CME 213, ME 339—Spring 2021

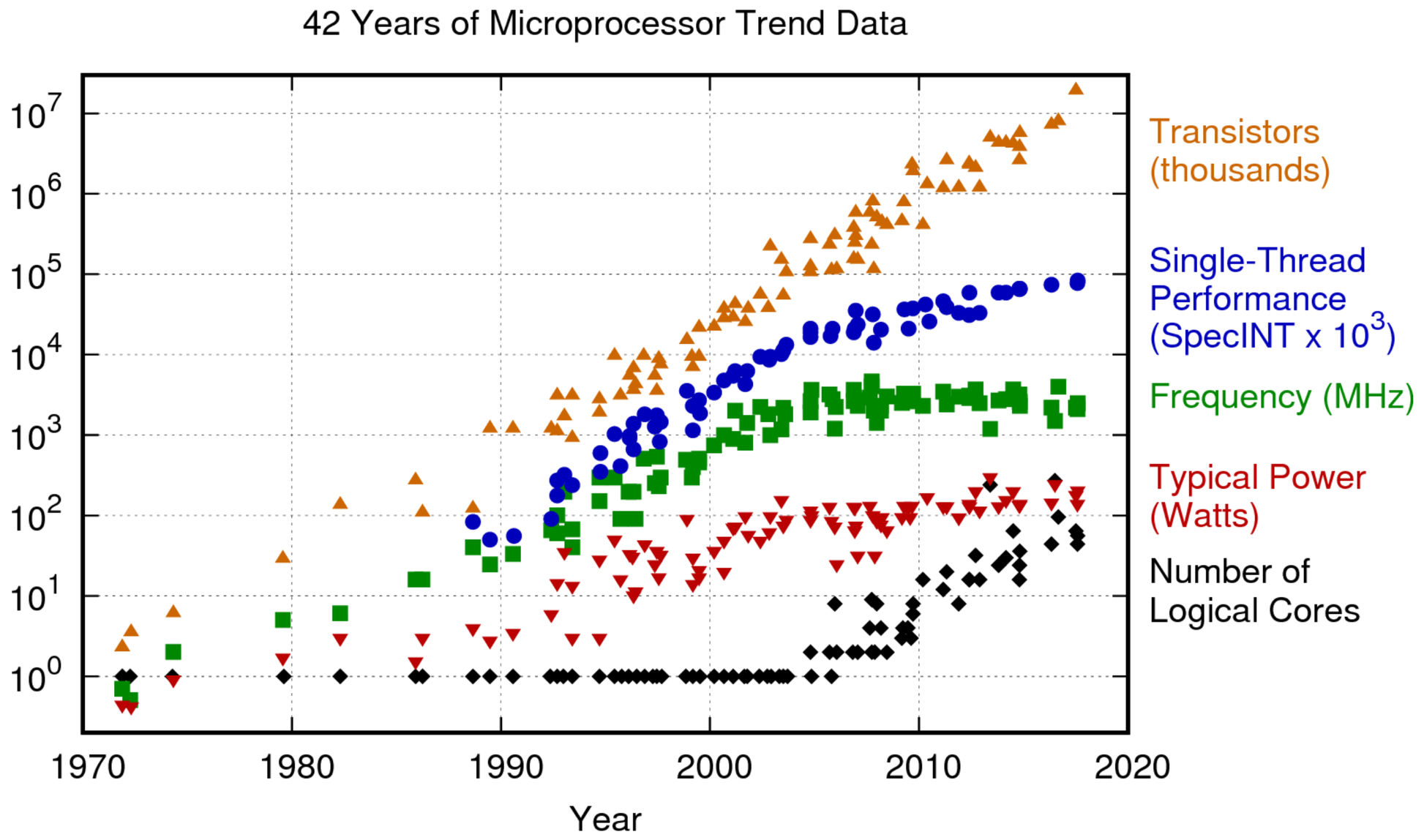
Eric Darve, ICME, Stanford

"If debugging is the process of removing bugs, then programming must be the process of putting them in." (Edsger W. Dijkstra)

CME 213 so far:

- C++ threads
- OpenMP: `for` loop and `task`
- Sorting algorithms on shared memory

Onwards to GPU computing!



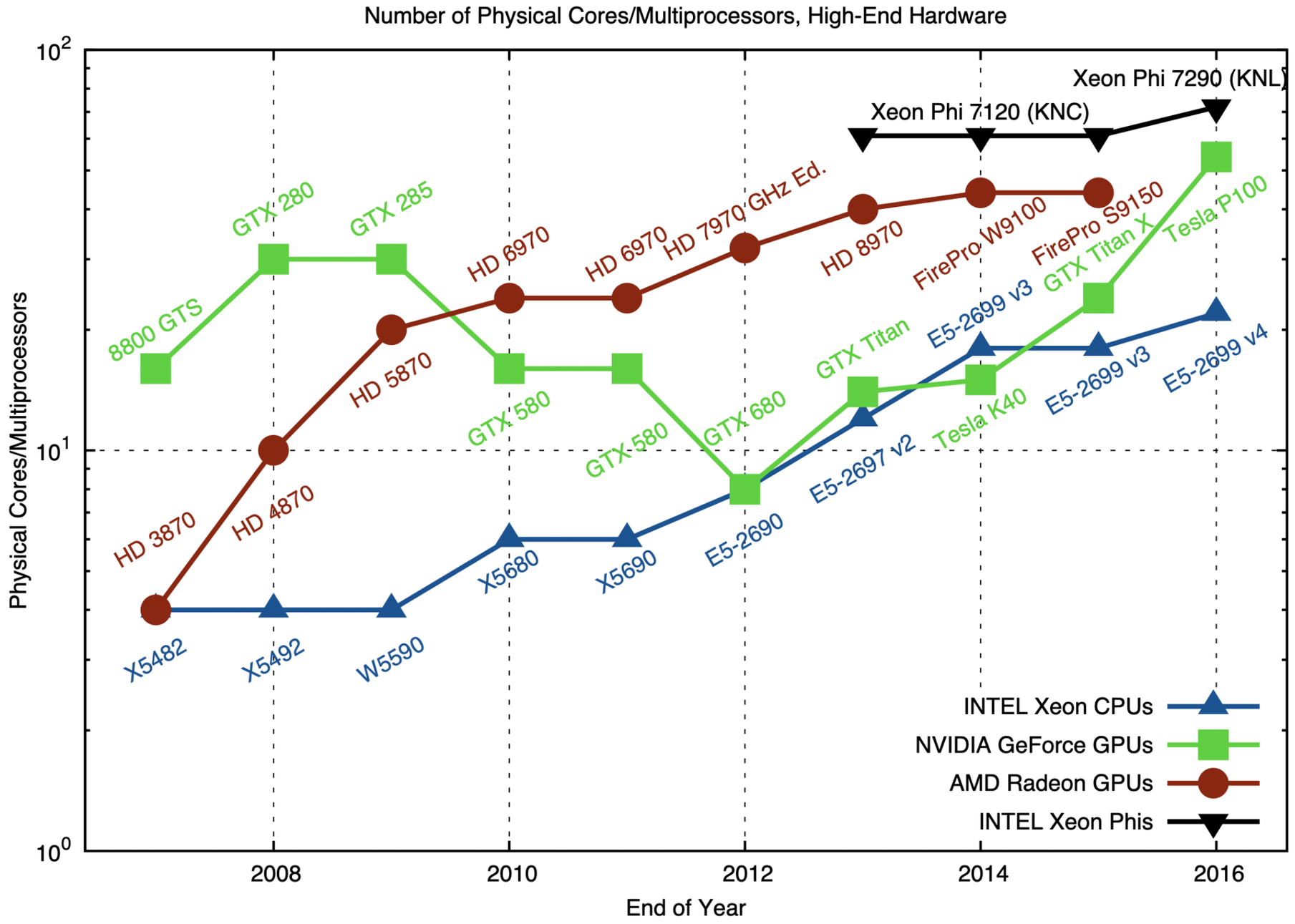
Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

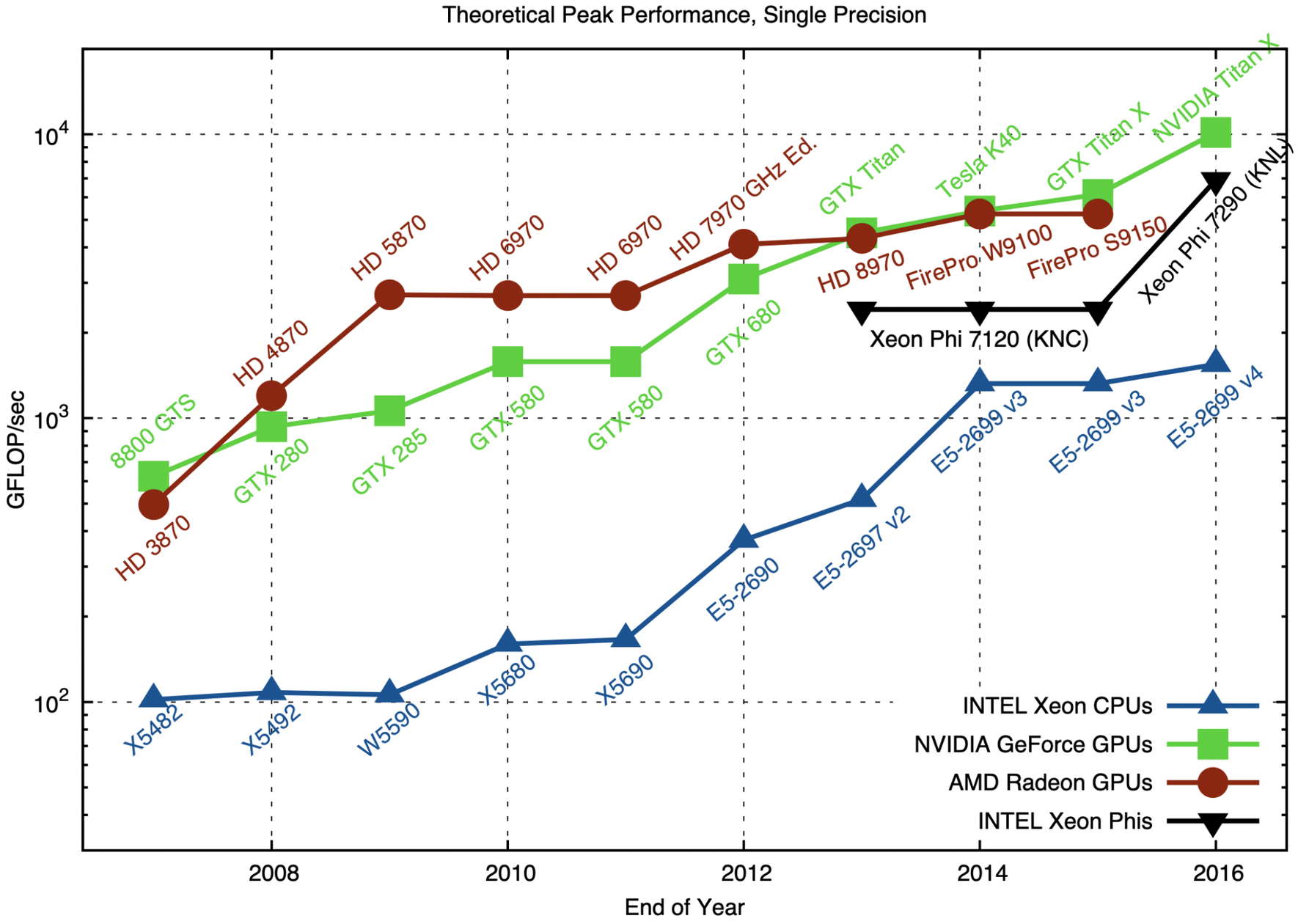
Reference

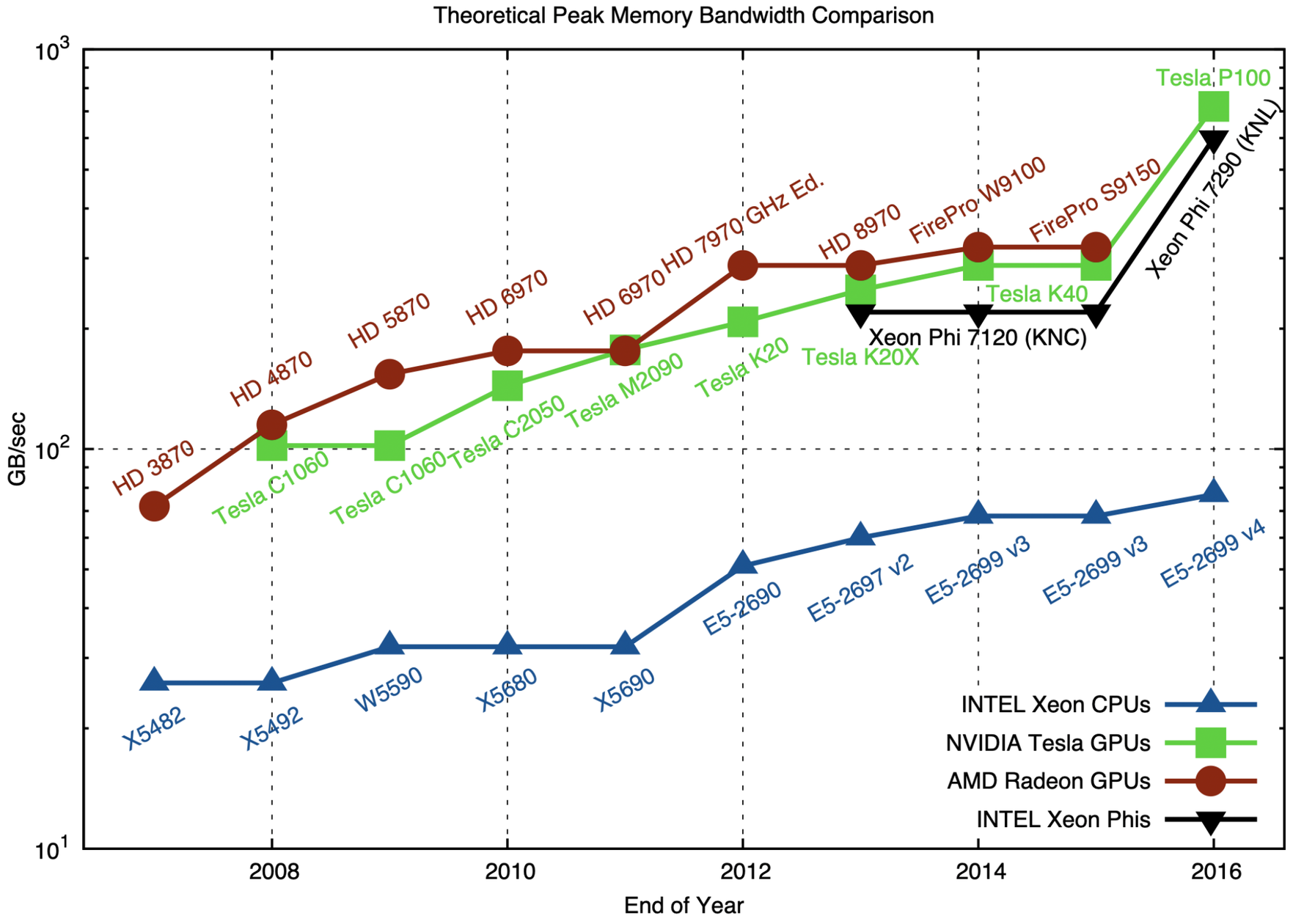
https://github.com/karlrupp/cpu-gpu-mic-comparison

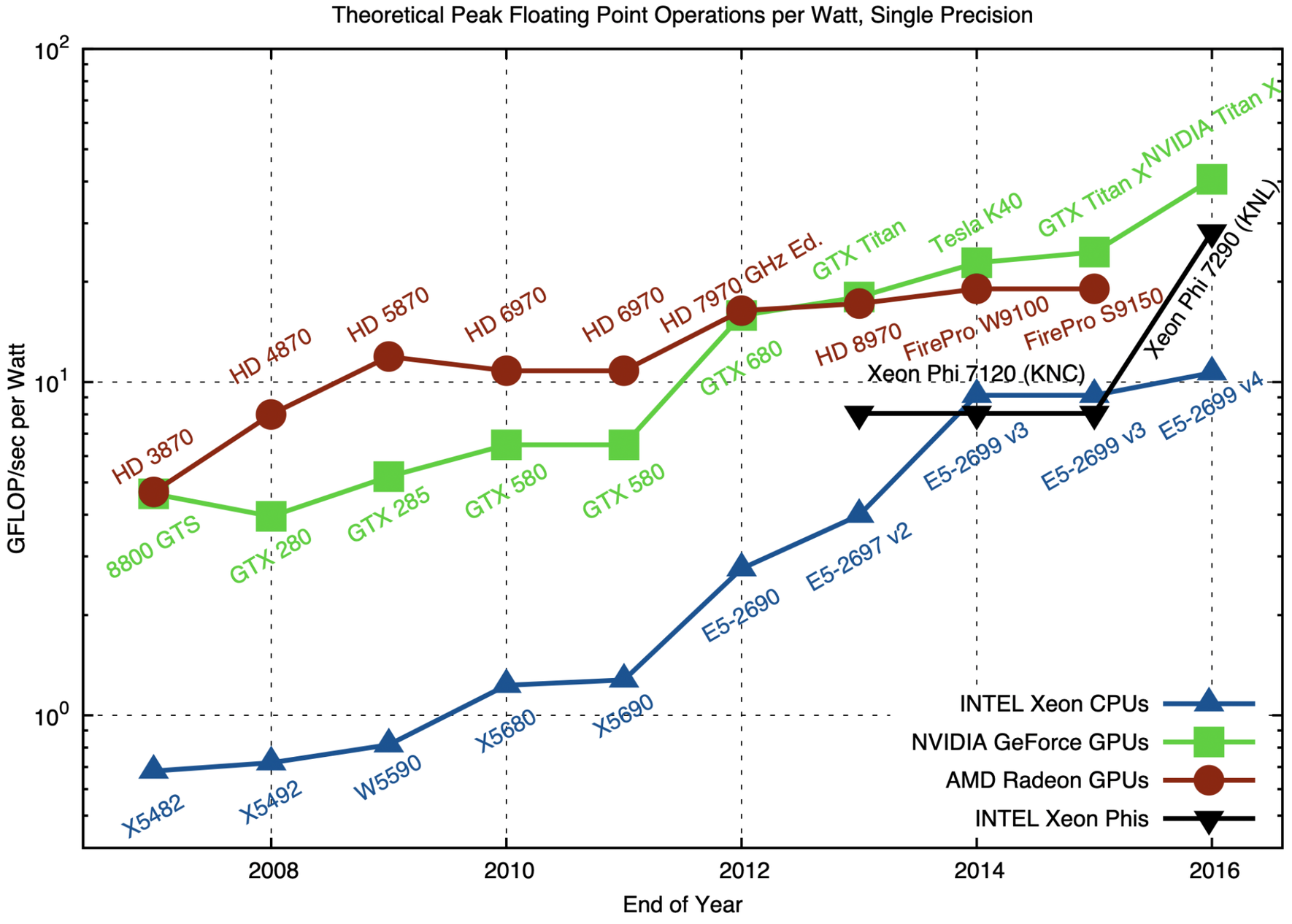
https://www.karlrupp.net/2013/06/cpu-gpu-and-mic-hardware-characteristics-over-time/

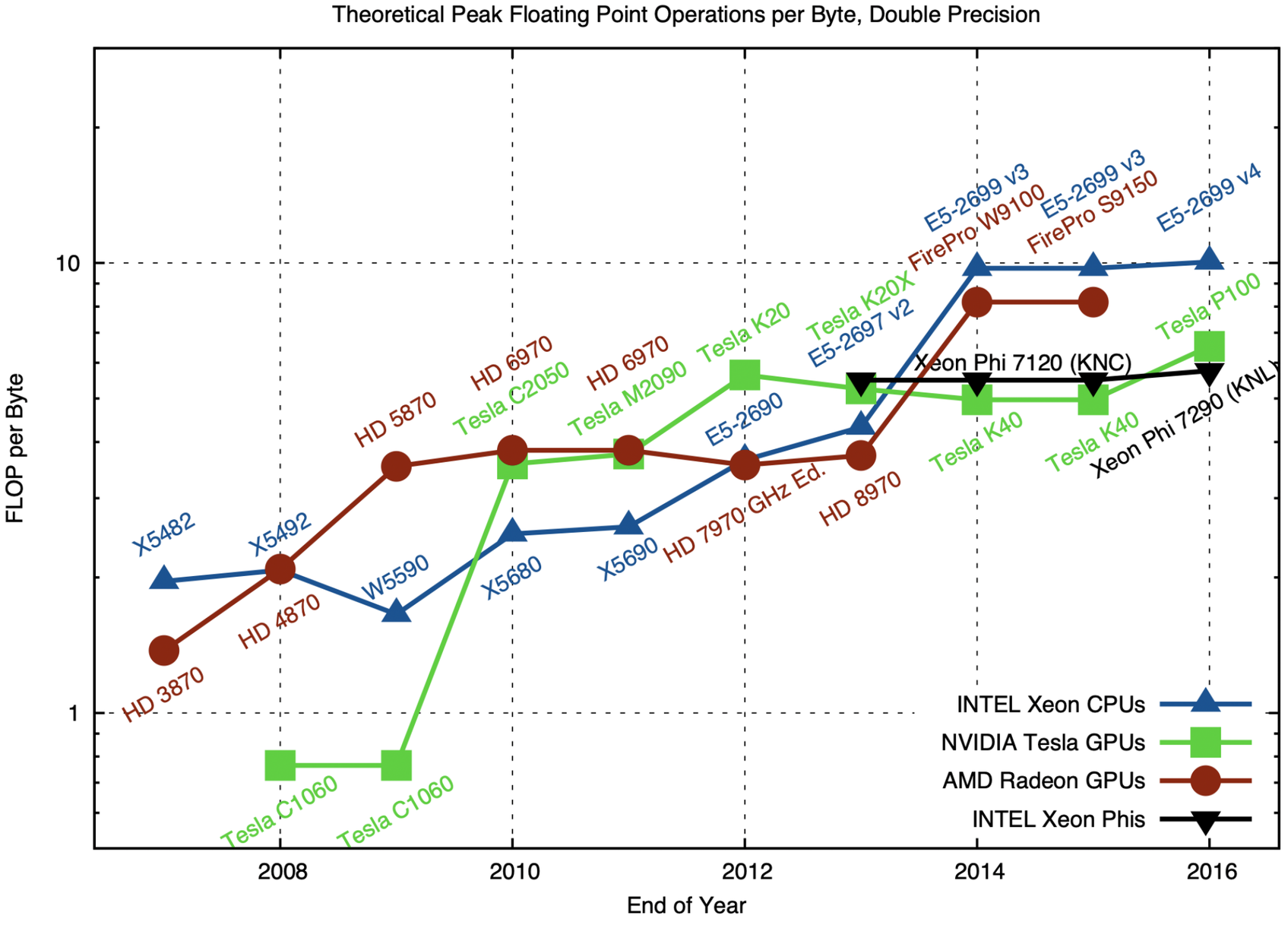
https://www.karlrupp.net/2015/06/40-years-of-microprocessor-trend-data/


Number of Physical Cores/Multiprocessors, High-End Hardware
Physical Cores/Multiprocessors
End of Year
10^2
10^1
10^0
2008
2010
2012
2014
2016
Xeon Phi 7290 (KNL)
Xeon Phi 7120 (KNC)
8800 GTS
GTX 280
GTX 285
GTX 580
GTX 580
GTX 680
GTX Titan
GTX Titan X
Tesla P100
Tesla K40
HD 3870
HD 4870
HD 5870
HD 6970
HD 6970
HD 7970 GHz Ed.
HD 8970
FirePro W9100
FirePro S9150
X5482
X5492
W5590
X5680
X5690
E5-2690
E5-2697 v2
E5-2699 v3
E5-2699 v3
E5-2699 v4
INTEL Xeon CPUs
NVIDIA GeForce GPUs
AMD Radeon GPUs
INTEL Xeon Phis


Theoretical Peak Performance, Single Precision
GFLOP/sec
10^4
10^3
10^2
2008
2010
2012
2014
2016
End of Year
8800 GTS
HD 3870
HD 4870
HD 5870
GTX 280
GTX 285
HD 6970
GTX 580
HD 6970
GTX 580
HD 7970 GHz Ed.
GTX 680
GTX Titan
HD 8970
Tesla K40
FirePro W9100
GTX Titan X
FirePro S9150
NVIDIA Titan X
Xeon Phi 7120 (KNC)
Xeon Phi 7290 (KNL)
X5482
X5492
W5590
X5680
X5690
E5-2690
E5-2697 v2
E5-2699 v3
E5-2699 v3
E5-2699 v4
INTEL Xeon CPUs
NVIDIA GeForce GPUs
AMD Radeon GPUs
INTEL Xeon Phis


Theoretical Peak Memory Bandwidth Comparison
GB/sec
End of Year
2008
2010
2012
2014
2016
HD 3870
HD 4870
HD 5870
HD 6970
HD 6970
HD 7970 GHz Ed.
HD 8970
FirePro W9100
FirePro S9150
Tesla P100
Tesla C1060
Tesla C1060
Tesla C2050
Tesla M2090
Tesla K20
Tesla K20X
Tesla K40
Xeon Phi 7120 (KNC)
Xeon Phi 7290 (KNL)
X5482
X5492
W5590
X5680
X5690
E5-2690
E5-2697 v2
E5-2699 v3
E5-2699 v3
E5-2699 v4
INTEL Xeon CPUs
NVIDIA Tesla GPUs
AMD Radeon GPUs
INTEL Xeon Phis


Theoretical Peak Floating Point Operations per Watt, Single Precision
GFLOP/sec per Watt
10^2
10^1
10^0
2008
2010
2012
2014
2016
End of Year
HD 3870
HD 4870
HD 5870
HD 6970
HD 6970
HD 7970 GHz Ed.
HD 8970
FirePro W9100
FirePro S9150
8800 GTS
GTX 280
GTX 285
GTX 580
GTX 580
GTX 680
GTX Titan
Tesla K40
GTX Titan X
NVIDIA Titan X
Xeon Phi 7120 (KNC)
Xeon Phi 7290 (KNL)
X5482
X5492
W5590
X5680
X5690
E5-2690
E5-2697 v2
E5-2699 v3
E5-2699 v3
E5-2699 v4
INTEL Xeon CPUs
NVIDIA GeForce GPUs
AMD Radeon GPUs
INTEL Xeon Phis

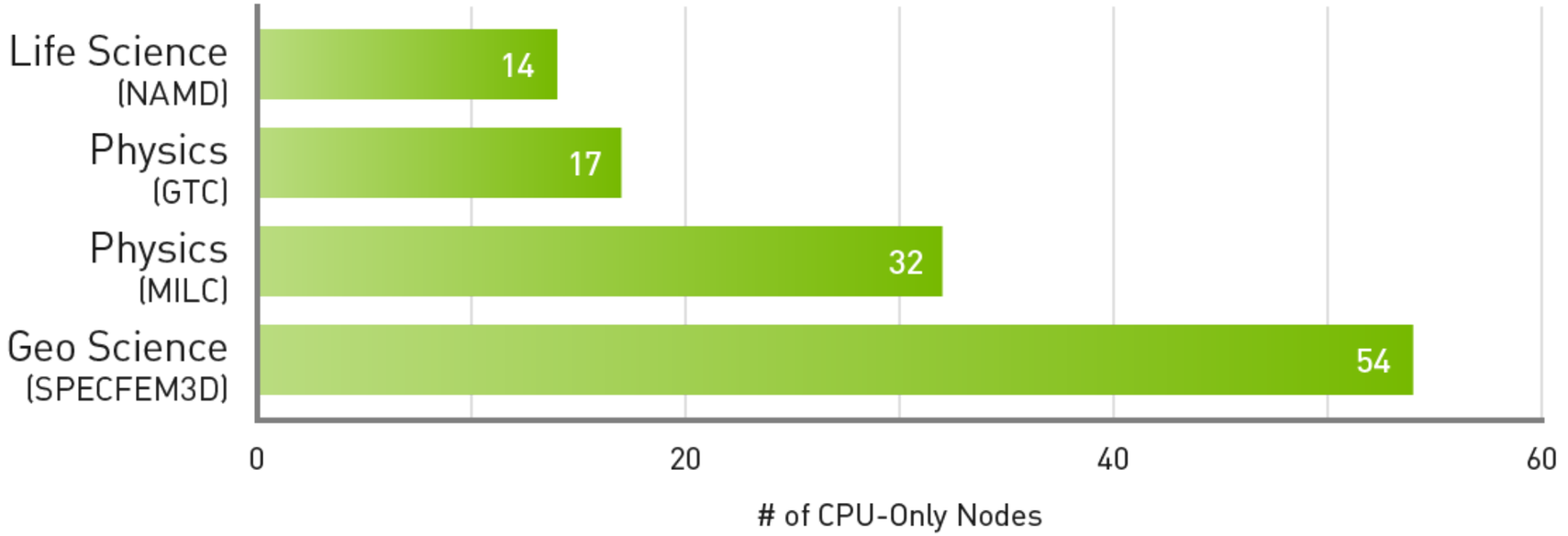

Theoretical Peak Floating Point Operations per Byte, Double Precision
FLOP per Byte
10
1
2008
2010
2012
2014
2016
End of Year
X5482
X5492
W5590
X5680
X5690
E5-2690
E5-2699 v3
E5-2699 v4
HD 3870
HD 4870
HD 5870
HD 6970
HD 7970 GHz Ed.
HD 8970
FirePro W9100
FirePro S9150
Tesla C1060
Tesla C2050
Tesla M2090
Tesla K20
Tesla K20X
E5-2697 v2
Tesla K40
Tesla P100
Xeon Phi 7120 (KNC)
Xeon Phi 7290 (KNL)
INTEL Xeon CPUs
NVIDIA Tesla GPUs
AMD Radeon GPUs
INTEL Xeon Phis

Example: Volta V100

- 8.2 teraflops double-precision performance
- 16.4 teraflops single-precision performance
- 130 teraflops for tensor (deep learning)
- 1134 GB/sec memory bandwidth
- 250 Watts power

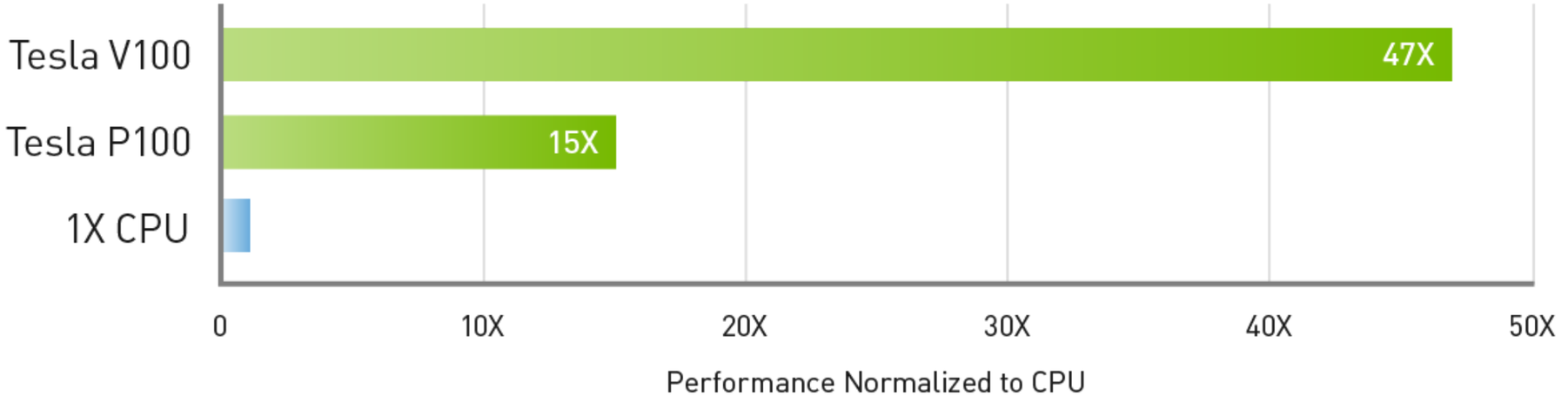
1 GPU Node Replaces Up To 54 CPU Nodes

Node Replacement: HPC Mixed Workload



CPU Server: Dual Xeon Gold 6140@2.30GHz, GPU Servers: same CPU server w/ 4x V100 PCIe | CUDA Version: CUDA 9.x| Dataset: NAMD (STMV), GTC (mpi#proc.in), MILC (APEX Medium), SPECFEM3D (four_material_simple_model) | To arrive at CPU node equivalence, we use measured benchmark with up to 8 CPU nodes. Then we use linear scaling to scale beyond 8 nodes.

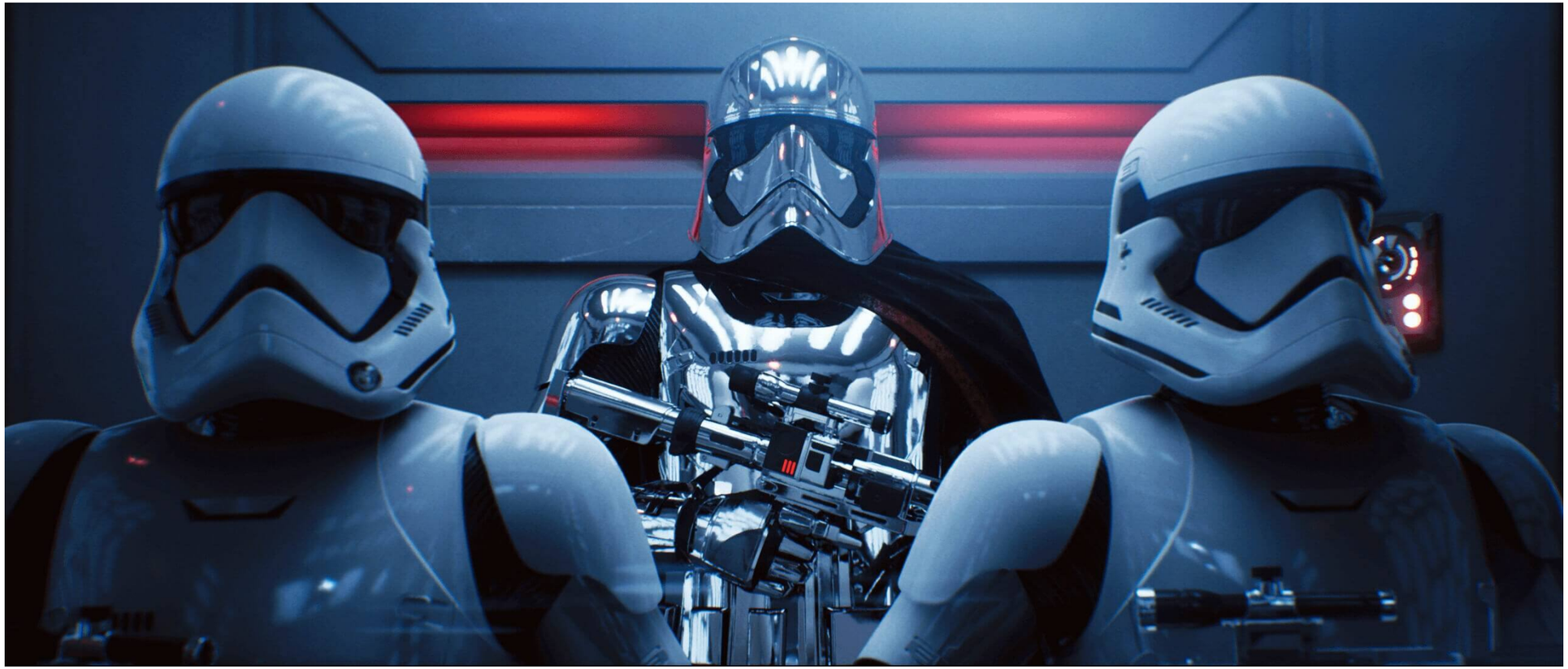
47X Higher Throughput Than CPU Server on Deep Learning Inference



Workload: ResNet-50 | CPU: 1X Xeon E5-2690v4 @ 2.6 GHz | GPU: Add 1X Tesla P100 or V100

What is the technology behind GPU processors?

It’s started with 3D graphics



GPUs were designed to perform the same instruction on a large amount of data

Graphics processing == scientific computing

The multicore processor

Fast processor because:

- Pipelining of execution
- Out of order execution
- Branch prediction
- Pre-fetching
- Large amount of multilevel cache
- ...

The GPU processor

Many computing units operating in parallel

Ideal for simple but repetitive tasks

Great for:

- Dense linear algebra
- Finite-difference
- Neural network

Bad when calculation involves branching

The secret behind the magic

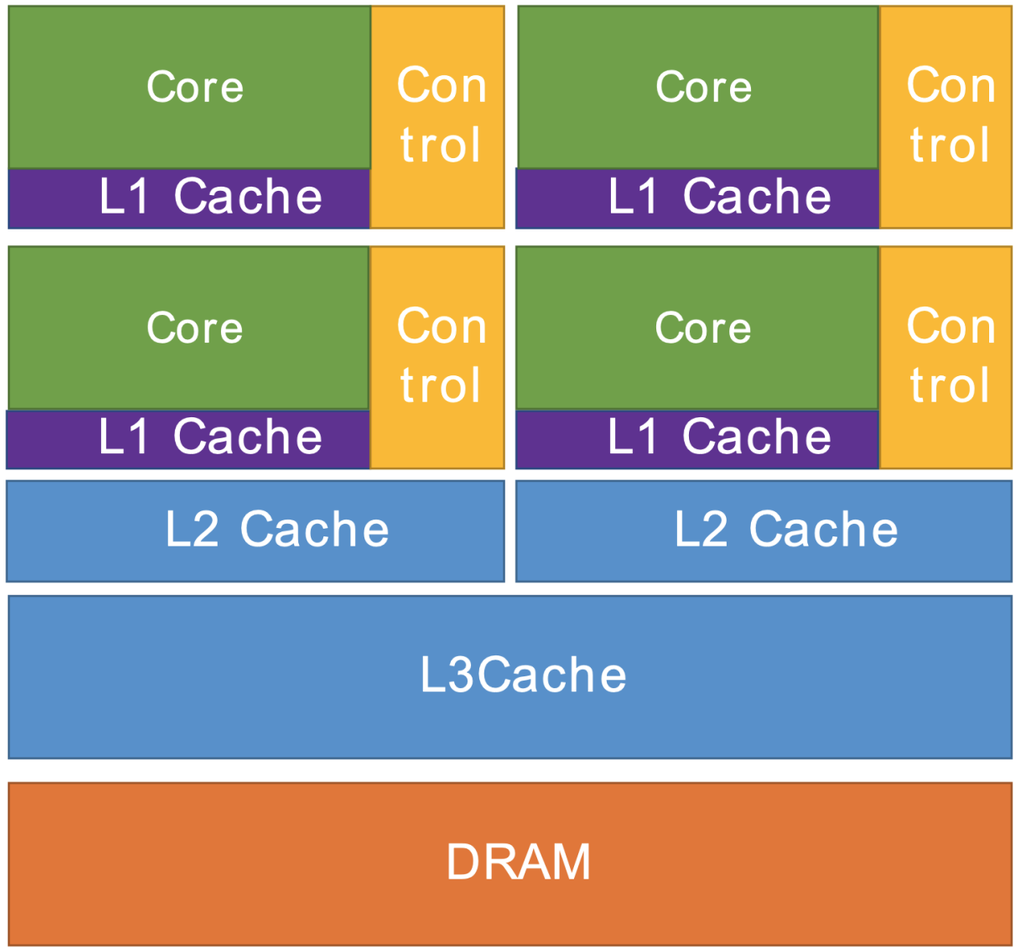
Striking the right trade-off

More computing units means you need to give up something

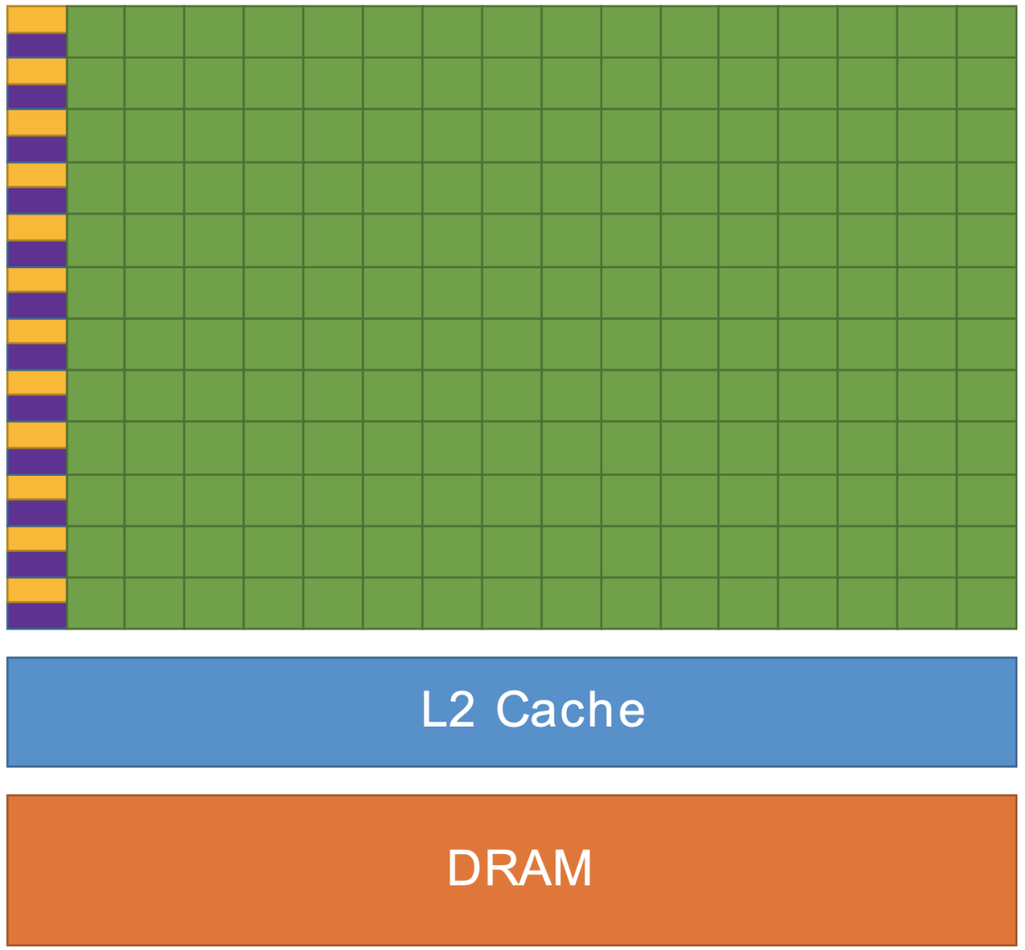
- No processor space dedicated to complex optimizations, e.g., out-of-order execution
- Cache/memory is limited because it has to be shared amongst the threads

- Light threads: hardware supports the ability to switch threads every cycle
- Limited logic for program control: 32 threads are grouped into warps; all threads in warp execute the same instruction at the same time.

Each computing unit is less powerful but there are more of them.

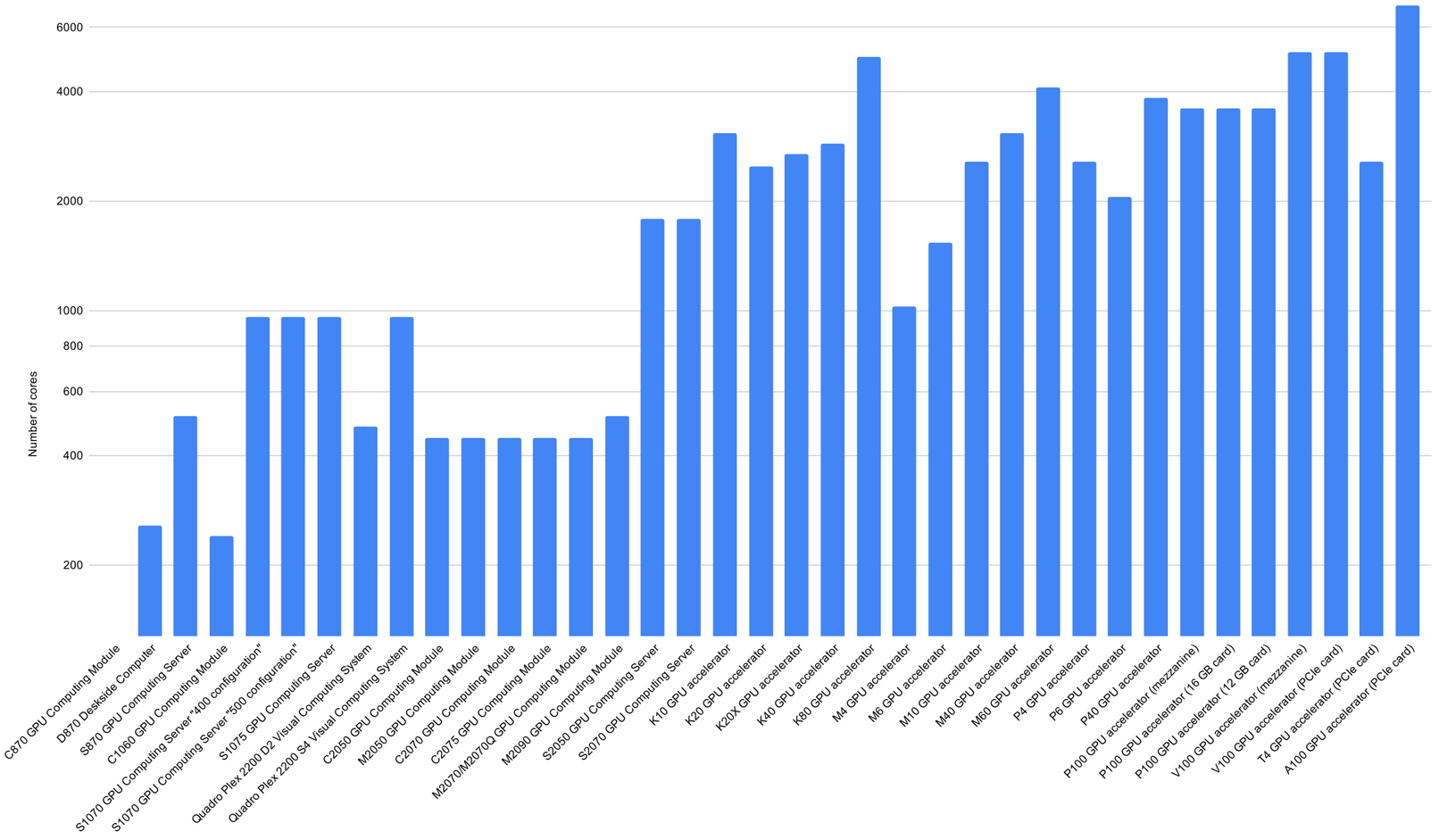

Core
Con
trol
L1 Cache
Core
Con
trol
L1 Cache
Core
Con
trol
L1 Cache
Core
Con
trol
L1 Cache
L2 Cache
L2 Cache
L3Cache
DRAM

CPU

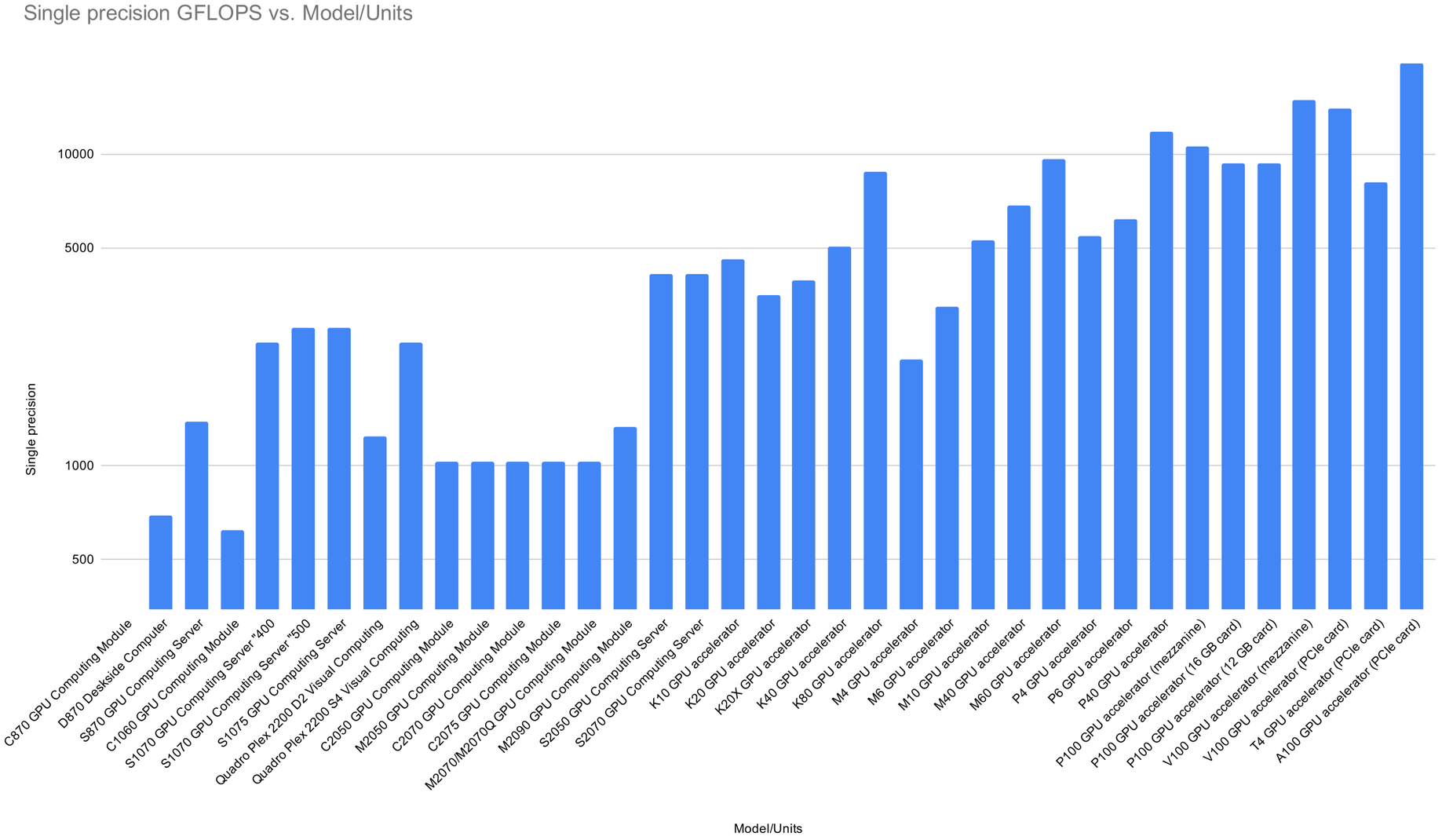

L2 Cache
DRAM

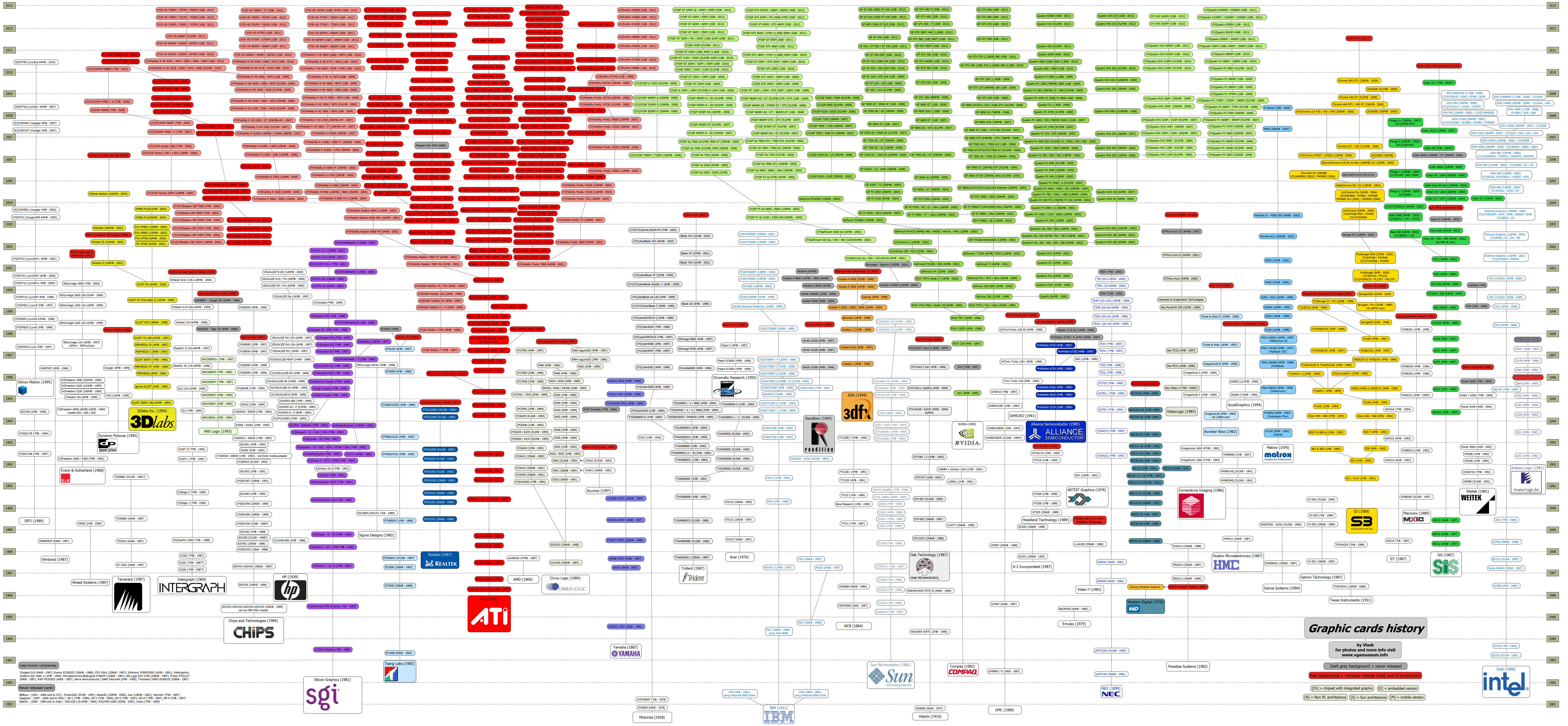
GPU

Micro-architecture	Release	Compute Capability	GPU code name
G70	2005		
Tesla	2006	1.0-1.3	GXX, GT2XX
Fermi	2010	2.0-2.1	GFXXX
Kepler	2012	3.0-3.7	GKXXX
Maxwell	2014	5.0–5.3	GMXXX
Pascal	2016	6.0-6.2	GPXXX
Volta	2017	7.0-7.2	GVXXX
Turing	2018	7.5	TUXXX
Ampere	2020	8.0-8.6	GAXXX
Lovelace	?	?	?
Hopper	?	?	?

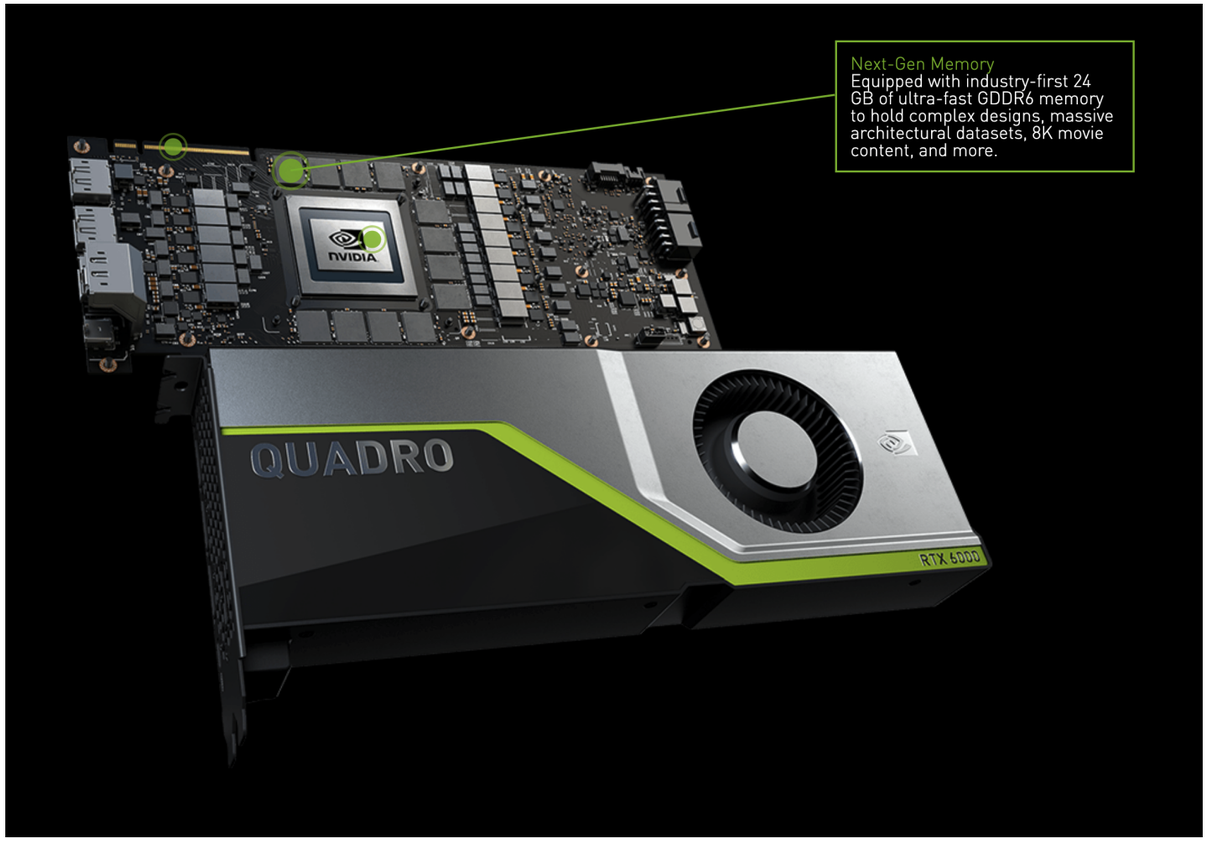

Number of cores
6000
4000
2000
1000
800
600
400
200
C870 GPU Computing Module
D870 Deskside Computer
S870 GPU Computing Server
C1060 GPU Computing Module
S1070 GPU Computing Server "400 configuration"
S1070 GPU Computing Server "500 configuration"
S1075 GPU Computing Server
Quadro Plex 2200 D2 Visual Computing System
Quadro Plex 2200 S4 Visual Computing System
C2050 GPU Computing Module
M2050 GPU Computing Module
C2070 GPU Computing Module
C2075 GPU Computing Module
M2070/M2070Q GPU Computing Module
M2090 GPU Computing Module
S2050 GPU Computing Server
S2070 GPU Computing Server
K10 GPU accelerator
K20 GPU accelerator
K20X GPU accelerator
K40 GPU accelerator
K80 GPU accelerator
M4 GPU accelerator
M6 GPU accelerator
M10 GPU accelerator
M40 GPU accelerator
M60 GPU accelerator
P4 GPU accelerator
P6 GPU accelerator
P40 GPU accelerator
P100 GPU accelerator (mezzanine)
P100 GPU accelerator (16 GB card)
P100 GPU accelerator (12 GB card)
V100 GPU accelerator (mezzanine)
V100 GPU accelerator (PCIe card)
T4 GPU accelerator (PCIe card)
A100 GPU accelerator (PCIe card)

Tesla GPUs


Single precision GFLOPS vs. Model/Units
Single precision
10000
5000
1000
500
C870 GPU Computing Module
D870 Deskside Computer
S870 GPU Computing Server
C1060 GPU Computing Module
S1070 GPU Computing Server "400
S1070 GPU Computing Server "500
S1075 GPU Computing Server
Quadro Plex 2200 D2 Visual Computing
Quadro Plex 2200 S4 Visual Computing
C2050 GPU Computing Module
M2050 GPU Computing Module
C2070 GPU Computing Module
C2075 GPU Computing Module
M2070/M2070Q GPU Computing Module
M2090 GPU Computing Module
S2050 GPU Computing Module
S2070 GPU Computing Server
K10 GPU accelerator
K20 GPU accelerator
K20X GPU accelerator
K40 GPU accelerator
K80 GPU accelerator
M4 GPU accelerator
M6 GPU accelerator
M10 GPU accelerator
M40 GPU accelerator
M60 GPU accelerator
P4 GPU accelerator
P6 GPU accelerator
P40 GPU accelerator
P100 GPU accelerator (mezzanine)
P100 GPU accelerator (16 GB card)
P100 GPU accelerator (12 GB card)
V100 GPU accelerator (mezzanine)
V100 GPU accelerator (PCIe card)
T4 GPU accelerator (PCIe card)
A100 GPU accelerator (PCIe card)
Model/Units


Graphic cards history
by Vlask
for photos and more info visit
www.vgamuseum.info
Dark grey background = never released
Red background = company change (sold, end of production)
[Ch] = chipset with integrated graphic
[E] = embedded version
[N] = Non PC architecture
[S] = Sun architecture
[M] = mobile version

Quadro RTX 6000 in `icme-gpu`



4,608 parallel processing cores; Turing architecture

- 72 Streaming multiprocessors (6 graphics processing clusters)
- 4,608 cores (64 per SM)
- Total global memory: 24 GB GDDR6
- Bandwidth: 672 GB/sec
- Peak performance: single 16.3 TFLOPS; double 510 GFLOPS (1/32)
- Compute capability: 7.5
- Power: 295 W
- Architecture: Turing
- L1 Cache: 96 KB (per SMX); L2 Cache: 6,144 KB

Quadro RTX 6000

Turing architecture

Whitepaper

Specs

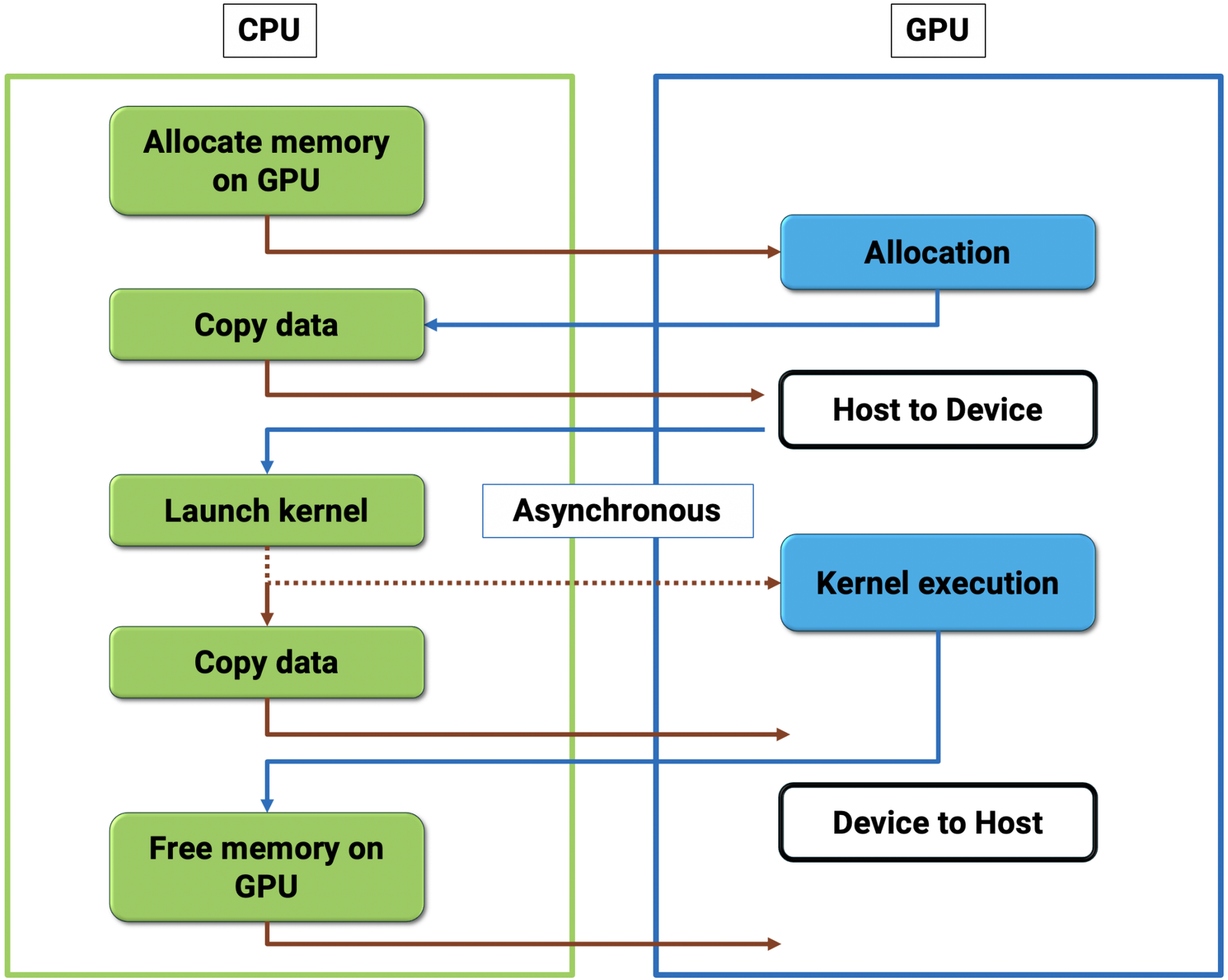
NVIDIA GPUs are different from conventional processors.

They only work as co-processors.

This means you need a host processor (e.g., Intel Xeon).

Your program runs on the host and uses the CUDA API to move data back and forth to the GPU and run programs on the GPU.

You cannot log on the GPU directly or run an OS on the GPU.


CPU
GPU
Allocate memory on GPU
Allocation
Copy data
Host to Device
Launch kernel
Asynchronous
Kernel execution
Copy data
Device to Host
Free memory on GPU

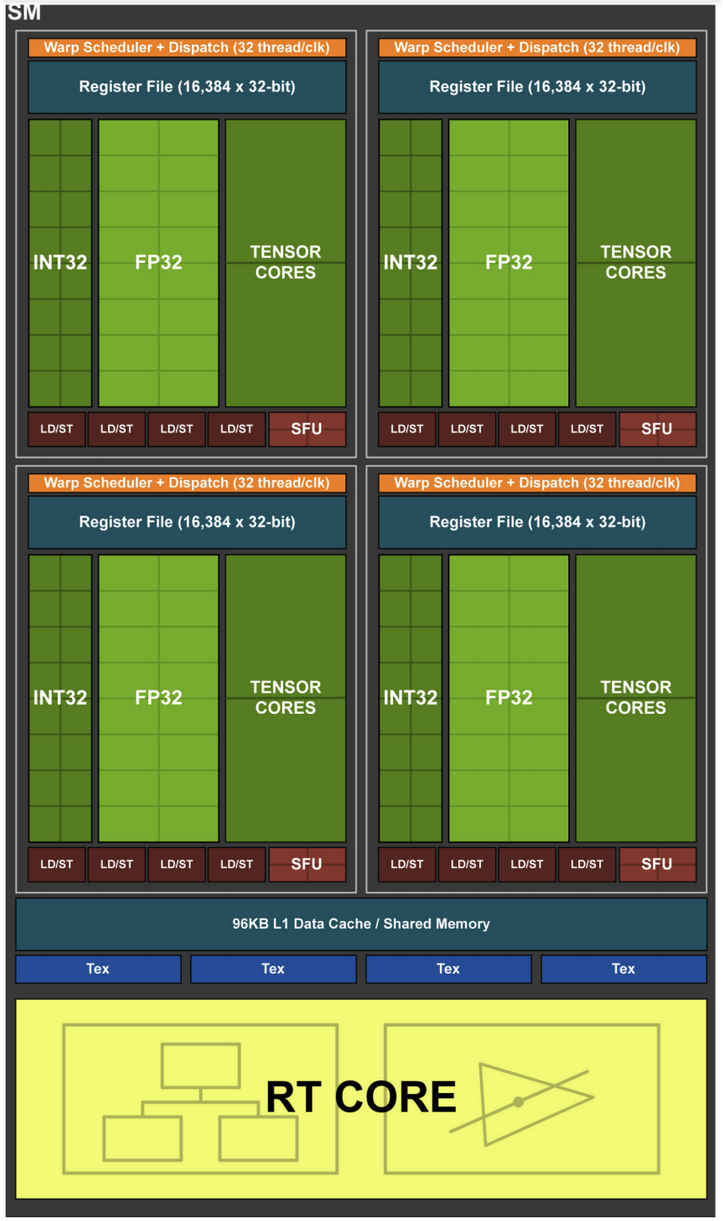
You cannot program a GPU without understanding the architecture of the processor.

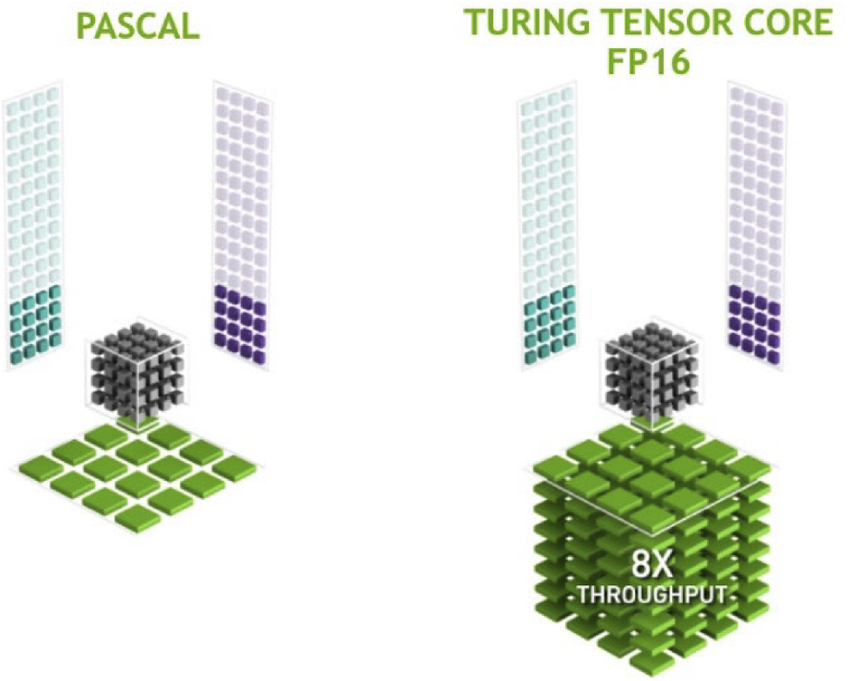
NVIDIA Turing

TU102 GPU architecture


PCI Express 3.0 Host Interface
GigaThread Engine
Memory Controller
Memory Controller
Memory Controller
Memory Controller
Memory Controller
Memory Controller
GPC
Raster Engine
TPC
PolyMorph Engine
SM
RT CORE
L2 Cache
High-Speed Hub
NVLink – Two x8 Links

- GPC: graphics processing cluster
- TPC: texture processing cluster
- SM: Streaming Multiprocessor
- RT core: ray-tracing core
- Tensor cores


SM
Warp Scheduler + Dispatch (32 thread/clk)
Register File (16,384 x 32-bit)
INT32
FP32
TENSOR CORES
LD/ST
SFU
96KB L1 Data Cache / Shared Memory
Tex
RT CORE



Tensor Core 4x4 Matrix Multiply and Accumulate

Hierarchical breakdown of threads

Warp

Bottom-most level: 32 cores = 32 threads, grouped in a warp.

The hardware is optimized to have all threads in a warp execute the same instruction at the same time.

SIMT (single instruction multiple threads) model.

Block

Groups of warps form a block. A block executes on 1 SM (streaming multiprocessor).

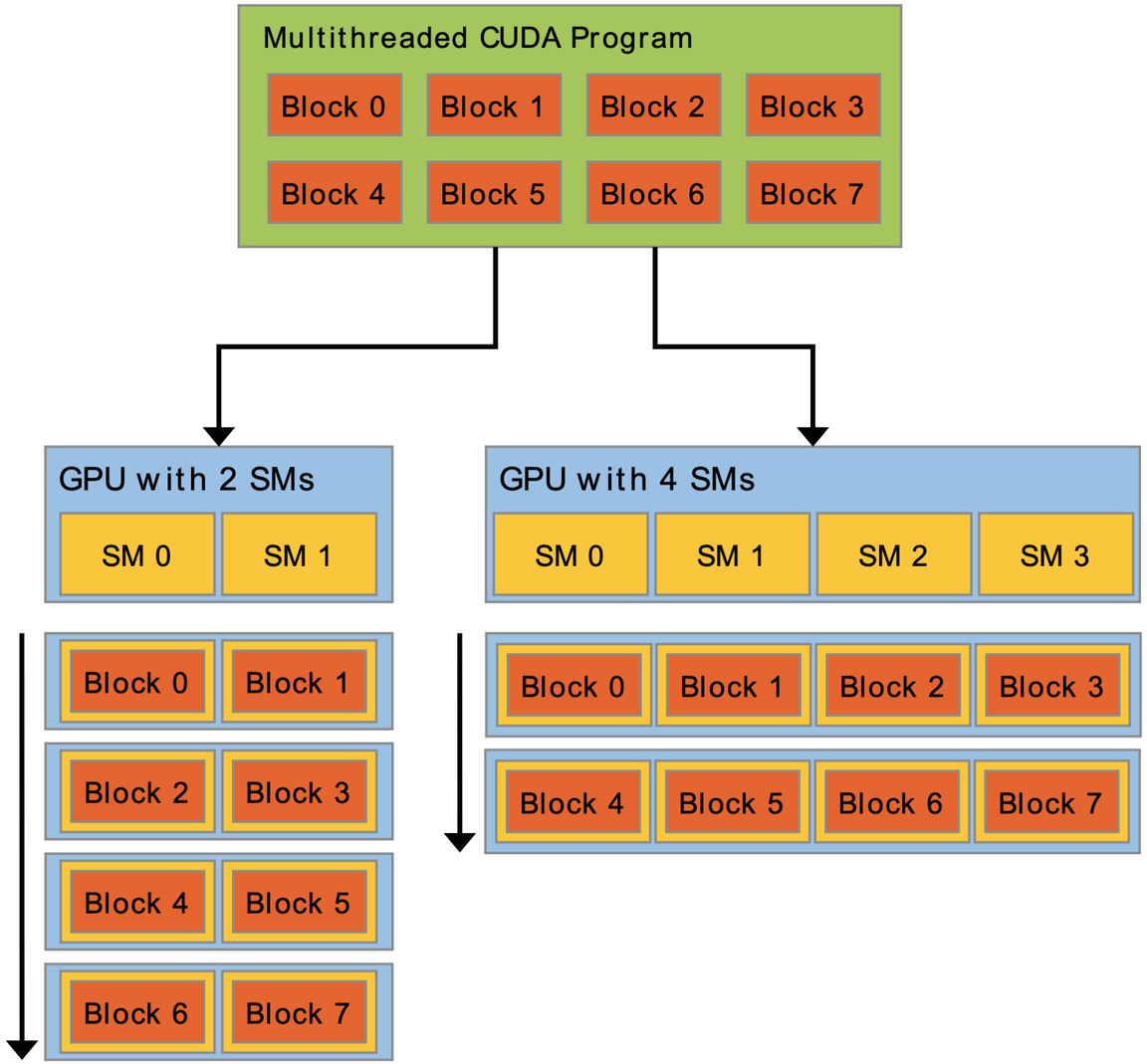
Threads within a block have some ability to exchange data and synchronize.


PCI Express 3.0 Host Interface
GigaThread Engine
Memory Controller
Memory Controller
Memory Controller
Memory Controller
Memory Controller
Memory Controller
Memory Controller
Memory Controller
GPC
Raster Engine
TPC
PolyMorph Engine
SM
RT CORE
L2 Cache
High-Speed Hub
NVLink – Two x8 Links

Grid

Collection of many blocks constitute the entire "dataset" that will be operated on by a kernel.


PCI Express 3.0 Host Interface
GigaThread Engine
Memory Controller
Memory Controller
Memory Controller
Memory Controller
Memory Controller
Memory Controller
Memory Controller
Memory Controller
GPC
Raster Engine
TPC
PolyMorph Engine
SM
RT CORE
L2 Cache
High-Speed Hub
NVLink – Two x8 Links


Multithreaded CUDA Program
Block 0
Block 1
Block 2
Block 3
Block 4
Block 5
Block 6
Block 7
GPU with 2 SMs
SM 0
SM 1
Block 0
Block 1
Block 2
Block 3
Block 4
Block 5
Block 6
Block 7
GPU with 4 SMs
SM 0
SM 1
SM 2
SM 3
Block 0
Block 1
Block 2
Block 3
Block 4
Block 5
Block 6
Block 7

Execution model