Heterogeneous Computing & GPU Introduction

National Tsing-Hua University

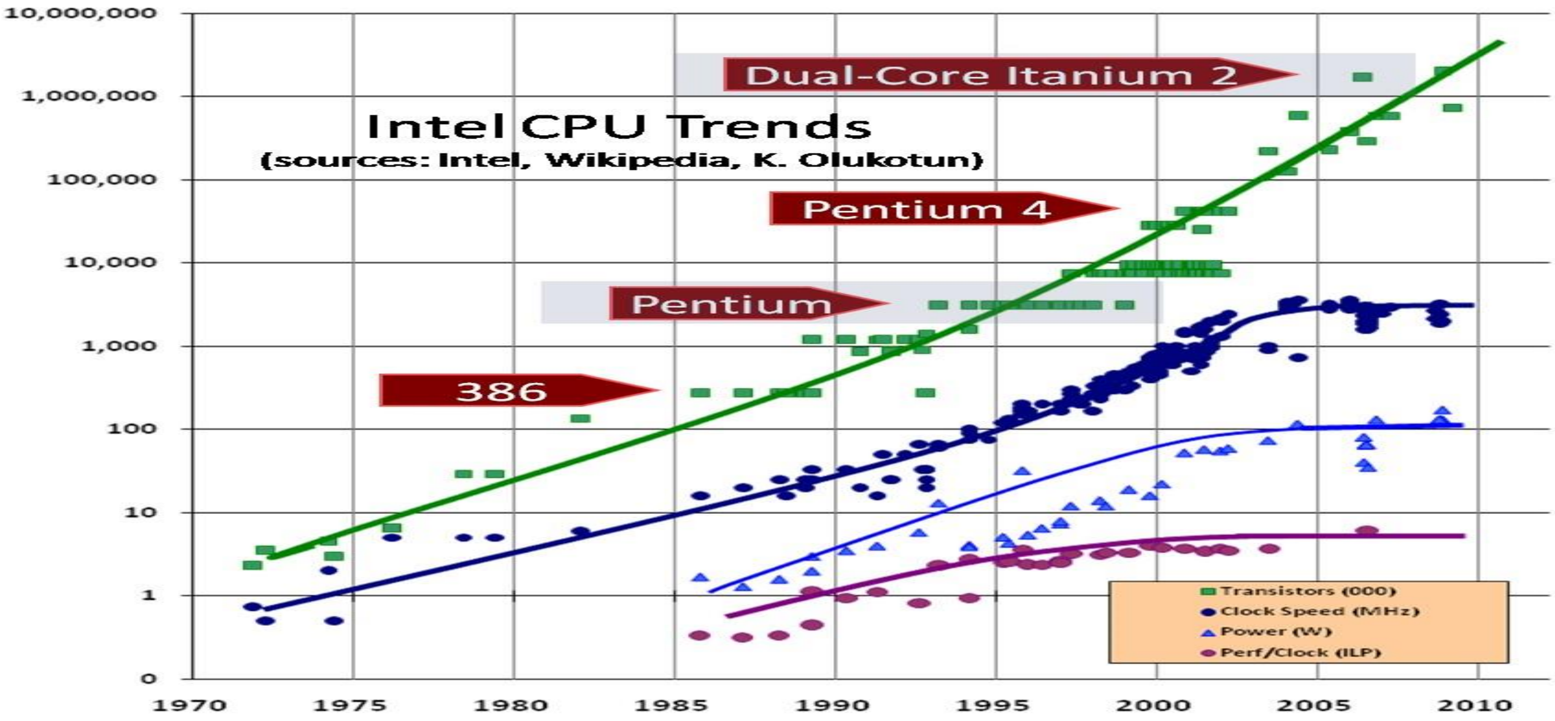
2017, Summer Semester

Outline

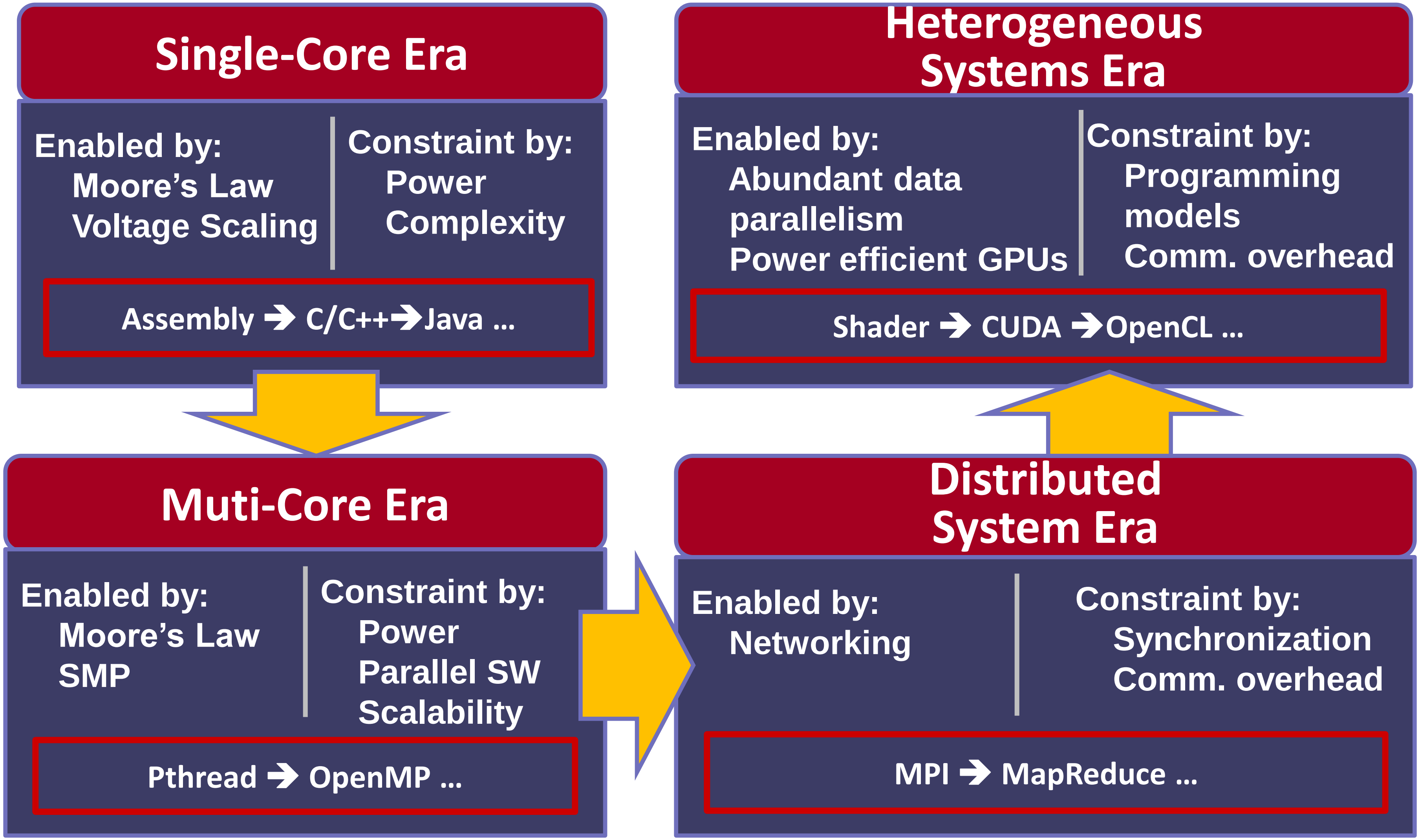
- Heterogeneous Computing
- GPU

The Death of CPU Scaling

- Increase of transistor density ≠ performance
  - The power consumption and clock speed improvements collapsed
  - Non-CPU bottleneck: memory and disk access speed



Trend of Parallel Computers



Single-Core Era

Enabled by:
Moore's Law
Voltage Scaling

Constraint by:
Power
Complexity

Assembly → C/C++→Java …

Muti-Core Era

Enabled by:
Moore's Law
SMP

Constraint by:
Power
Parallel SW
Scalability

Pthread → OpenMP …

Distributed System Era

Enabled by:
Networking

Constraint by:
Synchronization
Comm. overhead

MPI → MapReduce …

Heterogeneous Systems Era

Enabled by:
Abundant data parallelism
Power efficient GPUs

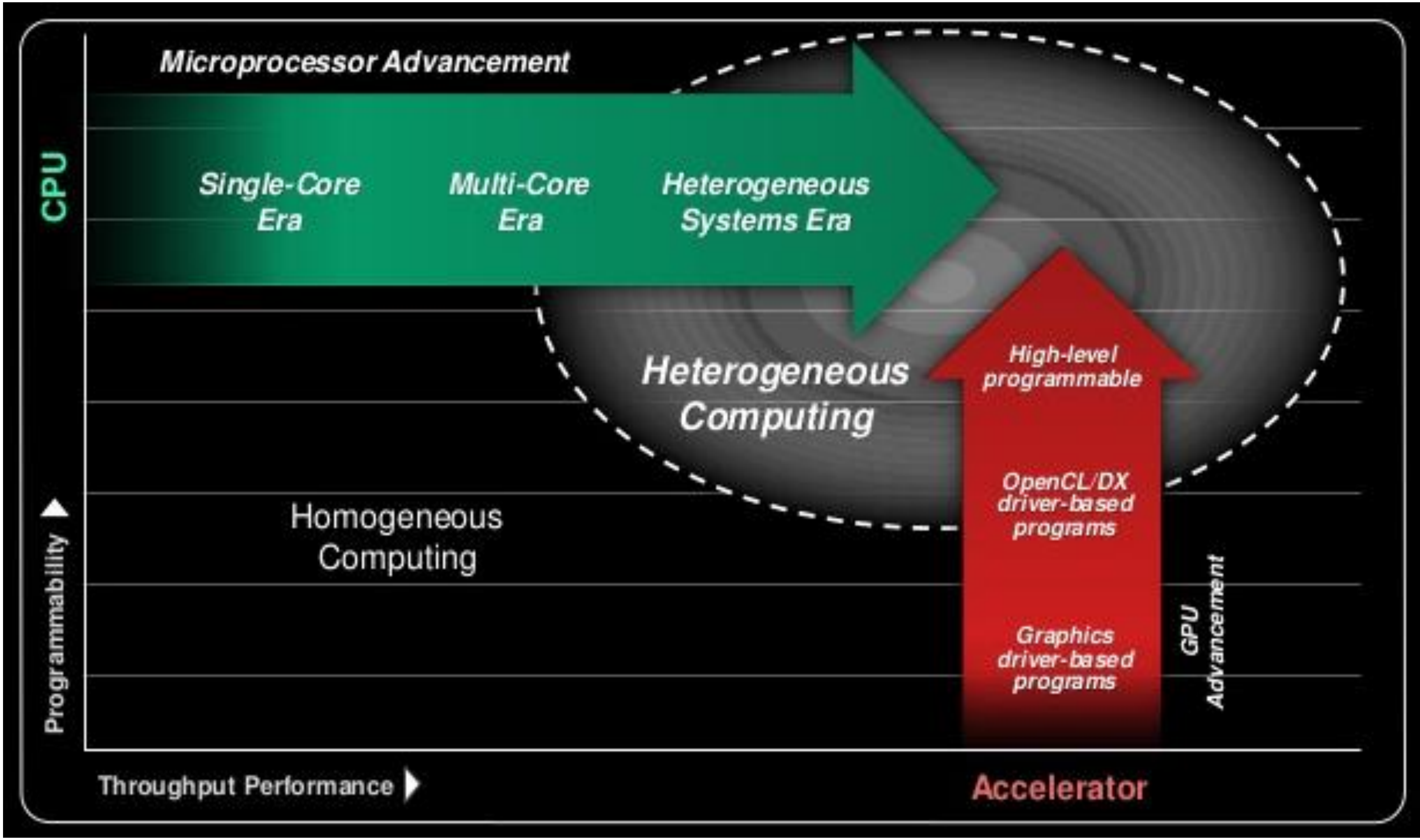
Constraint by:
Programming models
Comm. overhead

Shader → CUDA →OpenCL …

Heterogeneous Computing

- Heterogeneous computing is an integrated system that consists of different types of (programmable) computing units.
  - DSP (digital signal processor)
  - FPGA (field-programmable gate array)
  - ASIC (application-specific integrated circuit)
  - GPU (graphics processing unit)
  - Co-processor (Intel Xeon Phi)
- A system can be a cell phone or a supercomputer

Shift of Computing Paradigm



GPU/Xeon Phi in Top 500 list (rank world's fastest Supercomputer)

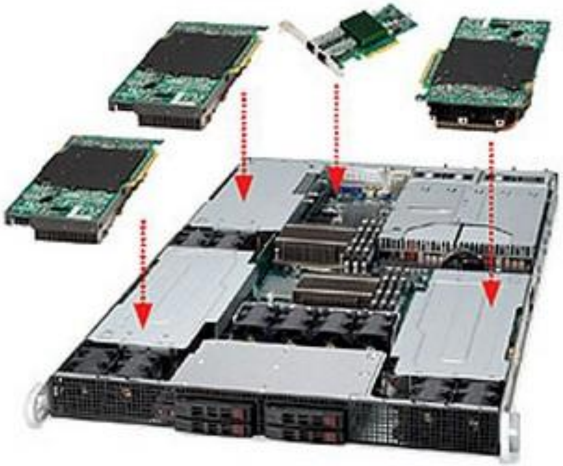
- Jaguar was upgraded with GPU and renamed to Titan
  - Increase computation power by a factor of **10** !!!
- 62 systems have accelerator(GPU) or co-processor (Phi)
- http://www.top500.org/lists/

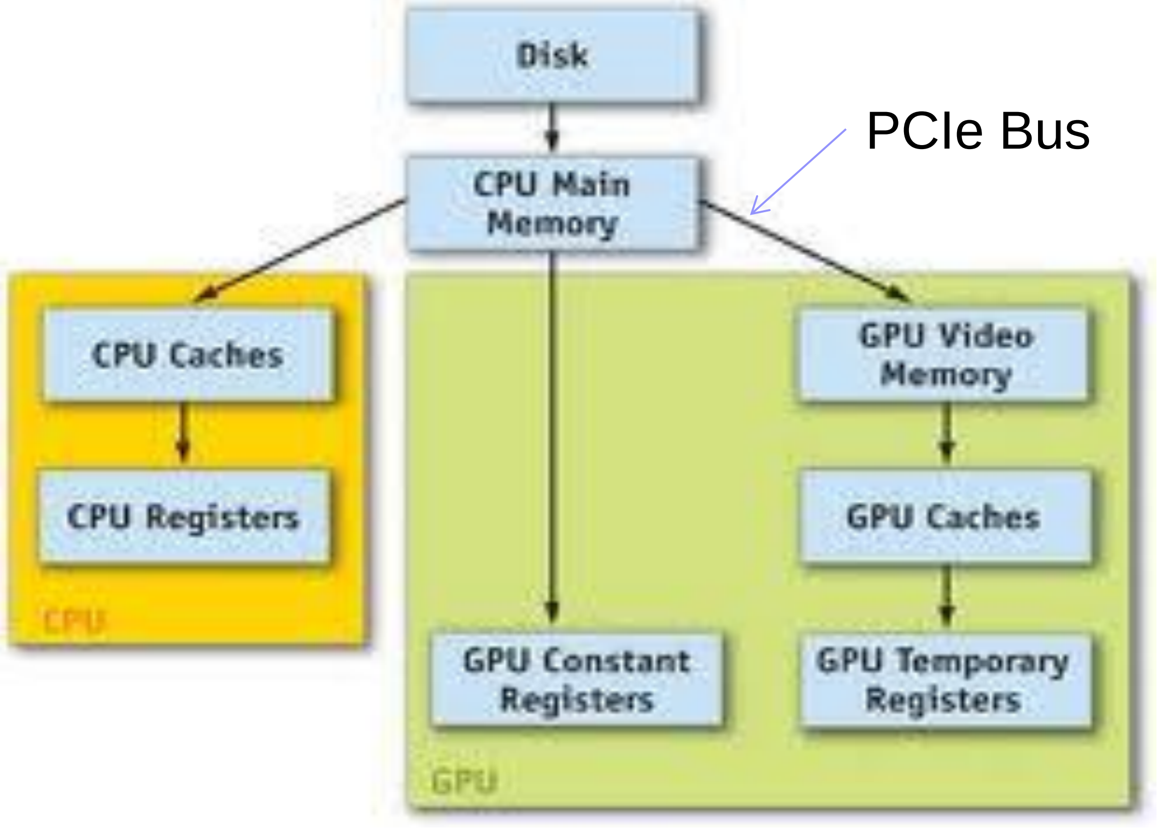
2014 Rank	Name	Country	Manuf-acture	Accelerator	Cores	Rmax (TFlops/s)
1	**Tianhe-2**	China	NUDT	Xeon Phi	3,120K	33.8K
2	**Titan**	US	Cray	NVIDIA K20x	560K	17.6K
3	**Sequoia**	US	IBM	N.A	1,572K	17.2K
4	**K computer**	Japan	Fujitsu	N.A	705K	10.5K

2012 Rank	Name	Country	Manuf-acture	Accelerator	Cores	Rmax (TFlops/s)
6	**Jaguar**	US	Cray	N.A	298K	1.9K

GPU Servers

- Same HW architecture as commodity server, but memory copy between CPU and GPU becomes the main bottleneck

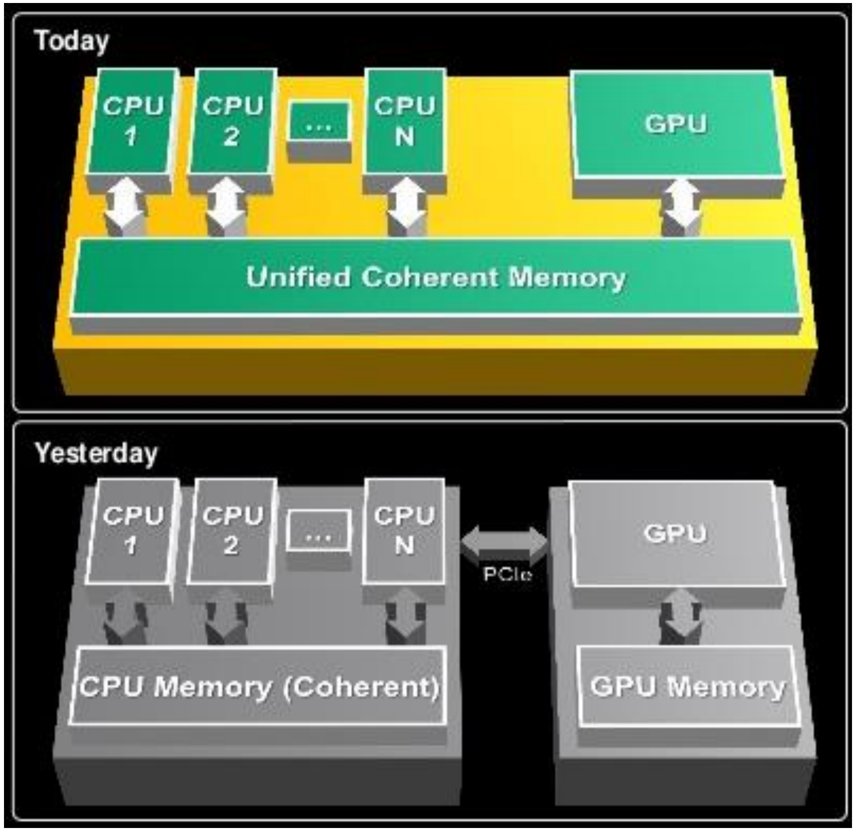




PCIe Bus

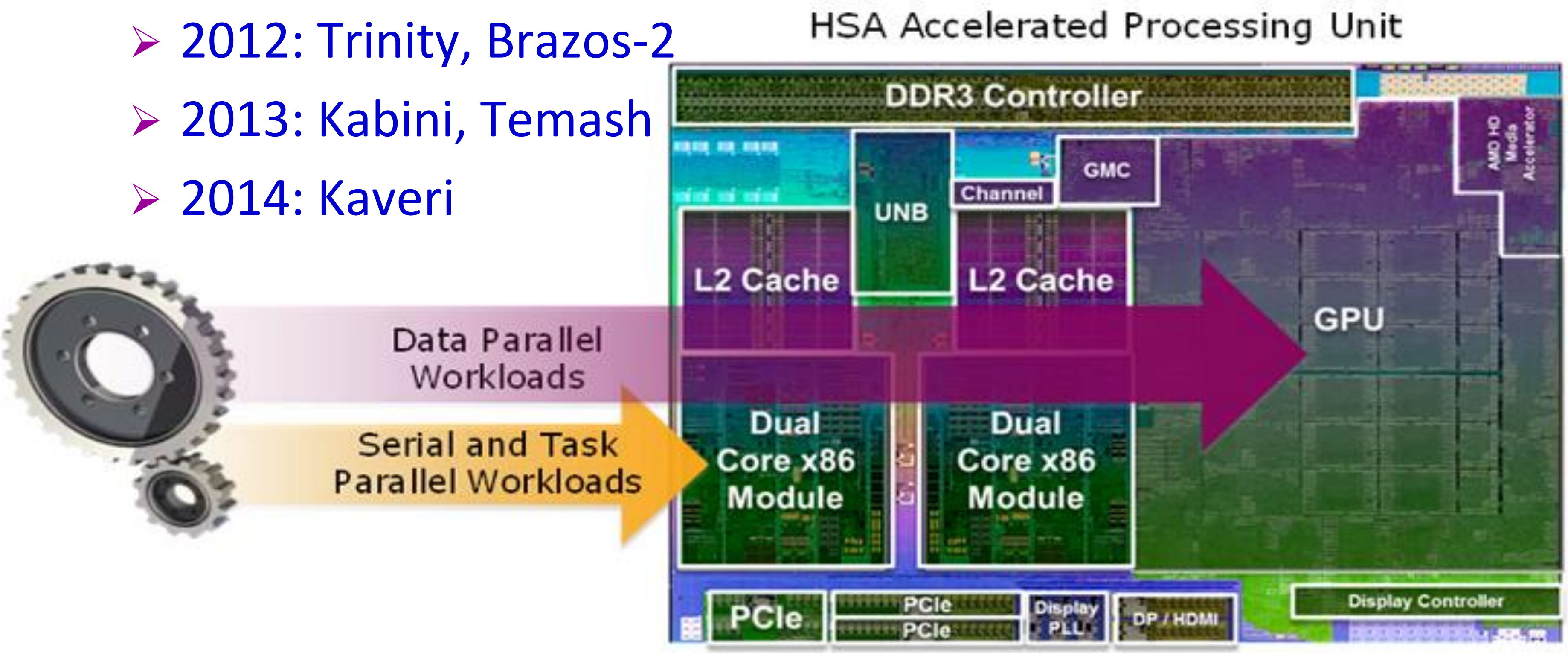
Heterogeneous System Architecture (HSA)

- Aim to provide a common system architecture for designing higher-level programming models for all devices
- Unified coherent memory
  - Single virtual memory address space
  - Prevent memory copy



AMD Accelerated Processing Unit (APU)

- A.k.a ***Fusion:*** a series of 64-bit microprocessors from AMD designed to act as a CPU and GPU on a single chip
  - 2011: Llano, Brazos
  - 2012: Trinity, Brazos-2
  - 2013: Kabini, Temash
  - 2014: Kaveri

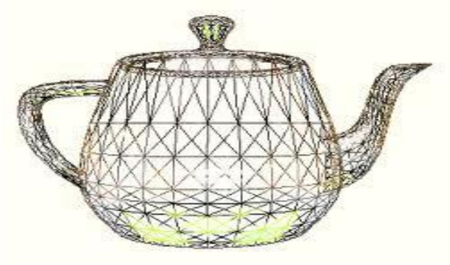


Outline

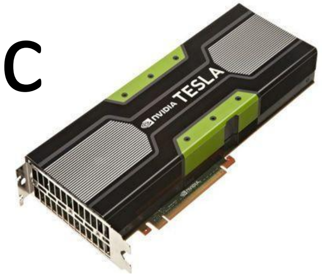
- Heterogeneous Computing
- GPU

GPU (Graphic Processing Unit)

- A specialized chip designed for rapidly display and visualization
  - SIMD architecture
- Massively multithreaded manycore chips
  - NVIDIA Tesla products have up to 128 scalar processors
  - Over 12,000 **concurrent** threads
  - Over 470 GFOLPS sustained performance
- Two major vendors: NVIDIA and ATI (now AMD)

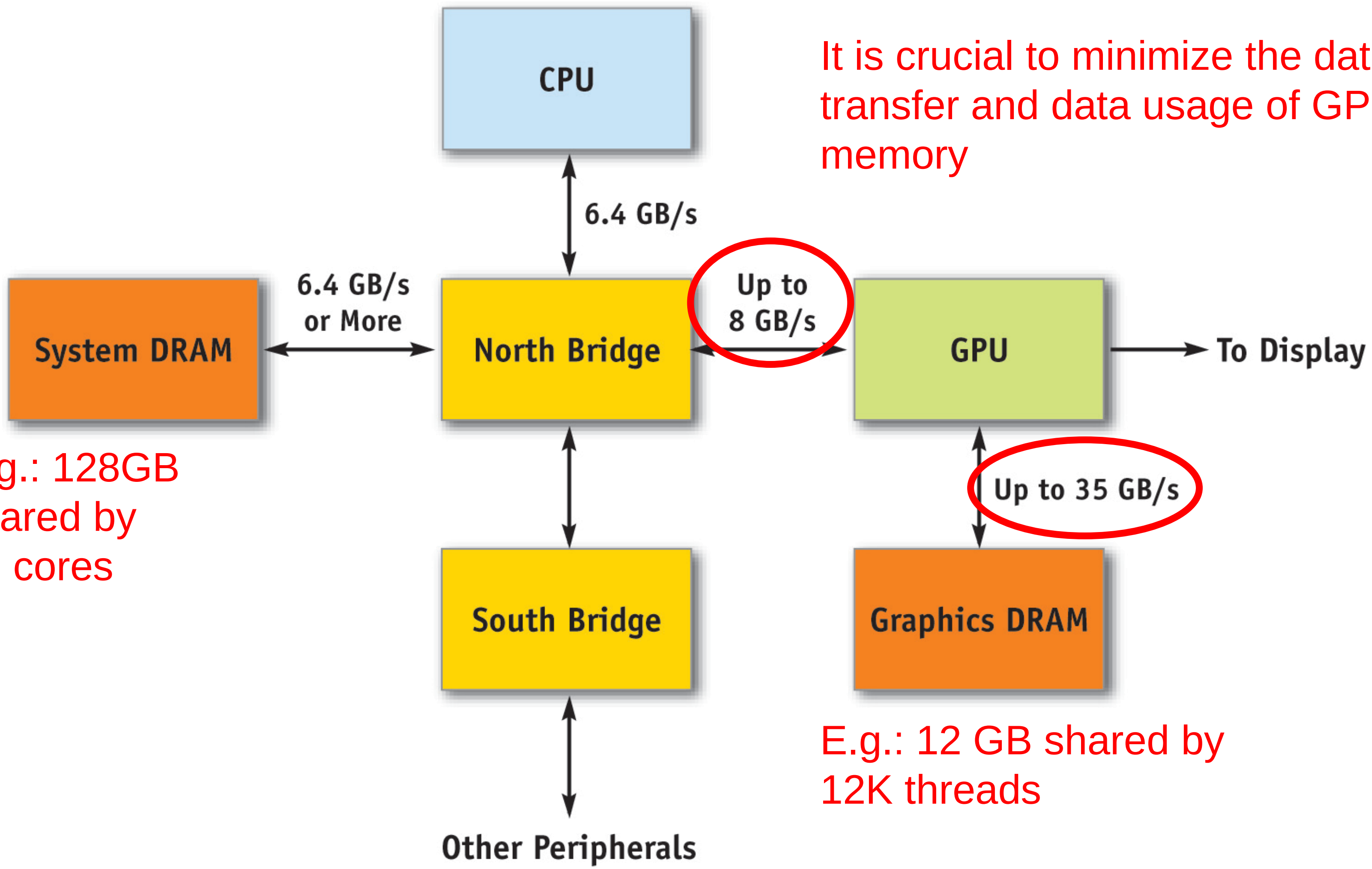


GPGPU (General-Purpose Graphic Processing Unit)



- **Expose** the horse power of GPUs for general purpose computations
  - Exploit **data parallelism** for solving embarrassingly parallel tasks and numeric computations
  - Users across science & engineering disciplines are achieving 100x or better speedups on GPUs
- Programmable
  - Early GPGPU: using the libraries in computer graphics, such as OpenGL or DirectX, to perform the tasks other than the original hardware designed for.
  - Now **CUDA** and openCL provides an extension to C and C++ that enables parallel programming on GPUs

System Architecture



It is crucial to minimize the data transfer and data usage of GPU memory

E.g.: 128GB shared by 16 cores

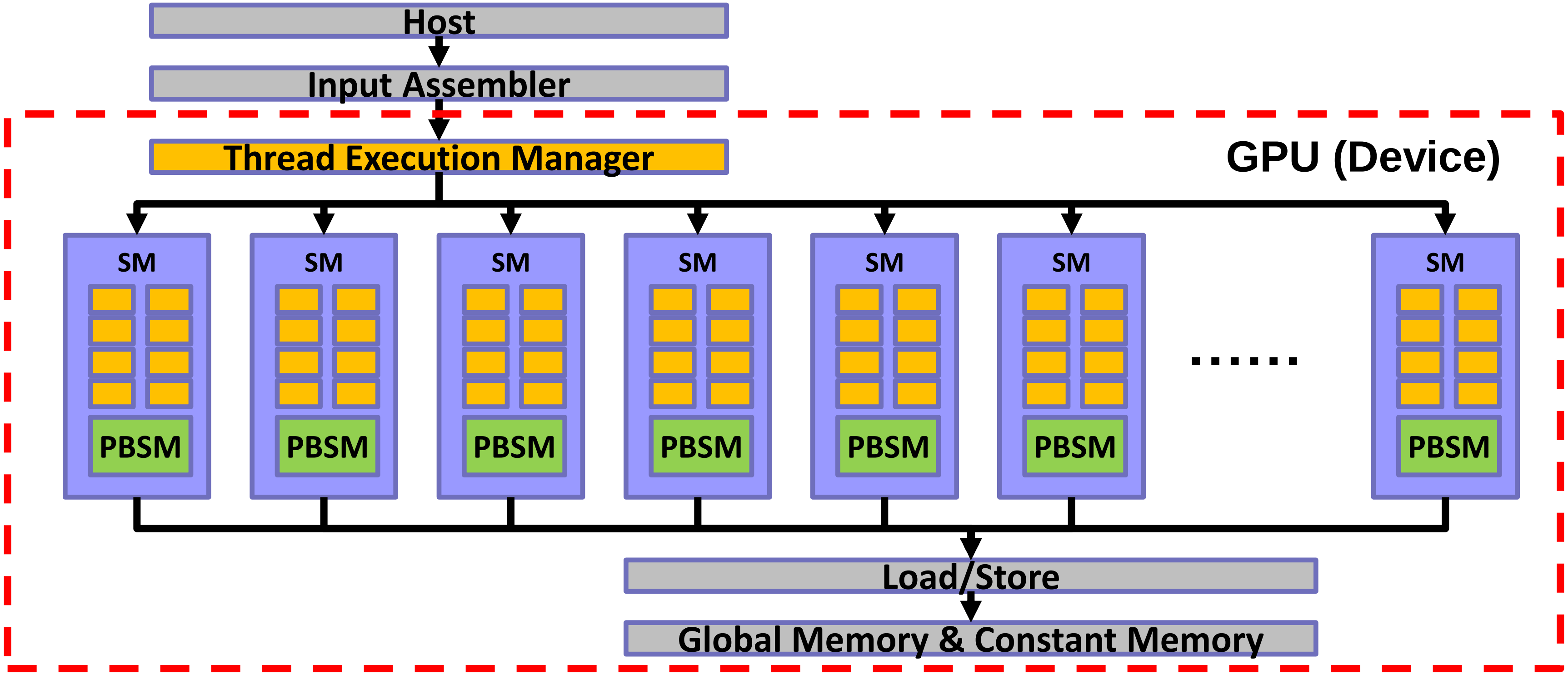
E.g.: 12 GB shared by 12K threads

Manycore GPU – Block Diagram

- Consist of multiple stream multi-processors (SM)
- Memory hierarchic:
  - global memory → PBSM/shared memory → local register

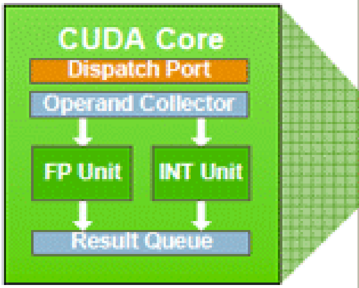
Slow, but large & shared

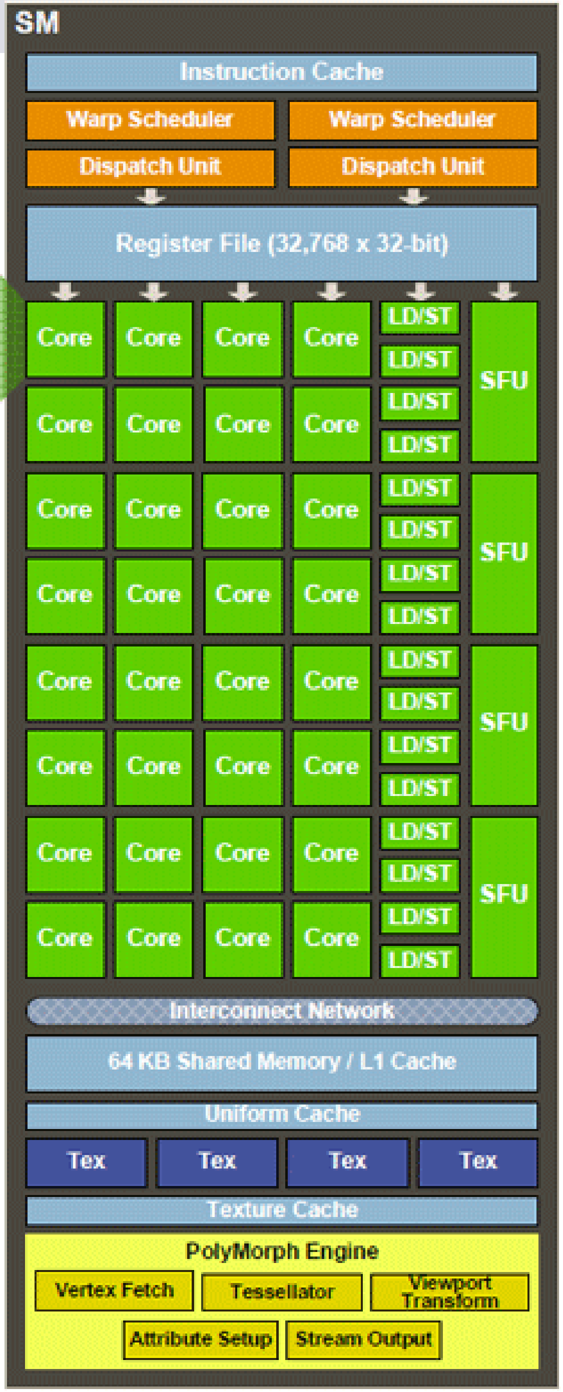
Fast, but small & local



Stream Multiprocessor

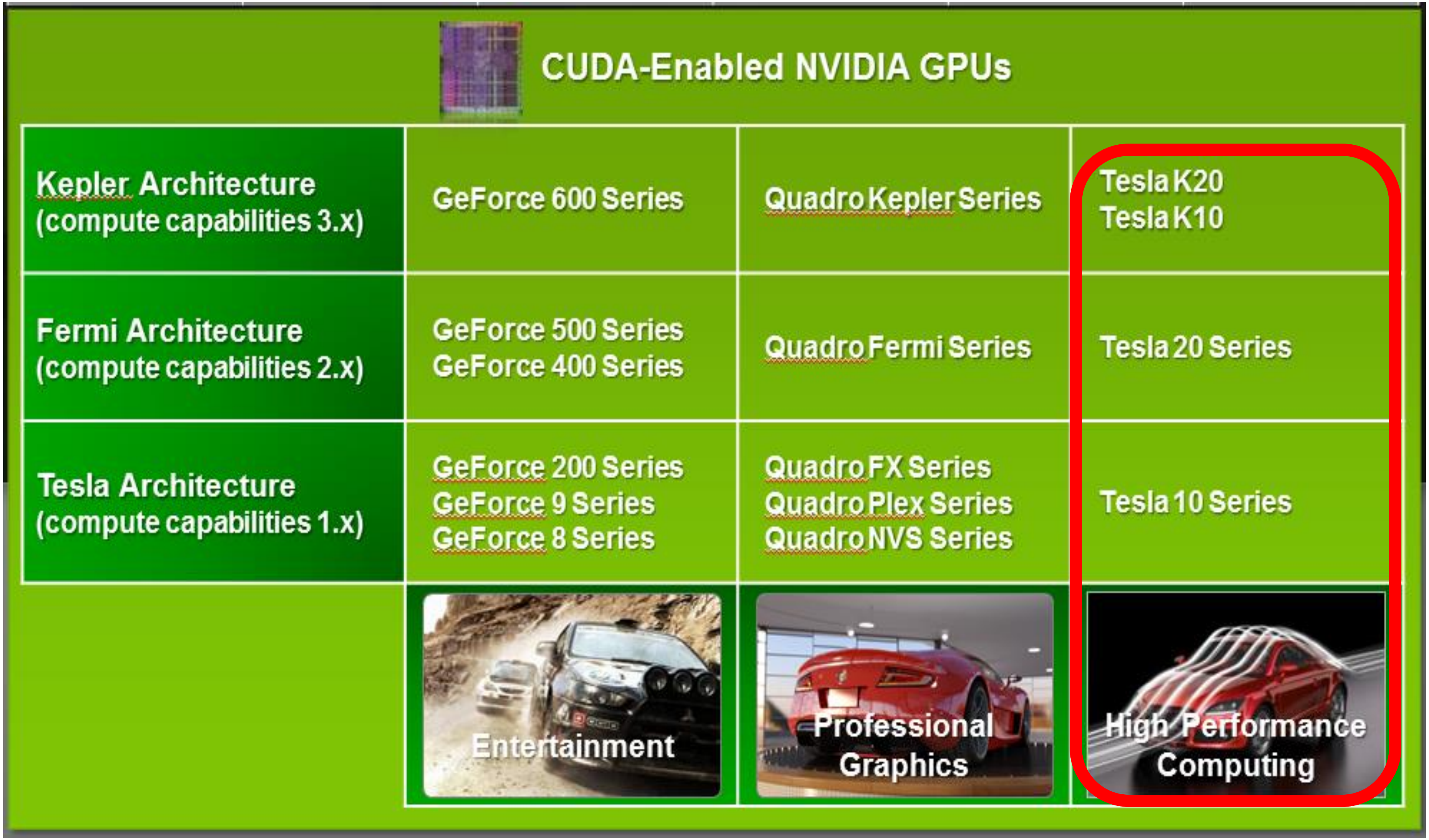
- Each SM is a vector machine
- Shared register files
  - Store local variables
- Programmable cache (shared memory)
  - Shared with a normal L1 cache.
- **Hardware scheduling** for thread execution and **hardware context switch**





http://hothardware.com/Articles/NVIDIA-GF100-Architecture-and-Feature-Preview/

NVIDIA CUDA-Enabled GPUs Products



CUDA-Enabled NVIDIA GPUs

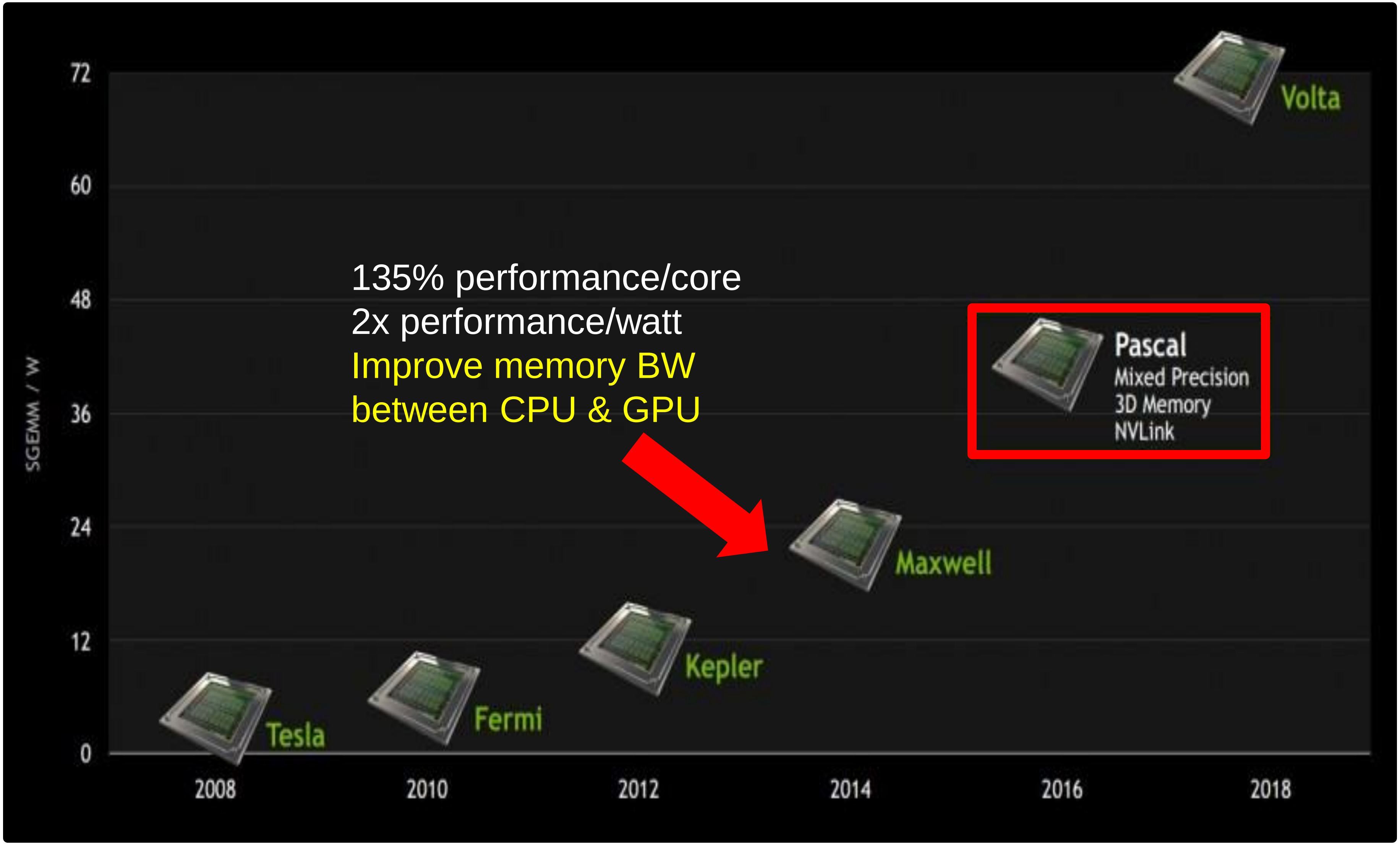
Kepler Architecture (compute capabilities 3.x)	GeForce 600 Series	Quadro Kepler Series	Tesla K20 Tesla K10
Fermi Architecture (compute capabilities 2.x)	GeForce 500 Series GeForce 400 Series	Quadro Fermi Series	Tesla 20 Series
Tesla Architecture (compute capabilities 1.x)	GeForce 200 Series GeForce 9 Series GeForce 8 Series	Quadro FX Series Quadro Plex Series Quadro NVS Series	Tesla 10 Series
	Entertainment	Professional Graphics	High Performance Computing

NVIDIA Tesla Family HW Specification

	Tesla K40	Tesla K20X	Tesla K20	Tesla M2090
Stream Processors	2880	2688	2496	512
Core Clock	745MHz	732MHz	706MHz	650MHz
Memory Clock	6GHz GDDR5	5.2GHz GDDR5	5.2GHz GDDR5	3.7GHz GDDR5
Memory Bus Width	384-bit	384-bit	320-bit	384-bit
VRAM	12GB	6GB	5GB	6GB
Single Precision	4.29 TFLOPS	3.95 TFLOPS	3.52 TFLOPS	1.33 TFLOPS
Double Precision	1.43 TFLOPS (1/3)	1.31 TFLOPS (1/3)	1.17 TFLOPS (1/3)	655 GFLOPS (1/2)
Transistor Count	7.1B	7.1B	7.1B	3B
TDP	235W	235W	225W	250W
Cooling	Active/Passive	Passive	Active/Passive	N/A
Manufacturing Process	TSMC 28nm	TSMC 28nm	TSMC 28nm	TSMC 40nm
Architecture	Kepler	Kepler	Kepler	Fermi
Launch Price	$5499?	~$3799	~$3299	N/A

http://www.anandtech.com/show/7521/
nvidia-launches-tesla-k40

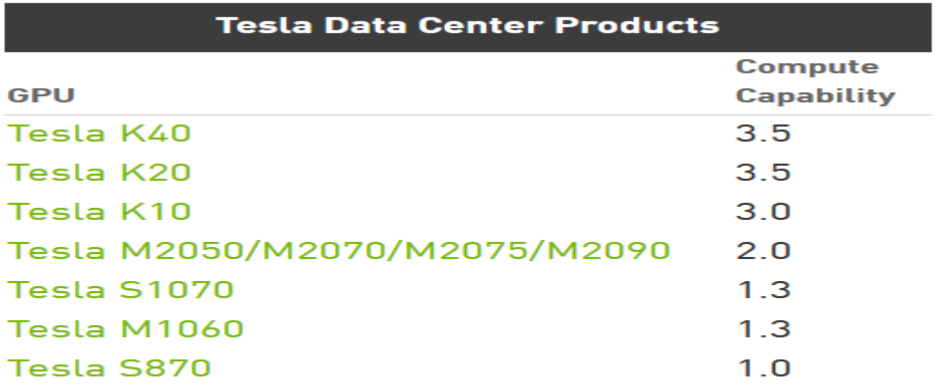
NVIDIA GPU Architecture Roadmap



135% performance/core
2x performance/watt
Improve memory BW
between CPU & GPU

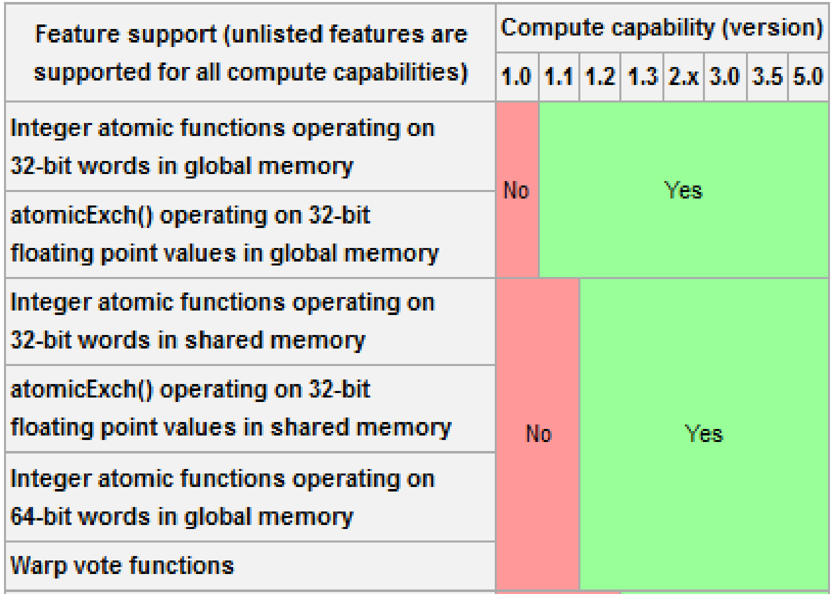
GPU Compute Capability

- Computing/Programming features & spec

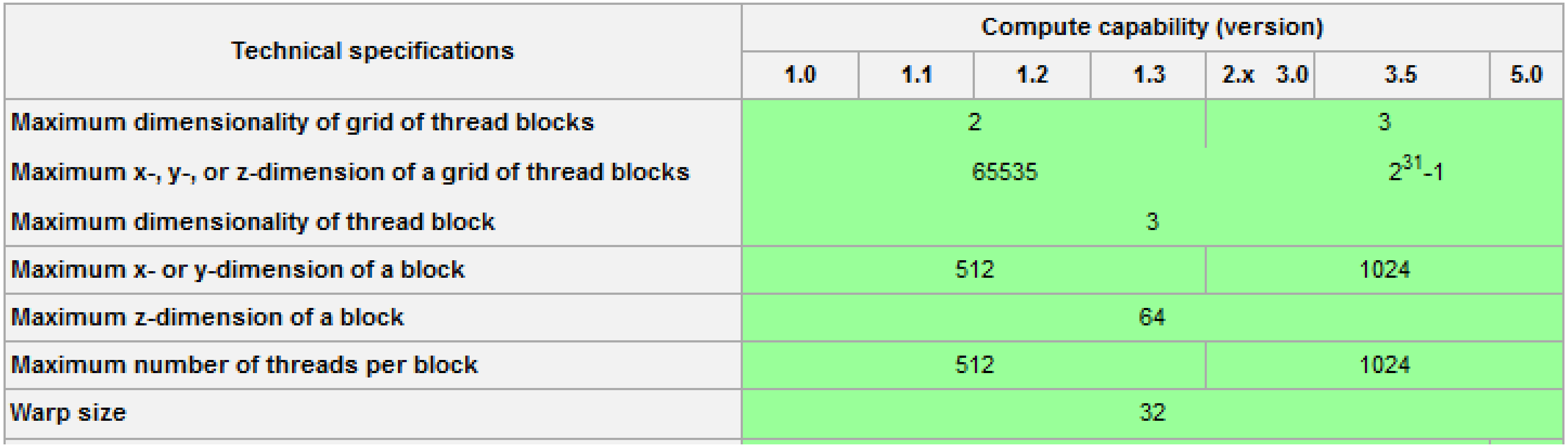


Tesla Data Center Products

GPU	Compute Capability
Tesla K40	3.5
Tesla K20	3.5
Tesla K10	3.0
Tesla M2050/M2070/M2075/M2090	2.0
Tesla S1070	1.3
Tesla M1060	1.3
Tesla S870	1.0



Feature support (unlisted features are supported for all compute capabilities)	Compute capability (version)							
	1.0	1.1	1.2	1.3	2.x	3.0	3.5	5.0
Integer atomic functions operating on 32-bit words in global memory	No	Yes						
atomicExch() operating on 32-bit floating point values in global memory								
Integer atomic functions operating on 32-bit words in shared memory	No		Yes					
atomicExch() operating on 32-bit floating point values in shared memory								
Integer atomic functions operating on 64-bit words in global memory								
Warp vote functions								

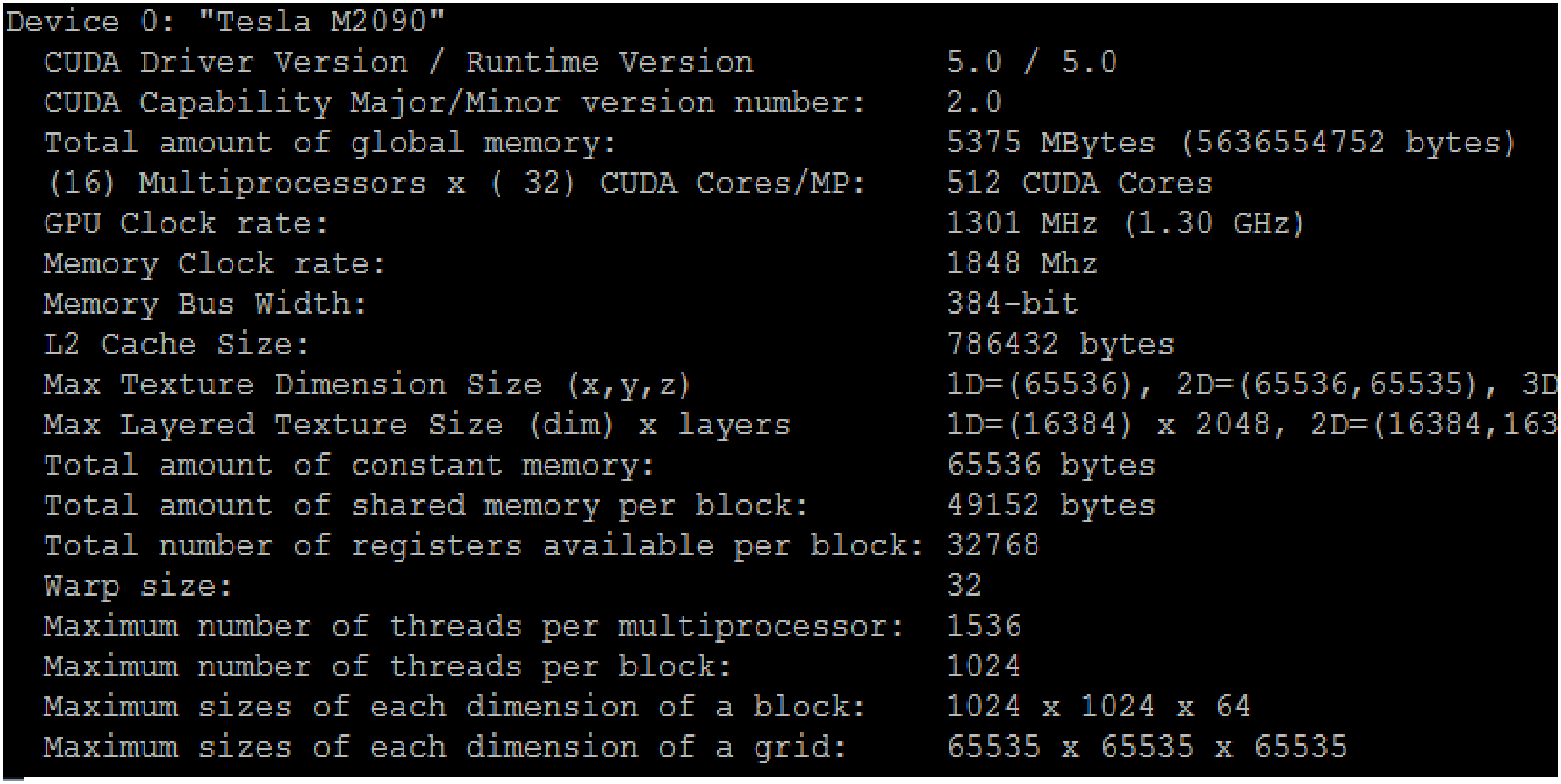


Technical specifications	Compute capability (version)							
	1.0	1.1	1.2	1.3	2.x	3.0	3.5	5.0
Maximum dimensionality of grid of thread blocks	2				3			
Maximum x-, y-, or z-dimension of a grid of thread blocks	65535					2^{31}-1		
Maximum dimensionality of thread block	3							
Maximum x- or y-dimension of a block	512				1024			
Maximum z-dimension of a block	64							
Maximum number of threads per block	512				1024			
Warp size	32							

source: http://en.wikipedia.org/wiki/CUDA

CUDA SDK Device Query

- deviceQuery.cpp



```
Device 0: "Tesla M2090"
  CUDA Driver Version / Runtime Version          5.0 / 5.0
  CUDA Capability Major/Minor version number:    2.0
  Total amount of global memory:                 5375 MBytes (5636554752 bytes)
  (16) Multiprocessors x ( 32) CUDA Cores/MP:    512 CUDA Cores
  GPU Clock rate:                                1301 MHz (1.30 GHz)
  Memory Clock rate:                             1848 Mhz
  Memory Bus Width:                              384-bit
  L2 Cache Size:                                 786432 bytes
  Max Texture Dimension Size (x,y,z)             1D=(65536), 2D=(65536,65535), 3D
  Max Layered Texture Size (dim) x layers        1D=(16384) x 2048, 2D=(16384,163
  Total amount of constant memory:               65536 bytes
  Total amount of shared memory per block:       49152 bytes
  Total number of registers available per block: 32768
  Warp size:                                     32
  Maximum number of threads per multiprocessor:  1536
  Maximum number of threads per block:           1024
  Maximum sizes of each dimension of a block:    1024 x 1024 x 64
  Maximum sizes of each dimension of a grid:     65535 x 65535 x 65535
```

Reference

- Cyril Zeller, NVIDIA Developer Technology slides
- Heterogamous computing course slides from Prof. Che-Rung Lee