<Slides download>
http://www.pf.is.s.u-tokyo.ac.jp/class.html

Advanced Operating Systems

#9

Shinpei Kato
Associate Professor

Department of Computer Science
Graduate School of Information Science and Technology
The University of Tokyo

Course Plan

- Multi-core Resource Management
- Many-core Resource Management
- GPU Resource Management
- Virtual Machines
- Distributed File Systems
- High-performance Networking
- Memory Management
- Network on a Chip
- Embedded Real-time OS
- Device Drivers
- Linux Kernel

Schedule

1. 2018.9.28 Introduction + Linux Kernel (Kato)
2. 2018.10.5 Linux Kernel (Chishiro)
3. 2018.10.12 Linux Kernel (Kato)
4. 2018.10.19 Linux Kernel (Kato)
5. 2018.10.26 Linux Kernel (Kato)
6. 2018.11.2 Advanced Research (Chishiro)
7. 2018.11.9 Advanced Research (Chishiro)
8. 2018.11.16 (No Class)
9. 2018.11.23 (Holiday)
10. 2018.11.30 Advanced Research (Chishiro)
11. 2018.12.7 Advanced Research (Kato)
12. 2019.12.14 (No Class)
13. 2018.12.21 Advanced Research (Kato)
14. 2019.1.11 (No Class)
15. 2019.1.18 10:25-12:10 Linux Kernel
16. 2019.1.25 13:00-14:45 Linux Kernel

GPU Resource Management

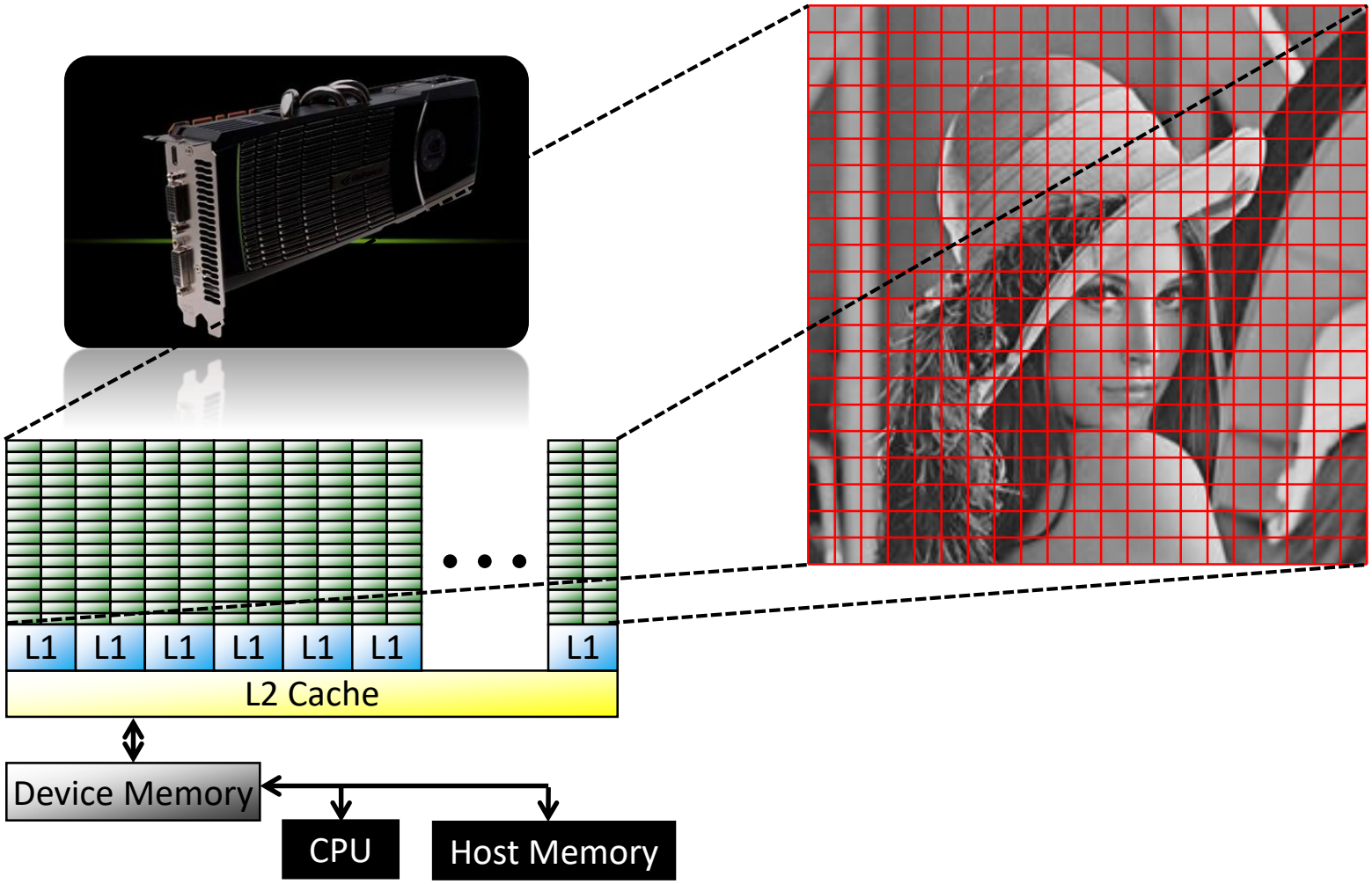
Abstracting GPUs as Compute Devices

/* The case for TimeGraph and Gdev */

Acknowledgement:

Linux Nouveau Community

Graphics Processing Unit (GPU)



NVIDIA GPU Trend

- GPU for HPC by NVIDIA
- Kepler > Maxwell > Pascal > Volta
- Mfg. process of Pascal 16nm FinFET+
- Mfg. process of Volta 12nm FFN
- Newest breakout features Tensor Core

Performance increase by shrinking die size

- Up to 12x faster for deep learning applications on V100 than P100 (FP16).
- DP(FP64), SP (FP32) is 1.5x performance improvement over P100.
- P100 was 3x faster than Kepler on FP64.
- Process shrink 16nm FinFET+ -> 12nm FFN.
- Retaining same power consumption (TDP 300W) for both P100 & V100.

Performance Comparison

GPU PERFORMANCE COMPARISON

	P100	V100	Ratio
DL Training	10 TFLOPS	120 TFLOPS	12x
DL Inferencing	21 TFLOPS	120 TFLOPS	6x
FP64/FP32	5/10 TFLOPS	7.5/15 TFLOPS	1.5x
HBM2 Bandwidth	720 GB/s	900 GB/s	1.2x
STREAM Triad Perf	557 GB/s	855 GB/s	1.5x
NVLink Bandwidth	160 GB/s	300 GB/s	1.9x
L2 Cache	4 MB	6 MB	1.5x
L1 Caches	1.3 MB	10 MB	7.7x

Tesla Product	Tesla K40	Tesla M40	Tesla P100	Tesla V100
GPU	GK110 (Kepler)	GM200 (Maxwell)	GP100 (Pascal)	GV100 (Volta)
SMs	15	24	56	80
TPCs	15	24	28	40
FP32 Cores / SM	192	128	64	64
FP32 Cores / GPU	2880	3072	3584	5120
FP64 Cores / SM	64	4	32	32
FP64 Cores / GPU	960	96	1792	2560
Tensor Cores / SM	NA	NA	NA	8
Tensor Cores / GPU	NA	NA	NA	640
GPU Boost Clock	810/875 MHz	1114 MHz	1480 MHz	1455 MHz
Peak FP32 TFLOP/s*	5.04	6.8	10.6	15
Peak FP64 TFLOP/s*	1.68	2.1	5.3	7.5
Peak Tensor Core TFLOP/s*	NA	NA	NA	120
Texture Units	240	192	224	320
Memory Interface	384-bit GDDR5	384-bit GDDR5	4096-bit HBM2	4096-bit HBM2
Memory Size	Up to 12 GB	Up to 24 GB	16 GB	16 GB
L2 Cache Size	1536 KB	3072 KB	4096 KB	6144 KB
Shared Memory Size / SM	16 KB/32 KB/48 KB	96 KB	64 KB	Configurable up to 96 KB
Register File Size / SM	256 KB	256 KB	256 KB	256KB
Register File Size / GPU	3840 KB	6144 KB	14336 KB	20480 KB
TDP	235 Watts	250 Watts	300 Watts	300 Watts
Transistors	7.1 billion	8 billion	15.3 billion	21.1 billion
GPU Die Size	551 mm^2	601 mm^2	610 mm^2	815 mm^2
Manufacturing Process	28 nm	28 nm	16 nm FinFET+	12 nm FFN

Table 1. Tesla V100 Compared to Prior Generation Tesla Accelerators. (* Peak TFLOP/s rates are based on GPU Boost clock.)

NVIDIA DRIVE PX2

- On-board AI engine
- The AI car computer for autonomous driving
- Fuse data from multiple cameras, as well as LiDAR, radar, and ultrasonic sensors.
- Built to support ASIL-D
- 4 product family variants

Past cards performance change

GPU PERFORMANCE COMPARISON

	P100	M40	K40
Double Precision TFlop/s	5.3	0.2	1.4
Single Precision TFlop/s	10.6	7.0	4.3
Half Precision Tflop/s	21.2	NA	NA
Memory Bandwidth (GB/s)	720	288	288
Memory Size	16GB	12GB, 24GB	12GB

Jetson: the low power embedded platform

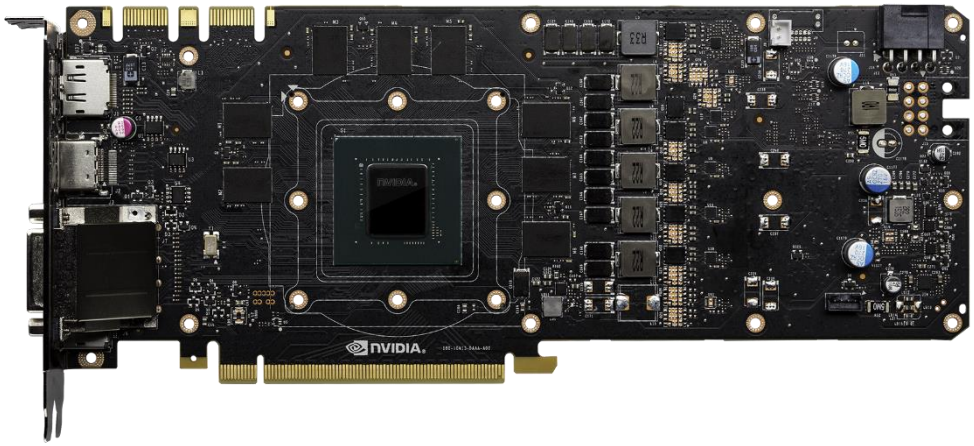
- Latest TX2 offers 2x performance of the predecessor.
- Tegra X2 is ARM Cortex-A57 (2GHz 4 core) + NVIDIA Denver2 (2GHz 2 core).
- Can put 24 TX2 in 1U box!
- Tegra X1 is on the “hard to buy” Nintendo Switch.

For PC/Games(non critical)

- Pascal based GeForce GTX 1080 released on June 2016.
- Fast but memory bandwidth & amount less than Titan-X.
- Water cooling as well as overlocking variants are available.

GeForce GTX 1080

- GM200 → GP204 core change (Maxwell/Pascal)
- 16FinFET+ is more expensive
- GDDR5X is more expensive
- Already more advanced 1080TI is out



GeForce GTX 1080 Ti

- Premium version of the GeForce GTX 1080
- 8GB → 11GB
- 8.2Gflops → 10.6Gflops FP
- 2560 cores → 3584 cores
- Same core, different world
- 30%+ price gap
- Shortage in supply

Titan-X (Pascal : Maxwell)

Spec comparison

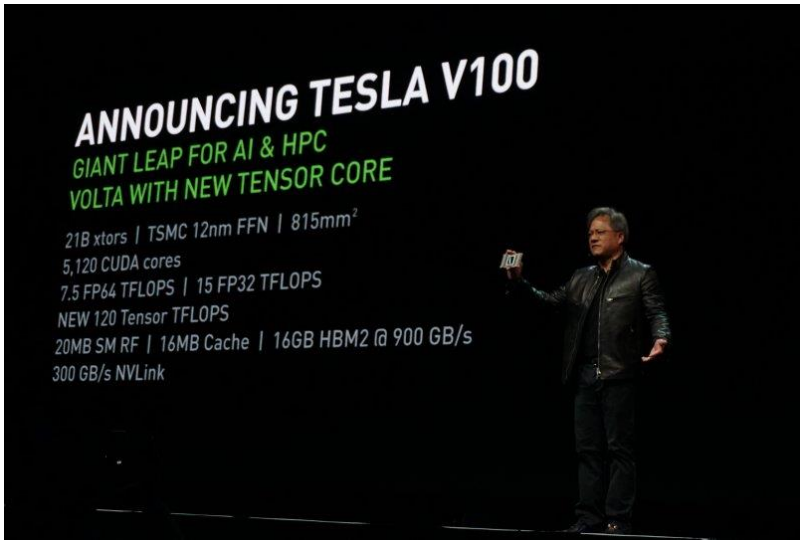
• # of cores	3584	3072
• Core clock	1417MHz	1000MHz
• TFLOPs (FMA)	11 TF	6.1 TF
• Mem clock	10Gbps	7Gbps
• FP64	1/32	1/32
• FP16 (Native)	1/64	N/A
• INT8	4:1	N/A
• TDP	250W	250W

Shoot in a foot

- Look at this timeline
- 2AUG2016 TITAN-X released
- 10MAR2017 1080ti released
- 6APR2017 TITAN-Xp released
  (Not released in Japan yet)
- Currently, TITAN-Xp is king of the consumer GPU
- Approx. US$2K in Japan

For high-end computing

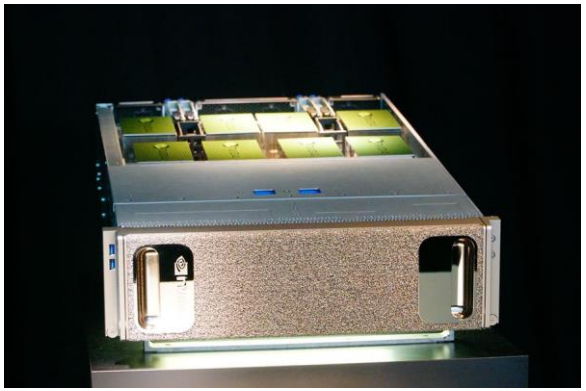
- Volta announced at NVIDIA GTC in May.
- For NVLink, NVIDIA does not tell you but needs mass-code change.
- SXM2 version(NVLink) needs special edition of NCCL libs to obtain an optimized perf. (Subscription req'd)



For workstations

- GP100
- Has display connector
- Can NVLink(up-to 2 cards)

Deep Learning DGX-1 Volta



- 8x V100
- Shipping now
- NVLink(SXM2)
- CPU-GPU NVLink ONLY by IBM Power9
- Previous P100 version is OLD already
- VERY EXPENSIVE

DGX-1 Pros/Cons

- Pros
  - NVIDIA brand
  - Better chassis design
  - Access to optimized libraries
- Cons
  - Too expensive (Low ROI, High TCO)
  - Warranty separately provided as subscription (Hostage)
  - Can not reconfigure*

TCO & ROI on AI

- DGX-1 Volta in Japan, approx. ¼ million USD plus 15% annual subscription.
- Fixed config.
- Smarter and bigger 3rd party solution with more robust system also available at lower cost, higher ROI.
- 2x GPU at same $!!
- Think/Act SMART!

And beyond…with OSS

- 1 system up-to 32 GPU (31 by CUDA)
- Single Root Complex!
- 2x performance of DGX-1 by single node and
  f DGX-1

WORLD FIRST & ONLY
16 GeForce also possible!

- Customisable CPU/RAM/STORAGE
- 120GPU in single rack

How many GPUs do you need?

- Currently, keeping x16 performance, without crossing CPUs, the maximum # of GPUs supported is 16.
- 8 is NOT ENOUGH in many cases.
- TESLA is not always an answer.
- US vendor made 16 GeForce recognized under Linux and running stable.
- Connecting 2 boxes can make 32 GPUs, but CUDA can only see up-to 31 GPUs.

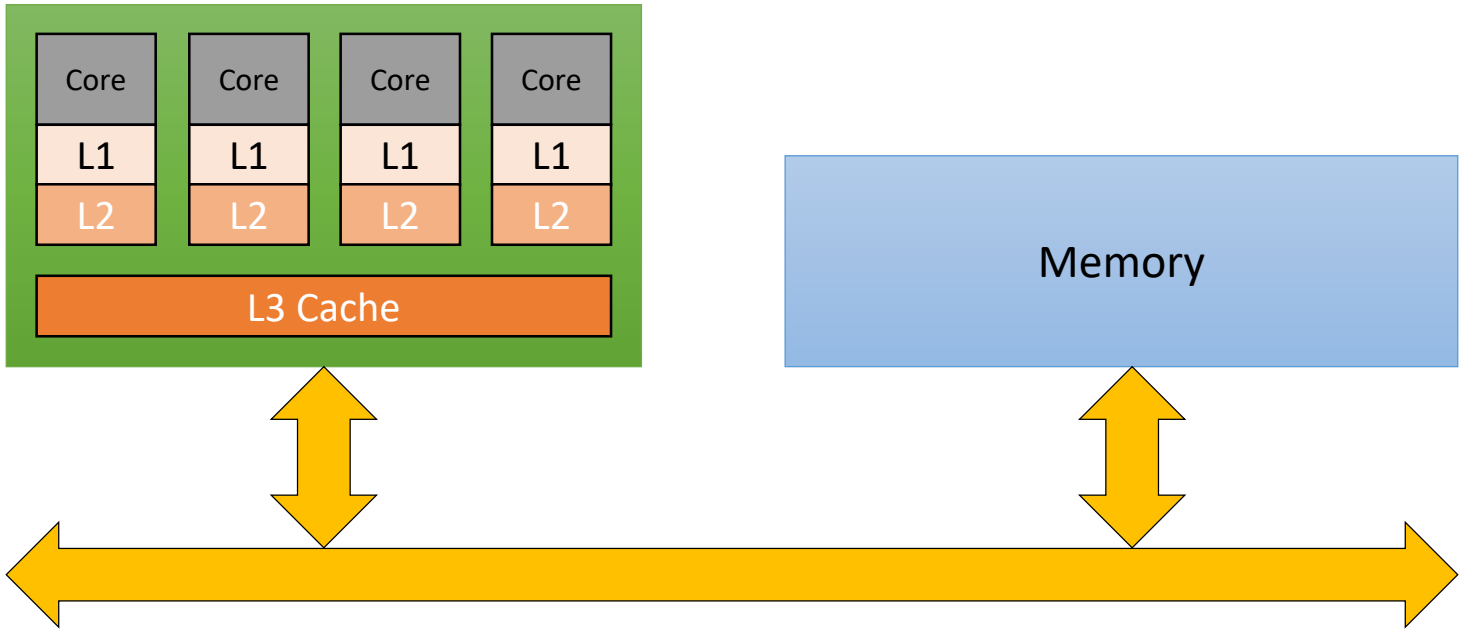
Problem solving

- Important point using GPUs
  - Performance decrease due to heating, and under current.
  - When Multi GPU environment, other GPU too will suffer throttling.
  - Often, users do not notice throttling and ust wonder why my code is slow!
- How to solve this?
  - Deployment of water block and strengthen the air cooling
  - Utilizing proper cables and grounding
  - Choose proper PSU and power source

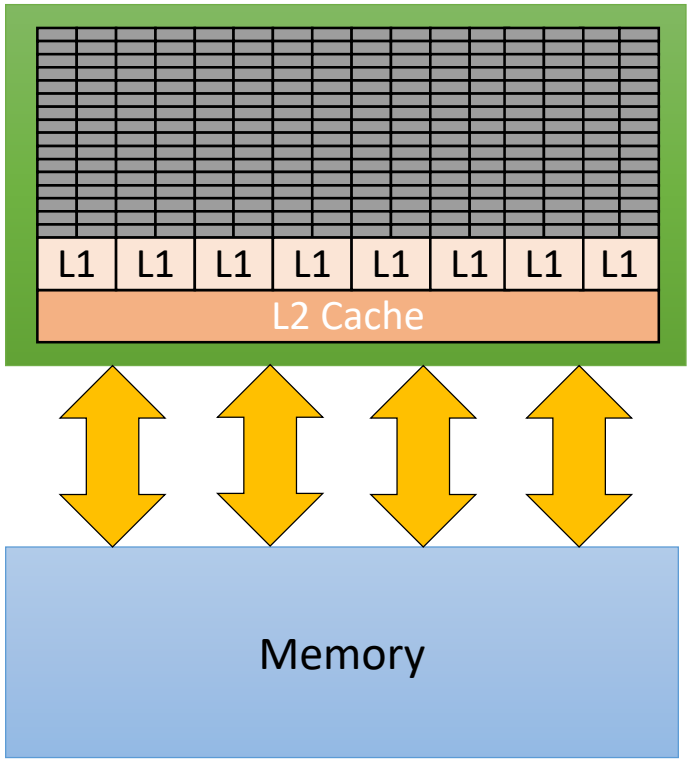
Theme for future movements

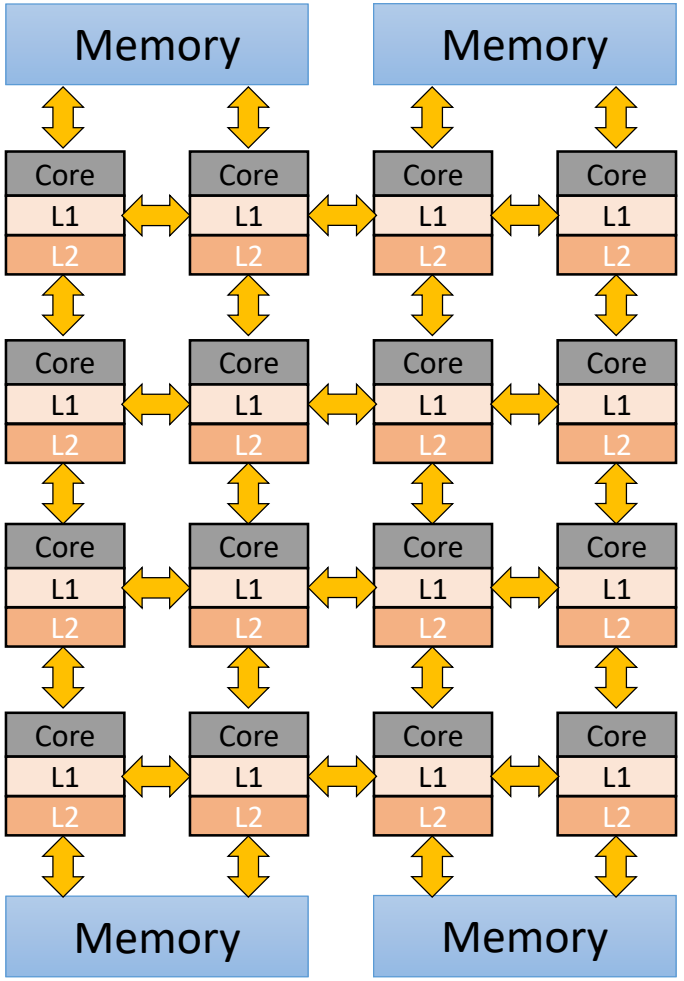
- Die size shrink will bring
  - Increase # of cores
  - Increase performance
  - Efficient energy consumption
- Deployment of faster memory (HBM2)
- Implementation of FP4/FP8
- Moving up to PCIe gen4/NVLink2
- Popularisation of NCCL2 and optimisation with NVLink
- Will be more expensive
- China to catch up to replace US technology

Multi-core

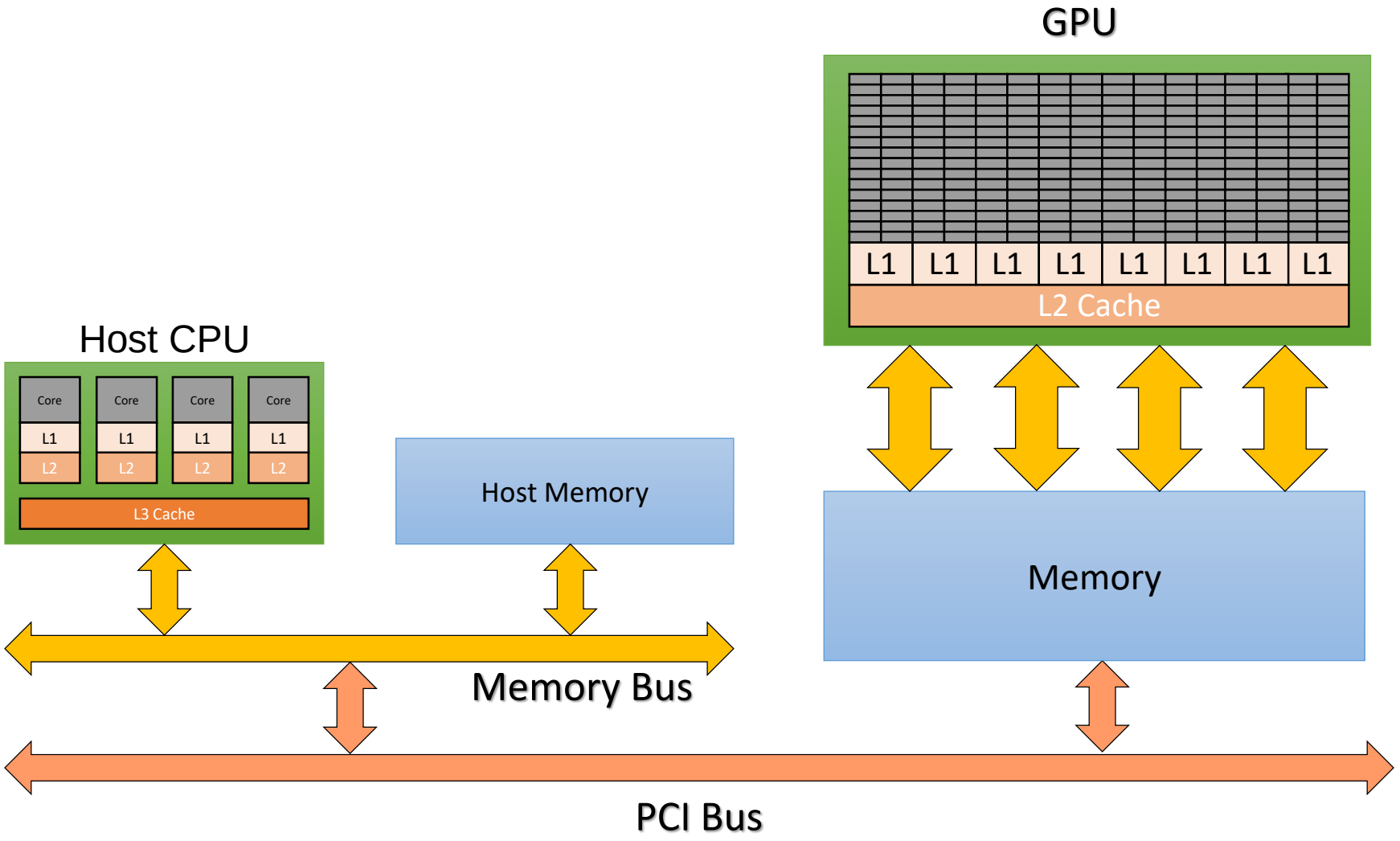


GPU vs. Many-core





GPU as Compute Device



GPGPU **G**eneral-**P**urpose computation on **G**raphics **P**rocessing **U**nits

NVIDIA CUDA

C

C++

Java

Tesla

2008

250 cores

Fermi

2010

500 cores

Kepler

2012

3000 cores

Maxwell

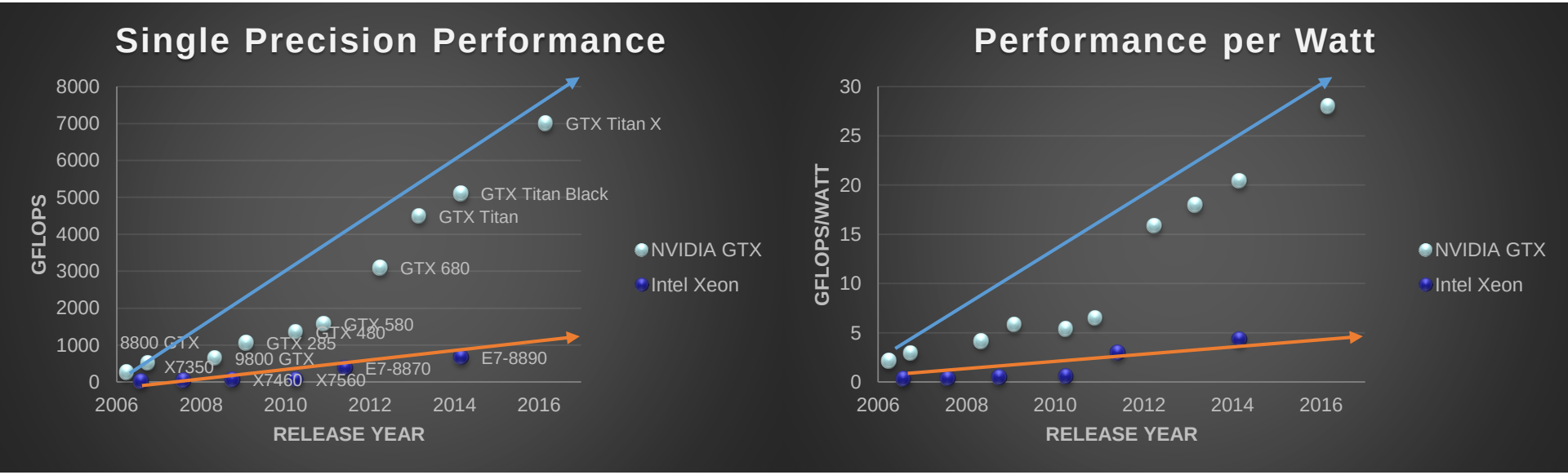
Pascal

2014

5000 cores

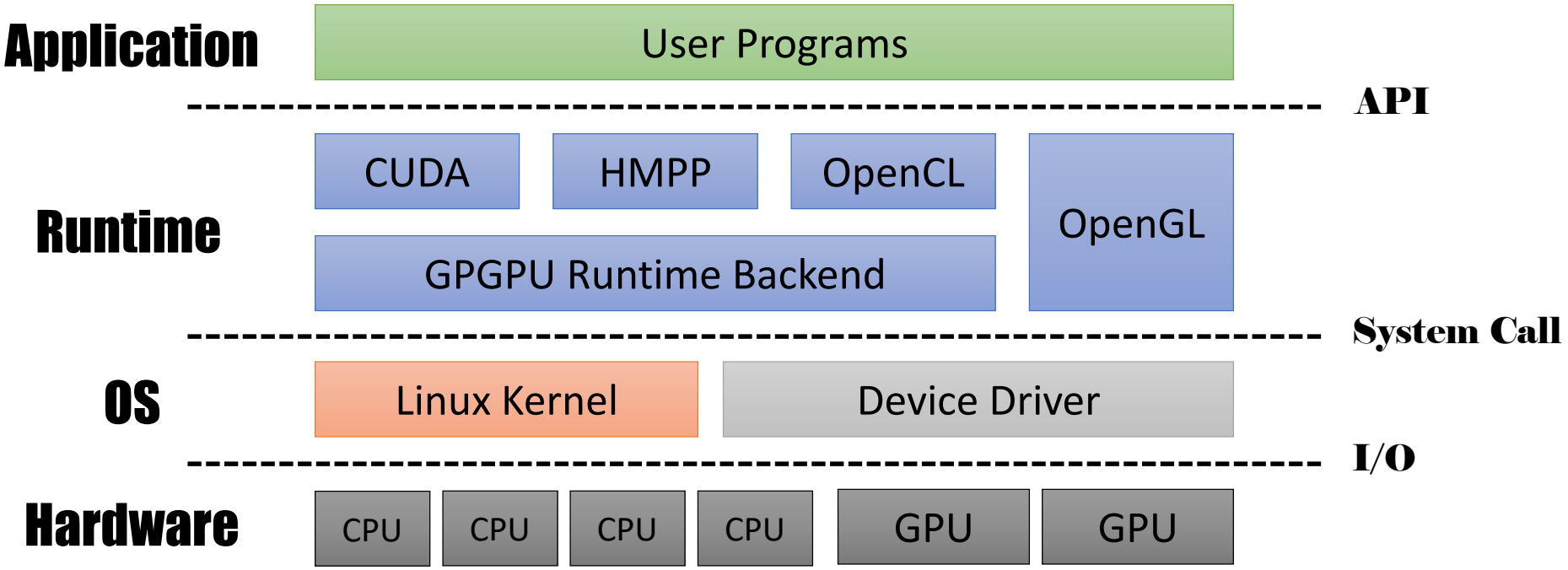
Low Power

GPGPU **G**eneral-**P**urpose computation on **G**raphics **P**rocessing **U**nits

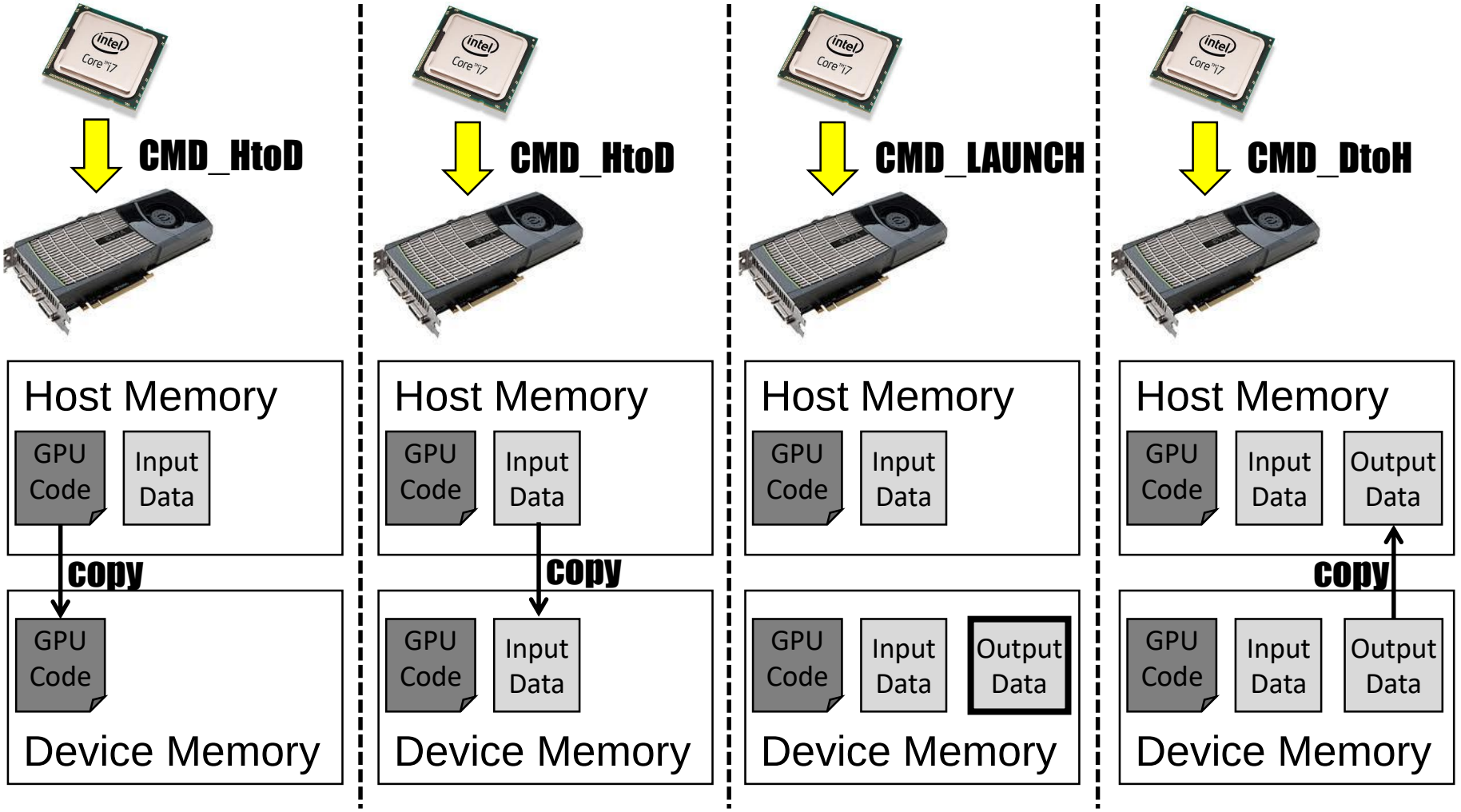


Trend on Performance and Power

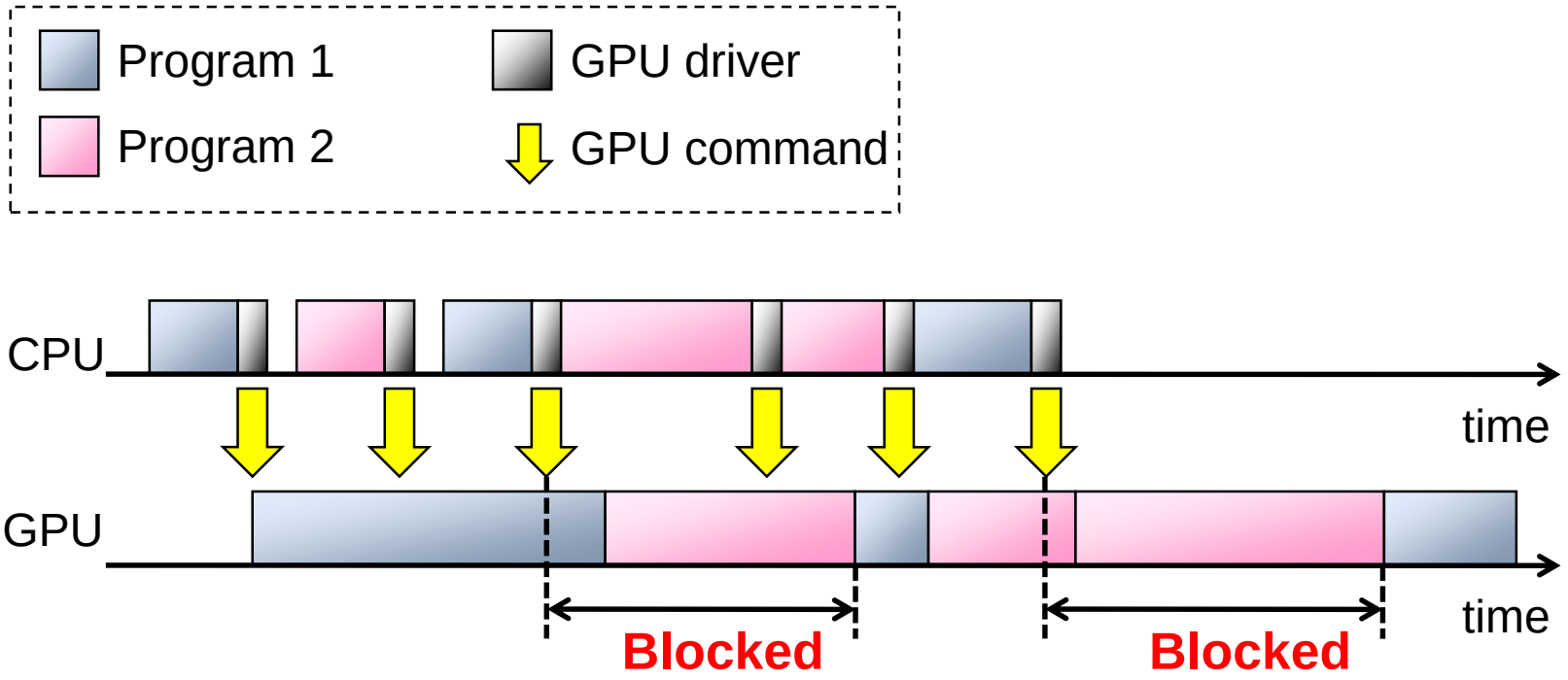
GPGPU Computing Stack

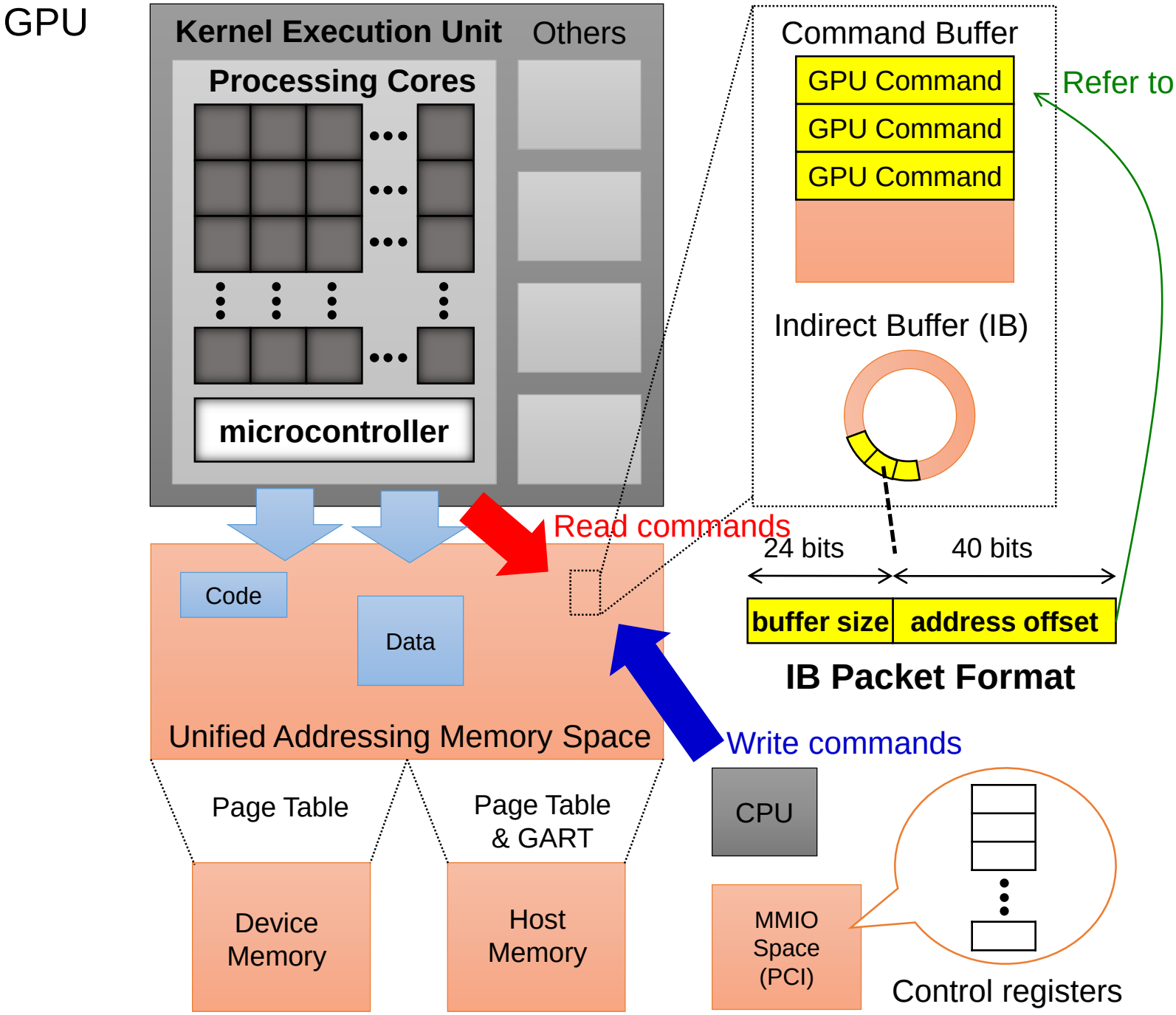


GPGPU Computing Model



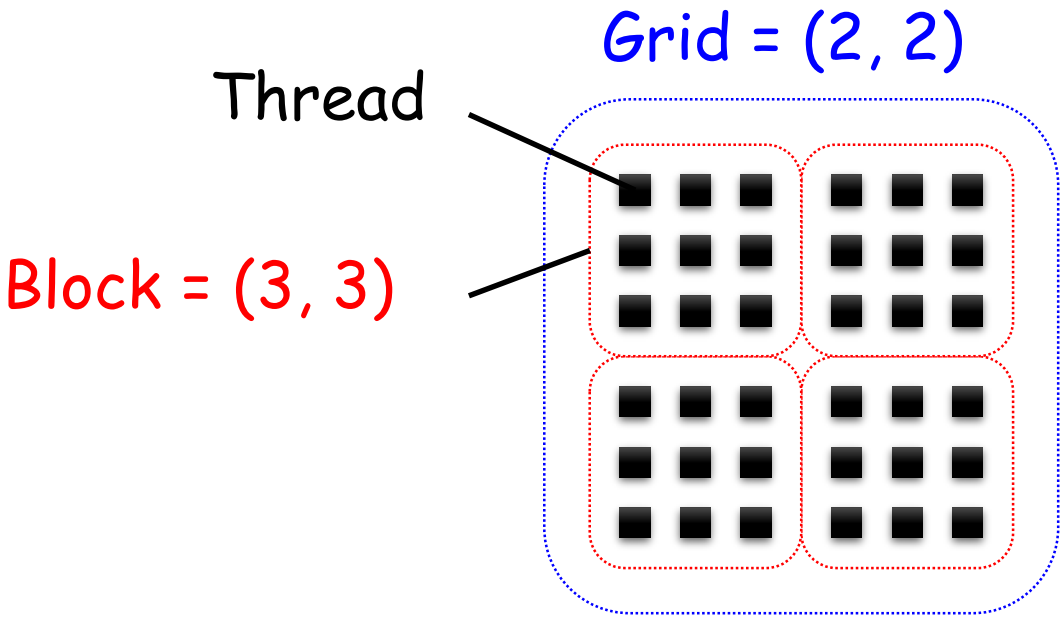
GPGPU Execution

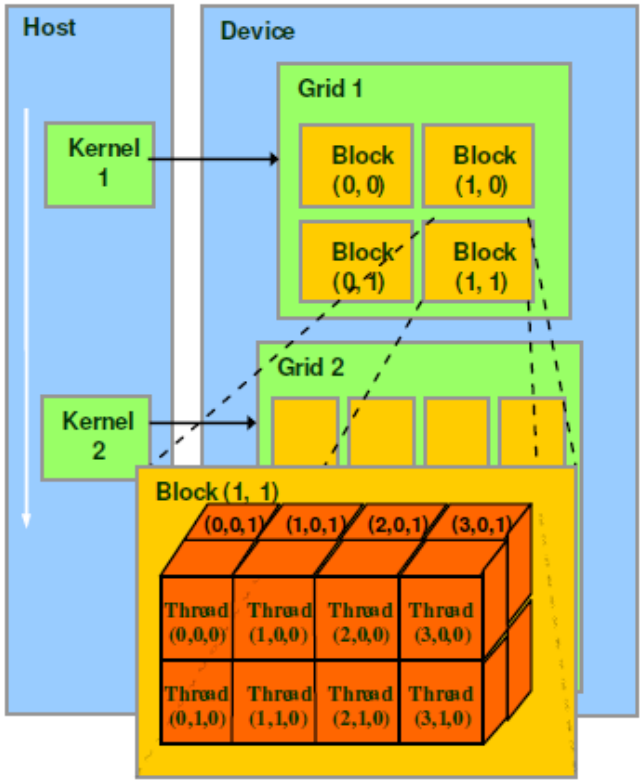



GPU
Kernel Execution Unit
Others
Processing Cores
microcontroller
Command Buffer
GPU Command
GPU Command
GPU Command
Refer to
Indirect Buffer (IB)
Read commands
24 bits
40 bits
buffer size
address offset
IB Packet Format
Code
Data
Unified Addressing Memory Space
Write commands
Page Table
Page Table
& GART
CPU
Device
Memory
Host
Memory
MMIO
Space
(PCI)
Control registers

CUDA

Compute **U**nified **D**evice **A**rchitecture





Abstracting computation by Grid/Thread/Block

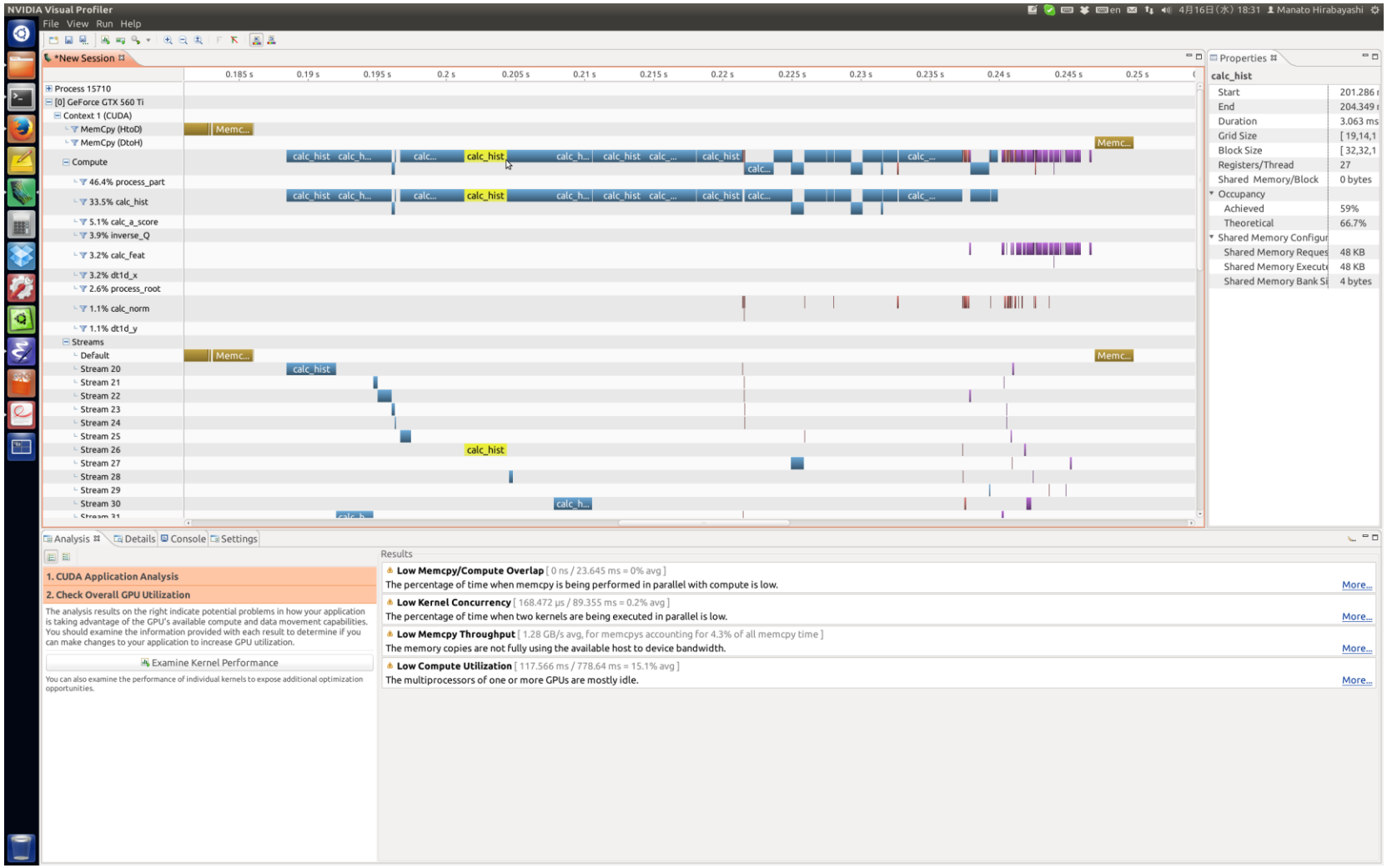
Programing Example of CUDA

```
void multiply(double *a, double *b, double *c, int n)
{
    double product = 0.0;
    int row = blockIdx.y * blockDim.y + threadIdx.y;
    int col = blockIdx.x * blockDim.x + threadIdx.x;
    int i, idx;

    for (i = 0; i < n; i++)
        product += a[row * n + i] * b[i * n + col];

    c[row * n + col] = product;
}
```

CUDA Profiler



State of the Art

- PTask: Operating System Abstractions To Manage GPUs as Compute Devices
  - Rossbach et. al., SOSP 2011
- TimeGraph: GPU Scheduling for Real-Time Multi-Tasking Environments
  - Kato et. al., USENIX ATC 2011
- Gdev: First-Class GPU Resource Management in the Operating System
  - Kato et. al., USENIX ATC 2012
- GPUvm: Why Not Virtualizing GPUs at the Hypervisor?
  - Suzuki et. al., USENIX ATC 2014
- GLoop: An Event-driven Runtime for Consolidating GPGPU Applications
  - Suzuki et. al., ACM SOCC 2017