GPGPU Architecture Comparison of ATI™ and NVIDIA® GPUs

Michael Fried
GPGPU Business Unit Manager
Microway, Inc.
Updated June, 2010
http://microway.com/gpu.html

Overview of Current Generation Architectures

June, 2010

ATI "Evergreen" / 5000 Series

- Up to 1600 SCs @ 725 - 850MHz
- 1600 SP, 320 DP, 320 SF
- 20 SIMDs * 16 TPs * 5 SCs issuing
  4 SP or 2 SP FMA or 2 DP or 1 DP FMA
  and 1 SP or 1 SF every cycle
- 32KB Local Memory
  8KB L1 Cache per SIMD
- OpenCL™ -> CAL/IL software stack
- 2.72 TFLOPs SP / 544 GFLOPs DP
  (Theoretical maximum, Radeon HD5870
  @ 850MHz, 1600 Cores)

NVIDIA "Fermi" / GF100

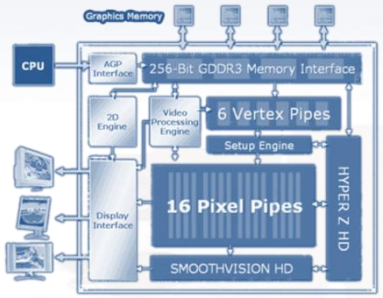
- Up to 512 CUDA™ cores @ 1.15 - 1.4GHz
- 512 SP, 256 DP, 64 SF
- 16 SMs each with 32 SP units,
  16 DP units, 4 SF units,
  and 16 L1 Cache Load / Store units
- 16 KB / 48 KB or 48 KB / 16 KB selectable
  L1 Cache / Shared memory per SM
- CUDA + OpenCL -> PTX software stack
- 1.03 TFLOPs SP / 515 GFLOPs DP
  (Theoretical maximum, Tesla C2050
  @ 1.15GHz, 448 Cores)

- SIMD = Single Issue Multiple Dispatch
- TP = Thread Processor
- SC = Scalar Core
- SP = Single Precision
- DP = Double Precision
- FMA = Fused Multiply/Add
- SF = Special Function (transcendental: sin, cos, arc tan, sqrt, pow, etc)
- SM = Streaming Multiprocessor
- OpenCL = Open Compute Language
- CAL/IL = Compute Abstraction Layer / Intermediate Language
- CUDA = Compute Unified Device Architecture
- PTX = Parallel Thread eXecution

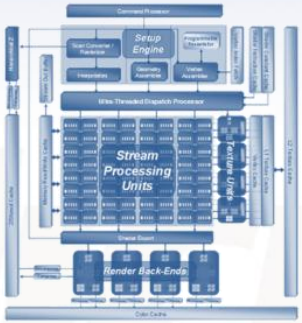
History of ATI GPUs with Stream Computing support

Architecture	Competes with NVIDIA Architecture	GPU Chip (Card Name)	SIMD Engines (Thread Processors)	Compute Cores SP / DP / SFU (5/1/1 per TP)	Max TFLOPs (SP / DP)
R600 Family	G80	RV670 (HD3870)	4 (64)	320 / 64 / 64	.496 / .099
R700 Family	GT200	RV770 (HD4870)	10 (160)	800 / 160 / 160	1.20 / .24
Evergreen Family	GT200	Juniper (HD5770)	10 (160)	800 / 0 / 160	1.36 / 0
	GF100	Cypress (HD5870)	20 (320)	1600 / 320 / 320	2.72 / .544
	GF100	Hemlock (HD5970)	20 x2 (640)	3200 / 640 / 640	4.64 / .928

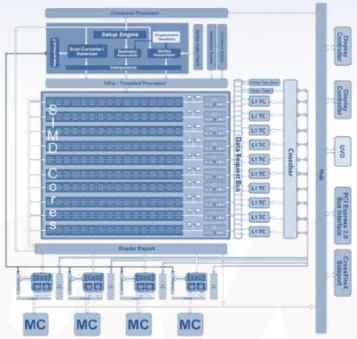
X800 Architecture



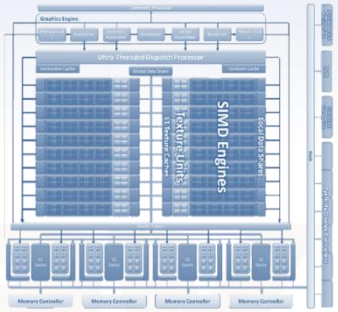
R600 Architecture



RV770 Architecture



Cypress Architecture



Diagrams and information from sources including: GPU Computing: Past, Present, and Future, ATI Stream SDK OpenCL Programming Guide, AMD: Unleashing the Power of Parallel Compute, The RV870 Story: AMD Showing up to the Fight, Wikipedia

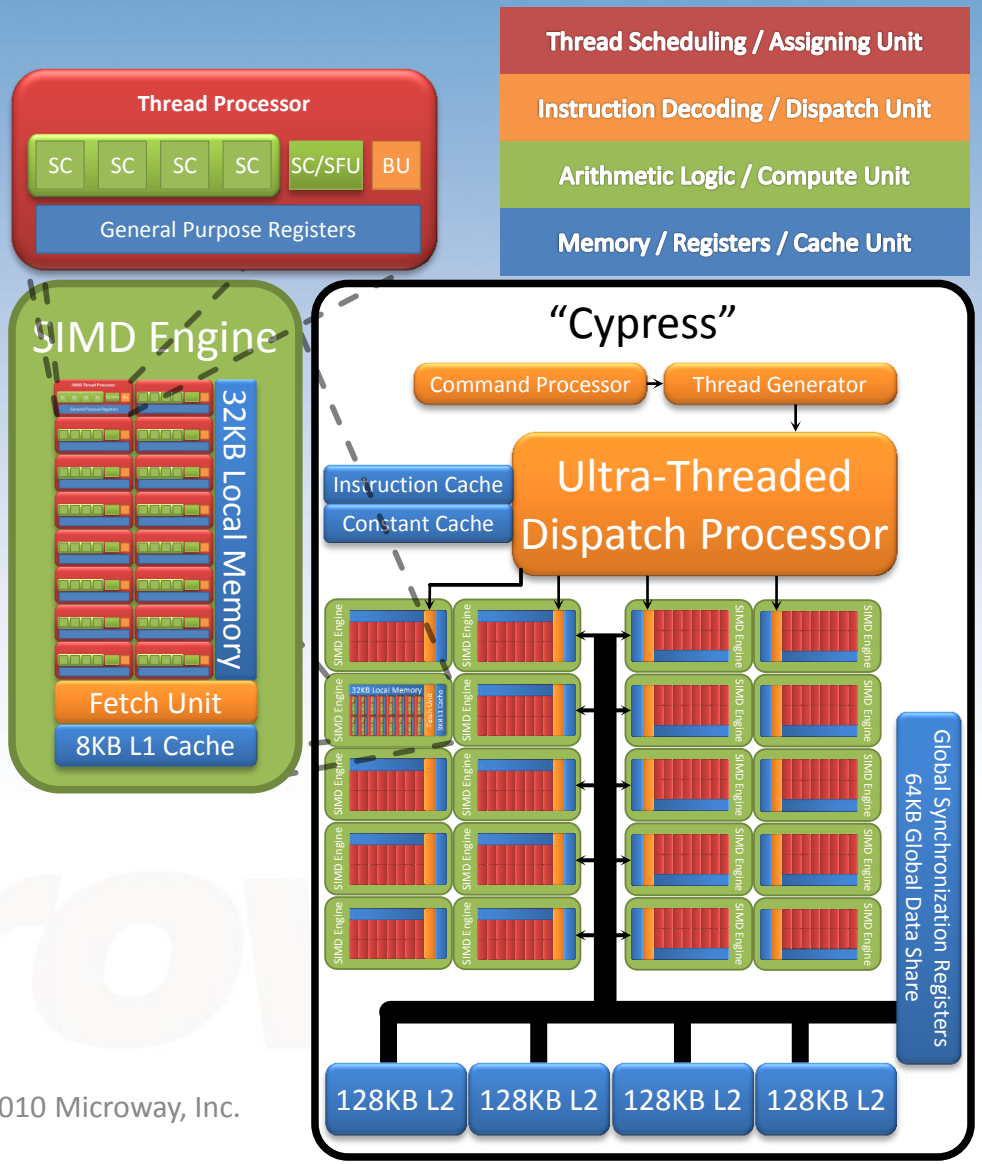
ATI Evergreen GPU Architecture

- Compute Units
  - SIMD Engines
  - BU = Branch Unit
  - SFU = Special Function Unit
  - SC = Stream Core

Group of 4 SC's can produce per clock:

4x 32-bit FP Fused MUL + ADD
or 2x 64-bit DP MUL or ADD
or 1x 64-bit DP Fused MUL + ADD
or 4x 24-bit Integer MUL or ADD+

- 20 SIMD Engines * 16 Thread Processors * 5 SCs = 1600 SIMD Stream Cores



ATI Radeon HD5870

GPU Information (Evergreen / Cypress XT)

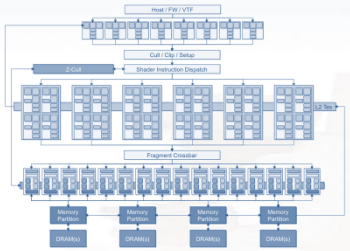
1600 cores in 20 SIMD Engines
850 MHz core clock
32KB Registers per SIMD
8KB L1 Cache per SIMD
32KB * 20 = 640KB local memory
8KB * 20 = 160KB L1 cache
64 KB Shared memory (total)
128 KB * 4 banks = 512KB L2 cache
1 GB global memory
GDDR5 – 1200 MHz, 153.6 GB/s
2.15 Billion transistors

History of NVIDIA GPUs with CUDA
Unified Graphics and Computing Architecture

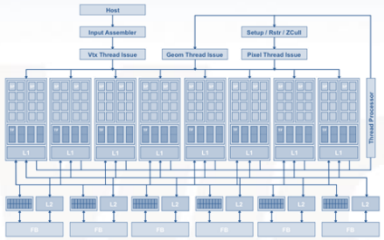
CUDA Compute Capability	GPU Chip Series	TPCs (Fully populated)	MPs per TPC (see note)	Compute cores per Multiprocessor: SP	DP	SFU	Warps (Groups of 32 Threads) active per MP	Atomic Operations	Memory Access and Coalescing	Compute cores per GPU (Fully populated) SP	DP	SFU
1.0	G80	8	2	8	0	2	24	None	16 thread alignment or penalty incurred	128	0	32
1.1	G84/G92	8	2	8	0	2	32	32-bit		128	0	32
1.3	GT200	10	3	8	1	2	32	32 and 64-bit	Degrades gracefully if unaligned	240	30	60
2.0	GF100	16	1	32	16	4	64 (2 Warp Schedulers per MP)	32 and 64-bit	Cache used to prevent degradation	512	256	64
				16 Load / Store						256 Load / Store		

Note: Thread Processing Clusters (TPCs) in the GPU units below combine MultiProcessors (MPs) in the "Scalable Processor Array" Framework with other graphics related units to do graphics processing. Some of these units are repurposed with different names to do High Performance Computation (HPC), and some new units are added. The number of TPCs in each product implementation as well as computational features may vary by market segment (budget, gaming, HPC, CAD, etc). For example, the GTX 260 has 9 of the 10 TPCs → 27 of 30 MPs → 216 of the 240 cores, and the GTX 480 has 15 of 16 TPCs → 15 of 16 MPs → 480 of 512 cores.

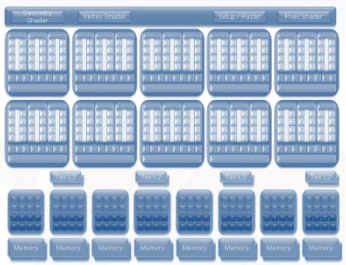
GeForce 7800 (Pre Unified Shader Architecture)



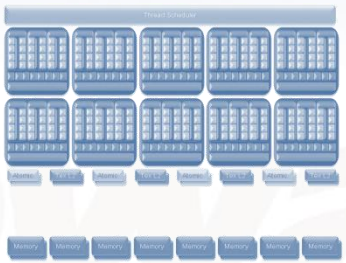
GeForce 8800 (First generation Unified Shader / Unified Graphics and Computing Architecture)



GeForce GTX 280 Graphics Processing Architecture



GeForce GTX 280 Parallel Computing Architecture



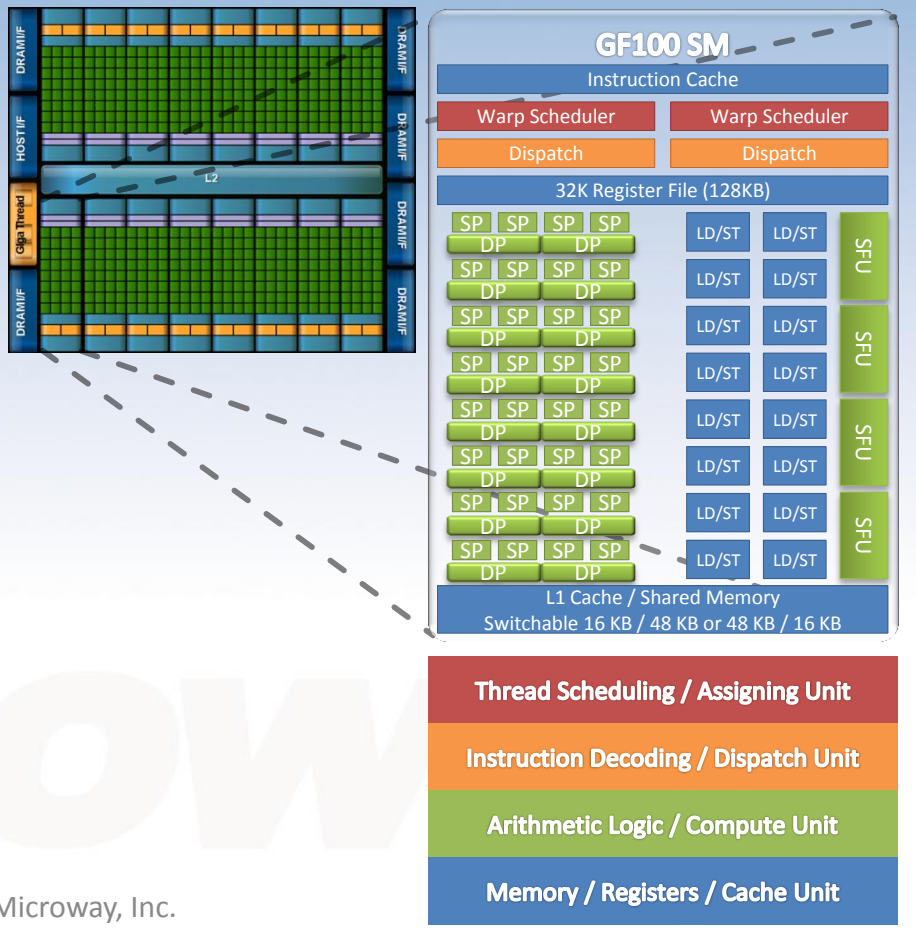
GF100 "Fermi" Architecture

Diagrams and information from sources including: Getting Started with CUDA, CUDA Optimization, NVIDIA CUDA Programming Guide 3.0, NVIDIA Fermi Compute Architecture Whitepaper, Wikipedia

NVIDIA Fermi/GF100 GPU Architecture

- Compute Unified Device Architecture
- 1 Warp = 32 Threads
- 32 max warps active per Warp Scheduler
  1024 threads active at once per Scheduler
  Actual number of threads managed depends on amount of memory used per thread
- Compute and Memory Units
  - SP = Streaming Processor - CUDA Core
  - LD/ST = Load/Store Unit
  - SFU = Special Function Unit
  - DP = Double Precision Logic Unit
- Global memory latency is on the order of 400 to 800 cycles. Optimal HPC performance is achieved by hiding latency in calculations by organizing warps in a memory coherent manner.

Streaming Multiprocessors (SM)



NVIDIA Tesla C2050

GPU Information (GF100 Architecture)

448 CUDA cores in 14 SMs

1.15 GHz core clock

128 KB registers per SM

16+48 KB shared memory/L1 per SM

14*64KB = 896KB shared memory/L1

768 KB L2 cache

3 GB or 6 GB global memory

GDDR5 – 1.55GHz, 148 GB/s

ECC: 2.625GB or 5.25GB

3.0 Billion Transistors

NVIDIA Tesla C1060

GPU Information (GT200 Architecture)

240 CUDA cores in 30 SMs

1.296 GHz core clock

32 KB registers per SM

32 KB shared memory per SM

30*32KB = 960KB shared memory

768 KB L2 cache

4 GB global memory

GDDR3 – 800 MHz, 102.4 GB/s

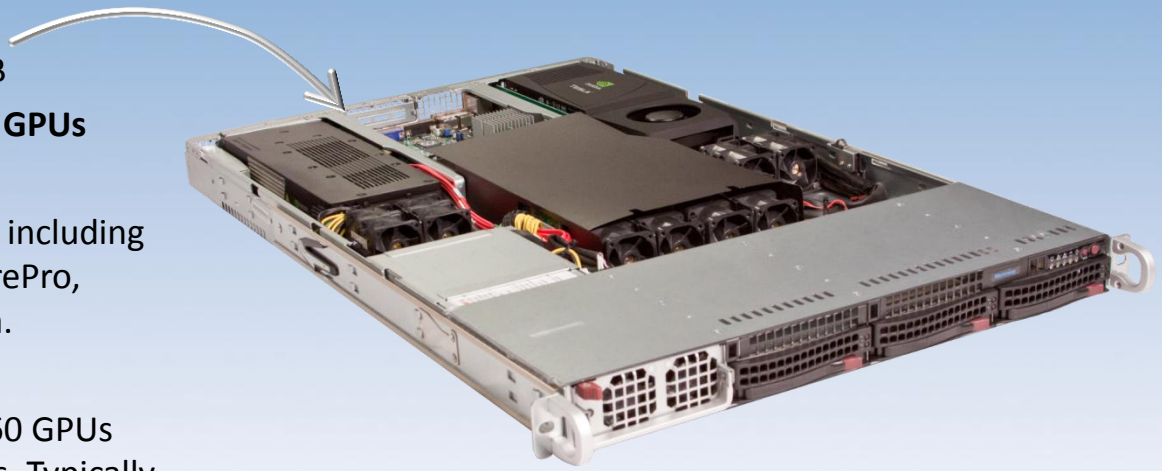
1.2 Billion transistors

Microway Multiple GPU Servers with CPUs in the Same Chassis

Solution Name	# of GPUs	# of CPUs (Nodes)	Solution size	PCIe lanes from CPUs	Notes
1U GPU Server	2	2 CPUs (1 Node)	1U	32	Full PCIe bandwidth per GPU, 2x AMD/Intel CPUs + Half-height PCIe Gen 2 x8/x4 expansion card
4U GPU Server	4	2 CPUs (1 Node)	4U	64	Full PCIe bandwidth per GPU
Microway OctoPuter™	8	2 CPUs (1 Node)	4U	64	8 GPUs on 1 PCIe bus + 4x 16 to 32 PCIe switches Supports ATI or NVIDIA

Microway 1U GPU Server

- 2x PCIe x16 devices
- 2 AMD / 2 Intel CPUs
- Slim PCIe x16 expansion slot (x8 on AMD / x4 on Intel) for IB
- **Full PCIe x16 Bandwidth to all GPUs**



- Supports ATI and NVIDIA GPUs including ATI Radeon, FireStream and FirePro, NVIDIA GTX, Quadro, and Tesla.

- Shown here with 2x Tesla C1060 GPUs and 2x Intel CPUs under baffles. Typically configured with “passively cooled” NVIDIA Tesla M1060 GPUs.

Microway 4U GPU Server or Workstation

- 4x PCIe x16 devices and 2CPUs in 4U
- **Full PCIe x16 Bandwidth to all GPUs**
- Supports replacing individual GPUs
- Shown with 4x Tesla C1060 GPUs and workstation accessories
- Additional PCIe x4 slot supports DDR InfiniBand

Workstation model shown. Server chassis replaces front panel, top cover, and bottom feet with mounting rails in horizontal orientation.

Microway Whisperstation - PSC

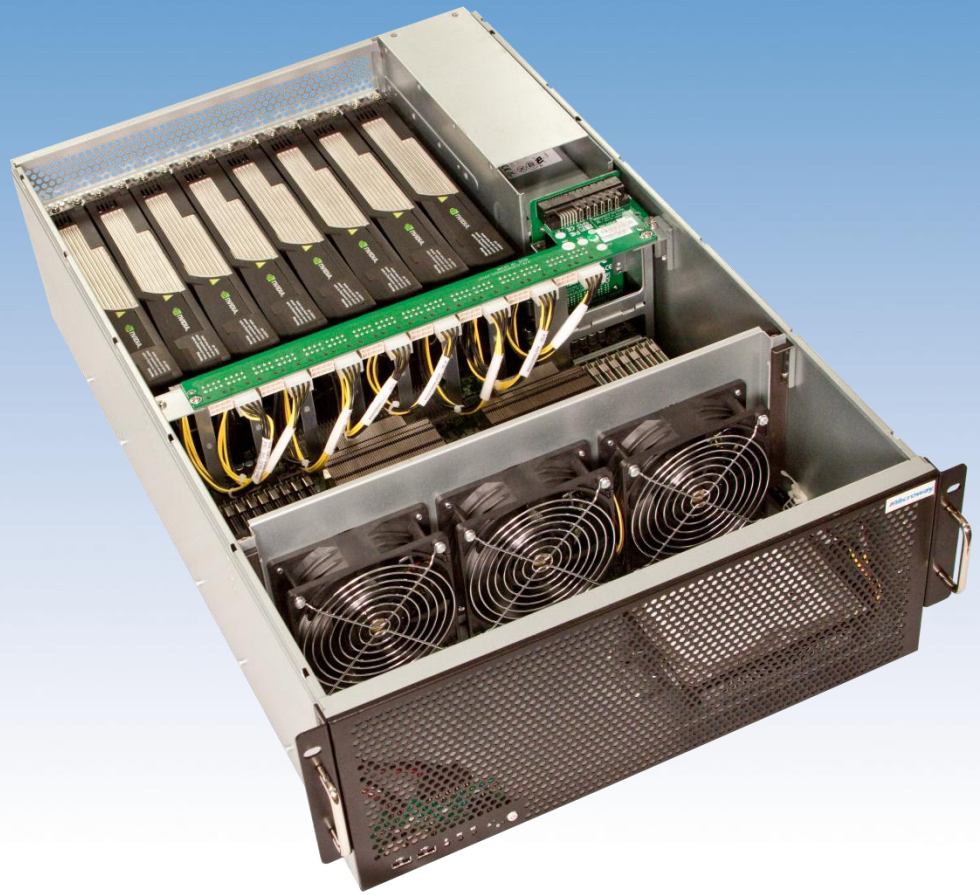
- 4x PCIe x16 devices and 2CPUs in 4U
- **Full PCIe x16 Bandwidth to all GPUs**
- Supports replacing individual GPUs
- Tesla C2050 support (shown) includes Microway custom cooling
- Shown with 5 SATA HDDs in RAID configuration and fully populated memory

Microway OctoPuter™

- 4U Chassis
- 2 Chipsets deliver 64 lanes of PCIe
- 4 PCIe 16 to 32 lane switches deliver 128 lanes to 8 PCIe x16 slots

- **8 GPUs per Node**
- 2 + 1 redundant 1250W Power
- Highest GPU/CPU density in 4U

- Unpopulated GPU slots can also be used for RAID or InfiniBand HCAs.
- Note: It is possible to drive 4x S1070s with one OctoPuter compute node for 16 GPUs in one compute node for embarrassingly parallel applications with very low bandwidth to compute ratios.



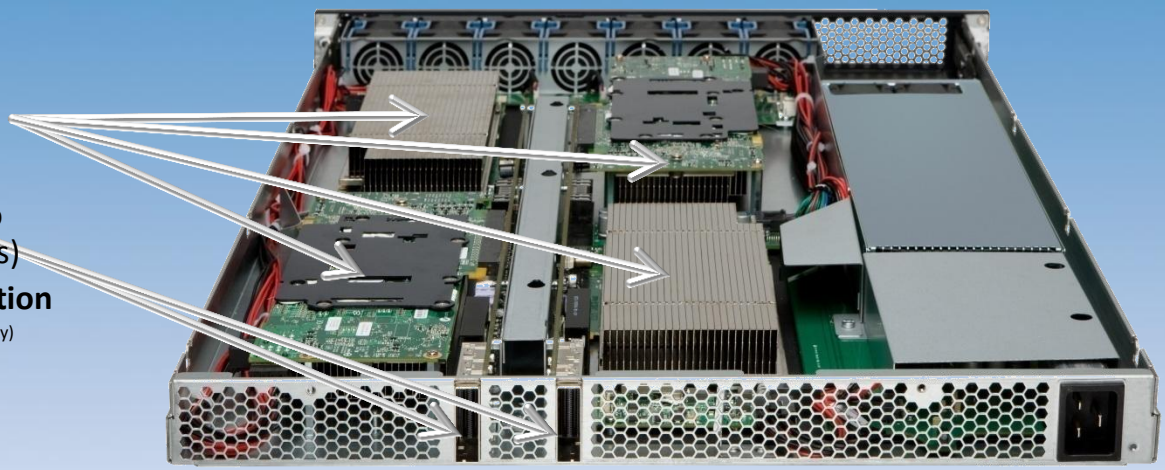
Microway Multiple GPU Servers with NVIDIA GPUs in Separate Chassis

Solution Name	# of GPUs	# of CPUs (Nodes)	Solution size	PCIe lanes from CPUs	Notes
S1070 / S2050 + 1U GPU Server	4	2 CPUs (1 Node)	2U	32	1 Node with 4 GPUs
S1070 / S2050 + 1U Twin Server	4	4 CPUs (2 Nodes)	2U	32	2 Nodes with 2 GPUs each
S1070 / S2050 + 2U Server	4	2 CPUs (1 Node)	3U	32	2 additional slim x16 cards can be used for RAID, IB, and graphics. 2U node MBs offer more memory.
2x S1070 / S2050 + 2U Twin^2 Server	8	8 CPUs (4 Nodes)	4U	64	2x 1U Twin Server solution + Redundant power supplies, hot swappable nodes
2x S1070 / S2050 + 2U Server	8	2 CPUs (1 Node)	4U	64	More contention between GPUs and CPUs due to PCIe switch configuration.

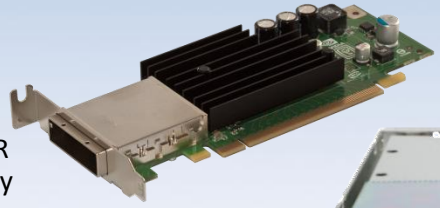
NVIDIA Tesla S1070

- 4x Tesla GPUs in 1U Chassis
- 2x PCIe 16 to 32 lane switches
- 2x PCIe x16 Cables connected to external chassis (1U - 4U options)
- **2 GPUs sharing one x16 connection**

(Pictures courtesy of NVIDIA, Opening S1070 voids warranty)



- With Microway 1U GPU Server:
  4 GPUs per Node (2 CPUs) in 2U
  InfiniBand HCA Option (PCIe x8 or x4)

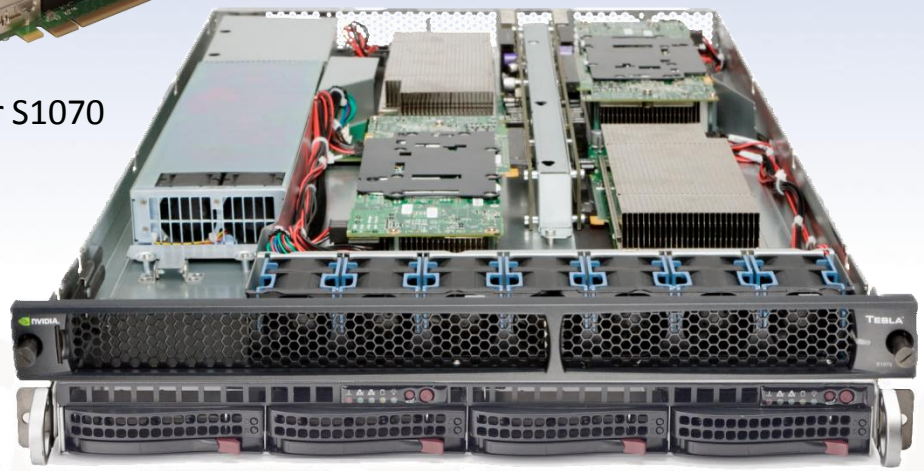


HBA for S1070

- With Microway 2U Server:
  4 GPUs per Node (2CPUs) in 3U with QDR InfiniBand HCA Option and more memory and HDD capacity.

- 2x S1070 with Microway 4U Server:
  8 GPUs per Node(2CPUs) in 6U with DDR InfiniBand HCA Option (PCIe x4)

- With 1U Twin or 2U Twin^2:
  2 GPUs per Node (2 CPUs) in 2U or 4U with integrated DDR or QDR HCAs on MB 2U Twin^2 MBs are hot-swappable



S1070 with 1U Twin (Popular Configuration)

Cluster Software Design Considerations

- Optimal GPGPU performance requires a large investment in design, development, and testing of software algorithms against actual hardware implementations to realize the huge potential of these special purpose processors. In general, this process can take 6 to 24 months.
- Problems which run well on clusters tend to be well suited for GPUs. Memory utilization and PCIe latency are the current bottlenecks that determine how many operations can execute in parallel and if going to GPUs will be faster than using multi-core CPUs in a cluster for any given problem.
- The ATI GPGPUs presented have a rate of 5 SP FLOP per DP FLOP. The NVIDIA GPUs post G80 and pre Fermi have a ratio of 8 to 1. The Fermi architecture supports a 2 to 1 ratio for the Tesla C2050, but the GTX 470 and GTX 480 have an 8 to 1 ratio like the GTX 200 series.
- Carefully consider your algorithm's need for DP vs. SP when choosing GPUs for a cluster. Also carefully consider the amount and type of RAM per card. The Tesla C1060 has 4GB DDR3, the Tesla C2050 has 3GB DDR5, the Radeon HD5970 has 2GB DDR5, and the Radeon HD5870 has 1GB DDR5.
- Most HPC market developers are currently using C for CUDA 3.0 because of the large amount of available software packages supporting it such as Accelereyes Jacket for MATLAB. While OpenCL offers the promise of "heterogeneous computing" (i.e. CPU and GPU and other Accelerators), you still need to carefully optimize your code to achieve the full potential on all devices you intend to support.

Cluster Hardware Design Considerations

- New GPUs come out every 6 - 12 months (holiday seasons + multiple competitors), so it's more important to optimize your software for a hardware architecture. GPU clock rates, memory transfer rates, sizes of global memory, and number of engines/TPCs vary by GPU.
- Carefully balance your budget between optimizing current gen software and purchasing next gen hardware. Consider your software budget and your operational expenses including power consumption and cooling over the lifetime of the hardware when purchasing new hardware. Hardware and the software targeting it become obsolete very quickly.
- Multiple GPU cards like the Radeon HD5970 and GTX 295 sacrifice PCIe bandwidth per GPU by using a switch. These GPUs are optimized for graphics rendering / gaming, and multiple GPU benefits on them apply to fewer HPC applications. Use Tesla / Quadro and FireStream / FirePro cards for HPC applications.
- A cluster is only as fast as its slowest node. Old hardware does not get cheaper. Used hardware gets cheaper. It is difficult to find new sources of the last generation of hardware. Memory prices fluctuate a lot.
- Avoid mixing new hardware with old hardware when designing clusters. Newer nodes with exponentially higher computing power and/or different network interfaces will be wasting utilization waiting on slower nodes.

Disclaimer and Attribution

The information presented in this document is provided “AS IS” and for informational purposes only. Microway makes no representations or warranties with respect to the contents hereof and assumes no responsibility for any inaccuracies, errors, or omissions that may appear in this information.

The opinions expressed herein are solely those of the author, and are not authorized by NVIDIA or AMD.

All trademarks are the property of their respective owners in the USA and abroad.

Questions or comments?

Email gpgpu-info@microway.com or

Call Microway at (508) 746-7341