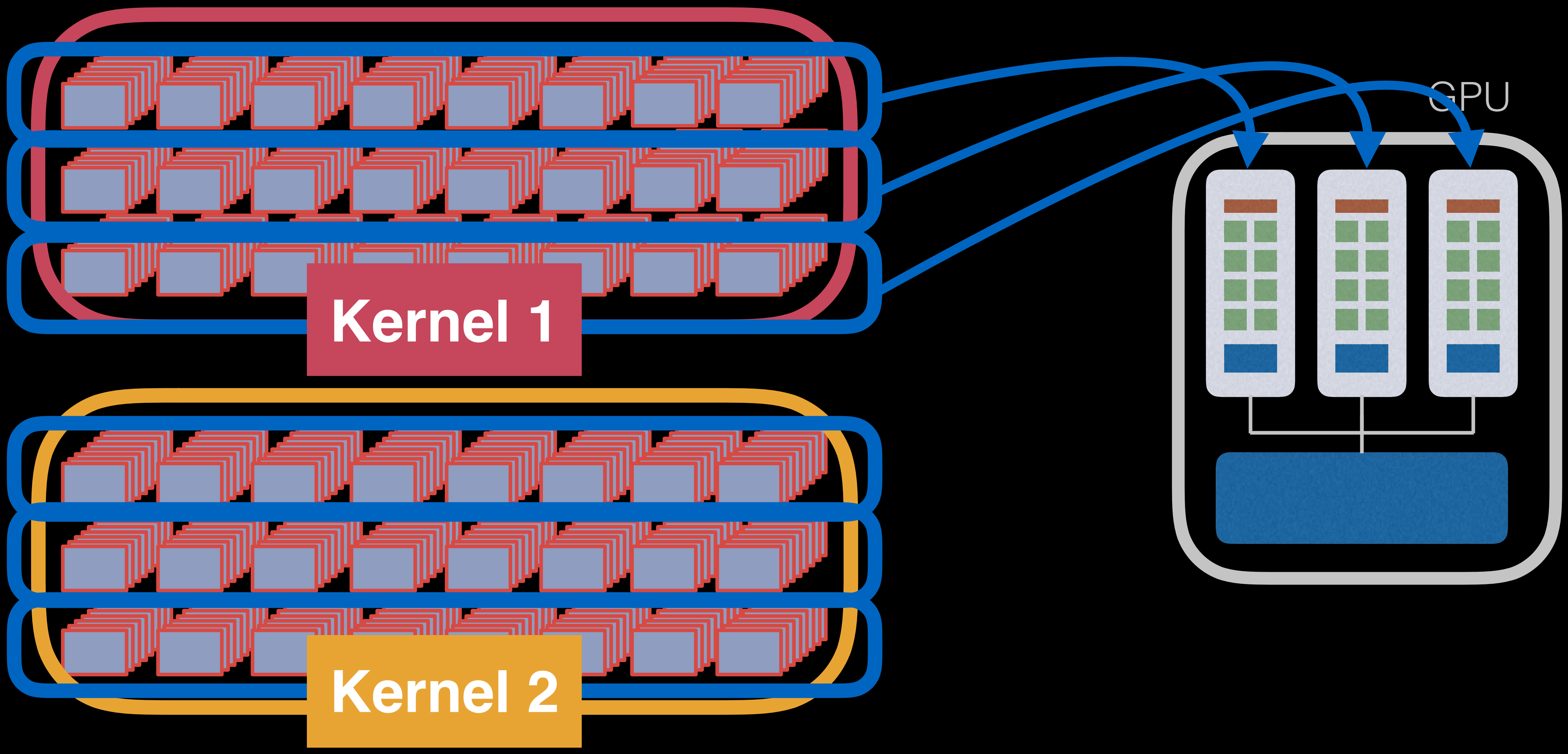
Does not address underutilisation within SM

Changing block granularity

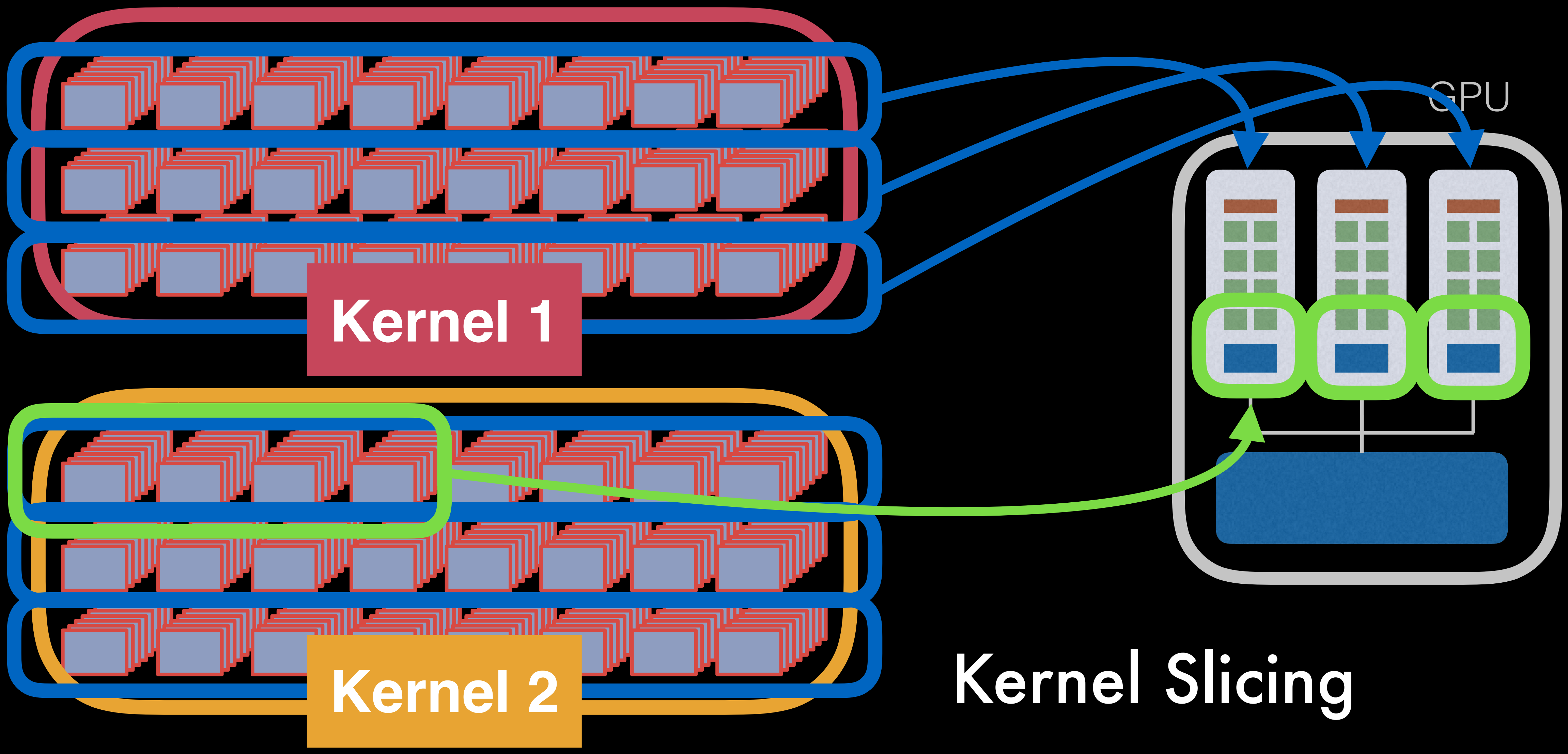


- Hardware maps blocks to SMs in Round-robin
- Resources are available in smaller chunk

Changing block granularity

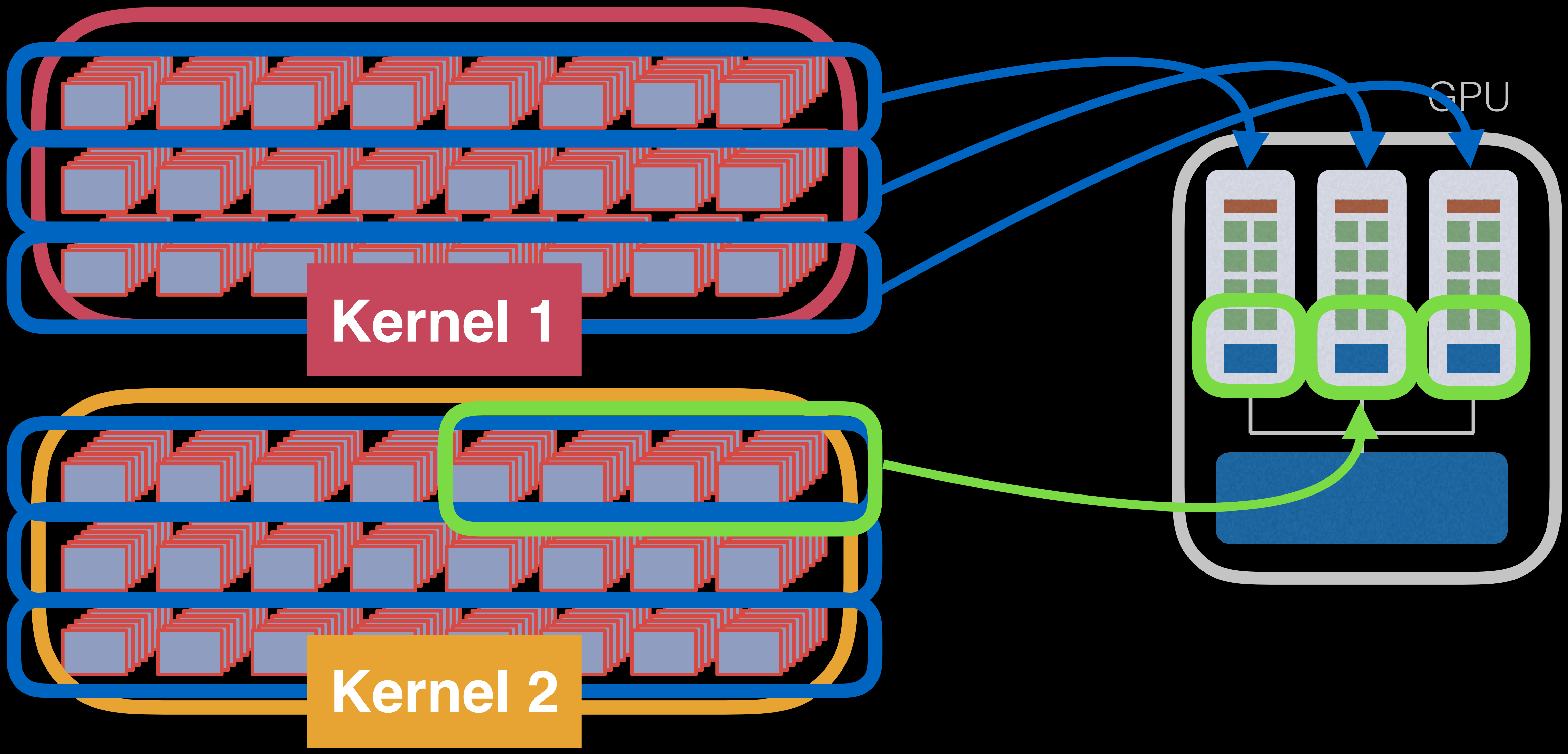


Changing block granularity

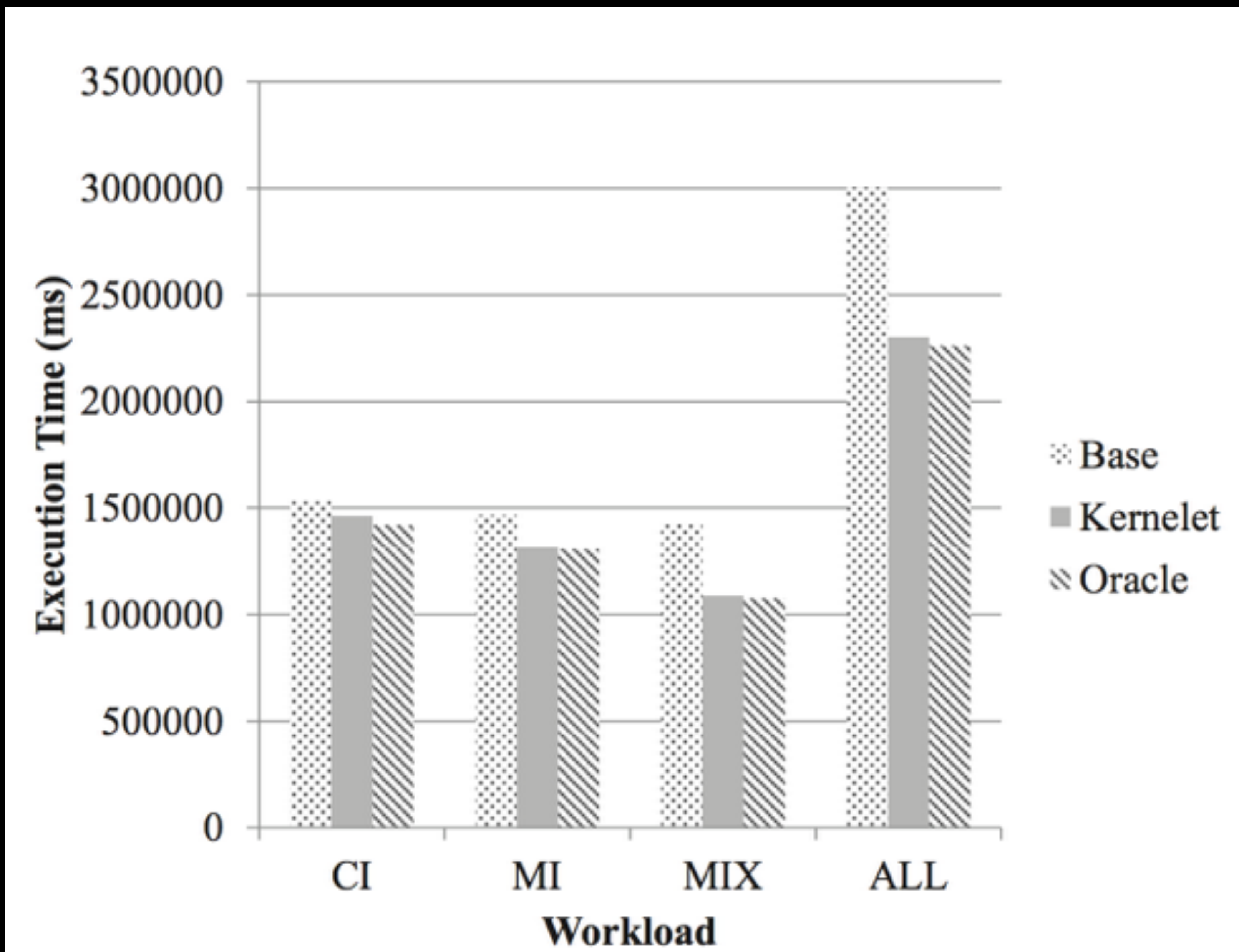


Kernel Slicing

Changing block granularity



Changing block granularity



Zhong, Jianlong, and Bingsheng He. "Kernelet: High-throughput GPU kernel executions with dynamic slicing and scheduling." IEEE Transactions on Parallel and Distributed Systems 25.6 (2013): 1522-1532.

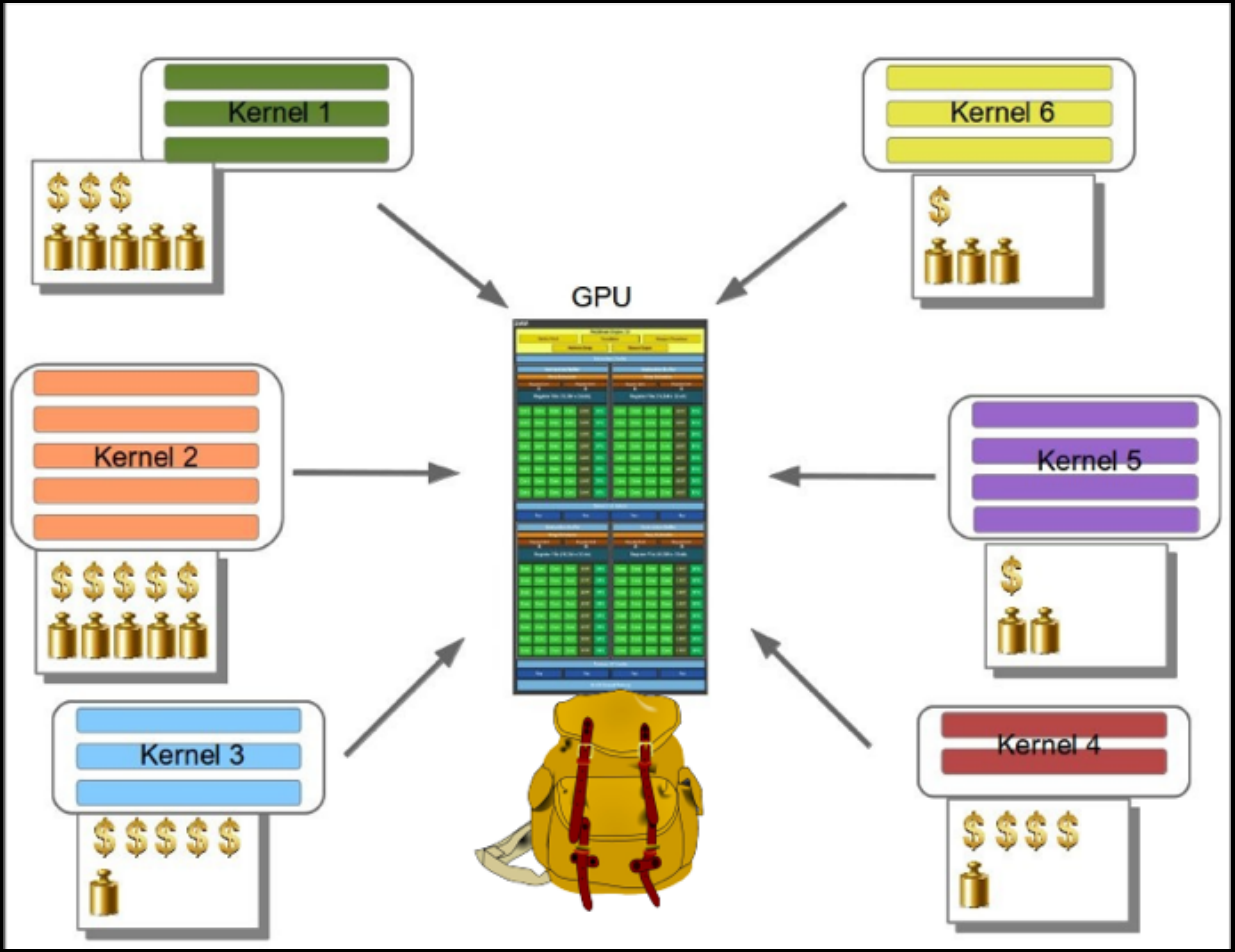
Change application code

2. Dealing with submission order

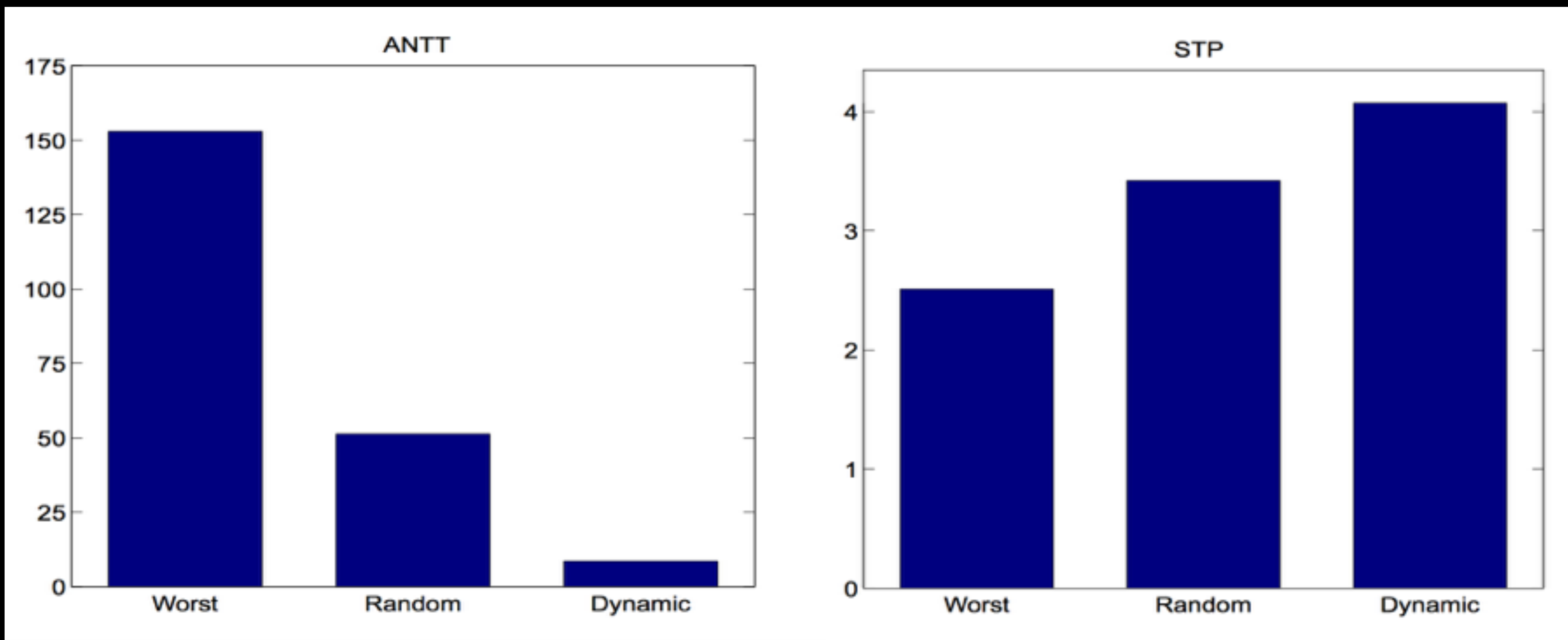
Submission order

- Optimization strategy to determine the best order to submit the kernels
- Model as a series of knapsack problems - items to put on a knapsack that maximize the profit without exceeding the capacity
- Knapsack capacity = Available resources
- Items = Kernels

Submission order



Submission order



Cruz, R. A., Bentes, C., Breder, B., Vasconcellos, E., Clua, E., de Carvalho, P. M., & Drummond, L. M. (2019). Maximizing the GPU resource usage by reordering concurrent kernels submission. Concurrency and Computation: Practice and Experience, 31(18), e4409.

3. Which kernels are most appropriate to be launched concurrently?

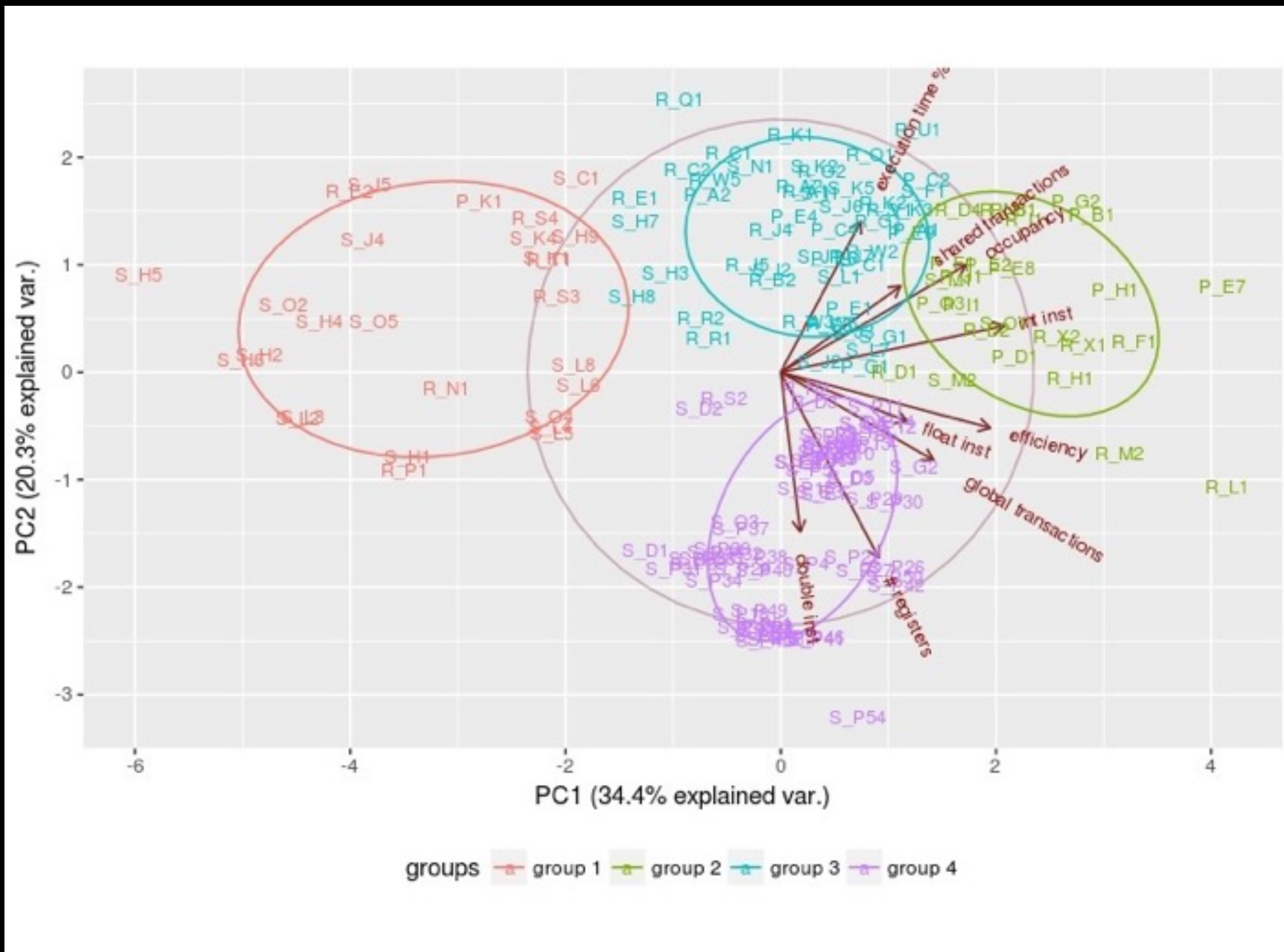
Kernel Characterization

- Analyze the individual behaviour of the kernels in terms of resource usage
- Kernel profiling (nvprof)
- Guide decisions on more efficient concurrent execution
- Classify the kernels with similar characteristics in terms of resource usage from Parboil, SHOC and Rodinia

Kernel Characterization

- Resource requirements: integer, single and double precision floating point operations, SM efficiency, GPU occupancy and memory operations
- Principal Component Analysis (PCA) statistical method for reducing dimensionality
- K-means clustering for creating the groups

Kernel Characterization



- G1: small kernels
- G2: arithmetic intensive
- G3: medium kernels
- G4: low occupancy

Carvalho, P., Cruz, R., Drummond, L. M., Bentes, C., Clua, E., Cataldo, E., & Marzulo, L. A. (2020). Kernel concurrency opportunities based on GPU benchmarks characterization. Cluster Computing, 23(1), 177-188.

Co-scheduling

- Effects of the concurrent execution of the kernels from the different groups
- Execute concurrently a sample of pairs of kernels from different groups

	G2			
G4	P_E2	P_E8	R_D2	R_X2
S_P4	4.704	0.804	0.502	0.711
S_P15	0.638	0.619	1.996	0.507
S_E4	0.679	0.768	0.490	0.555
S_P38	1.439	0.411	0.682	0.521

	G2			
G3	P_E2	P_E8	R_D2	R_X2
P_C4	0.189	0.749	0.738	0.572
R_J4	2.926	0.416	0.637	0.499
R_J7	5.516	0.555	3.395	19.683
S_K5	0.321	1.724	0.538	2.032

	G3			
G4	P_C4	R_J4	R_J7	S_K5
S_P4	20.410	0.381	0.123	0.901
S_P15	2.780	0.123	0.098	0.340
S_E4	0.440	0.334	0.182	0.827
S_P38	0.733	0.788	0.085	0.961

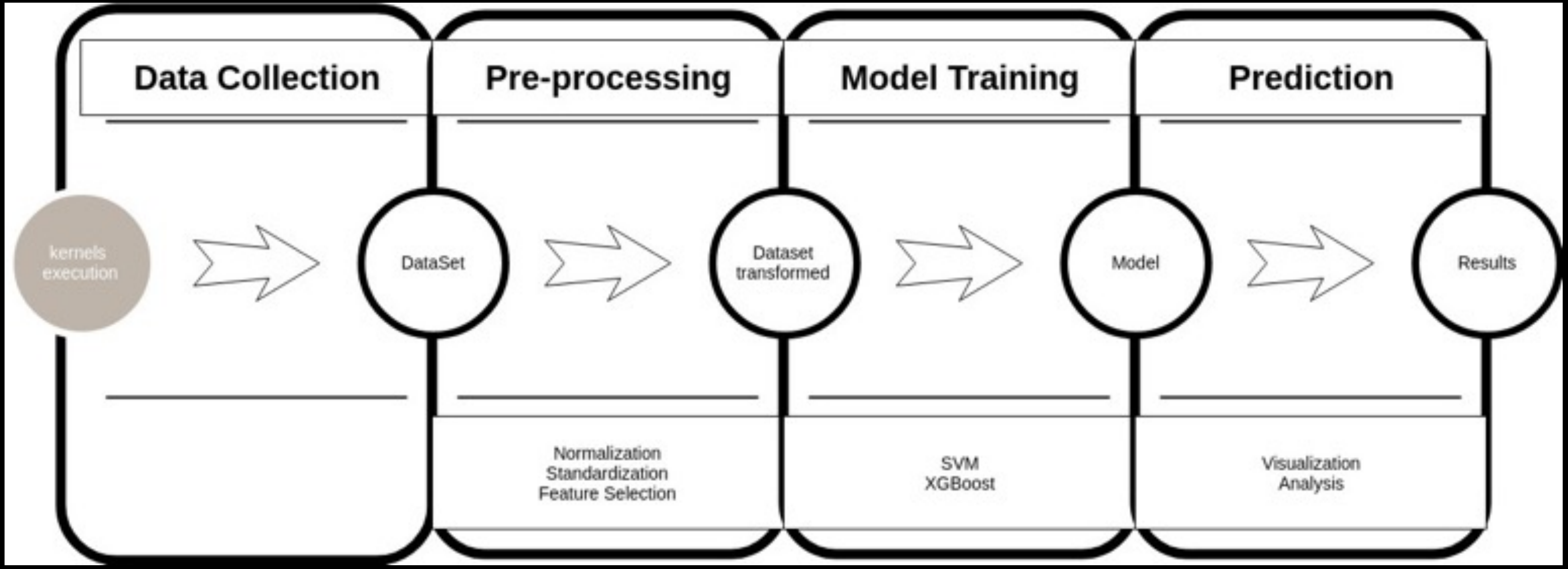
Co-scheduling

- Kernels with heavy requirements on one resource may prevent concurrent execution
- Synchronization or global memory access time can make the other kernel dominate the SM
- Inconclusive results

4. Kernels interference

Kernel interference

- Kernel resource requirements relations are tricky
- Machine learning techniques to model and predict concurrency and interference



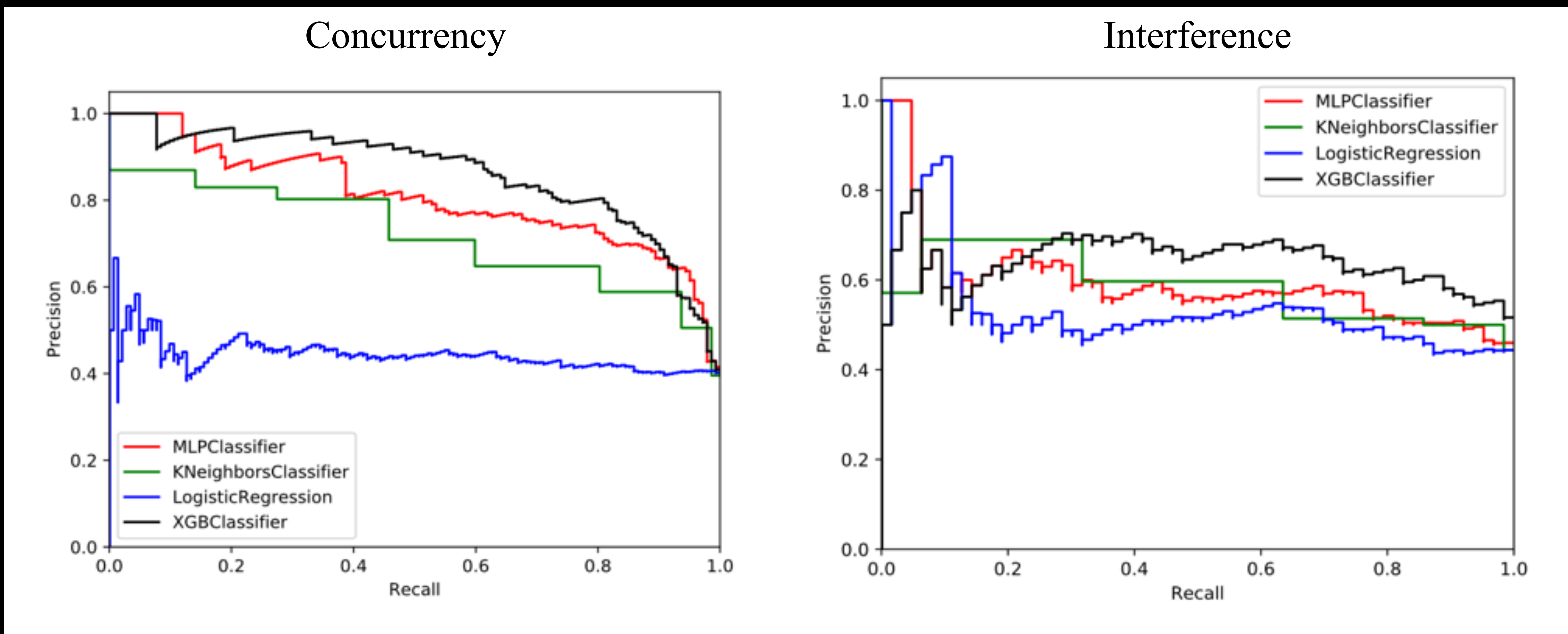
Kernel Interference

- Selected 60 kernels (15 representative kernels from each category) and executed all possible 3,600 permutations

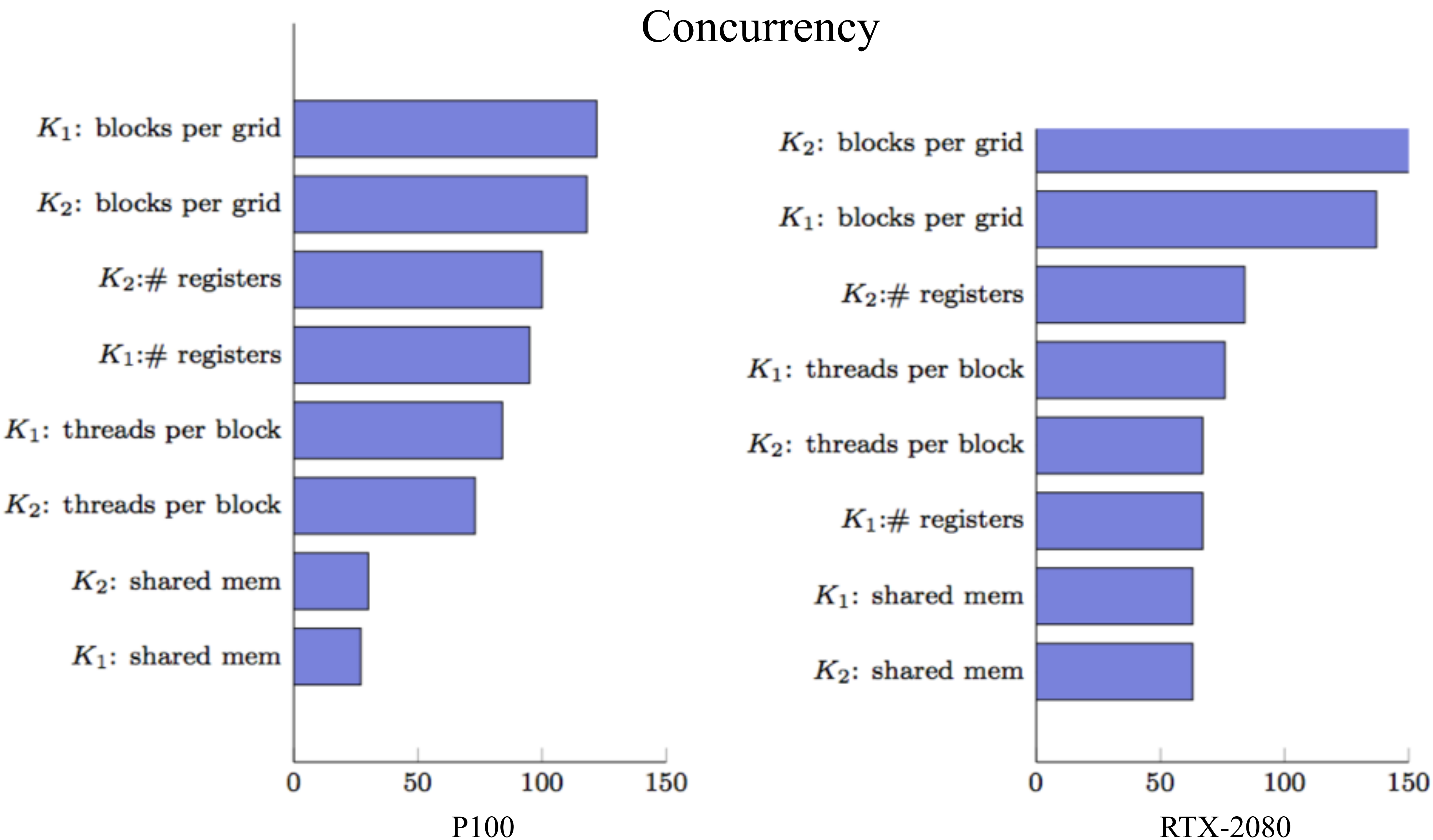
- Resource variables for each kernel: blocks per grid, threads per block, number of registers and shared memory

- Variables selected are exposed to the developer before the kernel execution

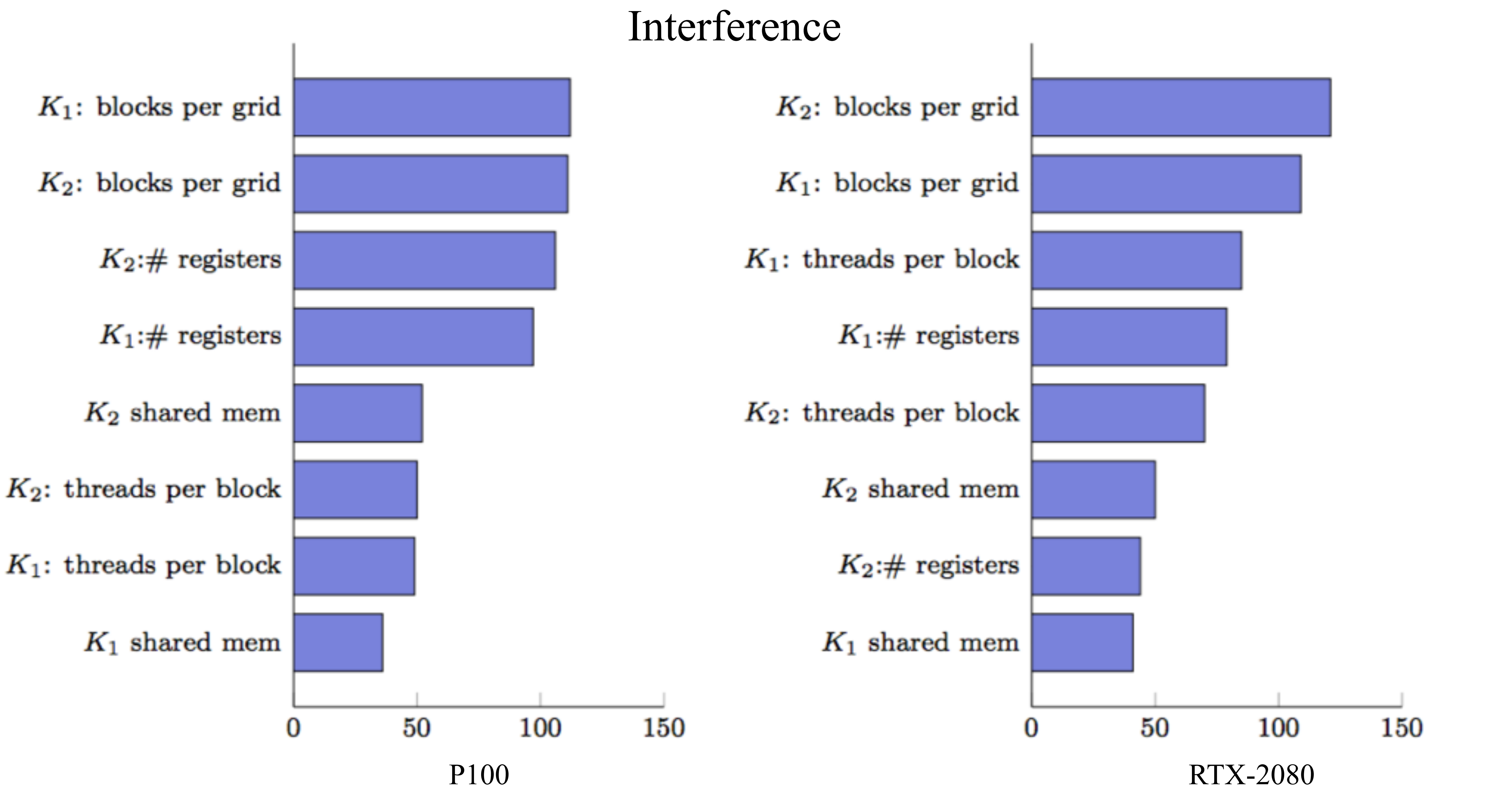
Kernel Interference



Kernel Interference



Kernel Interference

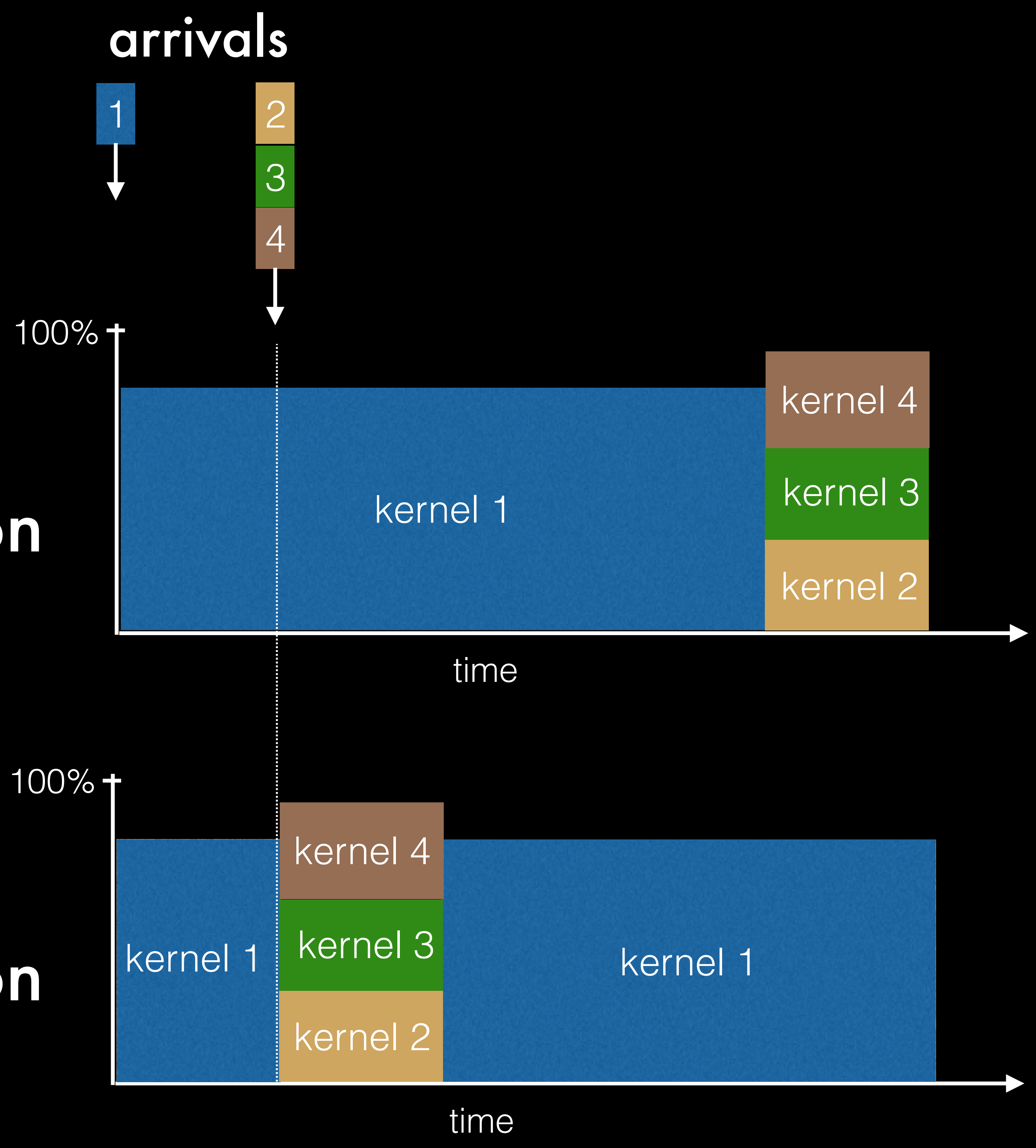


5. Dealing with preemption in co-scheduling

Preemption

- Modern GPUs provide hardware preemption
- Coarse-grain (thread level) - reduces the amount of context to be saved
- Fine-grain (instruction level) - substantially more state information

Preemption



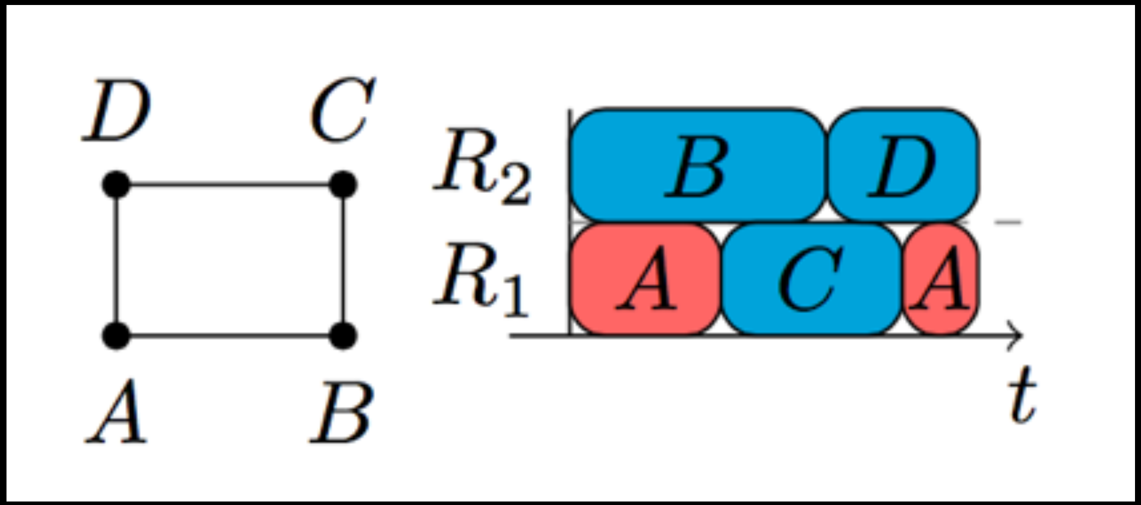
without preemption

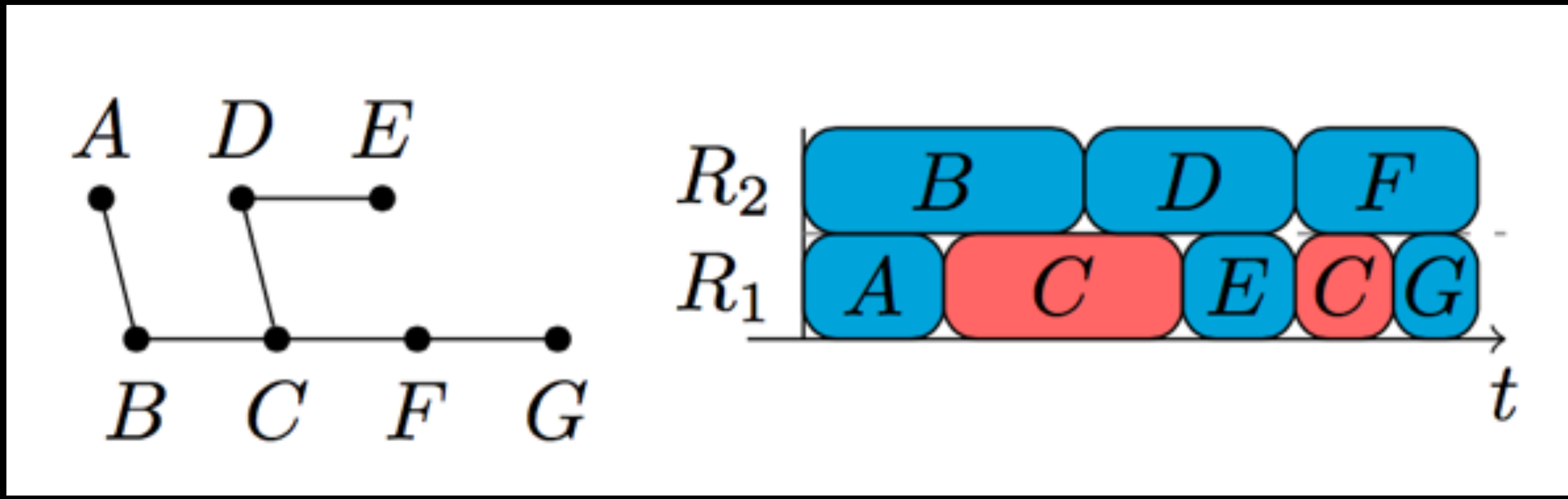
with preemption

improve responsiveness

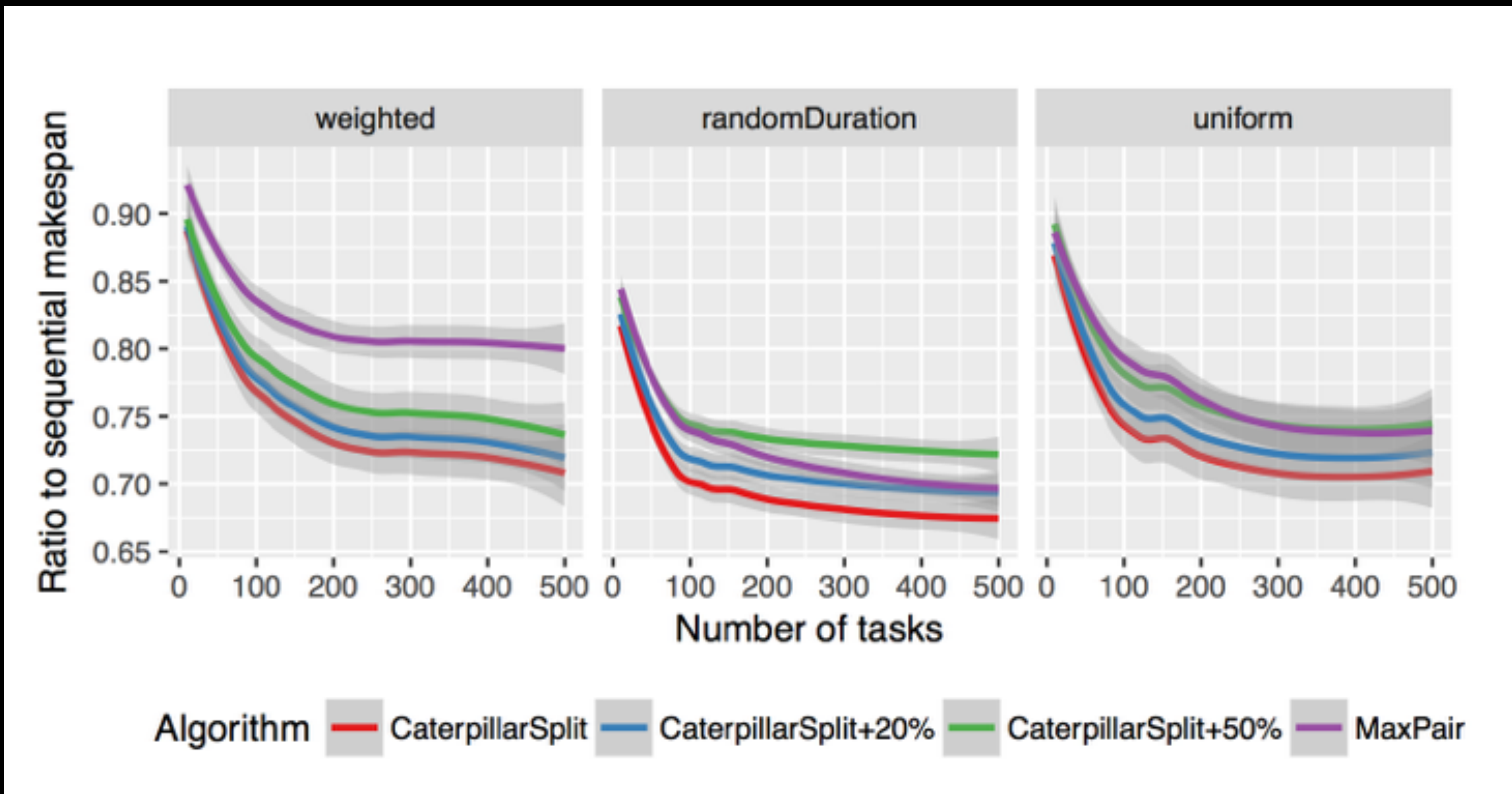
Preemption

- Linear Programming solve optimally preemptive co-scheduling minimizing makespan
- Optimal solution produces a co-execution graph
- Graph-based Caterpillar algorithm to reduce the number of preemptions





Preemption



Eyraud-Dubois, L.and Bentes, C.,(2020). Algorithms for Preemptive Co-scheduling of Kernels on GPUs. Submitted to HiPC.

Concurrency: 20 to 30% reduction

Preemption: further 10 to12% reduction

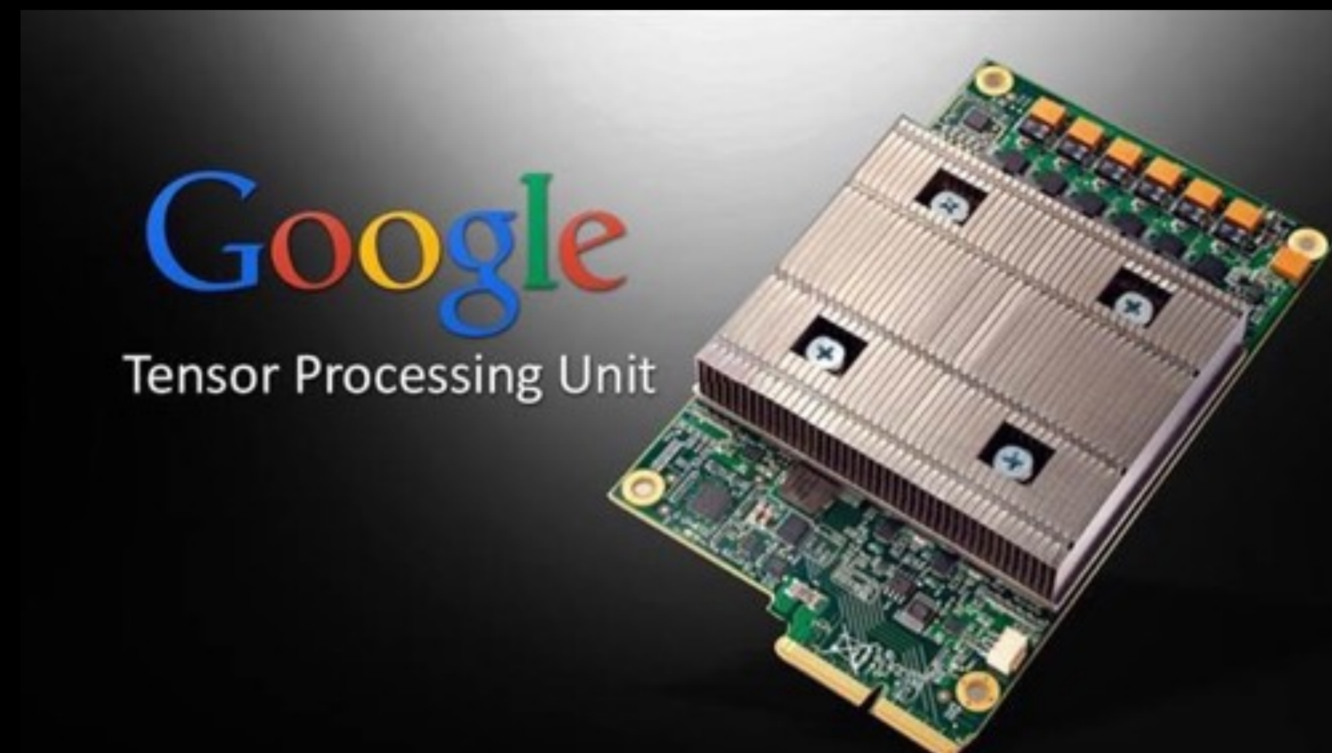
Concluding

- New era of heterogeneous architecture
- AI Big bang - GPUs play a fundamental role
- Exploit GPU to the fullest - concurrent kernel execution
- Hardware support still rudimentary
- Co-scheduling is challenging
- GPU virtualization - more competition and need for preemption

Thank you

cris@eng.uerj.br

Specialization



Kernel interference

- The number of blocks per grid is the most relevant feature to define if the kernels will execute concurrently
- The second most important feature depends on the GPU architecture:
  - For the GPU with more resources - the number of registers
  - For the GPU with less SM resources, but the same amount of registers - the number of threads per block