面向多用户深度学习共享计算优化研究

组会-2020-20-8

Motivation：

目前不同深度学习模型对精度的依赖不同，压缩前和压缩后

1. GPU异构（核内异构）
   1. V100架构：

Tensor core

1. 模型喜好精度不同

原模型

压缩模型

不同模型倾向的精度不同

调研问题：

GPU在体系结构的优化相关paper

调查模型INT8时的敏感度

优化指标：

Makespan

性能和公平性

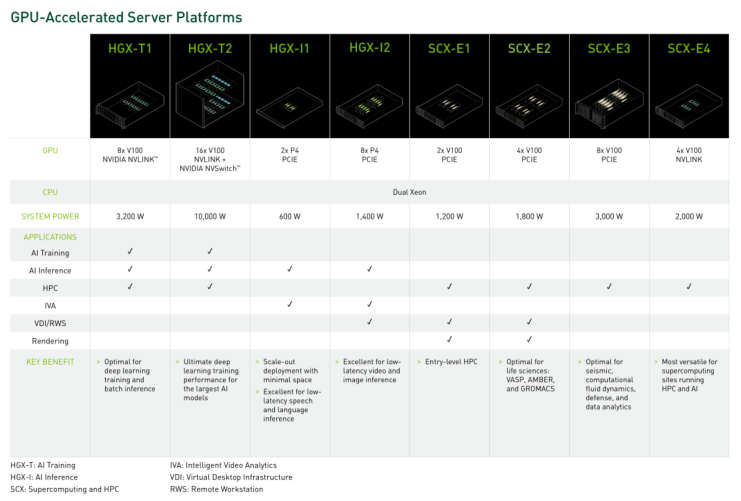
1. [Paddle-lite](https://paddle-lite.readthedocs.io/zh/latest/user_guides/model_optimize_tool.html) 提供多种策略来自动优化原始的训练模型，量化，子图融合，混合调度，kernel优选

[模型优化工具opt](https://paddle-lite.readthedocs.io/zh/latest/user_guides/opt/opt_python.html)

Protobuf格式的模型文件转化为naïve\_buffer格式的模型文件，降低模型体积

执行量化，子图融合，混合调度，kernel优选等图优化操作，提升在paddlelite的运行速度，内存占用

1. [Paddle-slim](https://paddlepaddle.github.io/PaddleSlim/)模型压缩
   1. PaddleSlim是一个模型压缩工具库，包含模型剪裁、定点量化、知识蒸馏、超参搜索和模型结构搜索等一系列模型压缩策略。



V100

GV100 GPU包括211亿个晶体管，芯片尺寸为815毫米。它是在一个新的TSMC 12nm FFN (FinFET NVIDIA)高性能制造工艺定制NVIDIA。GV100提供了相当多的计算性能，并增加了许多新的功能，比之前的Pascal GPU一代。GV100进一步简化了GPU编程和应用程序移植，还提高了GPU资源利用率。GV100是一款极高效的处理器，每瓦特功率提供卓越的性能。

特征：

1. 为深度学习而优化的新型流媒体多处理器(SM multiprocessor)架构

Volta的特点是对位于**GPU中心的SM处理器架构**进行了重大的新设计。与上一代Pascal设计相比，新的Volta SM的能源效率提高了50%，在相同的功率范围内，**FP32和FP64**的性能得到了显著提升。专为深度学习设计的新张量核心可以提供高达12x的高峰值TFLOPS用于训练，**6x高峰值TFLOPS用于推理**。由于具有**独立的并行整数和浮点数据路径**，Volta SM在混合计算和寻址计算的工作负载上效率更高。

1. 第二代NVIDIA NVLink™

第二代NVIDIA的NVLink高速互连提供了更高的带宽，更多的链路，并改善了多gpu

和多gpu /CPU系统配置的可伸缩性。

Volta GV100支持最多6个NVLink链接和300 GB/秒的总带宽，相比之下，GP100支持4个NVLink链接和160 GB/s的总带宽。NVLink现在在IBM Power 9基于CPU的服务器上支持CPU控制和缓存一致性功能。新NVIDIA DGX-1与V100 AI超级计算机使用NVLink提供更大的可扩展性超快速深度学习训练。

1. HBM2内存:更快，更高的效率

Volta高度调优的16gb HBM2内存子系统提供900gb /秒的峰值内存带宽。三星的新

一代HBM2内存和Volta的新一代内存控制器的结合，提供1.5倍的内存带宽，相比Pascal GP100，可在多种工作负载下使用高达95%的内存带宽。

1. Volta多进程服务

多进程服务(MPS)是Volta GV100架构的一个新特性，提供关键部件的硬件加速为共

享GPU的多个计算应用程序提供改进的性能、隔离和更好的服务质量(QoS)。Volta MPS还将MPS客户端的最大数量增加了两倍，从Pascal上的16个增加到**Volta上的48个。**

1. 增强的统一内存和地址翻译服务

GV100统一内存技术包括新访问计数器允许更精确的内存页面迁移处理器访问最频

繁,提高效率之间的内存范围共享处理器。在IBM Power平台上，新的地址转换服务(ATS)支持允许GPU直接访问CPU的页表。

1. 最大性能和最大效率模式
2. 合作组和新的合作启动api

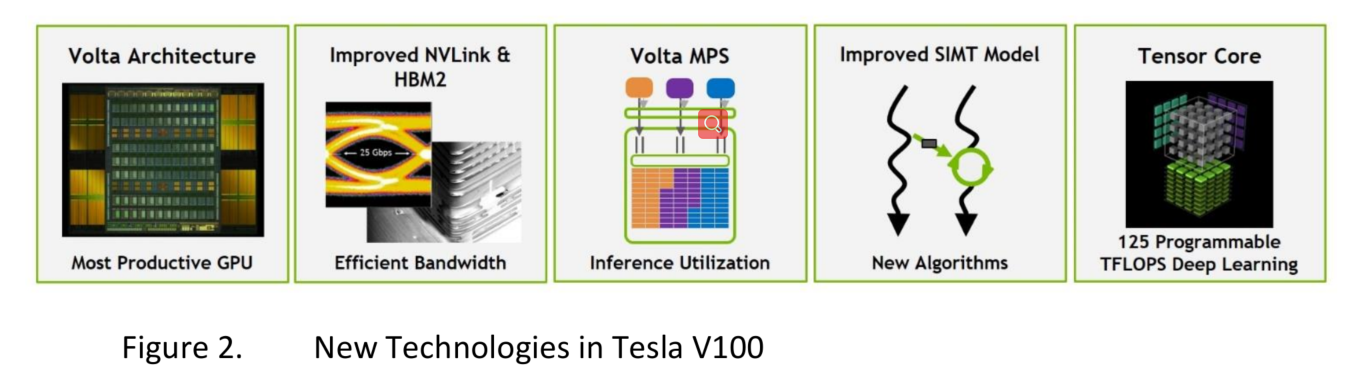
协作组是cuda9中引入的一种新的编程模型，用于组织通信线程的组。协作组允许开

发人员表达线程通信的粒度，帮助他们表达更丰富、更有效的并行分解。基本的协作组功能是支持的所有NVIDIA图形处理器开普勒。Pascal和Volta支持新的合作启动api，支持CUDA线程块之间的同步。Volta增加了对新同步模式的支持。

1. Volta优化软件

深度学习框架的新版本，如Caffe2、MXNet、CNTK、TensorFlow和其他人

利用Volta的性能来提供更快的训练时间和更高的多节点训练性能。Volta优化的GPU加速库版本例如cuDNN、cuBLAS和TensorRT利用了Volta GV100的新功能，为深度学习推理和高性能计算（HPC）应用。NVIDIA CUDA工具包版本9.0包括新的API和对Volta特性的支持提供了更简单的编程功能。



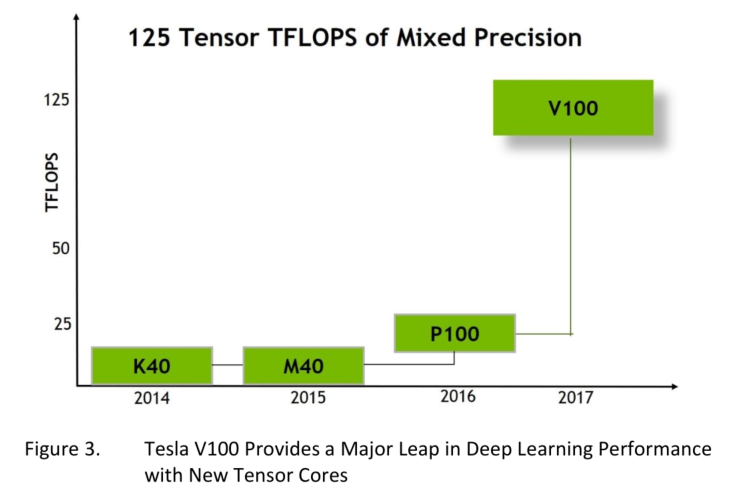
AI和HPC的极端性能

特斯拉V100提供业界领先的浮点数和整数性能。下面是峰值计算率。图3显示了使用新的张量核进行深度学习的Tesla V100性能。

7.8双精度浮点（FP64）性能的TFLOPS

15.7单精度（FP16）性能的TFLOPS

125Tensor TFFLOPS



GPU硬件架构

NVIDIA Tesla V100加速器，以Volta GV100 GPU为特色，是当今世界上性能最高的并行计算处理器。GV100具有重要的新的硬件创新，为深度学习算法和框架提供了巨大的加速，并为高性能计算系统和应用程序提供了更强的计算能力。

与上一代Pascal GP100 GPU一样，GV100 GPU由多个GPU处理集群(GPCs)、纹理处理集群(TPCs)、流多处理器(SMs)和内存控制器组成。一个完整GV100 GPU包括:

1、6个GPCs

每个GPC含有：

7个TPCs（每个含有两个SMs）

14个SMs

2、84个Volta SMs

每个SM：

64 FP32 cores

64 INT32 cores

32 FP64 cores

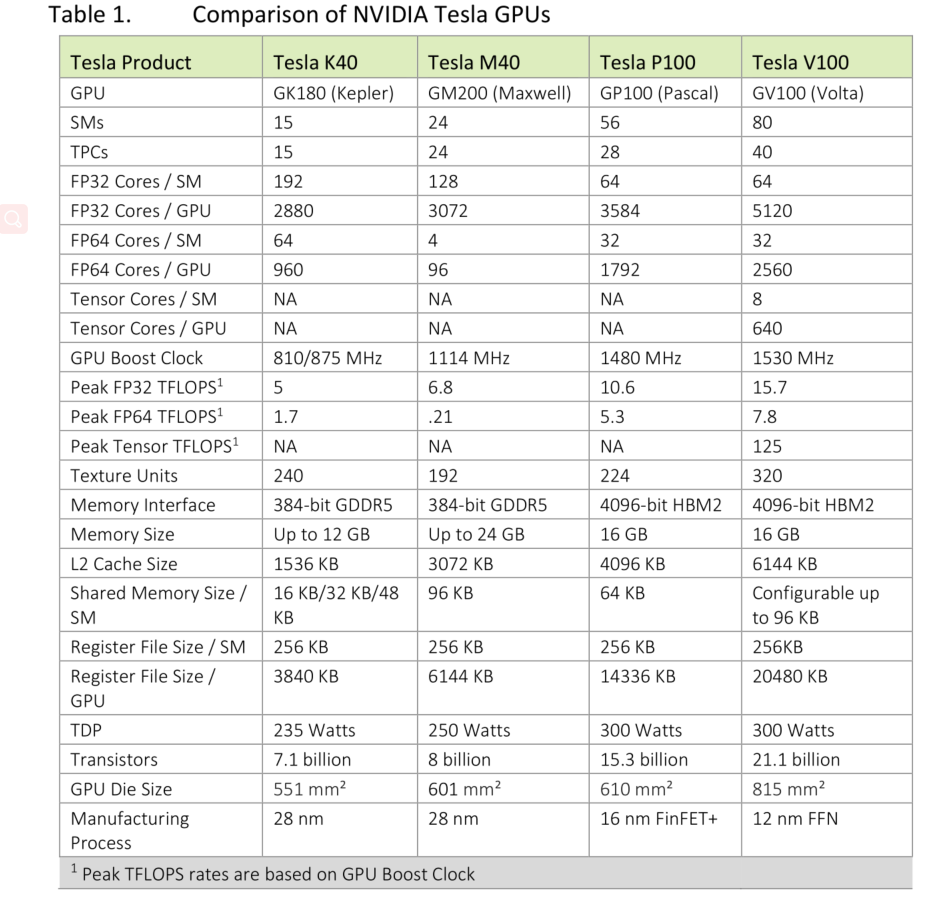
8 Tensor cores

四个结构单元

3、8个512位内存控制器(总共4096位)

一个完整的GV100 GPU拥有84条SMs，共5376个FP32核心、5376个INT32核心、2688个FP64核心、672个张量核心、336个结构单元。每个HBM2 DRAM堆栈由一对内存控制器控制。完整的GV100 GPU包括总共6144 KB的L2缓存。图4显示了一个完整的GV100 GPU 84SMs(GV100不同产品可以使用不同的配置)。特斯拉V100加速器使用了80个SMs。表1对比了NVIDIA Tesla gpu在过去五年的表现。





极致的性能和效率

gpu，同时也提高了能源效率。特斯拉V100为数据中心架构师提供了一个新的设计灵活性维度，它可以配置为提供绝对最大的性能，或最节能的性能。在特斯拉V100中，这两种运行模式被称为Maximum Performance Mode和Maximum Efficiency Mode。

在最大性能模式下,特斯拉V100加速器运营计划书水平的300 W加速应用程序需要最快的计算速度和最高的数据吞吐量

最大效率模式是一种运行模式，使数据中心管理人员运行他们的特斯拉V100加速器的最佳性能每瓦特。可以将V100设置为在功率/性能曲线上运行，以获得最高的功率效率和最佳性能。例如，曲线上最有效的范围可能是TDP的50-60%，而GPU仍然可以达到75-85%的最大性能。数据中心管理人员可以在机架上的所有gpu上设置一个不超过的功率上限，从而显著降低功耗，同时仍然可以获得优异的性能。该功能允许数据中心设计人员在机架的功率预算范围内最大限度地提高整机性能。在某些情况下，这种优化甚至可以在机架中启用额外的服务器节点。

power limit可以由NVIDIA-SMI(数据中心管理器可以使用的命令行实用程序)或NVML(基于c的API库，公开power limit控件，特斯拉OEM合作伙伴可以将其与工具集集成在一起)设置。最大效率模式不降低高峰时钟或记忆时钟在正常运行期间，而GPU工作，以获得最高的时钟速度驻留在指定的功率限制。许多工作负载并不会消耗特斯拉V100的全部300w TDP，因此在某些情况下可能会有更高的功率限制。然而，数据中心设计者应该根据最大的预期工作负载来设置GPU的功率级别，这样就不会超出他们的机架功率预算。

Volta 流多处理器 SMs

Volta的特点是一种新的流式多处理器(SM)架构，提供了性能、能源效率和易于编程方面的重大改进。

主要特性：

新的混合精度张量核专为深度学习矩阵算法，提供12x TFLOPS培训，相比GP100，在相同的功率包络

在一般计算工作负载上提高50%的能源效率

增强的高性能L1数据缓存

一个新的SIMT线程模型，它消除了以前SIMT和SIMD处理器设计中存在的限制

与Pascal GP100类似，GV100 SM包含64个FP32内核，每个SM包含32个FP64内核。然而，GV100 SM使用了一种新的分区方法来提高SM的利用率和整体性能。注意，GP100 SM被划分为两个处理块，每个处理块有32个FP32内核、16个FP64内核、一个指令缓冲区、一个warp调度器、两个分派单元和一个128 KB的寄存器文件。GV100 SM划分为四个处理块,每个都有16 FP32核心,8 FP64核心,16 INT32核心,深入学习两个新mixed-precision张量的核心矩阵运算,一个新的L0指令缓存,经调度程序,一个调度单位,和一个64 KB的注册文件。注意,新L0指令缓存现在用于每个分区提供更高的效率比以前的NVIDIA gpu指令缓冲。(参见图5中的Volta SM)。

一个GV100 SM和一个Pascal GP100 SM有相同数量的寄存器，整个GV100 GPU有更多的SMs，因此总体上有更多的寄存器。总的来说,GV100支持更多的线程,扭曲和线程块飞行相比之前GPU代。

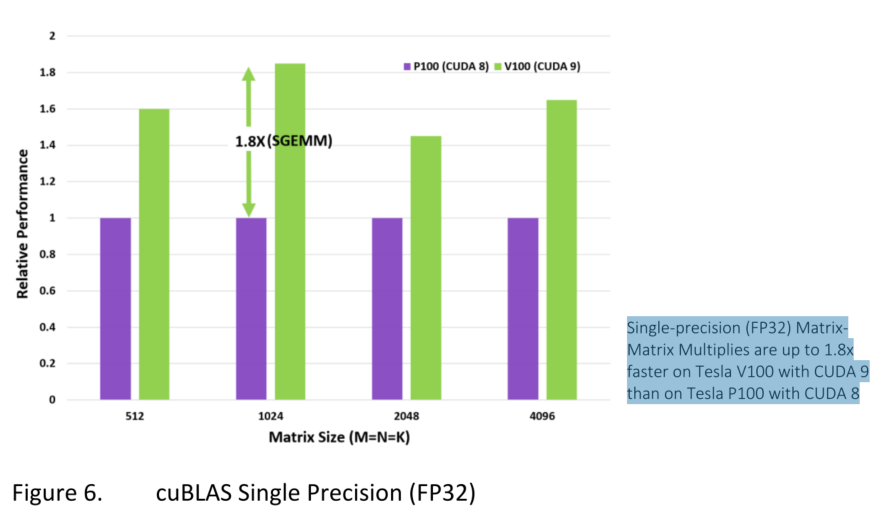
共享内存的合并和L1资源允许共享内存容量的增加每Volta SM 96 KB,而GP100到64 KB。

Tensor Cores

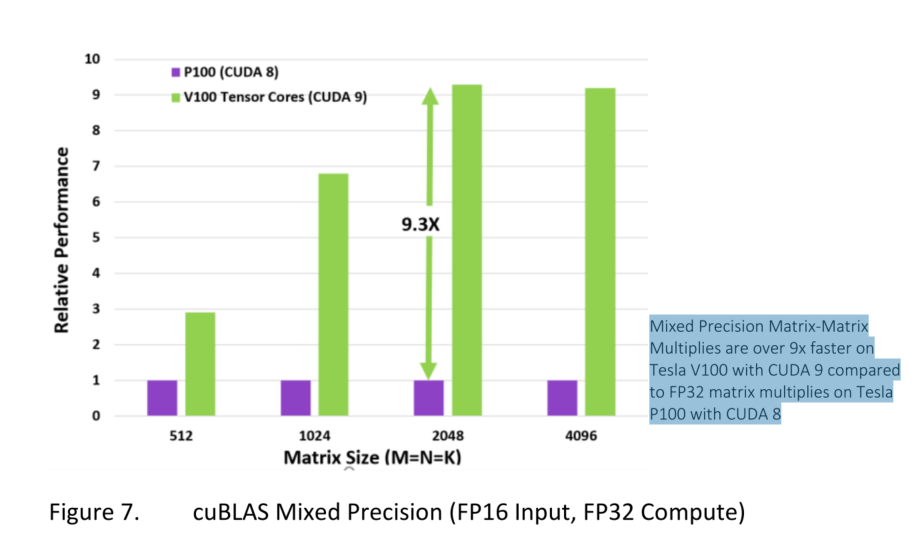
特斯拉V100 GPU包含640个张量核:每个SM 8个(8)，每个SM内每个处理块(分区)2个(2)。在Volta GV100中，每个张量核心每个时钟执行64个浮点FMA操作，而在一个SM中8个张量核心每个cloc执行总共512个FMA操作(或1024个单独的浮点操作)

特斯拉V100张量核心提供多达125张量TFLOPS训练和推理应用。与在P100上使用标准FP32操作相比，Tensor core在特斯拉V100上可以提供高达12倍的峰值TFLOPS，可用于深度学习训练。对于深度学习推理，与P100上的标准FP16操作相比，V100张量核提供了高达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，用于在网络的连接层中乘法输入数据和权重的大型矩阵。对于使用单精度矩阵乘法的应用，图6显示，配备CUDA 9的特斯拉V100的性能比配备CUDA 8的特斯拉P100高1.8倍。对于用于训练和推理操作的具有半精度输入的矩阵乘法，图7显示，对于具有FP16输入和FP32累加的矩阵运算的情况，与P100相比，Volta的混合精度张量核的性能提高了9x以上。



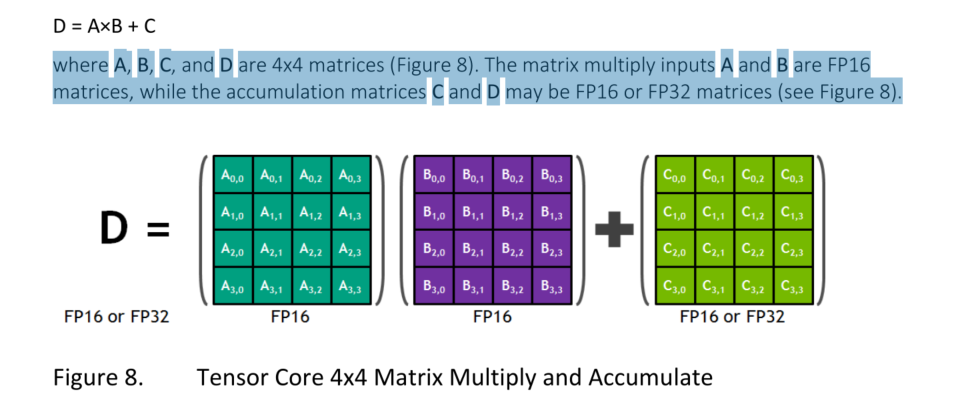
单精度(FP32)矩阵矩阵乘法在配备CUDA 9的特斯拉V100上比配备CUDA 8的特斯拉P100快1.8倍



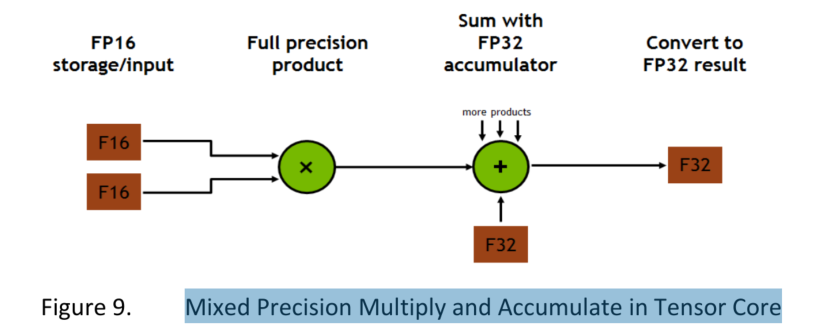
混合精度矩阵-矩阵乘法在特斯拉V100和CUDA 9上比FP32矩阵乘法在特斯拉P100和CUDA 8上快9倍以上

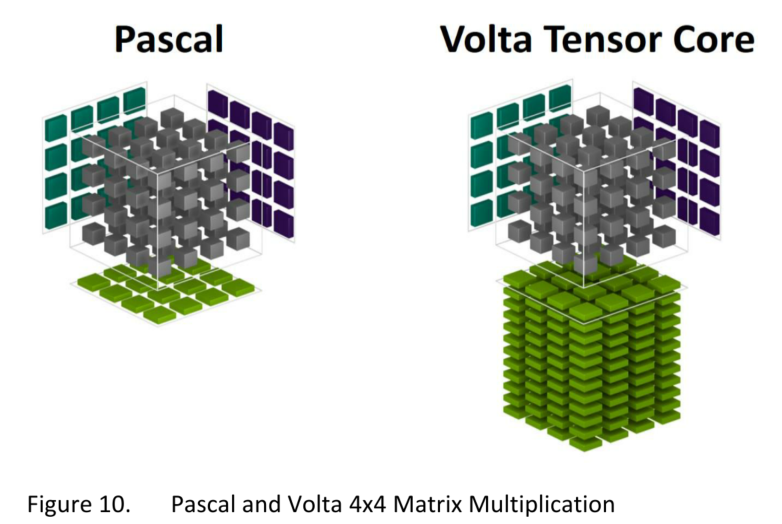
张量核心及其关联的数据路径是定制设计的，以高能效显著提高浮点计算吞吐量。

每个张量核在一个4x4矩阵上操作，并执行以下操作: 其中A、B、C和D是4×4矩阵(图8)。矩阵乘法输入A和B是FP16矩阵，而累积矩阵C和D可以是FP16或FP32矩阵(见图8)。



张量核使用FP32累加对FP16输入数据进行运算。FP16乘法产生一个全精度乘积，然后使用FP32与其他中间乘积相加进行4×4矩阵乘法(见图9)。在实践中，张量核用于执行更大的2D或更高维度的矩阵运算，由这些更小的元素构成。





显示了4x4矩阵乘法(使用立方体外的两个源4x4矩阵)，需要64次运算(由立方体表示)才能生成4x4输出矩阵(显示在立方体下方)。基于伏特的V100加速器配有张量核心，执行这种计算的速度比基于帕斯卡的特斯拉P100快12倍。

伏打张量核心是可访问的，并在CUDA 9 C++应用编程接口中显示为warp级矩阵运算。该应用编程接口公开了专门的矩阵加载、矩阵乘法和累加以及矩阵存储操作，以有效地使用CUDA-C++程序中的张量核。

在CUDA级，warp级接口采用16x16大小的矩阵，跨越翘曲的所有32个线程。除了CUDA-C++接口直接对张量核进行编程之外，cuBLAS和cuDNN库也进行了更新，提供了新的库接口，以便将张量核用于深度学习应用程序和框架。NVIDIA与许多流行的深度学习框架(如Caffe2和MXNet)合作，使张量核心能够用于基于Volta GPU的系统的深度学习研究。英伟达正在努力在其他框架中增加对张量核心的支持。

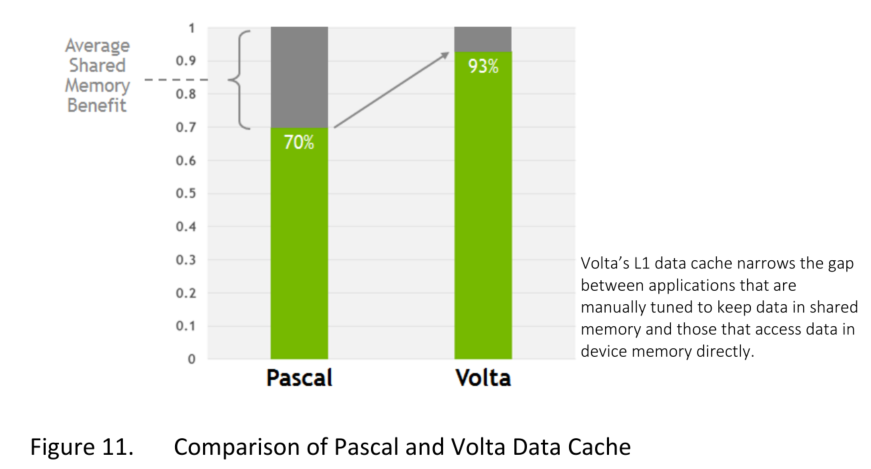
增强型L1数据缓存和共享内存

伏打SM的新组合L1数据缓存和共享内存子系统显著提高了性能，同时简化了编程，减少了达到或接近峰值应用性能所需的调整。

将数据缓存和共享内存功能结合到单个内存块中，可以为两种类型的内存访问提供最佳的整体性能。总容量为128 KB/SM，比GP100数据缓存大七倍多，所有这些都可以被不使用共享内存的程序用作缓存。纹理单元也使用缓存。例如，如果共享内存配置为64 KB，纹理和加载/存储操作可以使用剩余的64kb L1。

共享内存块内的集成确保伏打GV100 L1缓存比过去NVIDIA GPUs中的L1缓存具有更低的延迟和更高的带宽。L1 In Volta作为一个高吞吐量的数据流管道，同时为频繁重用的数据提供高带宽和低延迟的访问，这是两全其美的。这种组合是沃尔特独有的，提供了比过去更易接近的性能。

在GV100中将L1数据缓存与共享内存合并的一个关键原因是允许L1缓存操作获得共享内存性能的优势。共享内存提供了高带宽、低延迟和一致的性能(没有缓存未命中)，但是**CUDA程序员**需要明确地管理这个内存。Volta缩小了明确管理共享内存的应用程序和直接访问设备内存中数据的应用程序之间的差距。为了证明这一点，我们修改了一套程序，用设备内存阵列替换共享内存阵列，这样访问就可以通过L1缓存。如图11所示，在Volta上，这些代码在不使用共享内存的情况下运行时，性能损失仅为7%，而在Pascal上，性能损失为30%。虽然共享内存仍然是获得最高性能的最佳选择，但新的沃尔特L1设计使程序员能够以更少的编程工作量快速获得出色的性能。



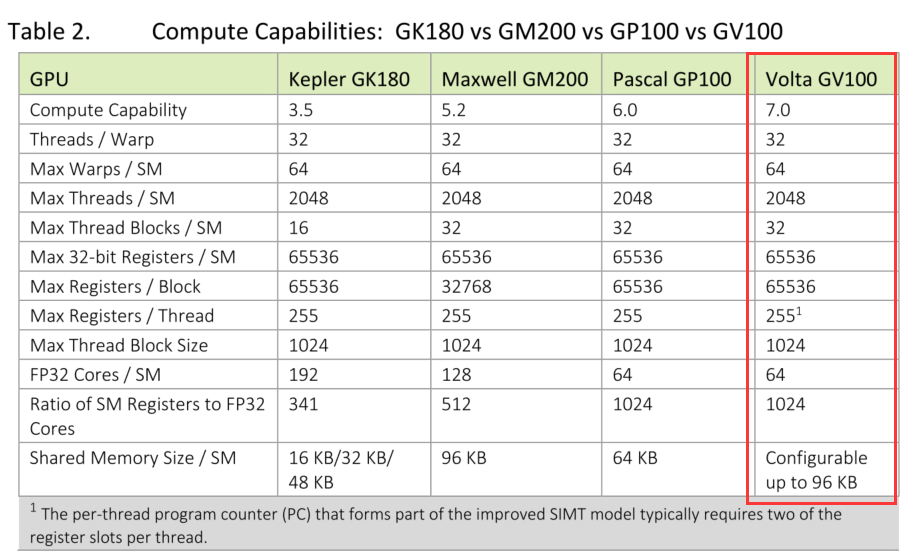
GV100 L1高速缓存在各种共享内存不是最佳选择或无法工作的情况下提高了性能。通过Volta GV100，共享内存和L1的合并提供了一条通往全局内存的高速路径，能够在无限的高速缓存缺失中进行流式访问。以前的NVIDIA GPUs只执行加载缓存，而GV100引入写缓存(存储操作的缓存)来进一步提高性能。

FP32和INT32操作的同时执行

与不能同时执行FP32和INT32指令的Pascal GPUs不同，Volta GV100 SM包括独立的FP32和INT32内核，允许在全吞吐量下同时执行FP32和INT32操作，同时还增加了指令发布吞吐量。对于核心FMA(融合乘加)数学运算，相关指令发出延迟也减少了，volta只需要四个时钟周期，而帕斯卡需要六个周期。

许多应用程序都有内部循环，执行指针算术(整数内存地址计算)和浮点计算，浮点计算将受益于FP32和INT32指令的同时执行。流水线循环的每次迭代可以更新地址(INT32指针算法)，并为下一次迭代加载数据，同时在FP32中处理当前迭代。

计算能力7.0



更高的带宽，更多的链接，更多的功能

NVLink是NVIDIA的高速互连技术，最初于2016年推出，配备特斯拉P100加速器和帕斯卡GP100。与使用PCIe互连相比，NVLink为图形处理器到图形处理器和图形处理器到中央处理器的系统配置提供了更高的性能。有关NVLink技术的基本详细信息，请参考帕斯卡架构白皮书。特斯拉V100推出第二代NVLink，提供更高的链接速度、每GPU更多链接、CPU主控、缓存一致性和可扩展性改进。

更多链接，更快链接

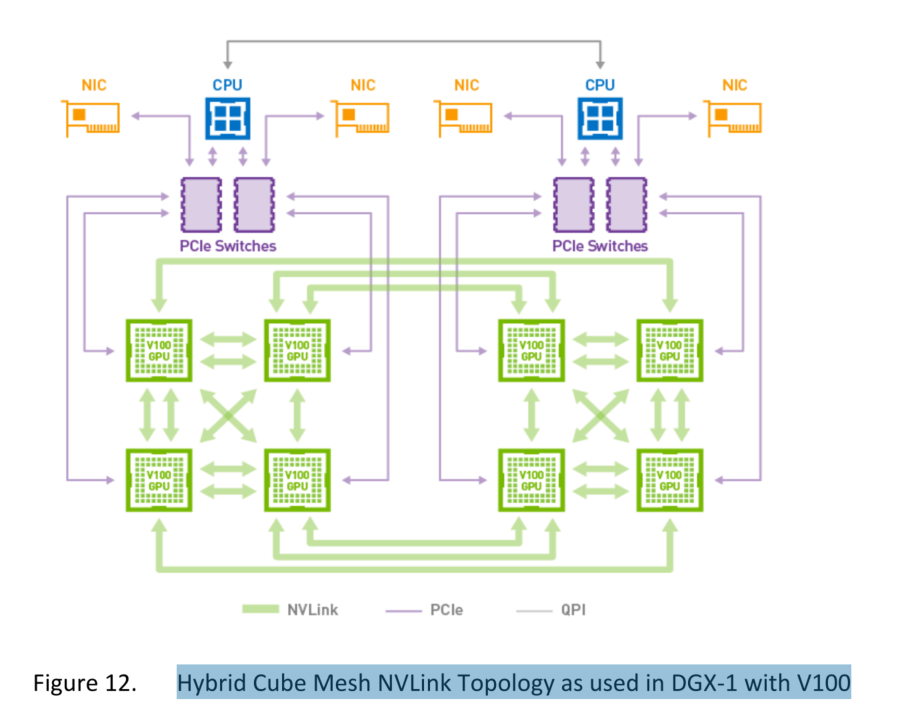
随着开发人员在人工智能计算等应用中暴露和利用更多的并行性，具有多个GPU和CPU的系统在各种行业中变得越来越普遍。这种趋势增加了对更快、更可扩展的多处理器互连的需求。同样，在数据中心、研究机构和超级计算机中部署了具有数万到数千个计算节点的高性能GPU加速系统，以解决更大的问题。英伟达自己的DGX-1系统与P100和V100结合了英伟达技术。2016年，英伟达与IBM密切合作，利用英伟达帕斯卡GPU和IBM Power 8+CPU构建高性能服务器。

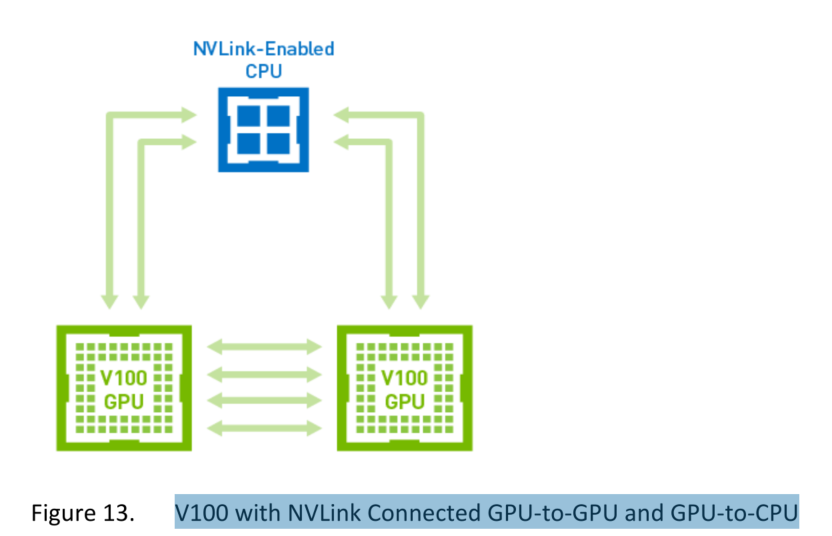
如今，NVIDIA正与IBM合作，使用特斯拉V100加速器和与NVLink连接的Power 9 CPUs来构建更高性能的服务器。与帕斯卡上的NVLink相比，V100上的NVLink将信令速率从20千兆位/秒提高到25千兆位/秒。每个链路现在在每个方向上提供25千兆字节/秒。支持的链路数量从4个增加到6个，将支持的GPU NVLink带宽提高到300千兆字节/秒。这些链路可以专门用于图形处理器到图形处理器的通信，如图12所示的V100拓扑的DGX-1，或者如图13所示的图形处理器到图形处理器和图形处理器到中央处理器通信的某种组合。

更多功能

第二代NVLink允许从CPU直接加载/存储/原子访问每个GPU的HBM2内存。结合新的中央处理器控制功能，NVLink支持一致性操作，允许从图形内存中读取的数据存储在中央处理器的缓存层次结构中。从中央处理器高速缓存访问的较低延迟是中央处理器性能的关键。虽然P100支持对等GPU原子，但不支持通过NVLink发送GPU原子并在目标CPU上完成。NVLink增加了对由GPU或CPU启动的原子的支持。增加了对地址转换服务的支持，允许图形处理器直接访问中央处理器的页表。增加了链路的低功耗工作模式，当链路未被大量使用时，可以显著节省功耗(参见图14)。

链接数量的增加、更快的链接速度和第二代NVLink的增强功能，加上沃尔特的新张量核心，导致多GPU特斯拉V100系统的深度学习性能显著提高，超过了具有特斯拉P100 GPU的系统。





HBM2内存架构

特斯拉P100是世界上第一个支持高带宽HBM2存储技术的GPU架构。特斯拉V100采用了更快、更高效的HBM2实现。HBM2内存由与GPU位于同一物理封装上的内存堆栈组成，与传统的GDDR5内存设计相比，节省了大量功耗和面积，从而允许在服务器中安装更多GPU。

特斯拉V100中的HBM2每个HBM2堆栈使用四个内存芯片，四个堆栈，最大16GB GPU内存。HBM2内存在四个堆栈中提供900 GB/秒的峰值内存带宽。相比之下，特斯拉P100的最高速度为732 GB/s。HBM2技术的更多细节包含在我们的帕斯卡架构白皮书中。

与特斯拉P100相比，特斯拉V100的峰值动态随机存取存储器带宽更高，此外，V100图形处理器的HBM2效率也得到了显著提高。三星的新一代HBM2内存和沃尔特的新一代内存控制器相结合，提供了比帕斯卡GP100高1.5倍的内存带宽，以及运行许多工作负载时超过95%的内存带宽效率(见图15)。

ECC内存弹性

特斯拉V100 HBM2内存子系统支持单错误纠正双错误检测(SECDED)纠错码(ECC)来保护数据。ECC为对数据损坏敏感的计算应用程序提供了更高的可靠性。这在大规模集群计算环境中尤其重要，在这种环境中，图形处理器处理非常大的数据集和/或长时间运行应用程序。HBM2支持本机或边带纠错码，其中一个独立于主存储器的小存储区域用于纠错码位。这与内联纠错码相比，在内联纠错码中，主存储器的一部分被分割出来用于纠错码位，如特斯拉K40 GPU的GDDR5存储器子系统，其中整个GDDR5的6.25%被保留用于纠错码位。有了V100和P100，纠错码可以在没有带宽或容量损失的情况下有效。对于内存写入，纠错码位是在写入请求的32字节数据中计算的。为每八个字节的数据创建八个纠错码位。对于内存读取，32个纠错码位与每个32字节的数据读取并行读取。纠错码位用于纠正单比特错误或标记双比特错误。GV100中的其他关键结构也受到SECDED ECC的保护，包括SM寄存器文件、L1缓存和L2缓存。帕斯卡尔GP100的相同结构提供了相同的SECDED纠错码保护，以确保高水平的错误检测和纠正以及整体内存弹性。

复制引擎增强功能

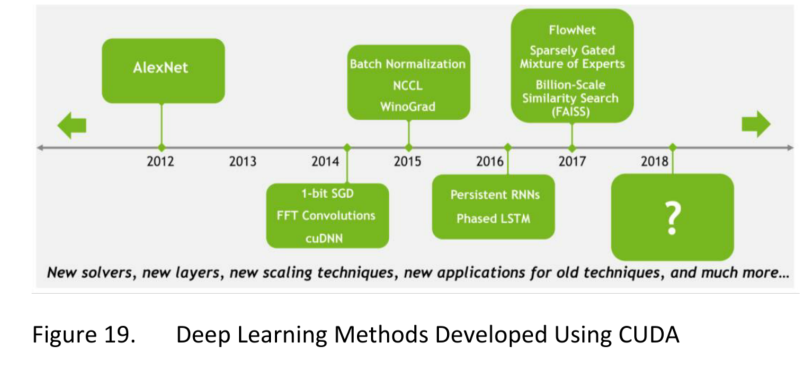
NVIDIA GPU复制引擎在多个GPU之间或GPU与CPU之间传输数据。在以前的图形处理器中，如果源或目标内存地址没有在图形处理器页表中映射，执行复制引擎传输(类似于直接存储器存取传输)可能会导致致命错误。以前的复制引擎要求源和目标内存区域都被固定(不可分页)。新的Volta GV100 GPU复制引擎可以为没有映射到页表中的地址生成页错误。然后，内存子系统可以处理页面错误，将地址映射到页面表中，之后复制引擎可以执行传输。这是一个重要的增强，尤其是在大型多GPU /多CPU系统中，因为在多个处理器之间为多个复制引擎操作锁定内存会大大减少可用内存。有了硬件页面错误，地址可以传递给复制引擎，而不用担心它们是否是驻留的，复制过程就可以工作了。这一特性可能会在今天的自动测试系统中使用。

特斯拉V100保持了与特斯拉P100相同的SXM2板外形。主要区别是使用GV100 GPU而不是GP100。SXM2主板提供NVLink和PCIe 3.0连接。一个或多个V100加速器可用于工作站、服务器和大规模计算系统。V100加速器为140毫米x 78毫米，包括向GPU提供各种所需电压的高效电压调节器。V100额定功率为300瓦TDP(热设计功率)。图16显示了特斯拉V100加速器的正面，图17显示了特斯拉V100加速器的背面。图18显示了英伟达特斯拉V100 SXM2模块的风格化分解图。

GV100 CUDA软硬件体系结构进展

NVIDIA CUDA是NVIDIA创建的并行计算平台和编程模型，旨在让应用程序开发人员能够访问NVIDIA GPUs的海量并行处理能力。CUDA是GPU加速深度学习以及广泛的其他计算和内存密集型应用的基础，从天文学到分子动力学模拟，再到计算金融。数以千计的GPU加速应用建立在CUDA并行计算平台上。

NVIDIA CUDA工具包提供了一个全面的环境，使开发人员能够使用对C和C++编程语言的扩展来构建大规模并行应用程序。CUDA的灵活性和可编程性使其成为研究新的深度学习和并行计算算法的首选平台。图19显示了基于CUDA平台的深度学习创新的时间表。



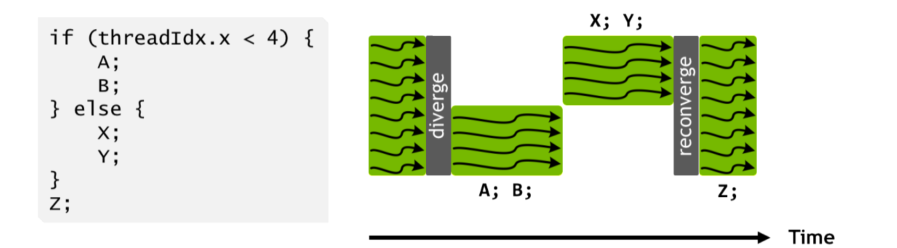
本节中描述的Volta架构的进步进一步扩展了CUDA应用程序中并行线程的可能性，从而在CUDA平台的能力、灵活性、生产率和可移植性方面实现了重大改进。

独立线程调度

沃尔特架构的设计比以前的图形处理器更容易编程，使用户能够高效地处理更复杂和多样的应用程序。Volta GV100是第一个支持独立线程调度的GPU，它支持程序中并行线程之间的细粒度同步和协作。沃尔特的主要设计目标之一是减少在GPU上运行程序所需的工作量，并在线程合作中实现更大的灵活性，从而提高细粒度并行算法的效率。

以前的英伟达图形处理器SIMT型号

帕斯卡和早期的英伟达图形处理器以SIMT(单条指令，多线程)的方式执行32个线程(称为扭曲)的组。帕斯卡扭曲使用所有32个线程共享的单个程序计数器，并结合一个活动掩码来指定在任何给定时间扭曲的哪些线程是活动的。这意味着不同的执行路径会让一些线程处于非活动状态，从而序列化扭曲的不同部分，如图20所示。原始遮罩会一直保存到扭曲重新收敛，通常是在分叉部分结束时，此时遮罩会恢复，线会再次一起运行

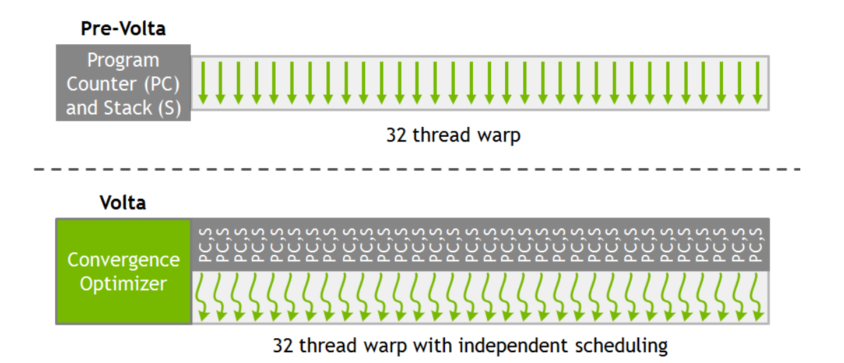


帕斯卡和早期NVIDIA GPUs的SIMT扭曲执行模型下的线程调度。大写字母代表程序伪代码中的语句。扭曲中的不同分支被序列化，这样分支一端的所有语句在另一端的任何语句被执行之前被一起执行到完成。在else语句之后，经线通常会重新聚合。

帕斯卡state执行模型通过减少跟踪线程状态所需的资源数量和积极地重新收集线程以最大化并行性来最大化效率。然而，跟踪整个扭曲的线程状态意味着当执行路径分叉时，采用不同分支的线程会失去并发性，直到它们重新聚合。这种并发性的丧失意味着来自不同区域或不同执行状态的相同扭曲的线程不能相互发送信号或交换数据。这表现出一种不一致性，即来自不同经线的线继续并发运行，但是来自相同经线的分叉线顺序运行，直到它们重新汇合。这意味着，例如，要求细粒度共享由锁或互斥体保护的数据的算法很容易导致死锁，这取决于竞争线程来自哪个扭曲。因此，在Pascal和更早的GPU上，程序员需要避免细粒度同步或依赖无锁或翘曲感知算法。

Volta SIMT Model

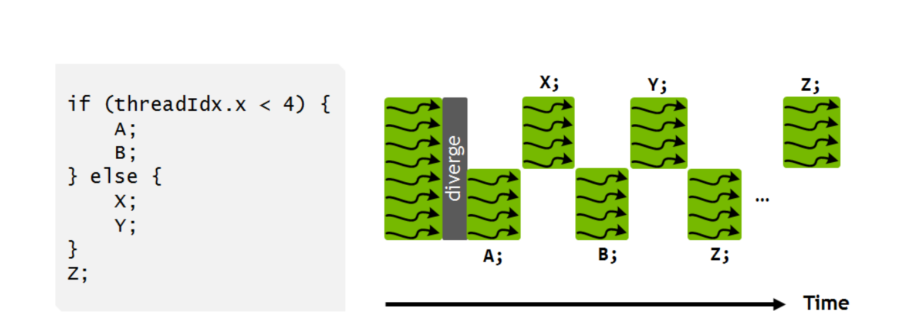
Volta通过在所有线程之间实现平等的并发性来改变这种情况，而不考虑扭曲。它通过维护每个线程的执行状态来做到这一点，包括程序计数器和调用堆栈，如图21所示。



Volta(底部)独立线程调度架构框图与Pascal和早期架构相比(顶部)。Volta维护每个线程的调度资源，如程序计数器(PC)和调用堆栈(S)，而早期的体系结构维护这些资源。

Volta的独立线程调度允许GPU产生任何线程的执行，要么更好地利用执行资源，要么允许一个线程等待另一个线程产生数据。为了最大限度地提高并行效率，沃尔特包括一个调度优化器，它可以确定如何将来自同一经线的活动线程分组到SIMT单元中。这保留了先前NVIDIA GPUs中SIMT执行的高吞吐量，但具有更大的灵活性:线程现在可以在亚扭曲粒度上分叉和重新聚合，而Volta中的收敛优化器仍然将执行相同代码的线程分组在一起，并并行运行它们以获得最大效率

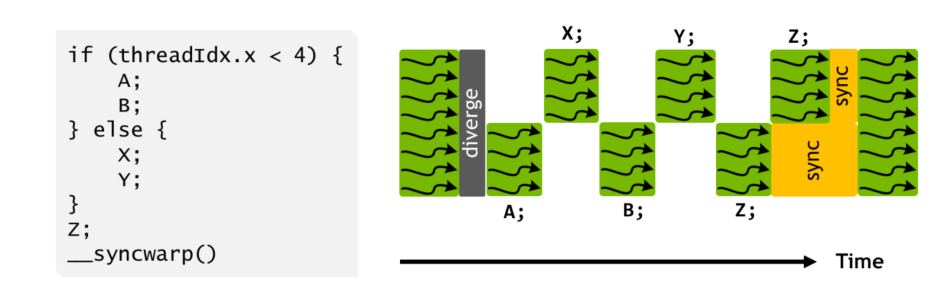
在Volta上执行图20中的代码示例看起来有些不同。程序中if和else分支的语句现在可以在时间上交错，如图22所示。请注意，执行仍然是SIMT:在任何给定的时钟周期，CUDA内核都像以前一样为warp中的所有活动线程执行相同的指令，保持了以前架构的执行效率。重要的是，沃尔特在一个扭曲中独立调度线程的能力使得以更自然的方式实现复杂、细粒度的算法和数据结构成为可能。虽然调度程序支持线程的独立执行，但它优化了非同步代码，以保持尽可能多的收敛性，从而实现最大的SIMT效率。



Volta独立线程调度支持从不同分支交错执行语句。这使得细粒度并行算法的执行成为可能，其中扭曲中的线程可以同步和通信。

有趣的是，图22并没有显示出翘曲中的所有线程同时执行语句Z。这是因为调度器必须保守地假设Z可能会产生其他执行分支所需的数据，在这种情况下，自动强制重新聚集是不安全的。在A、B、X和Y不包含同步操作的常见情况下，调度程序可以识别出翘曲在Z轴上自然重新聚合是安全的，就像在以前的体系结构中一样。

程序可以调用新的CUDA 9 warp同步函数\_\_syncwarp()来强制重新收敛，如图23所示。在这种情况下，扭曲的不同部分可能不会一起执行Z，但是在任何线程到达\_\_syncwarp()之后的语句之前，扭曲内的所有线程执行路径都将完成。类似地，在执行Z之前调用\_\_syncwarp()会在执行Z之前强制重新聚集，如果开发人员知道这对他们的应用程序是安全的，就有可能实现更高的SIMT效率。



程序使用显式同步在扭曲中重新聚合线程

无饥饿算法

无饥饿算法是独立线程调度支持的关键模式。这些是并行计算算法，只要系统确保所有线程都有足够的权限访问一个竞争资源，就能保证正确执行。例如，如果试图获取互斥体的线程保证最终成功，互斥体(或锁)可以在无饥饿算法中使用。在不支持starvationfreedom的系统中，一个或多个线程可能会重复获取和释放互斥体，同时使另一个线程无法成功获取互斥体。

Page 28

<https://www.sohu.com/a/214439561_560178>

TensorRT

<https://docs.nvidia.com/deeplearning/tensorrt/developer-guide/index.html>

TensorRT 通过组合层和优化内核选择来优化网络，从而改善延迟、吞吐量、能效和内存消耗。如果应用程序指定，它将进一步优化网络以较低的精度运行，进一步提高性能并降低内存要求。

