TESLA V100

The Fastest and Most Productive GPU for Deep Learning and HPC

Volta Architecture
Most Productive GPU

Improved NVLink & HBM2
Efficient Bandwidth

Volta MPS
Inference Utilization

Improved SIMT Model
New Algorithms

Tensor Core
120 Programmable TFLOPS Deep Learning
TESLA V100

21B transistors
815 mm²

80 SM
5120 CUDA Cores
640 Tensor Cores

16 GB HBM2
900 GB/s HBM2
300 GB/s NVLink

*full GV100 chip contains 84 SMs*
## GPU PERFORMANCE COMPARISON

<table>
<thead>
<tr>
<th></th>
<th>P100</th>
<th>V100</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training acceleration</td>
<td>10 TOPS</td>
<td>120 TOPS</td>
<td>12x</td>
</tr>
<tr>
<td>Inference acceleration</td>
<td>21 TFLOPS</td>
<td>120 TOPS</td>
<td>6x</td>
</tr>
<tr>
<td>FP64/FP32</td>
<td>5/10 TFLOPS</td>
<td>7.5/15 TFLOPS</td>
<td>1.5x</td>
</tr>
<tr>
<td>HBM2 Bandwidth</td>
<td>720 GB/s</td>
<td>900 GB/s</td>
<td>1.2x</td>
</tr>
<tr>
<td>NVLink Bandwidth</td>
<td>160 GB/s</td>
<td>300 GB/s</td>
<td>1.9x</td>
</tr>
<tr>
<td>L2 Cache</td>
<td>4 MB</td>
<td>6 MB</td>
<td>1.5x</td>
</tr>
<tr>
<td>L1 Caches</td>
<td>1.3 MB</td>
<td>10 MB</td>
<td>7.7x</td>
</tr>
</tbody>
</table>
NEW HBM2 MEMORY ARCHITECTURE

HBM2 stack

1.5x Delivered Bandwidth

STREAM: Triad Delivered GB/s

P100
76% DRAM Utilization

V100
95% DRAM Utilization

V100 measured on pre-production hardware.
VOLTA NVLINK

300GB/sec
50% more links
28% faster signaling
PROGRAMMABILITY
PASCAL UNIFIED MEMORY

GPU Optimized State

Page Migration Engine

CPU Optimized State

GPU

CPU

Unified Memory

Memory
VOLTA + PCIE CPU UNIFIED MEMORY

Page Migration Engine

+ Access counters
VOLTA + NVLINK CPU UNIFIED MEMORY

Page Migration Engine

+ Access counters
+ New NVLink Features
  (Coherence, Atomics, ATS)
VOLTA MULTI-PROCESS SERVICE

Volta MPS Enhancements:

- Reduced launch latency
- Improved launch throughput
- Improved quality of service with scheduler partitioning
  - More reliable performance
- 3x more clients than Pascal
NEW SM MICROARCHITECTURE
VOLTA GV100 SM

GV100

- FP32 units: 64
- FP64 units: 32
- INT32 units: 64
- Tensor Cores: 8
- Register File: 256 KB
- Unified L1/Shared memory: 128 KB
- Active Threads: 2048
VOLTA GV100 SM
Redesigned for Productivity

Completely new ISA
Twice the schedulers
Simplified Issue Logic
Large, fast L1 cache
Improved SIMT model
Tensor acceleration

The easiest SM to program yet
RECAP: PASCAL L1 AND SHARED MEMORY

- **Shared Memory**: 64 KB
- **L1$**: 24 KB
- **L2$**: 4 MB

**Load/Store Units**

- **Pascal SM**
- **Low Latency**
- **Streaming**: *Unlimited* cache misses in flight
UNIFYING KEY TECHNOLOGIES

Pascal SM
- Load/Store Units
- Shared Memory 64 KB
- L1$ 24 KB
- L2$ 4 MB

Volta SM
- Load/Store Units
- L1$ and Shared Memory 128 KB
- L2$ 6 MB

Low Latency
Streaming
VOLTA L1 AND SHARED MEMORY

Volta Streaming L1$:  
Unlimited cache misses in flight  
Low cache hit latency  
4x more bandwidth  
5x more capacity

Volta Shared Memory:  
Unified storage with L1  
Configurable up to 96KB
NARROWING THE SHARED MEMORY GAP with the GV100 L1 cache

Cache: vs shared
- Easier to use
- 90%+ as good

Shared: vs cache
- Faster atomics
- More banks
- More predictable

Directed testing: shared in global

Average Shared Memory Benefit

<table>
<thead>
<tr>
<th></th>
<th>Pascal</th>
<th>Volta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benefit</td>
<td>70%</td>
<td>93%</td>
</tr>
</tbody>
</table>
INDEPENDENT THREAD SCHEDULING
VOLTA: INDEPENDENT THREAD SCHEDULING

Communicating Algorithms

Pascal: Lock-Free Algorithms
Threads cannot wait for messages

Volta: Starvation Free Algorithms
Threads may wait for messages
The SIMT model: enable thread-parallel programs to execute with vector efficiency

<table>
<thead>
<tr>
<th></th>
<th>CPU</th>
<th>Pascal GPU</th>
<th>Volta GPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thread-parallelism</td>
<td>MIMD</td>
<td>SIMT (lock-free)</td>
<td>SIMT</td>
</tr>
<tr>
<td>Data-parallelism</td>
<td>SIMD</td>
<td>SIMT</td>
<td>SIMT</td>
</tr>
</tbody>
</table>
VOLTA TENSOR CORE
TENSOR CORE

Mixed Precision Matrix Math
4x4 matrices

\[ D = \begin{pmatrix}
  A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} \\
  A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} \\
  A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} \\
  A_{3,0} & A_{3,1} & A_{3,2} & A_{3,3}
\end{pmatrix}
\begin{pmatrix}
  B_{0,0} & B_{0,1} & B_{0,2} & B_{0,3} \\
  B_{1,0} & B_{1,1} & B_{1,2} & B_{1,3} \\
  B_{2,0} & B_{2,1} & B_{2,2} & B_{2,3} \\
  B_{3,0} & B_{3,1} & B_{3,2} & B_{3,3}
\end{pmatrix} + \begin{pmatrix}
  C_{0,0} & C_{0,1} & C_{0,2} & C_{0,3} \\
  C_{1,0} & C_{1,1} & C_{1,2} & C_{1,3} \\
  C_{2,0} & C_{2,1} & C_{2,2} & C_{2,3} \\
  C_{3,0} & C_{3,1} & C_{3,2} & C_{3,3}
\end{pmatrix}

D = AB + C
TENSOR SYNCHRONIZATION

Full Warp 16x16 Matrix Math

Warp-synchronizing operation

Composed Matrix Multiply and Accumulate for 16x16 matrices

Result distributed across warp
TESLA V100

More V100 Features: 2x L2 atomics, int8, new memory model, copy engine page migration, and more ...

The Fastest and Most Productive GPU for Deep Learning and HPC