NVIDIA TESLA V100 GPU 架构

全球领先的数据中心 GPU

WP-08608-001_v1.1 | 2017 年 8 月

目录

NVIDIA Tesla V100 GPU 架构简介 1
Tesla V100：AI 计算和 HPC 的动力之源 2
主要特性 2
为 AI 和 HPC 实现出众性能 5
NVIDIA GPU – 兼具出众速度与灵活性的深度学习平台 6
深度学习背景知识 6
GPU 加速的深度学习 7
深入了解 GV100 GPU 硬件架构 8
卓越的性能与能效 11
Volta 流多处理器 12
Tensor 核心 14
L1 数据缓存和共享内存的性能提升 17
同时执行 FP32 和 INT32 运算 18
计算能力 18
NVLink：更高带宽、更多链路、更多功能 19
更多的链路、更快的链接速度 19
更多功能 20
HBM2 内存架构 22
ECC 内存弹性 23
复制引擎增强 23
Tesla V100 主板设计 24
GV100 CUDA 硬件和软件架构改进 26
独立线程调度 27
NVIDIA 早期 GPU SIMT 模型 27
Volta SIMT 模型 28
无饥饿现象算法 30
VOLTA 多进程服务 31
统一内存寻址和地址转换服务 33
协作组 34

结束语 37
附录 A 搭载 Tesla V100 的 NVIDIA DGX-1 38
NVIDIA DGX-1 系统规格 39
DGX-1 软件 40
附录 B NVIDIA DGX 工作站 - 适用于深度学习的个人 AI 超级计算机 42
预加载最新的深度学习软件 44
推动 AI 计划 44
附录 C 通过 GPU 为深度学习和人工智能加速 45
深度学习概述 45
NVIDIA GPU：深度学习的引擎 48
训练深度神经网络 49
使用训练的神经网络进行推理 50
综合性深度学习软件开发工具包 51
自动驾驶汽车 52
机器人 53
医疗保健和生命科学 54

插图目录

图 1. 配备 Volta GV100 GPU 的 NVIDIA Tesla V100 SXM2 模块 1

图 2. Tesla V100 中的新技术 4

图 3. 配备全新 Tensor 核心的 Tesla V100 为深度学习性能实现重大飞跃 5

图 4. 含 84 个 SM 单元的完整 Volta GV100 GPU 9

图 5. Volta GV100 流多处理器 (SM) 13

图 6. cuBLAS 单精度 (FP32) 14

图 7. cuBLAS 混合精度（FP16 输入，FP32 计算） 15

图 8. Tensor 核心 4x4 矩阵乘积累加运算 15

图 9. Tensor 核心中的混合精度乘积累加运算 16

图 10. Pascal 和 Volta 4x4 矩阵乘法运算 16

图 11. Pascal 与 Volta 数据缓存的比较 17

图 12. "配备 V100 的 DGX-1"中使用的混合立体网络 NVLink 拓扑 20

图 13. V100 与以 NVLink 连接的 GPU 至 GPU 和 GPU 至 CPU 通信 21

图 14. 第二代 NVLink 性能 21

图 15. V100 上 HBM2 内存加速与 P100 的对比 22

图 16. Tesla V100 加速器（正面） 24

图 17. Tesla V100 加速器（背面） 24

图 18. NVIDIA Tesla V100 SXM2 模块 - 非写实剖析图 25

图 19. 使用 CUDA 开发的深度学习方法 26

图 20. Pascal 和早期 GPU 的 SIMT 线程束执行模型 27

图 21. Volta 线程束与每线程的程序计数器和调用栈 28

图 22. Volta 独立线程调度 29

图 23. 程序采用显式同步重新收敛线程束中的线程 29

图 24. 包含细粒度锁的双重链接列表.....30
图 25. Pascal 中基于软件的 MPS 服务与 Volta 中硬件加速 MPS 服务的对比.....32
图 26. 用于推理的 Volta MPS.....33
图 27. 粒子模拟的两个阶段.....36
图 28. NVIDIA DGX-1 服务器.....38
图 29. 与基于 GP100 的八路服务器相比，DGX-1 将训练速度提高了 3 倍.....39
图 30. NVIDIA DGX-1 全面集成的软件堆栈，可即时提高生产力.....41
图 31. 配备 Tesla V100 的 DGX Station.....42
图 32. NVIDIA DGX 工作站可将训练速度提高 47 倍.....43
图 33. 感知器是最简单的神经网络模型.....46
图 34. 复杂的多层神经网络模型需要更高的计算能力.....48
图 35. 训练神经网络.....49
图 36. 神经网络推理.....50
图 37. 为每个框架加速.....51
图 38. 在深度学习方面与 NVIDIA 合作的组织.....52
图 39. NVIDIA DriveNet.....53

表格列表

表 1. NVIDIA Tesla 系列各 GPU 比较.....10
表 2. 计算能力：GK180、GM200、GP100 与 GV100 之间的对比.....18
表 3. NVIDIA DGX-1 系统规格.....39
表 4. DGX 工作站规格.....43

NVIDIA TESLA V100 GPU 架构简介

自 10 年前推出开创性的 CUDA GPU 计算平台以来，NVIDIA® GPU 的每一次更新换代都将应用程序性能和能效推升到更高水平，同时还新增若干重要的计算功能，并简化 GPU 编程难度。如今，NVIDIA GPU 为数千个高性能计算 (HPC)、数据中心和机器学习应用程序提供加速动力。NVIDIA GPU 已成为推动人工智能 (AI) 革命的领先计算引擎。

NVIDIA GPU 可为大量深度学习系统和应用程序加速，包括自动驾驶车辆平台、高精度语音、图像和文本识别系统、智能视频分析、分子模拟、药物发现、疾病诊断、天气预测、大数据分析、财务建模、机器人、工厂自动化、实时语言翻译、在线搜索优化以及个性化用户推荐等等，不胜枚举。

全新 NVIDIA® Tesla® V100 加速器（如图 1 所示）搭载强大的新款 Volta™ GV100 GPU。GV100 不仅汲取了上一代产品 Pascal™ GP100 GPU 的精进之处，而且还显著提高性能和可扩展性，并新增许多能够改进可编程性的功能。这些改进将有效提升 HPC、数据中心、超级计算机以及深度学习系统和应用程序的性能。

本白皮书将针对 Tesla V100 加速器和 Volta GV100 GPU 架构进行介绍。

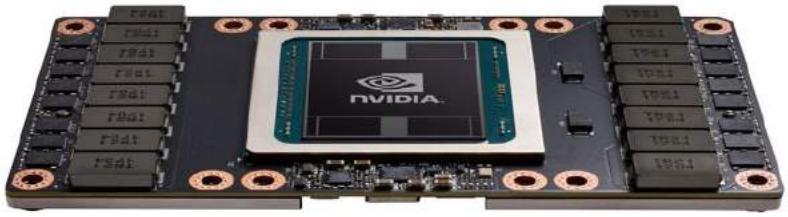


图 1.　配备 Volta GV100 GPU 的 NVIDIA Tesla V100 SXM2 模块

TESLA V100：AI 计算和 HPC 的动力之源

NVIDIA Tesla V100 加速器性能在全球并行处理器中堪称出类拔萃，设计目的旨在处理计算量巨大的 HPC、人工智能和图形工作负载。

GV100 GPU 包含 211 亿根晶体管，芯片大小为 815 mm^2。采用专为 NVIDIA 定制的全新 TSMC 12 nm FFN (FinFET NVIDIA) 高性能制造工艺精心打造而成。相比上一代 Pascal GPU，GV100 的计算性能显著提高，并且增加许多新功能。除了进一步简化 GPU 编程和应用程序移植，GV100 还可提高 GPU 资源利用率。GV100 是一款极为节能的处理器，可实现出色的性能功耗比。

主要特性

以下为 Tesla V100 部分主要计算特性：

- 专为深度学习优化的全新流多处理器 (SM) 架构

  Volta GPU 中央配备有全新设计的 SM 处理器架构。全新 Volta SM 的节能效率相较上一代 Pascal 产品提升 50%，在同一功率电路下可显著提高 FP32 和 FP64 的性能。专为深度学习设计的新 Tensor 核心在训练方面可提供高达 12 倍的 TFLOPS 峰值，而在推理方面则可提供 6 倍的 TFLOPS 峰值。此外，通过使用单独的并行整数和浮点数据路径，Volta SM 在处理包含计算和寻址计算的混合工作负载时也更为高效。Volta 新式独立线程调度功能，可在并行线程之间实现更精细的同步与合作。最后是 L1 数据缓存和共享内存单元的全新组合，在大幅提升性能之余更简化了编程。

- 第二代 NVIDIA NVLink™

  第二代 NVIDIA NVLink 高速互联功能提供更高带宽与更多链路，并可提升多 GPU 和多 GPU/CPU 系统配置的可扩展性。Volta GV100 最多支持六条 NVLink 链路，总带宽 300 GB/s，而 GP100 仅支持四条 NVLink 链路，总带宽仅为 160 GB/s。IBM Power 9 CPU 服务器借助 NVLink 可实现 CPU 主控和缓存一致性功能。搭载 V100 的 NVIDIA DGX-1 AI 超级计算机使用 NVLink 提高可扩展性，实现超快速的深度学习训练。

- HBM2 内存：高速、高效

  Volta 拥有经重点调整的 16 GB HBM2 内存子系统，可提供 900 GB/s 的内存带宽峰值。新一代 Samsung HBM2 内存与新一代 Volta 内存控制器的结合，能够提供比 Pascal GP100 高 1.5 倍的内存带宽，而运行多工作负载时的内存带宽利用率可达 95%。

- Volta 多进程服务

  Volta 多进程服务 (MPS) 是 Volta GV100 架构的新功能，可为 CUDA MPS 服务器的关键组件实现硬件加速，从而为共享 GPU 的多个计算应用程序提高性能、实现隔离并改进服务质量 (QoS)。此外，Volta MPS 还使 MPS 客户端的最大数量增至 3 倍，从 Pascal 时的 16 个增加到 Volta 的 48 个。

- 统一内存寻址和地址转换服务质量提升

  GV100 统一内存寻址技术包含新的存取计数器，可更准确地将内存分页迁移至对其读取最为频繁的处理器，同时提升处理器间共享内存范围的效率。在 IBM Power 平台上，新的地址转换服务 (ATS) 支持允许 GPU 直接读取 CPU 的分页表。

- 最大性能模式和最大效率模式

  最大性能模式下，Tesla V100 加速器将以 300 W 的 TDP（热设计功耗）级别运行，为需要最快计算速度和最高数据吞吐量的应用程序加速。在最大效率模式下，数据中心管理员可调节 Tesla V100 加速器的功率利用率，使加速器以最佳性能功耗比运行。可为同机架的所有 GPU 设置功率上限，从而大幅降低功耗，并使机架设备保持出色的性能。

- 协作组和新的协作启动 API

  协作组是 CUDA 9 中引入的新式编程模型，可用于组织线程通信群组。协作组允许开发者表示线程通信粒度，帮助他们表达更丰富、更高效的并行分解方法。自 Kepler 起的所有 NVIDIA GPU 均支持基本协作组功能。Pascal 和 Volta 包含对新协作启动 API 的支持，可在 CUDA 线程块中实现同步。Volta 添加了对全新同步模式的支持。

- 针对 Volta 优化的软件

  Caffe2、MXNet、CNTK、TensorFlow 等深度学习框架新版本以及其他框架皆可发挥出 Volta 的强大性能，缩短训练时间并获得更高的多节点训练性能。Volta 优化版的 GPU 加速库（如 cuDNN、cuBLAS 和 TensorRT）能够充分利用 Volta GV100 架构的新功能，为深度学习推理和高性能计算 (HPC) 应用程序提供更高性能。NVIDIA CUDA 工具包 9.0 版包含新的 API 以及对 Volta 功能的支持，并可实现更轻松的编程。

图 2 展示了整合入 Tesla V100 的新技术。

图 2.　Tesla V100 中的新技术

为 AI 和 HPC 实现出众性能

Tesla V100 可提供行业领先的浮点和整数性能。以下是计算速率峰值。图 3 展示 Tesla V100 配备全新 Tensor 核心后在深度学习中的出色性能。

- 7.8 TFLOPS[1] 的双精度浮点 (FP64) 性能
- 15.7 TFLOPS[1] 的单精度 (FP32) 性能
- 125 Tensor TFLOPS[1]

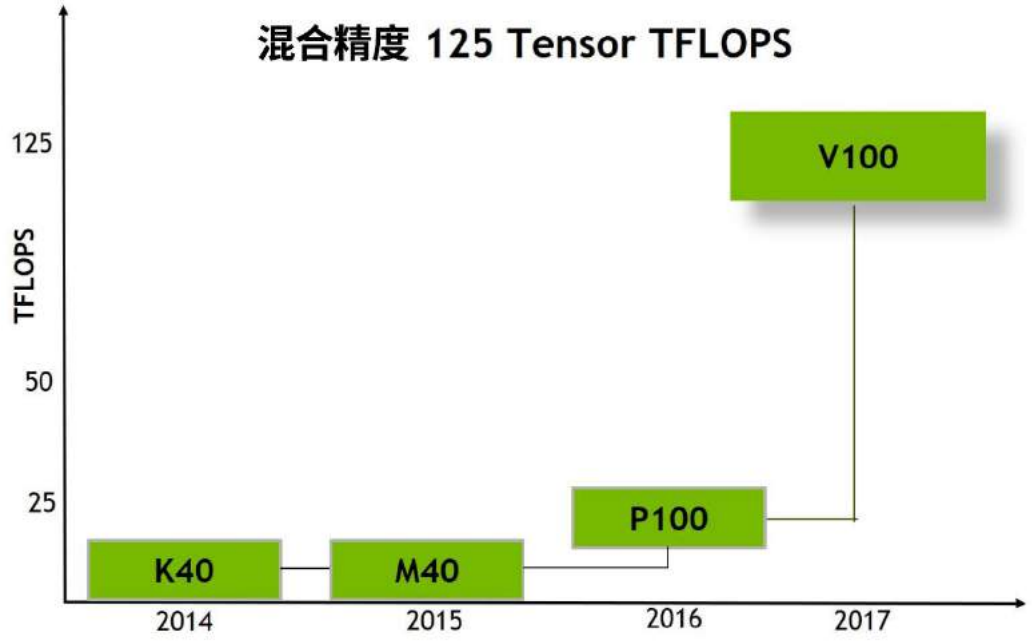


图 3.　配备全新 Tensor 核心的 Tesla V100 为深度学习性能实现重大飞跃

[1] 基于 GPU 加速时钟.

NVIDIA GPU – 兼具出众速度与灵活性的深度学习平台

无论是单 GPU 还是多 GPU 系统，GPU 加速都能让深度学习训练和推理运算受益匪浅。过去一年来，NVIDIA Pascal GPU 广泛应用于各种深度学习系统的加速，在训练和推理方面的运行速度大大领先于 CPU。新的深度学习架构功能结合 NVIDIA Tesla V100 GPU 的超强计算性能，可有效提升神经网络训练和推理性能。此外，配备 NVLink 的多 GPU 系统在性能上拥有卓越的可扩展性。

灵活的 GPU 可编程性使新算法的快速开发和部署成为可能。NVIDIA GPU 提供出色的性能、可扩展性和可编程性，可满足 AI、深度学习系统和算法在训练和推理方面不断增加的需求。

深度学习背景知识

多年来，在人工智能领域，人们一直使用许多不同的方法来建模人类智能。机器学习是非常流行的 AI 研究方法，可训练系统以学习如何自主制定决策并预测结果。深度学习则是受人类大脑神经学习过程启发而兴起的机器学习技术。深度学习使用深度神经网络 (DNN)，之所以这么称呼是因为它本质上是由许多互联人工神经元（有时称为*感知器*）组成的深度网络层，可通过大量输入数据进行训练，从而以较高的精确度快速解决复杂问题。神经网络经过训练后，便可进行部署并用于识别和分类对象或图案（这个过程称为*推理*）。请参见本白皮书中第 45 页开始的附录 C，深入了解神经网络工作原理。

大多数神经网络都由多层互联的神经元构成。每个神经元和层都在任务（即此网络的训练目的）中起到作用。例如，AlexNet，即赢得 2012 年 ImageNet 比赛的卷积神经网络 (CNN)，包含八个层、650,000 个互联神经元以及接近六千万个参数。如今，神经网络的复杂程度亦与日俱增，深度残差网络（例如 ResNet-152）等最新的网络拥有 150 多个层，互联的神经元和参数更是增加了数百万之多。

GPU 加速的深度学习

学术界和业界普遍认为，NVIDIA GPU 是极为杰出的深度神经网络训练引擎，因为它的速度和能效均优于传统 CPU 平台。神经网络由大量相同的神经元构建而成，因此本质上具有高度并行性。这种并行性可很自然地映射到 GPU，相比于仅使用 CPU 的神经网络训练，GPU 的速度大幅增加。

神经网络非常依赖矩阵数学运算，复杂的多层网络需要出色的浮点计算性能和极高带宽才能提高效率和速度。GPU 拥有成千上万个专为矩阵数学运算而优化的处理核心，可提供数十到数百 TFLOPS 的性能。因此，GPU 绝对是基于深度神经网络的人工智能和机器学习应用程序的理想计算平台。

Volta 架构专为运行深度学习工作负载而打造，可在上一代架构相同功率预算内实现性能的大幅提高。下面的架构部分介绍了实现此目标的技术详情。

深入了解 GV100 GPU 硬件架构

NVIDIA Tesla V100 加速器配备 Volta GV100 GPU，是世界领先的高性能并行计算处理器。除了为 HPC 系统和应用程序提供更强的计算能力之外，GV100 还采用全新的硬件创新设计，可显著提升深度学习算法和框架的运行速度。

与上一代 Pascal GP100 GPU 一样，GV100 GPU 由多个 GPU 处理集群 (GPC)、纹理处理集群 (TPC)、流多处理器 (SM) 以及内存控制器组成。完整的 GV100 GPU 包含以下组件：

- 6 个 GPC
  每个 GPC 拥有：
  - 7 个 TPC（各包含两个 SM）
  - 14 个 SM
- 84 个 Volta SM
  每个 SM 拥有：
  - 64 个 FP32 核心
  - 64 个 INT32 核心
  - 32 个 FP64 核心
  - 8 个 Tensor 核心
  - 4 个纹理单元
- 8 个 512 位内存控制器（总共 4096 位）

含 84 个 SM 的**完整** GV100 GPU，总共拥有 5376 个 FP32 核心、5376 个 INT32 核心、2688 个 FP64 核心、672 个 Tensor 核心以及 336 个纹理单元。每个 HBM2 DRAM 堆栈由一对内存控制器控制。完整的 GV100 GPU 总共包含 6144 KB 的 L2 缓存。图 4 展示了含 84 个 SM 的

完整 GV100 GPU（不同产品的 GV100 配置可能各不相同）。Tesla V100 加速器拥有 80 个 SM。表 1 为过去五年的 NVIDIA Tesla GPU 之比较。

图 4.　含 84 个 SM 单元的完整 Volta GV100 GPU

表 1. NVIDIA Tesla 系列各 GPU 比较

Tesla 产品	Tesla K40	Tesla M40	Tesla P100	Tesla V100
GPU	GK180 (Kepler)	GM200 (Maxwell)	GP100 (Pascal)	GV100 (Volta)
SM 数量	15	24	56	80
TPC 数量	15	24	28	40
FP32 核心数/SM	192	128	64	64
FP32 核心数/GPU	2880	3072	3584	5120
FP64 核心数/SM	64	4	32	32
FP64 核心数/GPU	960	96	1792	2560
Tensor 核心数/SM	NA	NA	NA	8
Tensor 核心数/GPU	NA	NA	NA	640
GPU 加速频率	810/875 MHz	1114 MHz	1480 MHz	1530 MHz
FP32 TFLOPS 峰值 [1]	5	6.8	10.6	15.7
FP64 TFLOPS 峰值 [1]	1.7	0.21	5.3	7.8
Tensor TFLOPS 峰值 [1]	NA	NA	NA	125
纹理单元数量	240	192	224	320
显存位宽	384 位 GDDR5	384 位 GDDR5	4096 位 HBM2	4096 位 HBM2
显存容量	最大为 12 GB	最大为 24 GB	16 GB	16 GB
L2 缓存大小	1536 KB	3072 KB	4096 KB	6144 KB
共享内存大小/SM	16 KB/32 KB/48 KB	96 KB	64 KB	最大可配置为 96 KB
寄存器文件大小/SM	256 KB	256 KB	256 KB	256KB
寄存器文件大小/GPU	3840 KB	6144 KB	14336 KB	20480 KB
TDP（热设计功耗）	235 W	250 W	300 W	300 W
晶体管数量	71 亿	80 亿	153 亿	211 亿
GPU 芯片大小	551 mm^2	601 mm^2	610 mm^2	815 mm^2
制造工艺	28 nm	28 nm	16 nm FinFET+	12 nm FFN

[1] TFLOPS 峰值速率基于 GPU 加速频率测试

卓越的性能与能效

NVIDIA GPU 每经一次开发换代，性能和能效都得到显著提升。Tesla V100 为数据中心架构师提供全新的设计灵活性，可设置为侧重最大性能或是最高能效。在 Tesla V100 中，这两种运行模式称为*最大性能模式*和*最大效率模式*。

*最大性能模式*下，Tesla V100 加速器将以 300 W 的热设计功耗 (TDP) 级别运行，为需要超高计算速度和数据吞吐量的应用程序加速。

*最大效率模式*允许数据中心管理员以最佳性能功耗比运行 Tesla V100 加速器。V100 可设置以一定的功率/性能比运行，以在最佳性能和最高能效间取舍。例如，功率/性能曲线上最高效的范围可能是 50% 至 60% 的 TDP，此时 GPU 仍可发挥 75% 至 85% 的最大性能。数据中心管理员可为同机架的所有 GPU 设置功率上限，在大幅降低功耗的同时，保持出色的性能。此功能允许数据中心设计人员在相应机架的功率预算内最大限度提升机架性能。在某些情况下，执行优化操作后甚至能在机架中启用额外节点。

功率限制可使用 NVIDIA-SMI（供数据中心管理员使用的命令行实用工具）进行设置或使用 NVML（基于 C 语言的 API 库，内含功率限制控制项，Tesla OEM 合作伙伴可将其集成至工具集）来设置。 最大效率模式并不会降低正常运行时的峰值频率或显存频率，GPU 会在指定的功率限制内尽可能保持最高的时钟频率。许多工作负载并不会完全占用 Tesla V100 的 300 W TDP，因此在某些情况下功率限制可以设得更高。但是，数据中心设计人员应根据最严苛的预期工作负载来设置 GPU 功率级别，防止超出机架功率预算。

VOLTA 流多处理器

Volta 新增流多元处理 (SM) 架构，该架构在性能、能效和可编程性方面实现重大改进。

主要特性包括：

- 新增专为深度学习矩阵算法构建的混合精度 Tensor 核心，相较 GP100 在同一功率电路下训练可提升 12 倍 TFLOPS
- 在通用计算工作负载中的能效提高 50%
- L1 数据缓存性能大幅提升
- 新增 SIMT 线程模型，可消除之前的 SIMT 和 SIMD 处理器设计中存在的限制

与 Pascal GP100 相似，GV100 的每个 SM 包含 64 个 FP32 核心和 32 个 FP64 核心。不过，GV100 SM 使用新的分区方法来提升 SM 利用率和整体性能。请注意，GP100 SM 分为两个处理块，每个拥有 32 个 FP32 核心、16 个 FP64 核心、一个指令缓冲器、一个线程束调度器、两个分配单元和一个 128 KB 的寄存器文件。GV100 SM 分为四个处理块，每个拥有 16 个 FP32 核心、8 个 FP64 核心、16 个 INT32 核心、两个用于深度学习矩阵运算的新型混合精度 Tensor 核心、一个 L0 指令缓存、一个线程束调度器、一个分配单元和一个 64 KB 的寄存器文件。请注意，每个分区现在使用新的 L0 指令缓存，相比较之前的 NVIDIA GPU 能提供更高的所用指令缓冲能效。（请参见图 5 中的 Volta SM）。

虽然 GV100 SM 的寄存器数量与 Pascal GP100 SM 相同，但是整个 GV100 GPU 拥有更多的 SM，因而总体的寄存器数量也更多。总体而言，相比之前几代的 GPU，GV100 支持同时运行更多的线程、线程束和线程块。

共享内存和 L1 资源的合并使每个 Volta SM 的共享内存容量增加至 96 Kb，而 GP100 仅有 64 KB 的共享内存。

图 5. Volta GV100 流多处理器 (SM)

Tensor 核心

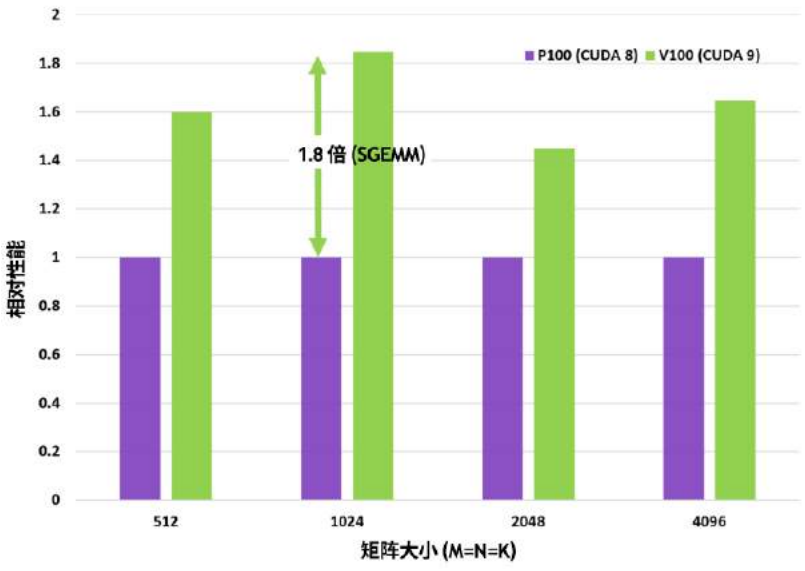
与上一代 NVIDIA Maxwell 和 Kepler 架构相比，Tesla P100 提供的神经网络训练性能明显更高，但是神经网络的复杂性和规模也在不断增加。如前所述，拥有数千层和数百万个神经元的新网络需要实现更高的性能和更快的训练时间。

新的 Tensor 核心是使 Volta GV100 GPU 架构提供大型神经网络训练所需性能的关键。

Tesla V100 GPU 包含 640 个 Tensor 核心：每个 SM 有 8 个核心，SM 内的每个处理块（分区）有 2 个核心。在 Volta GV100 中，每个 Tensor 核心每时钟执行 64 次浮点 FMA 运算，一个 SM 中的 8 个 Tensor 核心每时钟总共执行 512 次 FMA 运算（或 1024 次单个浮点运算）。

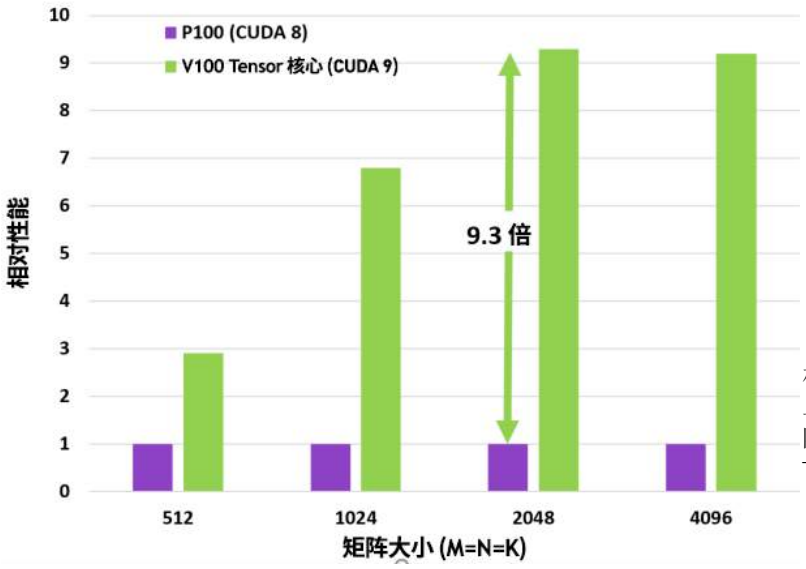
Tesla V100 的 Tensor 核心可为训练和推理应用提供高达 125 Tensor TFLOPS。与 P100 中使用标准 FP32 运算相比，Tesla V100 应用于深度学习训练时，Tensor 核心能够提供高 12 倍的峰值 TFLOPS。对于深度学习推理来说，相比于 P100 上的标准 FP16 运算，V100 Tensor 核心可提供高 6 倍的峰值 TFLOPS。

矩阵-矩阵乘法 (GEMM) 运算是神经网络训练和推理的核心，本质是在网络互联层中将大矩阵输入数据和权重相乘。对于使用单精度矩阵乘法的应用程序，在 图 6 中可以看到，与配备 CUDA 8 的 Tesla P100 相比，配备 CUDA 9 的 Tesla V100 可提供高 1.8 倍的性能。对于半精度输入的矩阵相乘（用于训练和推理运算），在图 7 中可以看到，针对 FP16 输入数据和 FP32 累加矩阵运算，Volta 混合精度 Tensor 核心可将性能提升至 P100 的 9 倍以上。



相比于配备 CUDA 8 的 Tesla P100，单精度 (FP32) 矩阵-矩阵乘法在配备 CUDA 9 的 Tesla V100 上的速度快 1.8 倍

图 6.　cuBLAS 单精度 (FP32)



相比于配备 CUDA 8 的 Tesla P100 上的 FP32 矩阵乘法，混合度矩阵-矩阵乘法在配备 CUDA 9 的 Tesla V100 上的速度快 9 倍

图 7.　cuBLAS 混合精度（FP16 输入，FP32 计算）

Tensor 核心及其关联的数据路径经自定义设计，可以较高能效大幅增加浮点计算吞吐量。

每个 Tensor 核心都在 4x4 矩阵中运行，并执行以下运算：

D = A×B + C

其中，A、B、C 和 D 为 4x4 矩阵（图 8）。矩阵乘积输入 A 和 B 为 FP16 矩阵，而累加矩阵 C 和 D 可以是 FP16 或 FP32 矩阵（请参见图 8）。

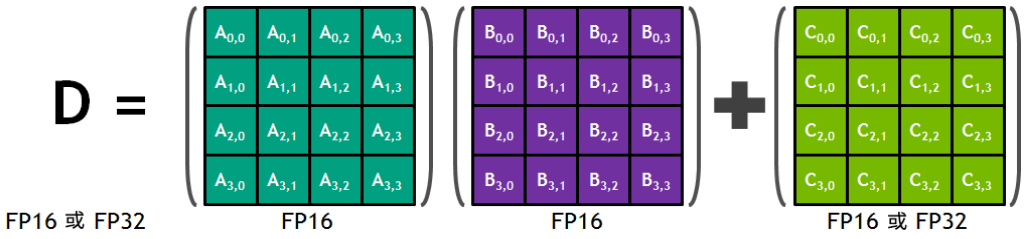


图 8.　Tensor 核心 4x4 矩阵乘积累加运算

Tensor 核心在 FP16 输入数据和 FP32 累加运算时都会发挥作用。使用 FP16 乘积得出全精度乘积，然后使用 FP32 累加将该乘积与其他中间乘积相加，从而得到 4x4x4 矩阵乘积（请参见图 9）。事实上，Tensor 核心会用于执行更大型的二维或更高维度的矩阵运算，这种运算都由这些较小元素构建而成。

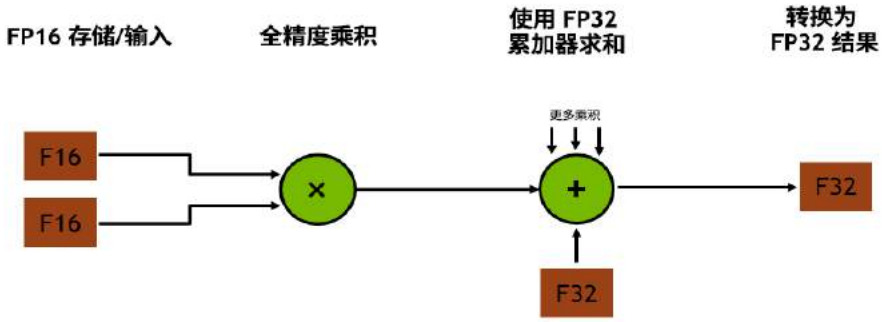


图 9.　Tensor 核心中的混合精度乘积累加运算

图 10 展示的是 4x4 矩阵乘法（使用立方体外部的双源 4x4 矩阵）需要进行 64 次运算才能生成 4x4 输出矩阵（如立方体下方所示）。配备 Tensor 核心的 Volta V100 加速器执行此类计算的速度可比 Pascal Tesla P100 快 12 倍。

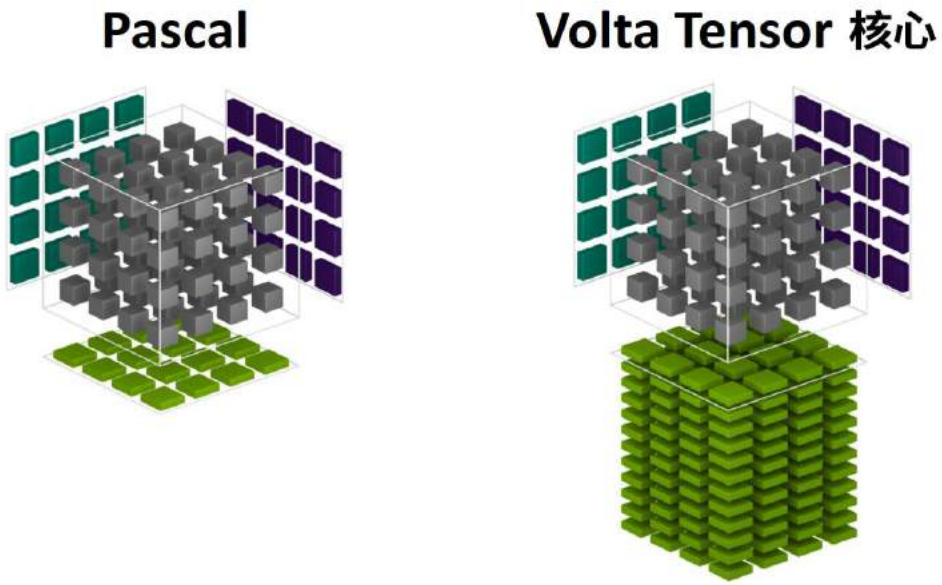


图 10.　Pascal 和 Volta 4x4 矩阵乘法运算

Volta Tensor 核心可在 CUDA 9 C++ API 中存取并作为线程束级矩阵运算公开。该 API 公开专门化矩阵负载、矩阵乘积累加和矩阵存储运算，以高效使用 CUDA-C++ 程序中的 Tensor 核心。在 CUDA 层面，线程束级接口假设 16x16 大小的矩阵跨越线程束的所有 32 个线程。

除了用于对 Tensor 核心直接编程的 CUDA-C++ 接口，cuBLAS 和 cuDNN 库也已经过更新，可提供新的库接口以将 Tensor 核心用于深度学习应用程序和框架。NVIDIA 与众多热门深度学习框架（如 Caffe2 和 MXNet）展开合作，成功实现将 Tensor 核心应用于 Volta GPU 系统的深度学习研究中。NVIDIA 也致力于在其他框架中添加对 Tensor 核心的支持。

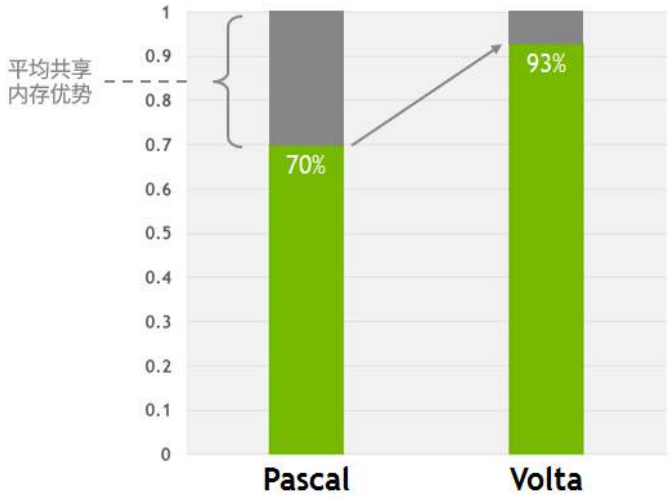
L1 数据缓存和共享内存的性能提升

Volta SM 中 L1 数据缓存和共享内存子系统的全新组合可大幅提升性能，同时还简化编程并减少实现应用程序性能峰值或接近峰值所需的调整。

将数据缓存和共享内存功能整合进单一内存块中，可为两种类型的内存访问提供出色的整体性能。整合后的容量达到 128 KB/SM，比 GP100 数据缓存大了七倍以上，不使用共享内存的程序可将其用作缓存。纹理单元也可使用该缓存。例如，如果共享内存配置为 64 KB，则纹理和加载/存储运算可使用 L1 缓存中剩余的 64 KB。

通过共享内存块内的整合，可确保相比过去 NVIDIA GPU 中的 L1 缓存，Volta GV100 L1 缓存拥有更低延迟和更高带宽。Volta 中的 L1 缓存可作为流式传输数据的高吞吐量管道，并同步提供对频繁重复使用数据的高带宽低延迟访问，实现两全其美的局面。这一组合是 Volta 独有的特性，可实现比以往更易获得的出色性能。

在 GV100 中将 L1 数据缓存与共享内存合并的主要原因，是为了让 L1 缓存运算从共享内存的出色性能中获益。共享内存可提供高带宽、低延迟和始终如一的出色性能（无缓存丢失），但是 CUDA 编程人员需要显式管理此共享内存。Volta 可缩小用于显式管理共享内存的应用程序与直接访问设备内存数据的应用程序之间的差距。为证明这一点，我们通过将共享内存阵列替换为设备内存阵列，修改了一套程序，从而实现通过 L1 缓存进行访问。如图 11 所示，当不使用共享内存运行时，这些代码在 Volta 上的性能只损失 7%，而在 Pascal 上性能会损失 30%。共享内存仍然是实现极限性能的绝佳选择，同时新的 Volta L1 缓存设计还允许编程人员通过更少的编程步骤，快速获得卓越性能。



Volta 的 L1 数据缓存可缩小将数据保存于共享内存（需手动调整）的应用程序与直接访问设备内存数据的应用程序之间的差距。

图 11. Pascal 与 Volta 数据缓存的比较

许多情况下，共享内存并非绝佳选择或者无法用到，而 GV100 L1 缓存可有效改进此时的性能。在 Volta GV100 中，共享内存与 L1 的合并可提供通往全局内存的高速通道，从而在运行时实现包含缓存丢失无限制的流式访问。之前的 NVIDIA GPU 仅可执行加载缓存，而 GV100 引入了写入缓存（存储操作缓存），从而进一步提升性能。

同时执行 FP32 和 INT32 运算

与无法同时执行 FP32 和 INT32 指令的 Pascal GPU 不同，Volta GV100 SM 包含单独的 FP32 和 INT32 核心，允许以完整吞吐量同时执行 FP32 和 INT32 运算，并且还可增加指令发送吞吐量。进行核心 FMA（混合乘加）数学运算产生的相关指令发送延迟也随之减少，Volta 上只需四个时钟周期即可实现该目标，而 Pascal 需要六个时钟周期。

许多应用程序都具有内环路，这种环路可执行结合浮点计算的指针运算（整数内存地址计算），而同时执行 FP32 和 INT32 指令将为内环路带来益处。管道化环路的每次迭代均可更新地址（INT32 指针运算）和为下一次迭代加载数据，并同时处理 FP32 中的当前迭代。

计算能力

GV100 GPU 支持全新的计算能力 7.0。表 2 比较了 NVIDIA GPU 架构不同计算能力的参数。

表 2.　计算能力：GK180、GM200、GP100 与 GV100 之间的对比

GPU	Kepler GK180	Maxwell GM200	Pascal GP100	Volta GV100
计算能力	3.5	5.2	6.0	7.0
线程/线程束	32	32	32	32
最大线程束数/SM	64	64	64	64
最大线程数/SM	2048	2048	2048	2048
最大线程块数/SM	16	32	32	32
最大 32 位寄存器数/SM	65536	65536	65536	65536
最大寄存器数/块	65536	32768	65536	65536
最大寄存器数/线程	255	255	255	255[1]
最大线程块大小	1024	1024	1024	1024
FP32 核心数/SM	192	128	64	64

SM 寄存器与 FP32 核心的比率	341	512	1024	1024
共享内存大小/SM	16 KB/32 KB/ 48 KB	96 KB	64 KB	最大可配置为 96 KB

[1] 对于组成经改进 SIMT 模型的一部分的每线程程序计数器 (PC)，每个线程通常需要两个寄存器槽。

NVLINK：更高带宽、更多链路、更多功能

NVLink 是 NVIDIA 的高速互联技术，最初于 2016 年随 Tesla P100 加速器和 Pascal GP100 GPU 一起推出。与 PCIe 互联技术相比，NVLink 可为 GPU 至 GPU 和 GPU 至 CPU 的系统配置提供更佳的性能。请参阅 *Pascal 架构白皮书*，了解 NVLink 技术的基本详情。Tesla V100 引入了第二代 NVLink，可提供更高的链路速度以及每个 GPU 更多的链路，并在 CPU 主控、缓存一致性和可扩展性方面实现改进。

更多的链路、更快的链接速度

随着开发者在 AI 计算等应用领域中公开并运用越来越多的并行结构，各行各业中的多 GPU 和 CPU 系统愈发普及。这一趋势增加了对更快速、更具可扩展性的多处理器互联技术的需求。同样地，包含数十到数千个计算节点的高性能 GPU 加速系统正广泛部署于数据中心、研究机构和超级计算机中，以解决前所未有的棘手问题。NVIDIA 配备 P100 和 V100 的 DGX-1 系统便采用 NVLink 技术。2016 年，NVIDIA 与 IBM 携手合作，使用 NVIDIA Pascal GPU 和 IBM Power 8+ CPU 构建了高性能服务器。目前，NVIDIA 正与 IBM 一道，使用 Tesla V100 加速器和以 NV Link 技术连接的 Power 9 CPU 构建性能更为优异的服务器。

与 Pascal 采用的 NVLink 技术相比，V100 上的 NVLink 可使信号发送速率从 20 GB/s 增加到 25 GB/s。现在，每条链路在每个方向均可提供 25 GB/s 的速率。支持的链路数量从四条增加至六条，进而将支持的 GPU NVLink 带宽速率提升至 300 GB/s。这些链路可专门用于“配备 V100 的 DGX-1”拓扑结构中的 GPU 至 GPU 通信（如图 12 所示），或如图 13 所示的 GPU 至 GPU 与 GPU 至 CPU 通信的组合。

更多功能

第二代 NVLink 允许从 CPU 对每个 GPU 的 HBM2 内存进行直接加载/存储/原子访问。结合全新的 CPU 主控功能，NVLink 可支持一致性运算，允许读取自图形内存的数据存储在 CPU 的缓存层次结构中。CPU 缓存低延迟访问是 CPU 性能的关键。尽管 P100 支持对等 GPU 原子，但并不支持通过 NVLink 发送 GPU 原子并在目标 CPU 完成相关操作的功能。NVLink 新增对 GPU 或 CPU 发起的原子支持，以及对地址转换服务 (ATS) 的支持，允许 GPU 直接访问 CPU 分页表。同时新增低功耗链路操作模式，当链路使用率较低时，可以大幅节省能耗（参见图 14）。

链路数量增加、链路速度提升、第二代 NVLink 的功能改进，这一切结合 Volta 全新的 Tensor 核心，使多 GPU Tesla V100 系统的深度学习性能相比使用 Tesla P100 GPU 的系统得到显著提升。

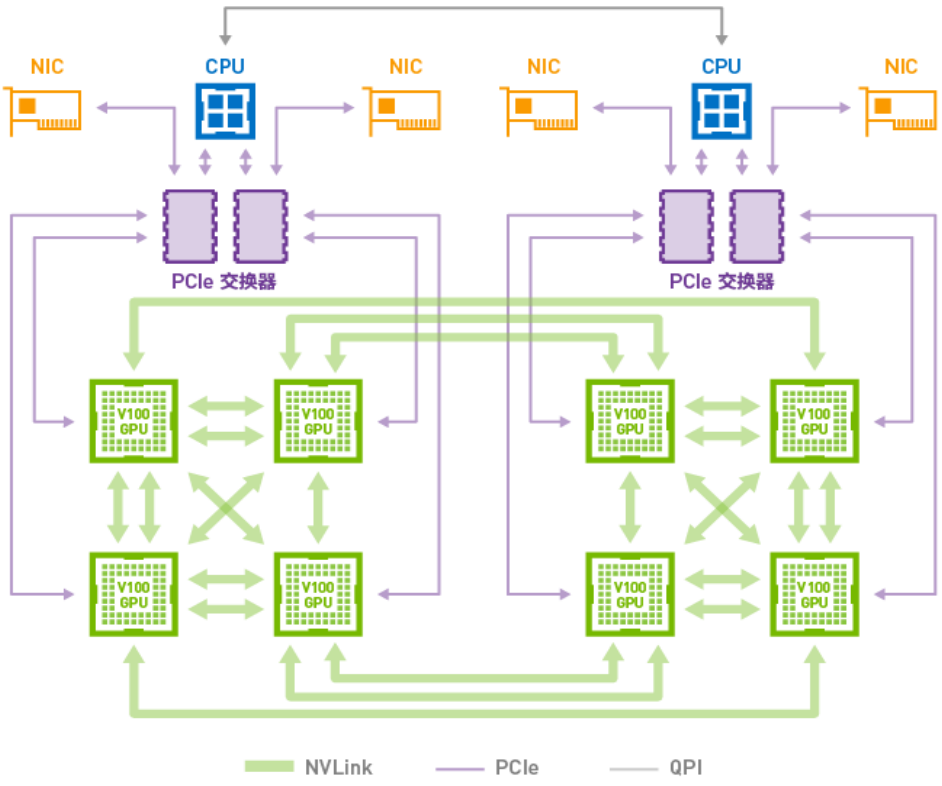


图 12. “配备 V100 的 DGX-1”中使用的混合立体网络 NVLink 拓扑

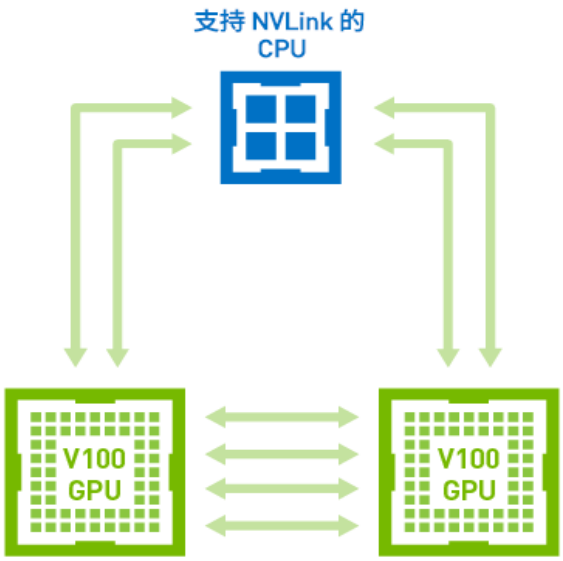


图 13. V100 与以 NVLink 连接的 GPU 至 GPU 和 GPU 至 CPU 通信

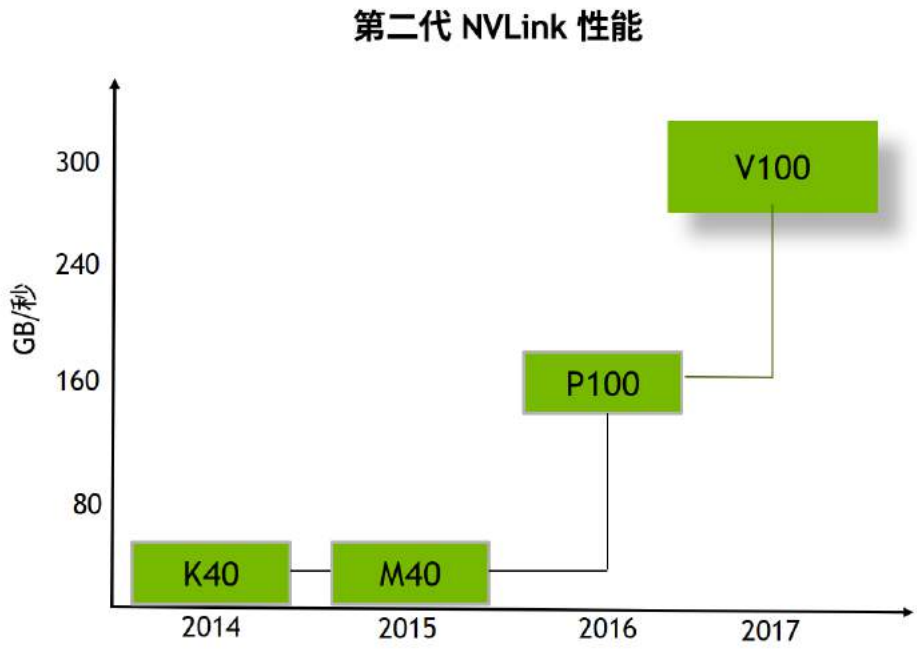


图 14. 第二代 NVLink 性能

HBM2 内存架构

Tesla P100 是全球首个支持高带宽 HBM2 内存技术的 GPU 架构。Tesla V100 则采用更快、更高效的 HBM2 架构。HBM2 内存由内存堆栈（与 GPU 位于相同的物理包）组成，与传统 GDDR5 设计相比，可显著节省能耗和占用空间，从而允许在服务器中安装更多 GPU。

Tesla V100 中的 HBM2 的每个 HBM2 堆栈使用四个存储器晶片，从而获得最大为 16 GB 的 GPU 内存。HBM2 内存可在四个堆栈中提供 900 GB/s 的峰值内存带宽。而 Tesla P100 中的最大峰值内存带宽只能达到 732 GB/s。HBM2 技术的更多详情请参见我们的 *Pascal 架构白皮书*。

除了 Tesla V100 中的峰值 DRAM 带宽比 Tesla P100 中的高之外，HBM2 在 V100 GPU 中的效率也得到大幅提升。新一代 Samsung HBM2 内存与新一代 Volta 内存控制器的结合，可提供比 Pascal GP100 高 1.5 倍的内存带宽，而运行多工作负载时的内存带宽效率超过 95%（参见图 15）。

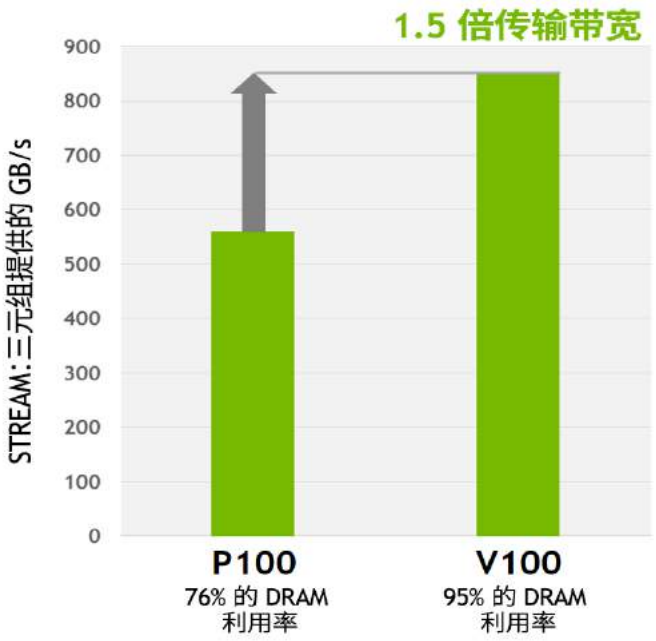


图 15.　V100 上 HBM2 内存加速与 P100 的对比

ECC 内存弹性

Tesla V100 HBM2 内存子系统支持通过纠一位检二位 (SECDED) 纠错码 (ECC) 来保护数据。ECC 可为已受数据损坏影响的计算应用程序提供更高可靠性。这在大型集群计算环境中尤为重要，因为其中的 GPU 需处理非常大的数据集亦或长时间运行应用程序。

HBM2 支持本地或*边带 ECC*，其中 ECC 位元使用的是独立于主内存的小型内存区域。与之相比的是*内联 ECC*，主内存会开拓出一部分空间用于 ECC 位元，在 Tesla K40 GPU 的 GDDR5 内存子系统中，GDDR5 总容量有 6.25% 是为 ECC 位元而保留。借助 V100 和 P100，ECC 可在不损失带宽和容量的情况下激活。对于内存写入，写入请求中可通过 32 字节的数据计算得出 ECC 位元。系统会为每八字节的数据创建八个 ECC 位元。对于内存读取，32 个 ECC 位元采用并行读取，一次读取 32 字节的数据。ECC 位元用于纠正单位错误或标记双位错误。

GV100 中的其他关键结构也受 SECDED ECC 的保护，包括 SM 寄存器文件、L1 缓存和 L2 缓存。Pascal GP100 的相同结构受到同样的 SECDED ECC 保护，以确保实现高级别的检错与纠错，以及整体内存弹性。

复制引擎增强

NVIDIA GPU 复制引擎可在多个 GPU 间或 GPU 与 CPU 间传输数据。在之前的 GPU 中，如果源内存地址或目标内存地址未映射至 GPU 分页表中，执行复制引擎传输（类似 DMA 传输）可能导致致命错误。之前的复制引擎需要固定（不可分页）源内存区域和目标内存区域。

而新的 Volta GV100 GPU 复制引擎可为没有映射至分页表的地址生成分页错误。然后，内存子系统可处理分页错误，并将地址映射至分页表，之后复制引擎便可执行传输。这是十分重要的提升，在大型的多 GPU 或多 CPU 系统中尤其如此，因为将内存固定以用于多个处理器之间的多复制引擎操作会大幅减少可用内存。通过硬件分页错误，地址可传递至复制引擎，而无需让用户担心它们是否会驻留，复制过程也会顺利进行。目前，此功能可用于 ATS 系统中。

TESLA V100 主板设计

Tesla V100 采用与 Tesla P100 相同的 SXM2 主板外形。主要区别在于 GPU 由 GV100 代替了 GP100。SXM2 主板支持 NVLink 和 PCIe 3.0 连接功能。工作站、服务器和大型计算系统中均可应用一个或多个 V100 加速器。V100 加速器大小为 140 x 78 毫米，包含可为 GPU 供应各种所需电压的高效电压调节器。V100 额定为 300 瓦热设计功耗 (TDP)。

图 16 为 Tesla V100 加速器的正面，图 17 为 Tesla V100 加速器的背面。图 18 为 NVIDIA Tesla V100 SXM2 模块的非写实剖析图。

图 16. Tesla V100 加速器（正面）

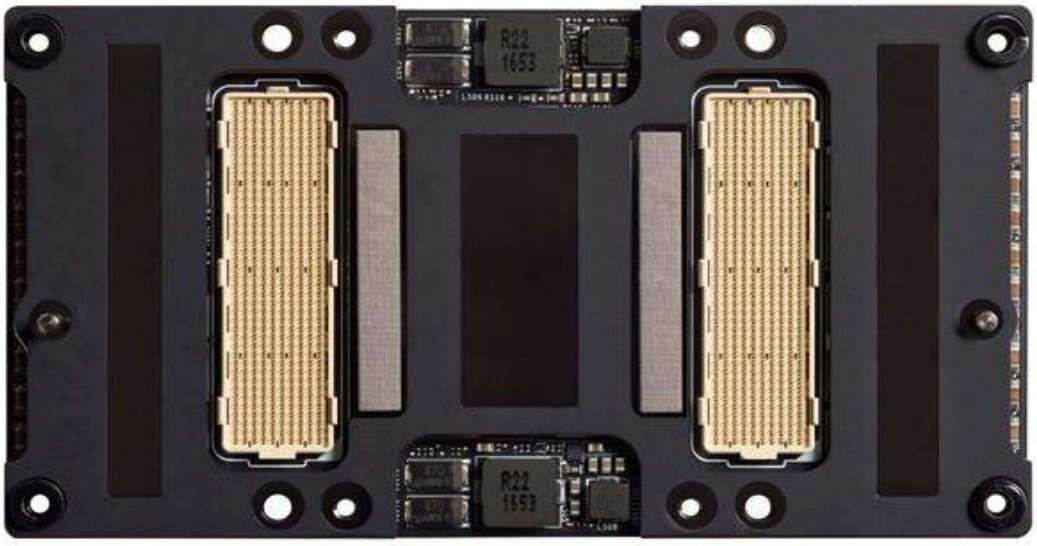


图 17. Tesla V100 加速器（背面）

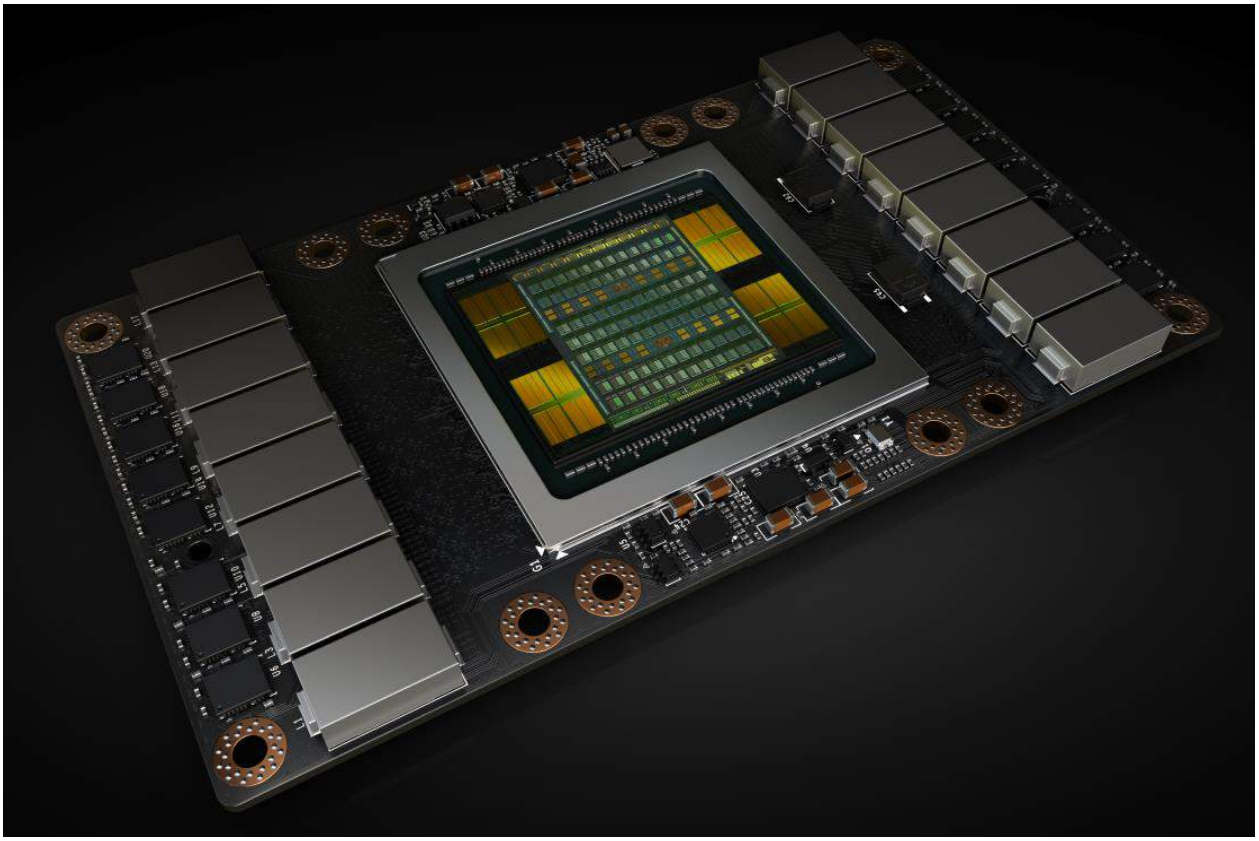


图 18.　NVIDIA Tesla V100 SXM2 模块 - 非写实剖析图

GV100 CUDA 硬件和软件架构改进

NVIDIA® CUDA® 是 NVIDIA 建立的并行计算平台和编程模型，允许应用程序开发者利用 NVIDIA GPU 强大的并行处理能力。CUDA 是深度学习以及很多其他计算及存储密集型应用的 GPU 加速基础，包括天文学、分子动力学模拟和计算金融学等。数以千计的 GPU 加速应用程序都构建于 CUDA 并行计算平台。

NVIDIA CUDA 工具包可为开发者提供全面的环境，以便其使用 C 和 C++ 编程语言扩展构建大规模并行应用程序。CUDA 的出色灵活性和可编程性使其成为研究深度学习和并行计算新算法的绝佳平台。图 19 展示了基于 CUDA 平台的深度学习创新时间线。

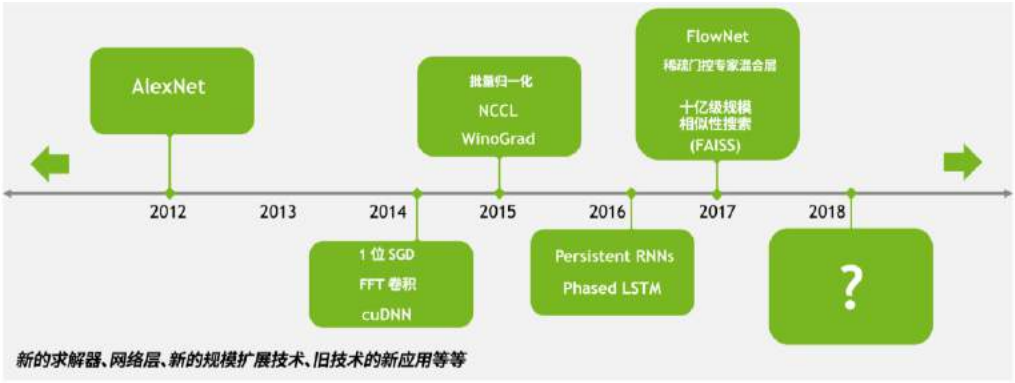


图 19. 使用 CUDA 开发的深度学习方法

本节介绍的 Volta 架构改进将进一步增强 CUDA 应用程序中并行线程的功能，使 CUDA 平台的能力、灵活性、生产力和可移植性得到显著提升。

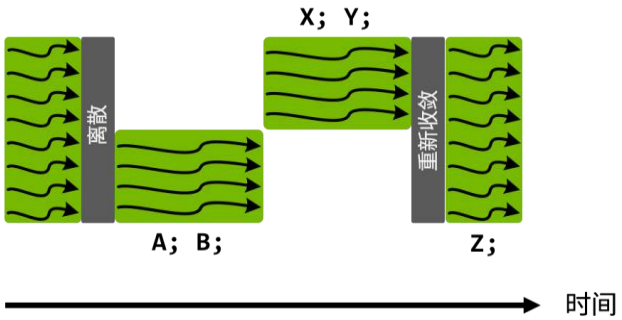
独立线程调度

Volta 架构旨在实现比以往 GPU 更高的可编程性，让用户能够在更复杂多样的应用程序中高效工作。Volta GV100 是首款支持独立线程调度的 GPU，可在程序中的并行线程之间实现更精细的同步与协作。Volta 的一大设计目标便是减少在 GPU 上运行程序所需的工作，同时提高线程协作的灵活性，最终实现更高效、更精细的并行算法。

NVIDIA 早期 GPU SIMT 模型

Pascal 和早期 NVIDIA GPU 均以 SIMT（单指令多线程）形式执行含 32 个线程的线程组（称为“线程束”）。Pascal 线程束使用所有 32 个线程中共享的单一程序计数器，并结合*激活掩码*，该掩码可指定某些线程束在特定时间内处于激活状态。这意味着离散的执行路径会让某些线程处于非激活状态，并序列化执行线程束的不同部分，如图 20 所示。原始掩码会存储起来直到线程束重新收敛（通常在离散部分末端），此时掩码恢复，线程再次一起运行。

```
if (threadIdx.x < 4) {
    A;
    B;
} else {
    X;
    Y;
}
Z;
```



Pascal 和早期 NVIDIA GPU 的 SIMT 线程束执行模型中的线程调度。大写字母代表以程序伪代码编写的语句。线程束中离散的分支经过序列化，因此分支一侧的所有语句一起执行直至完成，然后另一侧的语句才会执行。执行 else 语句后，线程束的线程通常将重新收敛。

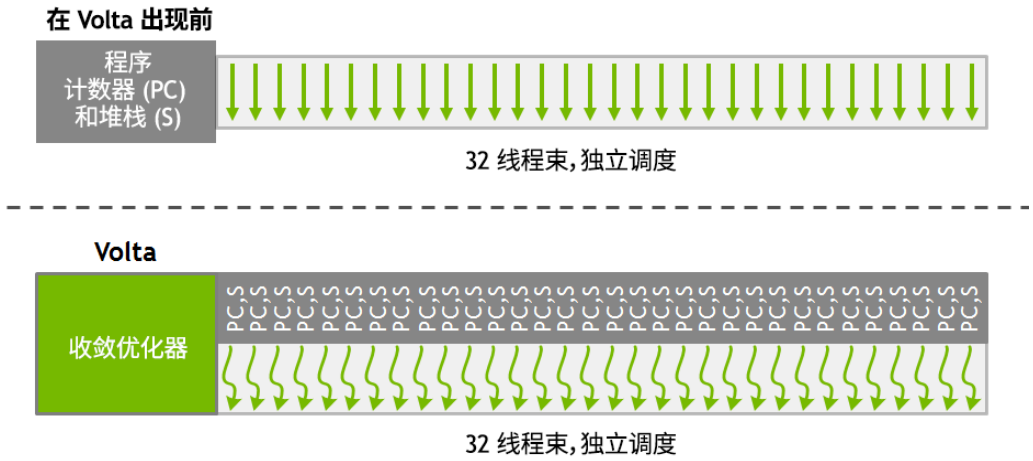
图 20.　Pascal 和早期 GPU 的 SIMT 线程束执行模型

Pascal SIMT 可通过减少跟踪线程状态所需的资源数量，及积极地重收敛线程以最大化并行性，从而最大程度地提高效率。然而，追踪整个线程束的线程状态，意味着当执行通道离散时，采用不同分支的线程会失去并发性，直至它们重新收敛。这种失去并发性的情况意味着，来自离散区相同线程束或不同执行状态的线程无法互相发送信号或交换数据。这将产生不一致性，其中，来自不同线程束的线程继续并发运行，但是相同线程束的离散线程会按顺序运行直到它们重新收敛。这表示，（例如）需要由锁或互斥体保护的数据进行精

细共享的算法很容易导致死锁，具体视争用线程来自哪个线程束。因此，在 Pascal 和早期 GPU 中，编程人员需要避免精细同步，或需要依赖无锁或线程束感知型算法。

Volta SIMT 模型

Volta 通过在所有线程之间实现等效并发（无论线程束为何），成功转变了这一格局。这一点全赖保持每个线程的执行状态（包括程序计数器和调用栈）而实现，如图 21 所示。



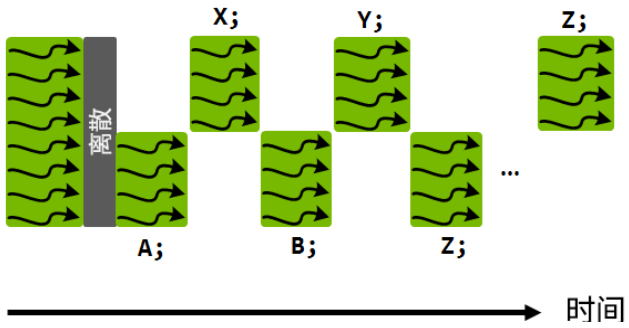
相较于 Pascal 和早期架构（顶部），Volta（底部）的独立线程调度架构块状图。Volta 会维持每个线程的调度资源，如程序计数器 (PC) 和调用栈 (S)，而早期架构是按每个线程束来维持这些资源。

图 21. Volta 线程束与每线程的程序计数器和调用栈

Volta 的独立线程调度功能允许 GPU 执行任何线程，从而更好地利用执行资源，或让一个线程等待另一个线程产生的数据。为最大程度地提高并行效率，Volta 采用了调度优化器，该优化器可确定如何将同一线程束中的活动线程一并分组到 SIMT 单元中。这样便保留如早期 NVIDIA GPU 一般的高 SIMT 执行吞吐量，但又同时大幅提高了灵活性：现在，线程能够以子线程束粒度进行离散和重新收敛，而 Volta 中的收敛优化器仍会将线程聚集在一起，这些线程将执行（且并行运行）相同的代码，以实现最大效率

图 20 中的代码执行示例看起来似乎与 Volta 中的有所不同。程序中 *if* 和 *else* 分支的语句现在可以及时交错执行，如图 22 所示。请注意，系统仍然采用 SIMT 执行方式：在任何特定时钟周期，CUDA 核心可像之前一样，为线程束中的所有活动线程执行相同指令，并保持如之前架构一般的执行效率。重要的是，Volta 能够独立调度线程束中的线程，因此其可通过更自然的方式实现复杂的精细算法和数据结构。调度器不仅支持独立执行线程，而且还可优化非同步代码，以最大限度维持较大的收敛，从而实现最大的 SIMT 执行效率。

```
if (threadIdx.x < 4) {
    A;
    B;
} else {
    X;
    Y;
}
Z;
```



通过 Volta 的独立线程调度功能，可交错执行离散分支中的语句。这可帮助执行精细的并行算法，使线程束中的线程实现同步和通信。

图 22. Volta 独立线程调度

有趣的是，我们可以看到，图 22 中并未显示线程束中的所有线程在同时执行语句 Z。这是因为调度器必须保守地假设，Z 可能生成其他离散执行分支需要的数据，在这种情况下，自动执行重新收敛并不安全。通常情况下，如果 A、B、X 和 Y 不包含同步运算，则调度器可确定线程束可以安全地在 Z 上自然重新收敛，就像之前的架构那样。

程序可调用新的 CUDA 9 线程束同步函数 __syncwarp()，以强制进行重新收敛，如图 23 所示。在这种情况下，线程束的离散部分可能不会一起执行 Z，但是同一线程束中线程的所有执行通道将在任何线程到达该语句（在执行 __syncwarp() 后）之前完成。同样地，在执行 Z 之前调用 __syncwarp()，将在执行 Z 之前强制进行重新收敛，从而有几率提高 SIMT 执行效率，前提是开发者知道这对应用程序是安全的。

```
if (threadIdx.x < 4) {
    A;
    B;
} else {
    X;
    Y;
}
Z;
__syncwarp()
```

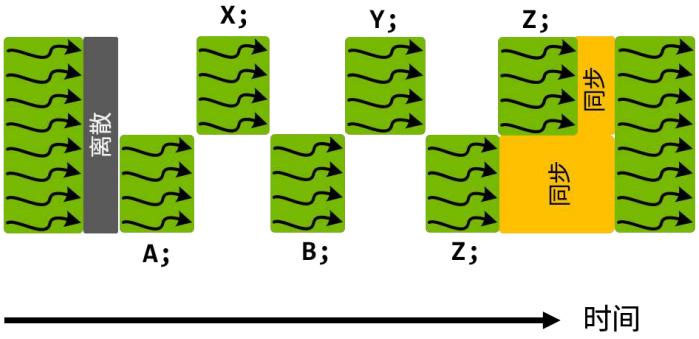


图 23. 程序采用显式同步重新收敛线程束中的线程

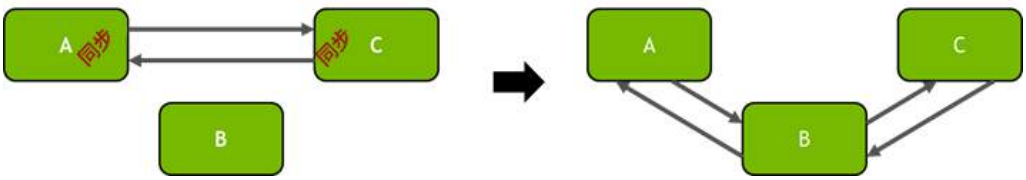
无饥饿现象算法

无饥饿现象算法是独立线程调度所实现的主要模式。只要系统确保所有线程对争用资源拥有相应的访问权限，这些并发算法便保证能正确执行。例如，如果尝试获取互斥体的线程保证最后能够成功执行，则互斥体（或锁）可用于无饥饿现象算法中。在不支持无饥饿现象的系统中，一个或多个线程可能重复获取或释放互斥体，同时使其他曾成功获得互斥体的线程处于饥饿状态。

请看一个简化的示例，其中 Volta 的独立线程调度功能可实现：将节点插入多线程应用程序的双重链接列表。

```
__device__ void insert_after(Node *a, Node *b)
{
    Node *c;
    lock(a); lock(a->next);
    c = a->next;

    a->next = b;
    b->prev = a;

    b->next = c;
    c->prev = b;

    unlock(c); unlock(a);
}
```

在本示例中，双重链接列表的每个元素都包含至少三个部分：*next 指针*、*previous 指针*和仅允许所有者更新节点的 *lock*。图图 24 显示的是在节点 A 之后插入节点 B，以及节点 A 和节点 C 的 *next* 和 *previous* 指针更新。



将节点 B 插入列表以前（右侧），需获取每节点的锁（左侧）。

图 24.　包含细粒度锁的双重链接列表

Volta 中的独立线程调度功能可确保即使线程 T0 当前持有节点 A 的锁，同一线程束中的另一个线程 T1 也可保持等待直到该锁变为可用，而不会阻碍线程 T0 的进展。但是需注意的是，由于线程束中的活动线程在同时执行，旋转于锁的线程可能降低持有该锁的线程的性能。

此外还必须注意的是，以上示例中每节点锁的使用对于 GPU 性能至关重要。传统的双重链接列表实现可能使用粗粒度锁，这种锁提供的是对整个结构的独占访问权限，而非单独保护个别节点。该方法通常会导致包含许多线程（Volta 可能包含多达 163,840 个并发线程）的应用程序的性能不理想 — 原因是对相应的锁存在高度争用的情况。通过在每个节点上使用细粒度锁，大型列表中的平均每节点争用通常较低，在某些病理性节点插入模式下除外。

这个包含细粒度锁的双重链接列表是个简单示例，展示了独立线程调度如何使开发者能够以自然方式在 GPU 中实现类似算法和数据结构。

VOLTA 多进程服务

Volta 多进程服务 (MPS) 是 Volta GV100 架构的一项新功能，可为共享 GPU 的多个计算应用程序提高性能并实现有效隔离。多个应用程序共享 GPU 的典型执行方式为时间片划分，即，每个应用程序获得一段时间的独占访问权限，然后再将访问权限授予另一个应用程序。当多个应用程序各自未充分利用 GPU 执行资源时，Volta MPS 允许这些应用程序同时共享 GPU 执行资源，从而提升 GPU 总体利用率。

从 Kepler GK110 GPU 开始，NVIDIA 引入了基于软件的多进程服务 (MPS) 和 MPS 服务器，让多个不同的 CPU 进程（应用程序上下文）得以结合为单一应用程序上下文，并在 GPU 上运行，实现更高的 GPU 资源利用率。

Volta MPS 可为 MPS 服务器的关键组件实现硬件加速，从而提升性能并改进隔离，同时增加 MPS 客户端的最大数量，将其从 Pascal 上的 16 个增加为 Volta 上的 48 个（参见图 25）。Volta 多进程服务专门用于在单一用户的应用程序中共享 GPU，而非针对多用户或多租户使用案例而设计。

对于 Pascal 而言，CUDA 多进程服务是一个 CPU 进程，可代表已请求同时与其他 GPU 应用程序共享执行资源的 GPU 应用程序执行操作。此进程可作为媒介，向 GPU 中的工作队列提交工作，实现并发内核程序执行。

Volta 多进程服务可为 CUDA MPS 实现硬件加速，使 MPS 客户端能够将工作直接提交至 GPU 中的工作队列。这一加速可显著降低提交延迟并增加总吞吐量。对于 Volta，CPU MPS 控制进程仍会进行配置并选择采用 MPS。

Volta MPS 基于两个关键指标来改进 MPS 客户端之间的隔离：服务质量 (QoS) 和独立地址空间。在 Volta 中，除了改进 QoS，不同 MPS 客户端 A、B 和 C 的工作还可使地址实现隔离，如图 25 所示。Volta MPS，与之前 NVIDIA GPU 上的 CUDA MPS 一样，不提供客户端之间的致命错误隔离功能。

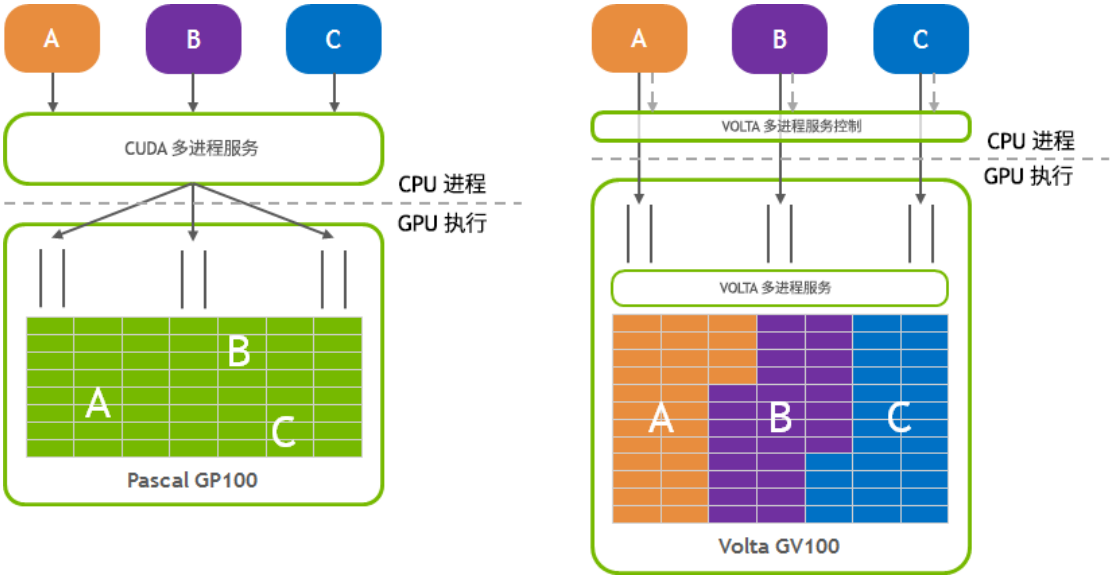


图 25. Pascal 中基于软件的 MPS 服务与 Volta 中硬件加速 MPS 服务的对比

服务质量系指提交工作后，GPU 执行资源能够在多快的速度内用于处理客户端工作。Volta MPS 为 MPS 客户端提供一种控制功能，允许其指定执行所需的 GPU 部分。将每个客户端仅限制在部分 GPU 执行资源范围内的这一控制功能，可有效减少或消除线头阻塞问题，在这种问题中，一个 MPS 客户端的工作可能耗尽 GPU 执行资源，阻止其他客户端取得进展，直到另一个 MPS 客户端完成之前的工作为止。QoS 的这一改进可减少系统中的平均延迟和抖动，这对 MPI/HPC 使用案例和深度学习推理使用案例都至关重要。

Volta 可为深度学习推理提供非常高的吞吐量和低延迟，尤其是使用批处理系统汇总图像，并同时提交至 GPU，以最大程度地提高性能的情况下，更是关键非常。若没有此类批处理系统，单独的推理工作便无法充分利用 GPU 的执行资源。通过允许多个单独的推理工作同时提交至 GPU 并提升 GPU 总体利用率，Volta MPS 可在满足延迟目标的同时轻松提高吞吐量。

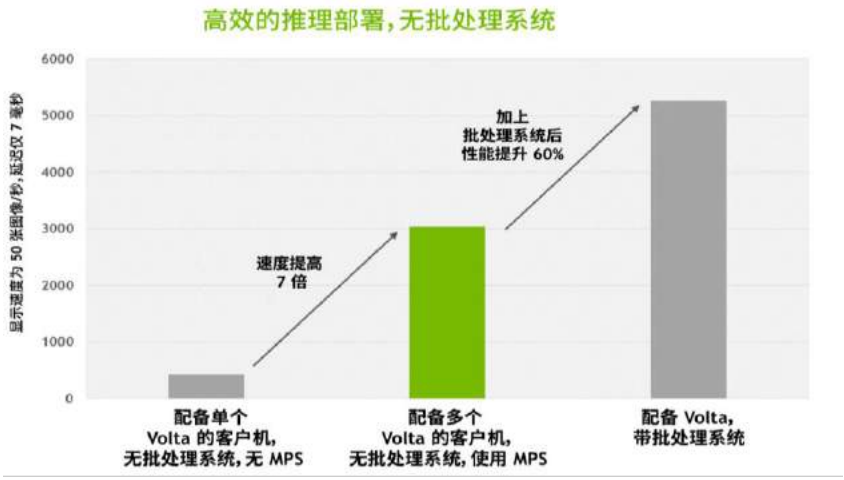


图 26. 用于推理的 Volta MPS

Volta MPS 的主要功能包括对具有 Linux 支持的统一内存寻址功能路线图的支持（例如，从 GPU 进行 malloc 内存访问）。之前的 NVIDIA GPU 架构中的 CUDA MPS 客户端在 GPU 中执行时均在单一地址空间下运行，这与访问独立 CPU 进程内存无法兼容。

统一内存寻址和地址转换服务

我们在 Kepler 和 Maxwell GPU 中随 CUDA 6 推出了有限形式的统一内存寻址，这项功能在 Pascal GP100 GPU 中通过硬件页面错误和更大的地址空间得到改进。统一内存寻址允许为 CPU 和 GPU 内存提供单一的统一虚拟地址空间，大大简化 GPU 编程，以及应用程序移植到 GPU 的过程。编程人员无需再为管理 GPU 与 CPU 虚拟内存系统之间的数据共享而烦恼。Pascal GP100 中的统一内存寻址可在 GPU 和 CPU 的整个虚拟地址空间之间实现透明的数据迁移。（如需 Pascal 统一内存寻址技术的详细说明，请参见我们的 *Pascal 架构白皮书*。）

尽管 Pascal GP100 中的统一内存寻址已通过许多方式改进 CUDA 编程，但 Volta GV100 还可进一步提升统一内存寻址的效率和性能。全新的存取计数器功能可跟踪 GPU 存取其他处理器内存的频率。存取计数器帮助确保内存页面移动至访问页面最频繁的处理器的物理内存。存取计数器功能可用于以 NVLink 或 PCIe 连接的 GPU-CPU 或 GPU-GPU 架构，并且可兼容不同类型的 CPU，包括 Power 9、x86 等。

此外，Volta 还支持地址转换服务 (ATS)（通过 NVLink）。ATS 允许 GPU 直接访问 CPU 的分页表。如果缺少 GPU MMU，系统将向 CPU 发出地址转换请求 (ATR)。CPU 访问其分页表以便为该地址进行虚拟至物理映射，然后将转换地址重新提供给 GPU。ATS 为 GPU 提供对 CPU 内存（例如通过“malloc”直接分配的内存）的完整访问权限。

协作组

在并行算法中，线程通常需要通过协作来执行集群计算。构建这些协作代码需要对协作线程进行分组和同步。因此，CUDA 9 引入了协作组 — 用于组织线程组的全新编程模式。

过去，CUDA 编程模型一直提供一个简单的结构用于同步协作线程：即在线程块的所有线程中设置一个障碍，就像通过 __syncthreads( ) 函数实现的那样。但是，编程人员通常希望通过“集合”整组函数接口的形式，以比线程块更小的粒度定义线程组，并在其中实现同步，以提高性能、设计灵活性和软件重用性。

使用协作组提供的功能，能够以子线程块和多线程块粒度显式定义线程组，并且可以执行集合运算，例如线程同步。此编程模型支持跨软件边界实现干净合成，这样库和实用函数便可在其本地上下文中安全地同步，而不必就收敛问题作出假设。这让开发者以安全、可支持的方式通过灵活同步功能针对硬件快速进行各种优化（例如优化 GPU 线程束大小），从而完美体现编程人员的意图。协作组原语在 CUDA 中实现全新模式的协作并行，包括生产者-消费者并行、机会型并行，以及整个网络的全局同步。

此外，协作组还实现了抽象化，让开发者能够编写灵活、可扩展的代码，这种代码可在不同的 GPU 架构中安全运行，包括扩展至未来 GPU 的功能。线程组的大小或许各不相同，可能包含几个线程（比一个线程束小），可能是整个线程块或者网格发射中的所有线程块，也可能包含跨多个 GPU 的网格。

尽管协作组适用于所有 GPU 架构，但是随着 GPU 功能的改进，某些功能不可避免地需要依赖架构。所有架构均支持基本功能，例如对小于线程块乃至线程束粒度的组进行同步，而 Pascal 和 Volta GPU 可启用新的涵盖整个网格的多 GPU 同步组。此外，Volta 的独立线程调度能够以任意交叉线程束和子线程束粒度，为线程组实现更灵活的选择和划分。Volta 同步真正实现了每线程操作：线程束中的线程可从发散的代码路径进行同步。

协作组编程模型由以下元素组成：

- 专为深度学习矩阵算法构建的全新混合精度 FP16/FP32 Tensor 核心；
- 表示协作线程组的数据类型；
- CUDA 启动 API 定义的默认组（例如，线程块和网格）；
- 将现有组划分为新组的运算；
- 同步组中所有线程的障碍运算；
- 检查群组属性以及特定于组的集合运算。

下面的简单示例说明了某些基本协作组运算。

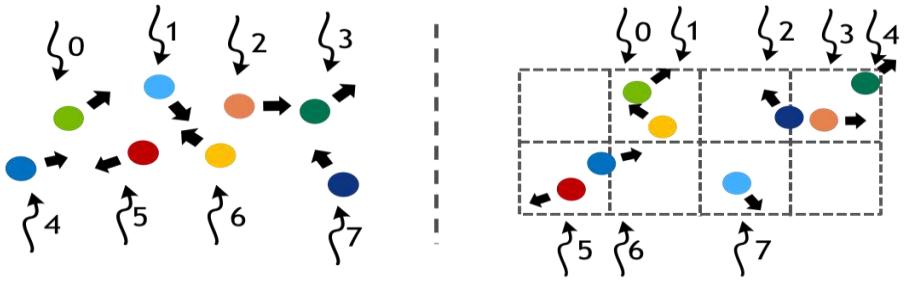
```
__global__ void cooperative_kernel(...)
{

// obtain default "current thread block" group
thread_group my_block = this_thread_block();

// subdivide into 32-thread, tiled subgroups
// Tiled subgroups evenly partition a parent group into
// adjacent sets of threads - in this case each one warp in size
thread_group my_tile = tiled_partition(my_block, 32);

// This operation will be performed by only the
// first 32-thread tile of each block
if (my_block.thread_rank() < 32) {
…
my_tile.sync();
}
}
```

协作组使用 C++ 模板提供类型和 API 过载，以表示静态确定大小的组，从而进一步提供效率。语言级接口得到了一套 PTX 组件扩展的支持，这些扩展可为 CUDA C++ 实现提供基底。此外，这些 PTX 扩展可用于希望提供类似功能的任何编程系统。最后，cuda-memcheck 中的竞争检测工具和 CUDA 调试器兼容协作组许可的更灵活的同步模式，可更轻松地发现细微的并行同步错误，如写后读 (RAW) 危险。

协作组让编程人员能够表达之前无法做到的同步模式。当同步粒度与自然架构粒度（线程束和线程块）相对应时，这种灵活性的开销可以忽略不计。使用协作组编写的集合通信原语库通常只需不怎么复杂的代码即可实现高性能运行。

以粒子模拟为例，模拟的每一步骤中有两个主要计算阶段。第一个阶段，及时向前集成每个粒子的位置和速度。第二个阶段，构建一个常规网格空间数据结构，加快发现粒子之间的冲突。

图 27 展示了这两个阶段



粒子模拟的两个阶段，带编号的箭头表示并行线程映射至粒子。请注意，集成并构建常规网格数据结构后，内存粒子和线程映射顺序会改变，并且需要在两个阶段之间同步。

图 27. 粒子模拟的两个阶段

在协作组出现之前，实现这样的模拟需要多个核心启动，因为阶段 1 到阶段 2 时线程映射会发生变化。构建常规网格加速结构的过程会对内存中的粒子进行重新排序，因而必需重新将线程映射至粒子。这样的重新映射需要在线程之间进行同步。如以下 CUDA 伪代码所示，连续核心启动之间的隐式同步可满足此要求。

```
// threads update particles in parallel
integrate<<<blocks, threads, 0, s>>>(particles);
// Note: implicit sync between kernel launches
// Collide each particle with others in neighborhood
collide<<<blocks, threads, 0, s>>>(particles);
```

协作组可提供灵活的、可扩展的线程组类型，而同步原语可在单一核心启动中实现如上述示例情况下的并行重新映射。以下 CUDA 核心简单说明了粒子系统更新如何在单一核心中完成。使用 this_grid() 可定义线程组，该组包含核心启动的所有线程，然后在两个阶段之间进行同步。

```
__global__ void particleSim(Particle *p, int N) {

  grid_group g = this_grid();
  // phase 1
  for (i = g.thread_rank(); i < N; i += g.size())
    integrate(p[i]);
  g.sync() // Sync whole grid
  // phase 2
  for (i = g.thread_rank(); i < N; i += g.size())
    collide(p[i], p, N);
}
```

此核心经过编写，可轻松将模拟扩展到多个 GPU。协作组函数 this_multi_grid() 返回一个线程组，该组涵盖跨多个 GPU 核心启动的所有线程。在此组中调用 sync()，可同步在多个 GPU 上运行核心的所有线程。请注意，在这两种情况下，thread_rank() 方法提供了线程组中当前线程的线性索引，核心会使用该索引对粒子进行并行迭代，以防粒子数量多于线程。

```
__global__ void particleSim(Particle *p, int N) {

  multi_grid_group g = this_multi_grid();
  // phase 1
  for (i = g.thread_rank(); i < N; i += g.size())
    integrate(p[i]);
  g.sync() // Sync whole grid
  // phase 2
  for (i = g.thread_rank(); i < N; i += g.size())
    collide(p[i], p, N);
}
```

为使用跨多个线程块或多个 GPU 的组，应用程序必须分别使用 cudaLaunchCooperativeKernel() 或 cudaLaunchCooperativeKernelMultiDevice() API。同步要求所有线程块同时驻留，因此应用程序还必须确保所启动线程块的资源（寄存器和共享内存）使用量不会超过 GPU 的总资源。

结束语

NVIDIA Tesla V100 加速器基于全新 Volta GV100 GPU，是当前世界上数据中心 GPU 中的精尖之作。V100 可加速 AI、HPC 和图形处理，让数据科学家、研究人员和工程师能够应对曾经无法解决的挑战。

Volta 是全球首款功能强大无比的 GPU 架构，而 GV100 是第一种突破 100 TFLOPS 深度学习性能极限的处理器。GV100 将 CUDA 核心和 Tensor 核心相结合，在 GPU 中提供 AI 超级计算机的出色性能。第二代 NVIDIA NVLink 可以高达 300 GB/s 的速度连接多个 V100 GPU，打造出全球惊艳的计算服务器。现在，借助 Tesla V100 加速的系统，过去需要消耗数周计算资源的 AI 模型只需几天即可完成训练。随着训练时间的大幅缩短，在 NVIDIA Tesla V100 加速器的助力下，AI 现在可以解决各类新型问题。

附录 A
搭载 TESLA V100 的 NVIDIA DGX-1

数据科学家和人工智能研究人员需要自己的深度学习系统在精确度、简单性和速度方面具备出色性能。更快的训练和迭代速度最终将推动创新速度加快、产品上市所需时间缩短。NVIDIA DGX-1（如图 28 所示）是世界上第一款专为深度学习而打造的服务器，具备全面集成的硬件和软件，可以轻松快速地完成部署。

图 28.　NVIDIA DGX-1 服务器

2016 年 NVIDIA 推出 DGX-1，配有 8 块 NVIDIA Tesla P100 GPU，在混合立体网络中通过 NVIDIA NVLink 互相连接。基于 P100 的 DGX-1 可搭配双路英特尔至强 CPU 及四块 100 Gb InfiniBand 网络适配器使用，为深度学习训练提供卓越的性能。凭借高达 170 FP16 TFLOPS，NVIDIA DGX-1 可显著加快训练速度，是首款单机箱 AI 超级计算机。有关基于 Tesla P100 的 DGX-1 系统的更多详情可参阅本白皮书。

DGX-1 系统构建于 3U 机架式机箱中，具有若干高性能/高可靠性组件，可供单独或集群使用。

随着 NVIDIA Tesla V100 的推出，NVIDIA 为 DGX-1 平台带来全新 SKU，即包含八块通过 NVLink 互联技术连接的 NVIDIA Tesla V100 GPU。基于 Tesla V100 的 DGX-1 平台可为深度学习应用程序提供 960 Tensor TFLOPS 的出色性能（参见图 29）。

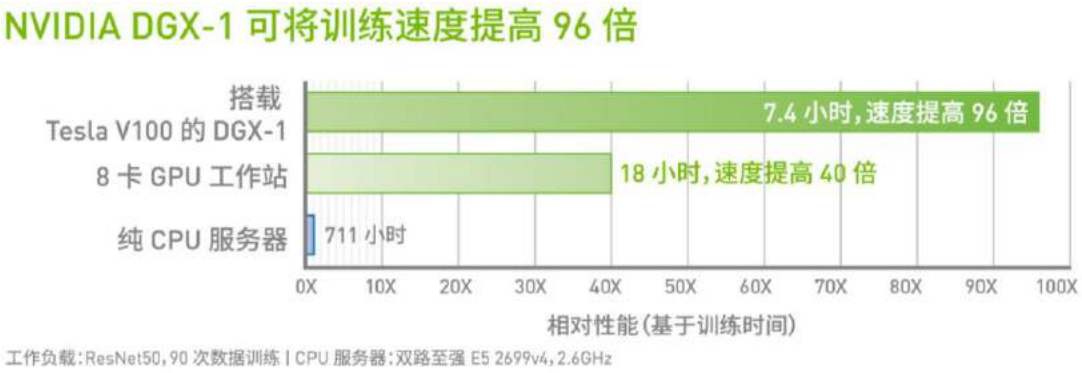


图 29. 与基于 GP100 的八路服务器相比，DGX-1 将训练速度提高了 3 倍

NVIDIA DGX-1 系统规格

NVIDIA DGX-1 是世界上第一款专为深度学习而打造的服务器，具备全面集成的硬件和软件，可以轻松快速地完成部署。NVIDIA DGX-1 革命性的性能可大幅加快训练速度，是首款单机箱 AI 超级计算机。表 3 列出了 NVIDIA DGX-1 的系统规格。

表 3. NVIDIA DGX-1 系统规格

规格	DGX-1 (Tesla P100)	DGX-1 (Tesla V100)
GPU	8 块 Tesla P100 GPU	8 块 Tesla V100 GPU
TFLOPS	170 (GPU FP16) + 3 (CPU FP32)	1 (GPU Tensor PFLOP) + 3 (CPU FP32)
GPU 内存	每个 GPU 16 GB 每个 DGX-1 节点 128 GB	每个 GPU 16 GB 每个 DGX-1 节点 128 GB
CPU	双路 20 核 Intel® Xeon® E5-2698 v4 2.2 GHz	双路 20 核 Intel® Xeon® E5-2698 v4 2.2 GHz
FP32 CUDA 核心	28,672	40,960
Tensor 核心数	--	5120
系统内存	高达 512MB 2133 MHz DDR4 LRDIMM	高达 512MB 2133 MHz DDR4 LRDIMM
存储	4 块 1.92TB SSD RAID 0	4 块 1.92TB SSD RAID 0

网络	双 10 GbE，4 IB EDR	双 10 GbE，4 IB EDR
系统重量	60 千克	60 千克
系统尺寸	866（长）x 444（宽）x 131（高）(毫米)	866（长）x 444（宽）x 131（高）(毫米)
包装尺寸	1180（长）x 730（宽）x 284（高）（毫米）	1180（长）x 730（宽）x 284（高）（毫米）
电源	3200 W（最大值）四个 1600 W 负载平衡电源（3+1 冗余），200-240 V(ac)，10 A	3200 W（最大值）四个 1600 W 负载平衡电源（3+1 冗余），200-240 V(ac)，10 A
操作温度范围	10 - 35°C	10 - 35°C

DGX-1 软件

除了强大的 DGX-1 硬件，该系统还包含具有开发工具和库的全面集成的软件堆栈，且专为大规模运行深度学习而优化。其主要目标是让从业者能够在 DGX-1 上部署深度学习框架和应用程序，同时将设置工作减至最少。

平台软件的设计旨在最大程度地减少服务器上的操作系统和驱动程序安装工作。所有应用程序和 SDK 软件的配置使用 NVIDIA Docker 容器，通过 NVIDIA 维护的 DGX 容器注册表[2]来实现。

DGX-1 的可用容器包括多个经优化的深度学习框架、NVIDIA DIGITS 深度学习训练应用程序、第三方加速解决方案及 NVIDIA CUDA 工具包。

该软件架构具有很多优势：

- 每个深度学习框架都位于单独的容器内，所以每个框架都能使用不同版本的库，比如 libc、cuDNN 等，并且不会相互影响。
- 为提高性能或修复问题，深度学习框架经过多次改进，现在 DGX 容器注册表中已有新版本容器。
- 系统易于维护，且由于应用程序并非直接安装于操作系统上，所以操作系统镜像非常干净。
- 可无缝提供安全更新、驱动程序更新及操作系统补丁。

[2] 容器注册表服务由 NVIDIA 提供. 请参阅：http://docs.nvidia.com/dgx/dgx-registry-guide/

这些深度学习框架和 CUDA Toolkit 都包含经定制调整的库，可以在 DGX-1 上提供较高的多 GPU 性能。图 30 显示的是 DGX-1 深度学习堆栈。

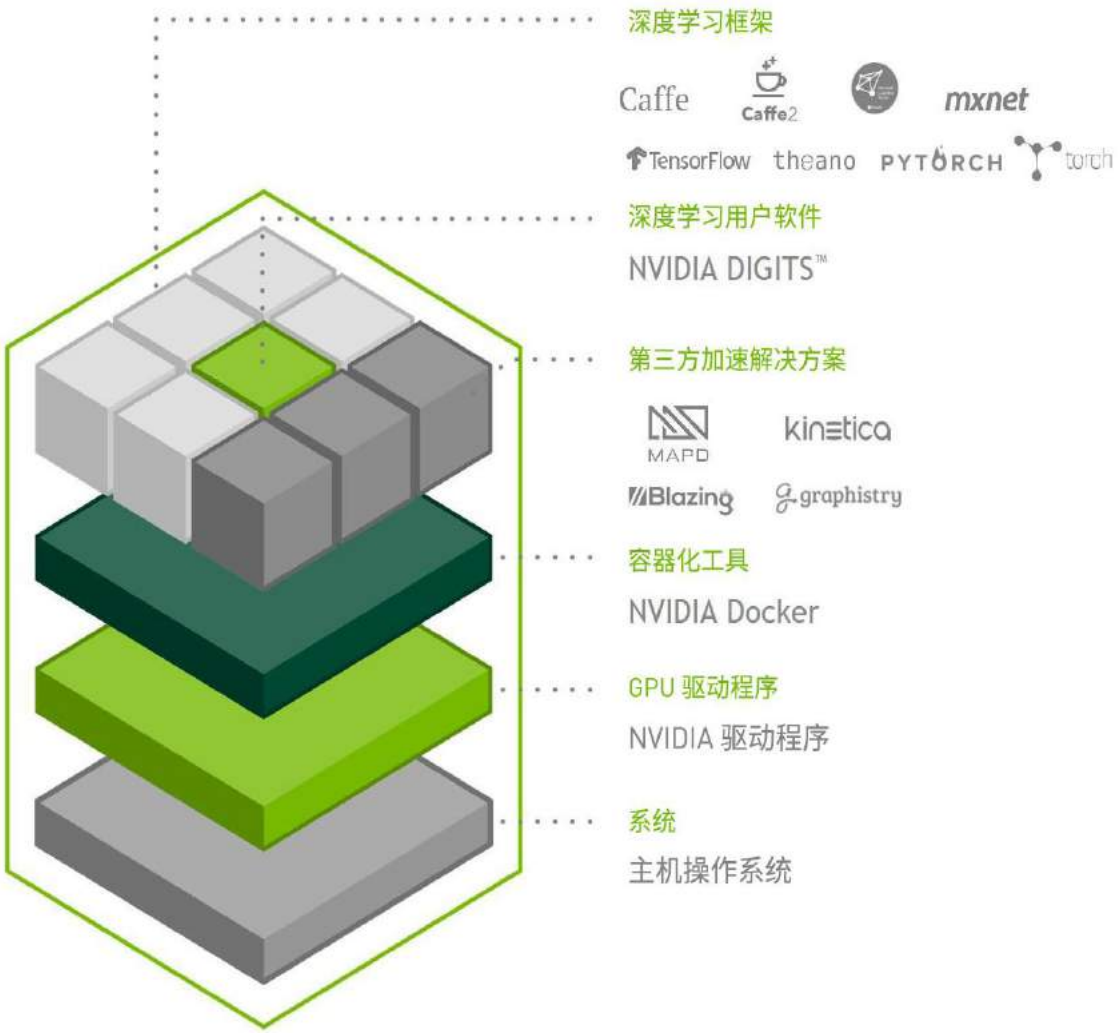


图 30. NVIDIA DGX-1 全面集成的软件堆栈，可即时提高生产力

NVIDIA DGX-1 将强大的硬件与为深度学习定制的软件相结合，为开发者和研究人员提供可实现高性能 GPU 加速深度学习应用程序开发、测试和网络训练的整体解决方案。

附录 B
NVIDIA DGX 工作站 - 适用于深度学习的个人 AI 超级计算机

NVIDIA DGX Station™ 是一款开创性的深度学习和分析专用超级计算机，可提供比拟 400 个 CPU 的强大计算能力，整个计算机置于可安装在办公桌下的便携式工作站中（请参见图 31）。DGX 工作站是一款静音水冷式工作站，包含四个采用 NVIDIA Volta 架构的 Tesla V100 GPU，可为深度学习应用程序实现 500 Tensor TFLOPS。

与当今的高速 GPU 工作站相比，DGX 工作站的深度学习训练性能提高了近 3 倍，推理性能提高了 3 倍。DGX 工作站中的四个 Tesla V100 GPU 通过 NVIDIA 第二代 NVLink 互联技术连接，相比基于 PCIe 连接的工作站，IO 带宽提高了近五倍。

图 31.　配备 Tesla V100 的 DGX Station

图 32 展示配备四路 Tesla V100 服务器的 DGX 工作站性能。Tesla V100 的性能比基于 CPU 的服务器快 47[3] 倍。表 4 列出了 DGX 工作站的规格。

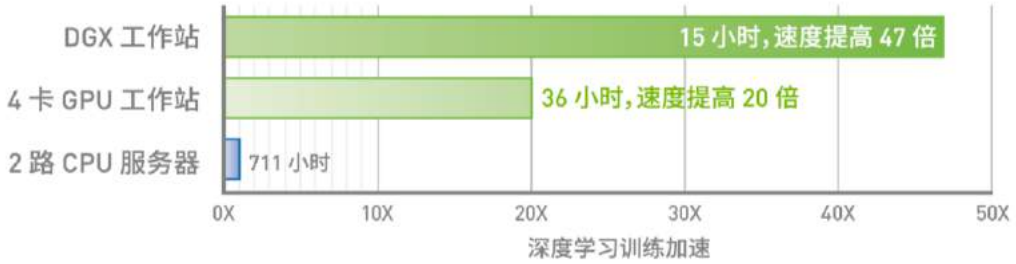


图 32. NVIDIA DGX 工作站可将训练速度提高 47 倍

表 4. DGX 工作站规格

规格	DGX 工作站
GPU	通过 NVLink 互联的 4 个 NVIDIA Tesla V100
TFLOPS	500 Tensor TFLOPS，15.7 FP32 TFLOPS
Tensor 核心	2560
CPU	英特尔至强 E5-2698 v4 2.2 GHz（20 核）
系统内存	256 GB LRDIMM DDR4
存储	数据：3 块 1.92 TB SSD RAID 0 操作系统：1 块 1.92 TB SSD
网络	Dual 10 Gb LAN
显示	3 个 DisplayPort 接口
声音	< 35 dB
系统重量	40 千克
系统尺寸	518 毫米（长）x 256 毫米（宽）x 639 毫米（高）
最大功率	1500 W
工作温度	10°C - 30°C
操作系统	Ubuntu 桌面 Linux 操作系统

[3] Workload: ResNet50, 90 epochs to solution | CPU Server: Dual Xeon E5-2699 v4, 2.6GHz

预加载最新的深度学习软件

NVIDIA DGX 工作站预加载的软件堆栈与所有 DGX 解决方案相同。这一创新的集成软件堆栈可访问热门深度学习框架，每个框架均经过 NVIDIA 深度学习专家优化，并且每月更新。该软件堆栈还包含 NVIDIA DIGITS 深度学习训练应用程序、第三方加速解决方案、NVIDIA 深度学习 SDK，例如 cuDNN、cuBLAS、CUDA 工具包、快速多 GPU 集合（称为 NCCL）以及 NVIDIA 驱动程序。

这一全面的深度学习软件堆栈不断得到调整和优化，并通过所有 DGX 平台适用的相同 NVIDIA Docker 容器和 NVIDIA 容器注册表服务提供。该单一的统一深度学习堆栈可有效简化工作流程。现在，数据科学家可轻松扩展其工作，并通过 DGX 工作站将开发的解决方案部署到数据中心的 DGX-1 服务器或 NVIDIA 深度学习云。

同样重要且值得注意的是，由 NVIDIA 维护和提供软件堆栈后，数据科学家现在可以只专注于训练和部署深度学习解决方案，而无需浪费时间调整和更新软件组件。生产力提高和对稀缺深度学习专业知识更合理的利用，可能节省数千美元的成本，从而大幅降低硬件的初始成本。

推动 AI 计划

NVIDIA DGX 工作站专为推动独立研究人员或组织的 AI 项目而设计，其精简优化的即插即启动的使用体验，让您当日即可开始深度神经网络的训练。

DGX 工作站提供卓越的计算性能，并配备包含以下功能的集成解决方案，让您高枕无忧：

- 企业级支持
- 对 NVIDIA 深度学习专业知识的访问权限
- 针对深度学习优化的工具库和软件
- 及时软件升级
- 优先解决关键问题

DGX 工作站完美结合 NVIDIA 工具和知识，让数据科学家能够出色地完成工作。

了解关于 NVIDIA DGX 工作站 的详细信息，请访问 https://www.nvidia.cn/dgx-station

附录 C
通过 GPU 为深度学习和人工智能加速

过去五年，部署于 GPU 上的深度神经网络 (DNN) 已经征服一个又一个算法领域。其潜在的使用案例不计其数：从无人驾驶汽车到更快速的药物开发，以及从在线图像数据库中的自动图像字幕到视频聊天应用程序中的智能实时语言翻译，深度学习正提供令人兴奋的人机交互机会。在本节中，我们将简单介绍深度学习，以及客户如何使用 GPU 实现全新的深度学习突破。

深度学习概述

深度学习是建模人脑神经学习过程的技术，随着不断学习，系统将变得越来越智能并能更快速地提供更准确的结果。儿童最初是在成年人的教导下学习正确地辨别和分类各种形状，最终才能够在无任何指导的情况下辨别形状。同样地，深度学习或神经学习系统需要在对象识别和分类方面接受训练，才能在识别基本对象、遮挡对象时变得更智能、更高效，同时还能将上下文分配至相应对象。

最简单地，人脑中的神经元关注馈送进来的各个输入项，并会为每个输入项分配重要性级别，然后输出项传递到其他神经元以采取相应行动。

图 33 中所示的感知器是最基本的神经网络模型，与人脑中的神经元类似。如图中所见，感知器有多个输入，这些输入项代表感知器经训练用于识别和分类对象的各种特征，每个特征根据定义对象形状中所占的重要性，分配特定的权重。

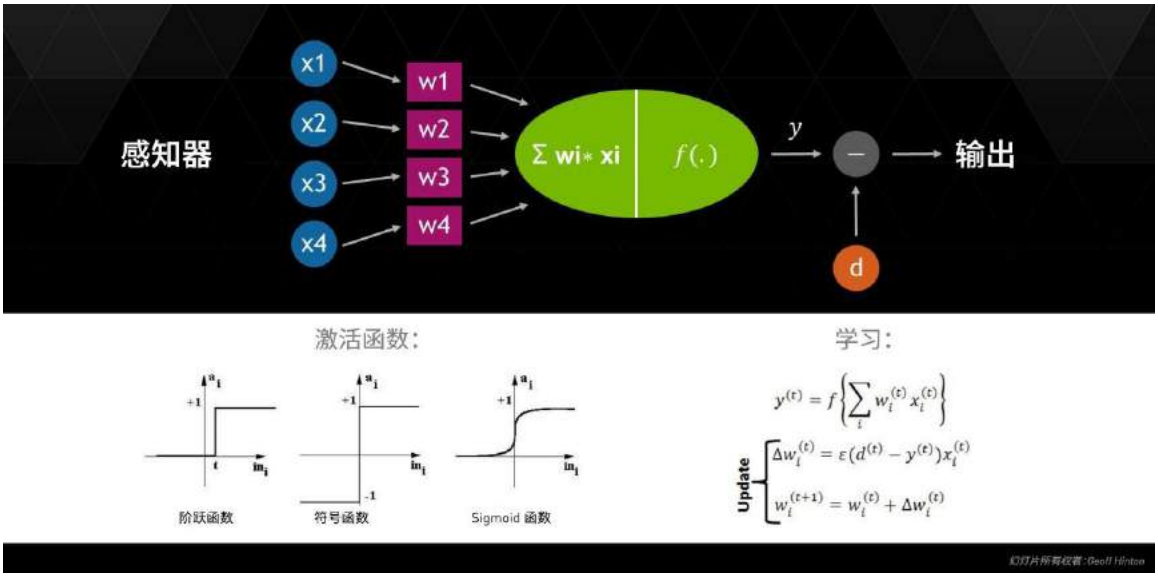


图 33. 感知器是最简单的神经网络模型

例如，假设感知器经训练用于识别手写的数字零。显然，根据不同的书法风格，数字零会有各种不同的方式。感知器将拍下数字零的图像，将其分解成各个部分并将这些部分分配到特征 x1 至 x4。数字零的右上角曲线可分配给 x1，底部曲线分配给 x2，以此类推。特定特征的权重值即代表此特征在正确决定手写数字是否为零中的重要性。在图表中央的绿色图块中，感知器会计算图像中所有特征的总权重，以确定该数字是否为零。然后会对此结果应用函数，就该数字是否为零，输出真值或假值。

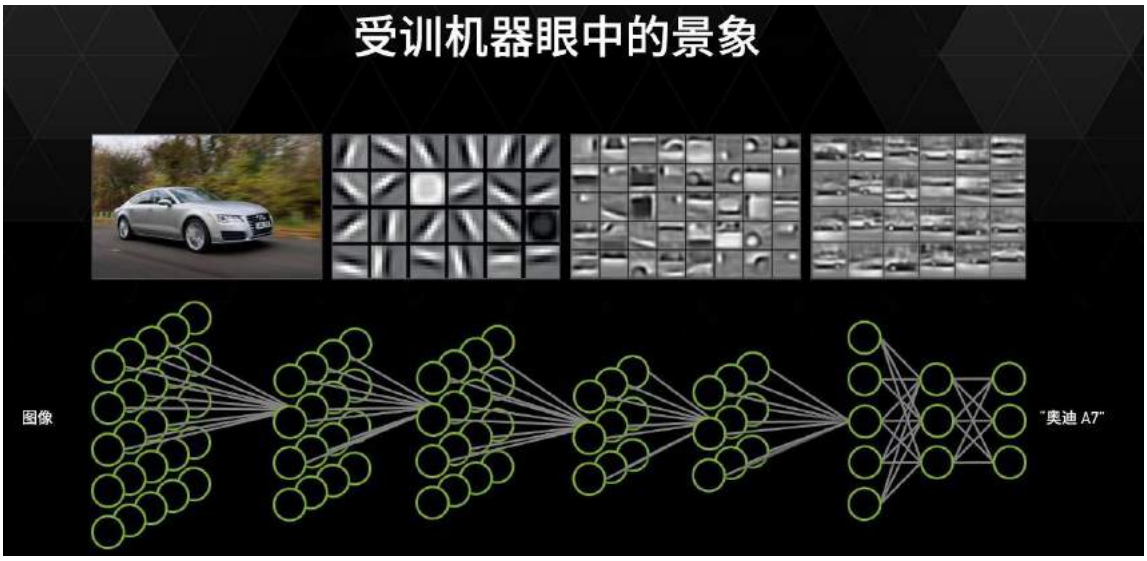
神经网络的关键在于训练网络使其能够进行更佳的预测。用于检测手写零的感知器模型（如图 33 所示）的训练过程，最初为通过向定义数字零的每个特征分配一组权重。然后会向感知器提供数字零，以确认其是否正确识别该数字。这种数据流经过网络直至其就该数字是否为零得出结论的过程，是为向前传播阶段。如果神经网络未正确识别该数字，则需了解识别错误的原因以及误差大小，并需要调整每个特征的权重，直到感知器正确识别零。权重需要进一步调整，直到其正确识别出以各种书写风格写出的零。这种反馈误差并调整定义数字零的每个特征权重的过程，称为“向后传播”。图表中所示的方程式看起来很复杂，但却是所述训练过程的基本数学表达式。

感知器虽然是非常简单的神经网络模型，但是基于类似概念的高级多层神经网络在今天已经广泛使用。一旦网络经过训练可正确识别和分类对象，便将部署于相应领域，并在其中反复运行推理运算。推理（DNN 从指定输入项提取有用信息的过程）示例包括：识别存放进 ATM 机中的支票上的数字，识别 Facebook 照片中朋友的图像，为超过五千万 Netflix 用户提供电影推荐，为无人驾驶汽车识别和分类不同类型的汽车、行人以及道路危险，或实时翻译人类语音。

如图 34 中所示的多层神经网络模型由多个类似感知器的互联复杂节点组成。每个节点会注意许多输入特征，并将其输出项馈送至下面几层的互联节点。

在图 34 所示的模型中，神经模型的第一层将汽车图像分解为各个部分，并寻找线条和角度等基本图案。第二层将这些线条组合起来，寻找更高级别的图案，如车轮、挡风玻璃和车镜。下一层识别车辆类型，最后几层识别特定汽车品牌的型号（本例中的车为奥迪 A7）。

与完全连接的神经网络层拥有类似地位的是卷积层。卷积层中的神经元仅连接至其下层较小区域中的神经元。通常，该区域可能为 5×5 网格的神经元（或者是 7×7 或 11×11）。此网格的大小称为“过滤大小”。因此，卷积层可理解为对其输入项执行卷积。这种类型的连接模式模拟的是人脑感知区中的模式，例如视网膜神经节细胞或初级视皮层中的细胞。



图像来源：Unsupervised Learning Hierarchical Representations with Convolutional Deep Brief Networks（卷积深信度网络无监督性学习层次表示），ICML 2009 & Comm。ACM 2011，Honglak Lee、Roger Grosse、Rajesh Ranganath 和 Andrew Ng。

图 34. 复杂的多层神经网络模型需要更高的计算能力

在 DNN 卷积层中，该层的每个神经元的过滤权重都相同。通常，卷积层以许多*子层*来实现，每个子层具有不同的过滤器。一个卷积层中可能使用数百个不同的过滤器。我们可以将 DNN 卷积层视为同时对其输入项执行数百次不同卷积，这些卷积的结果将供下一层卷积使用。整合卷积层的 DNN 称为“卷积神经网络 (CNN)”。

NVIDIA GPU：深度学习的引擎

先进的 DNN 和 CNN 可能有数百万乃至十亿以上的参数需要通过向后传播进行调整。而且，DNN 需要大量的训练数据才能实现较高的准确度，这意味着成千上万乃至数百万的输入样本必须同时进行向前和向后传输。

目前，学术界和业界普遍认为，GPU 在训练深度神经网络方面是最先进的技术，因为它的速度和能效均优于更传统的基于 CPU 的平台。神经网络由大量相同的神经元构建而成，因此本质上具有高度并行性。这种并行性可很自然地映射到 GPU，相比于仅使用 CPU 的网络训练，GPU 的速度大幅增加。

神经网络非常依赖矩阵数学运算，复杂的多层网络需要出色的浮点计算性能和极高带宽才能提高效率和速度。拥有数千个处理核心、面向矩阵数学运算优化并可实现几十到数百 TFLOPS 性能的 GPU，显然是基于深度神经网络的人工智能和机器学习应用程序的理想计算平台。

训练深度神经网络

先进的神经网络可能有数百万乃至十亿以上的参数需要通过向后传播进行调整。而且，神经网络需要大量的训练数据才能实现较高准确度的收敛，这意味着成千上万乃至数百万的输入样本必须同时进行向前推算和向后传输（参见图 35）。

训练复杂的神经网络需要强大的并行计算性能，而基本等级的神经网络就涉及数万亿次的浮点乘法和加法运算。早期在 GPU 上训练神经网络时，这些计算均通过 NVIDIA Fermi 和 Kepler GPU 架构提供的数千个核心以单精度浮点计算 (FP32) 并行完成。这些架构中的核心主要针对 HPC 优化，并且使用允许快速高精度浮点运算的 FMA 指令来支持单精度 FP32 和双精度 FP64 数据类型。

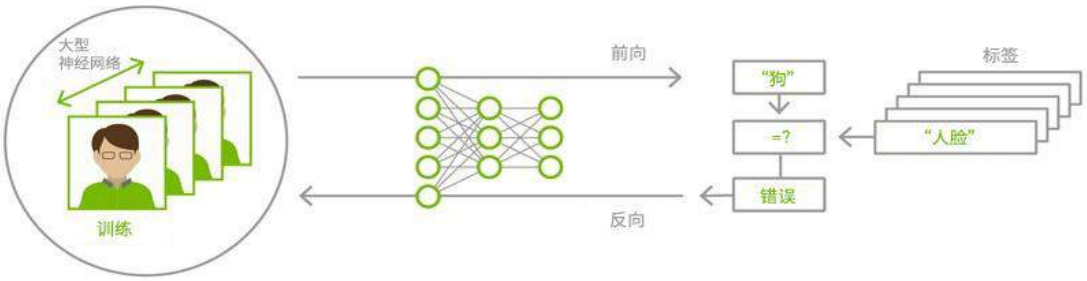


图 35.　训练神经网络

深度学习领域的进一步研究和发展表明，在许多情况下，可使用半精度 FP16 数据类型来训练神经网络，且能够达到与使用 FP32 数据训练相同的准确度。尽管只使用 FP16 数据，某些网络的训练不会收敛，但研究表明，对于网络的大多数卷积层，使用较低精度的数据类型即可解决这一问题，并且结果的累积通常可使用较高精度的数据类型来完成[4]

相比较高精度的 FP32 或 FP64，使用 FP16 数据可减少内存占用和神经网络的带宽要求，从而大幅提升速度。例如，在 NVIDIA Pascal GPU 架构中，与 FP32 算法相比，使用 FP16 运算可将性能提高 2 倍，而 FP16 数据传输所花的时间更少，内存带宽仅为 FP32 传输的一半。

[4] https://arxiv.org/abs/1412.7024

使用训练的神经网络进行推理

训练神经网络是一个计算量巨大的过程，需要使用一系列输入数据，进行误差检测向前传输，以及多次向后传输，以调整网络各层中数百万个神经元的权重。推理涉及的计算较少，但却是延迟敏感型过程，在该过程中，训练的网络会应用于以前未见过的新输入项，以识别图像、翻译语音，通常还会推理新的信息（请参见图 36）。

研究表明，使用半精度 FP16 数据进行推理的分类准确度与 FP32 [5]相同。相比于使用 FP32 数据类型，在 Pascal GPU 和 Tegra X1 SoC 架构中采用 FP16 数据类型[6]，推理吞吐量可提高 2 倍。此外，还可使用 8 位整数 (INT8) 精度进行推理，这在大幅增加推理吞吐量的同时还可最大程度地减少精确度损失。

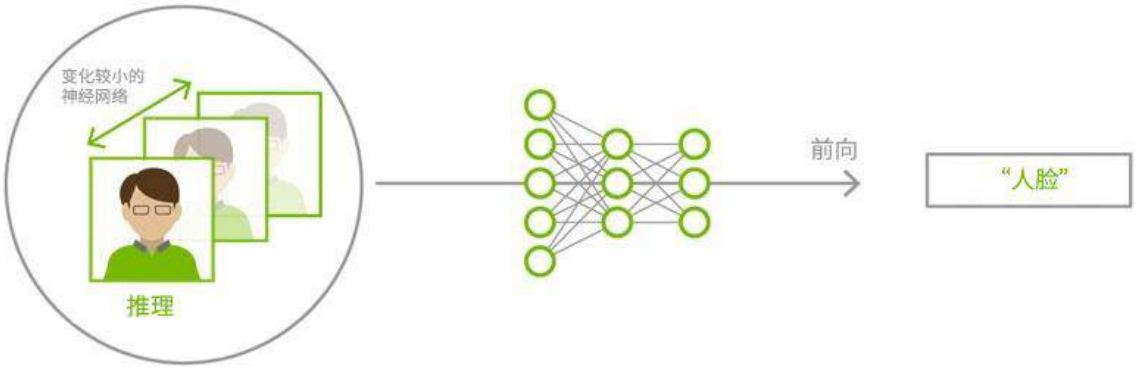


图 36.　神经网络推理

认识到这些好处后，NVIDIA 之前的 Pascal GP100 架构加入了对 FP16 数据格式的本地支持，其他基于 Pascal 的 GPU（如 NVIDIA Tesla P40 和 NVIDIA Tesla P4）也加入对 INT8 的支持，以进一步提升推理性能。

基于 Pascal GP100 的 Tesla P100 卡可提供 21.2 TFLOPS 的 FP16 性能。支持 INT8 运算的 NVIDIA Tesla P40 等 GPU 可提供接近 48 INT8 TOPS 的性能，进一步增强数据中心服务器的推理性能。如本白皮书前文所述，Volta 的 Tensor 核心将性能提升至全新水平，可为推理和训练实现高达 125 TFLOPS 的计算性能。

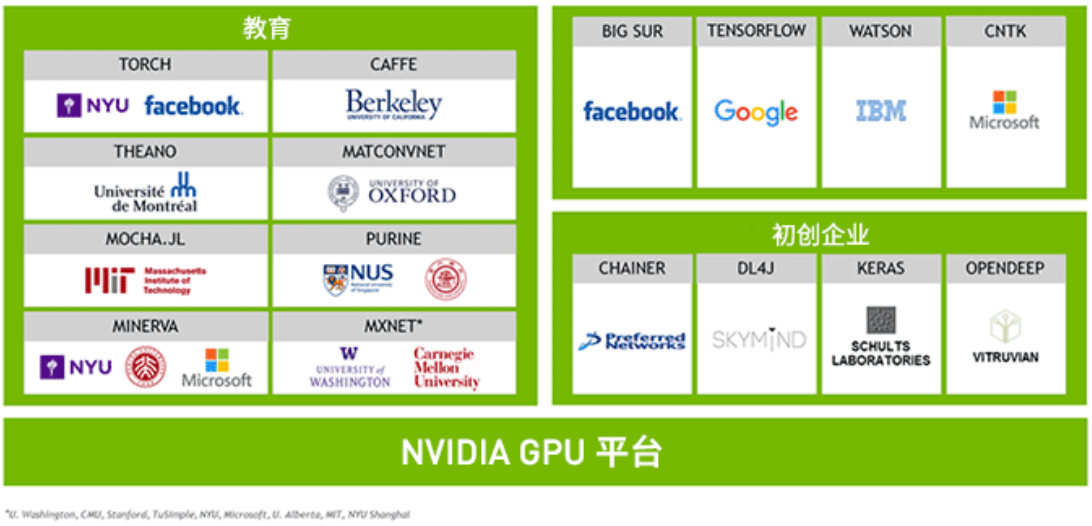
[5] https://arxiv.org/pdf/1502.02551.pdf
[6] https://www.nvidia.com/content/tegra/embedded-systems/pdf/jetson_tx1_whitepaper.pdf

综合性深度学习软件开发工具包

AI 正以极快的速度进行创新。简化编程和提高开发者的工作效率成为最重要的事情。NVIDIA CUDA 平台的出色可编程性以及包含的丰富工具让研究人员能够快速创新。NVIDIA 提供了高性能的工具和库（NVIDIA DIGITS™、cuDNN、cuBLAS 等），可在云端、数据中心、工作站和带有深度学习软件开发工具包 (SDK) 的嵌入式平台中为创新性 GPU 加速机器学习应用程序提供助力。

开发者想要随时随地进行创造和部署。您可以在世界各地的各家 PC OEM、台式机、笔记本电脑、服务器或超级计算机，以及由 Amazon、Google、IBM、Facebook、百度和 Microsoft 等各大公司提供的云服务中获得 NVIDIA GPU。所有主要的 AI 开发框架都支持 NVIDIA GPU 加速，从互联网公司到研究机构再到初创公司，不一而足。不管选择的是何种 AI 开发系统，都可以借助 GPU 加速来提速。

同时，NVIDIA 也为每种规模的计算机创建 GPU，以便 DNN 能为所有类型的智能机器提供支持。GeForce 专为 PC 打造；Tesla 适合云服务和超级计算机；Jetson 适用于机器人和无人机；DRIVE PX 2 适用于汽车。这些 GPU 都采用相同的架构，并能加速深度学习（图 37）。



*U. Washington, CMU, Stanford, TuSimple, NYU, Microsoft, U. Alberta, MIT, NYU Shanghai

图 37. 为每个框架加速

百度、Google、Facebook 和 Microsoft 是采用 NVIDIA GPU 进行深度学习和 AI 处理的第一批企业。事实上，AI 技术使这些公司开发的应用程序回应您说出的话、将语音或文本翻译为另一种语言、识别和自动标记图像，以及为每个用户推荐定制的新闻、娱乐和产品。

业内公司无论新旧，现在都争相使用 AI 来构建新产品和服务，或改善其运营状况。仅仅两年时间，与 NVIDIA 合作开发深度学习的公司数量已经增长将近 13 倍，达到 19000 多家（图 38）。

医疗保健、生命科学、能源、金融服务、汽车、制造和娱乐等行业将会通过洞察海量数据获益。随着 Facebook、Google 和 Microsoft 开放其深度学习平台供所有人使用，基于 AI 的应用程序将会飞速发展。

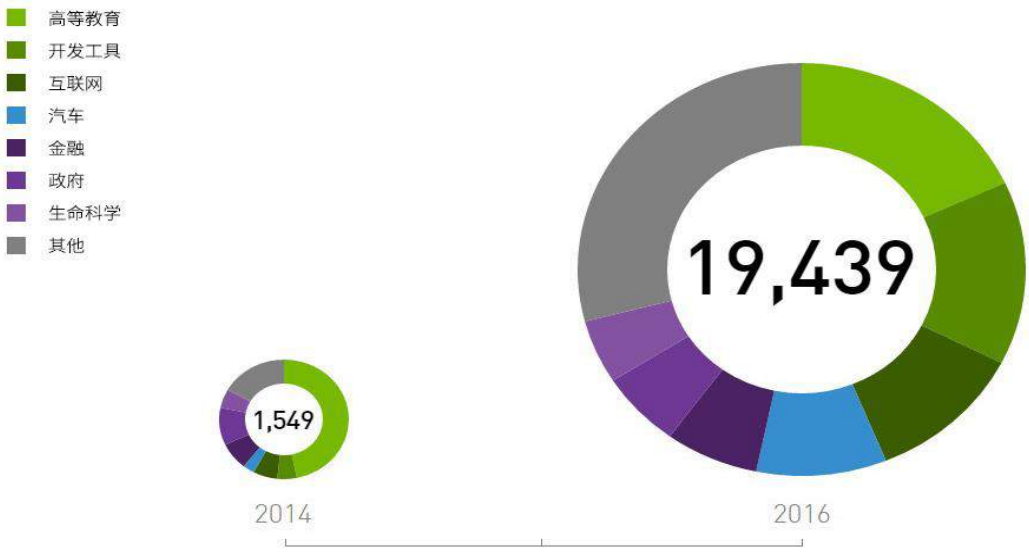
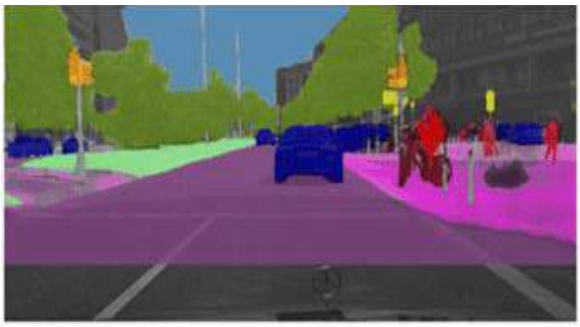


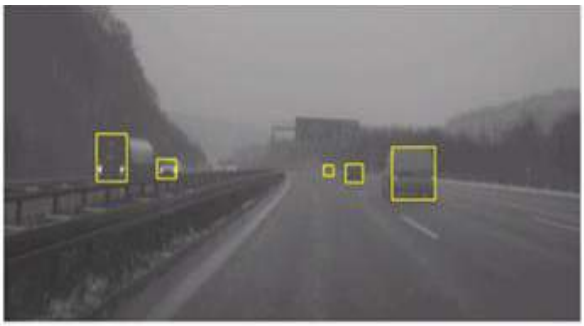
图 38.　在深度学习方面与 NVIDIA 合作的组织

自动驾驶汽车

无论是为人类增添一名超人副驾驶，还是变革个人出行服务，或是降低城市对大型停车场的需求，自动驾驶汽车都具有益于社会的潜质。驾驶是一件复杂的事情，会出现很多意外情况。冷冰冰的雨将路面变成溜冰场、通往目的地的道路封闭、有孩子突然跑到车前。您编写的软件无法预测自动驾驶汽车可能会遇到的所有可能情况，而那正是深度学习的价值所在，它可以学习、适应和改进。NVIDIA 目前正在通过 NVIDIA DRIVE PX 2、NVIDIA DriveWorks 和 NVIDIA DriveNet （请参见图 39）为自动驾驶汽车构建端到端深度学习平台解决方案，包括从训练系统到车内 AI 计算机等诸多方面。结果非常激动人心。拥有超人副驾驶和无人驾驶交通工具的未来将不再只是科幻作品里才有的剧情。



借助 NVIDIA DriveNet，Daimler 将车辆的环境感知性能推进到更接近人类表现的水平，大大超出典型的计算机视觉性能。



NVIDIA 工程师使用合作伙伴奥迪提供的数据集，快速训练了 NVIDIA DriveNet，在雨雪极端困难环境中检测车辆。

图 39.　NVIDIA DriveNet

机器人

业界领先的机器人制造商 FANUC 最近展示了一种流水线机器人，这种机器人通过学习能够从箱子中随机“捡取”目标物体。这款由 GPU 提供技术支持的机器人依靠反复试错进行学习。这项深度学习技术由 Preferred Networks 开发，《华尔街日报》通过标题为“Japan Seeks Tech Revival with Artificial Intelligence”（日本寻求通过人工智能实现技术复兴）的专题文章介绍了这家公司。

在 2017 年 5 月的 GTC 大会上，NVIDIA 宣布了重大创新，推出基于 AI 的全新虚拟机器人训练模拟系统 Isaac。Isaac 系统提供了一套开发工具，可进行高保真机器人模拟和先进的实时渲染。Isaac 允许开发人员使用详细、逼真的测试场景训练他们的虚拟机器人，而这些场景可在多个虚拟机器人中复制。曾经需耗时数月的模拟现在只需数分钟即可完成。并且，由于是完全虚拟的系统，因此不存在损坏或伤害的风险。模拟完成后，可以将经过训练的人工智能快速转移至真实世界中的机器人。然后，开发人员可以迭代并调整机器人测试方法，并在两种环境中来回交换信息。Isaac 基于 Epic Games 的 Unreal Engine 4 增强版本构建，使用 NVIDIA 先进的模拟、渲染和深度学习技术。

医疗保健和生命科学

Deep Genomics 公司正在应用基于 GPU 的深度学习了解基因变异如何导致疾病。Arterys 使用基于 GPU 的深度学习加快医学影像的分析速度。这项技术将部署在 GE Healthcare MRI 机器上，用以协助诊断心脏病。Enlitic 正在运用深度学习分析医学影像，以识别肿瘤、难以发现的骨折和其他病情。

以上仅列举了少许示例用于说明 GPU 和 DNN 如何为各领域的人工智能和机器学习带来革命性变化，相关的例子还有很多。

深度学习突破正在许多层面提升 AI 能力，GPU 加速的深度学习和 AI 系统及算法正推动各领域飞速发展。

通知

本规范中提供的信息在其发布日期之时是准确可靠的。但是，NVIDIA Corporation（“NVIDIA”）对此类信息的准确性或完整性不作任何明示或暗示的陈述或保证。对使用此类信息的后果或因使用此类信息而可能造成侵犯第三方专利权或其他权利的后果，NVIDIA 概不负责。本出版物将取代之前可能已提供的所有其他产品规范。

NVIDIA 保留随时对这一规范进行纠正、更改、增强、改进以及其他改动和/或终止任何产品或服务的权利，恕不另行通知。客户在下订单之前应获取最新的相关规范并验证这些信息是否为当前信息以及是否完整。

除非 NVIDIA 授权代表与客户另行签署销售协议，否则 NVIDIA 产品的销售受订单确认时所提供的 NVIDIA 标准销售条款与条件的制约。就购买这一规范中提到的 NVIDIA 产品而言，NVIDIA 在此明确拒绝应用客户的任何一般条款与条件。

NVIDIA 产品并非针对医学、军事、航空、航天或生命保障设备而设计，并未授权用于也不保证适合用于上述设备，亦不得用于 NVIDIA 产品之失效或故障合理预计会造成人身伤亡或财产或环境破坏的应用场合。客户如果在此类设备或应用场合中融入和/或使用 NVIDIA 产品，NVIDIA 不承担任何相关责任，风险由客户自行承担。

在未经进一步测试或改动的情况下，NVIDIA 并不表示也不担保基于这些规范的产品适合任何具体用途。每款产品所有参数的测试不一定由 NVIDIA 进行。确保产品适合客户所计划的应用场合并针对该应用场合进行必要的测试以避免应用场合出现问题或产品失灵，是客户单方面的责任。客户产品设计中的缺点可能会影响 NVIDIA 产品的质量和可靠性，并且可能会导致超出本规范以外的额外或不同的条件和/或要求。NVIDIA 不承担因下列情况造成失灵、损坏、成本或问题相关的任何责任：(i) 以违反本规范的方式使用 NVIDIA 产品或 (ii) 客户产品设计。

本规范对 NVIDIA 专利权、版权或其他 NVIDIA 知识产权并未作出任何明示或暗示的许可。NVIDIA 所发布的有关第三方产品或服务的信息并不构成 NVIDIA 对于使用该产品或服务的许可，亦不构成担保或支持。使用此类信息可能需要获得第三方的专利权或其他知识产权的许可，或者需要获得 NVIDIA 的专利权或其他知识产权的许可。只有在获得 NVIDIA 书面批准的情况下才可以复制本规范中的信息，而且必须毫无改动地复制并附带所有相应的条件、限制条款和通知。

所有 NVIDIA 设计规范、参考板、文件、图纸、诊断信息、列表和其他文档（统称与单称均为“资料”）均“如实”提供。NVIDIA 并未作出与资料相关的明示、暗示、法定或其他形式的保证，并明确否认与非侵权、适销性和特定用途适用性相关的所有暗示保证。尽管客户可能会因任何原因造成损失，但是 NVIDIA 针对本文所述产品向客户承担的全部责任应仅限于该产品的 NVIDIA 销售条款与条件。

VESA DisplayPort

DisplayPort 和 DisplayPort Compliance Logo、DisplayPort Compliance Logo for Dual-mode Sources 以及 DisplayPort Compliance Logo for Active Cables 是 Video Electronics Standards Association 在美国和其他国家/地区的商标。

HDMI

HDMI、HDMI 徽标和 High-Definition Multimedia Interface（高清多媒体接口）是 HDMI Licensing LLC 的商标或注册商标。

ARM

ARM、AMBA 和 ARM Powered 是 ARM Limited 的注册商标。Cortex、MPCore 和 Mali 是 ARM Limited 的商标。其他所有品牌或产品名称均为其各自所有者的资产。“ARM”用于表示 ARM Holdings plc、其运营公司 ARM Limited 和地区子公司 ARM Inc.、ARM KK、ARM Korea Limited、ARM Taiwan Limited、ARM France SAS、ARM Consulting (Shanghai) Co.Ltd.、ARM Germany GmbH、ARM Embedded Technologies Pvt.Ltd.、ARM Norway、AS 和 ARM Sweden AB。

OpenCL

OpenCL 是 Apple Inc. 的商标，经 Khronos Group Inc. 许可使用。

商标

NVIDIA、NVIDIA 徽标、TESLA、NVIDIA DGX Station、NVLink 和 CUDA 均为 NVIDIA Corporation 在美国和其他国家/地区的商标和/或注册商标。其他公司和产品名称可能是其各关联公司的商标。

版权所有