

AI 芯片 – GPU 详解

# NVLink基础与结构



ZOMI



NVIDIA®

# Talk Overview

## I. 硬件基础

- GPU 工作原理
- GPU AI 编程本质

## 2. 英伟达 GPU 架构

- GPU 基础概念
- 从 Fermi 到 Volta 架构
- Turing 到 Hopper 架构

## 3. GPU 详解

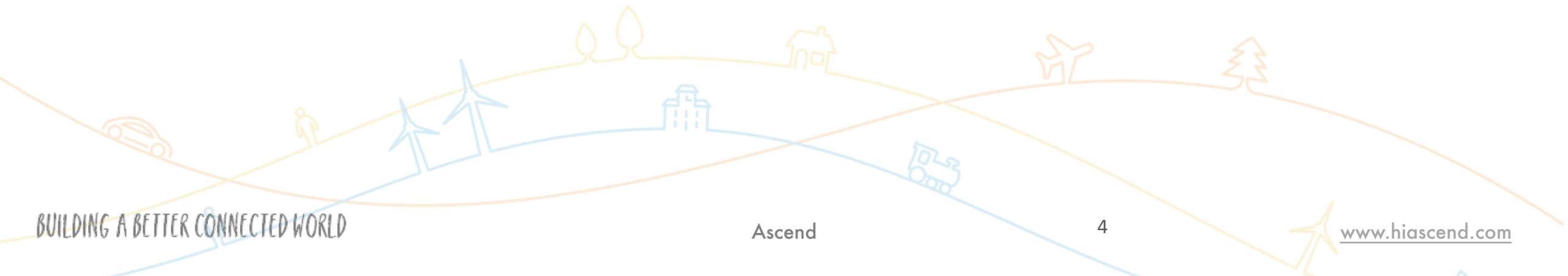
- Tensor Core 原理
- Tensor Core 架构演进
- Tensor Core 深度剖析
- 分布式训练和 NVLink 发展
- NVLink 原理
- NVSwitch 原理

# Talk Overview

## I. 内容介绍

- Before NVLink – NVLink 发展
- Architecture NVLink – NVLink 结构
- Topology NVLink– NVLink 拓扑

# Before NVLink



BUILDING A BETTER CONNECTED WORLD

Ascend

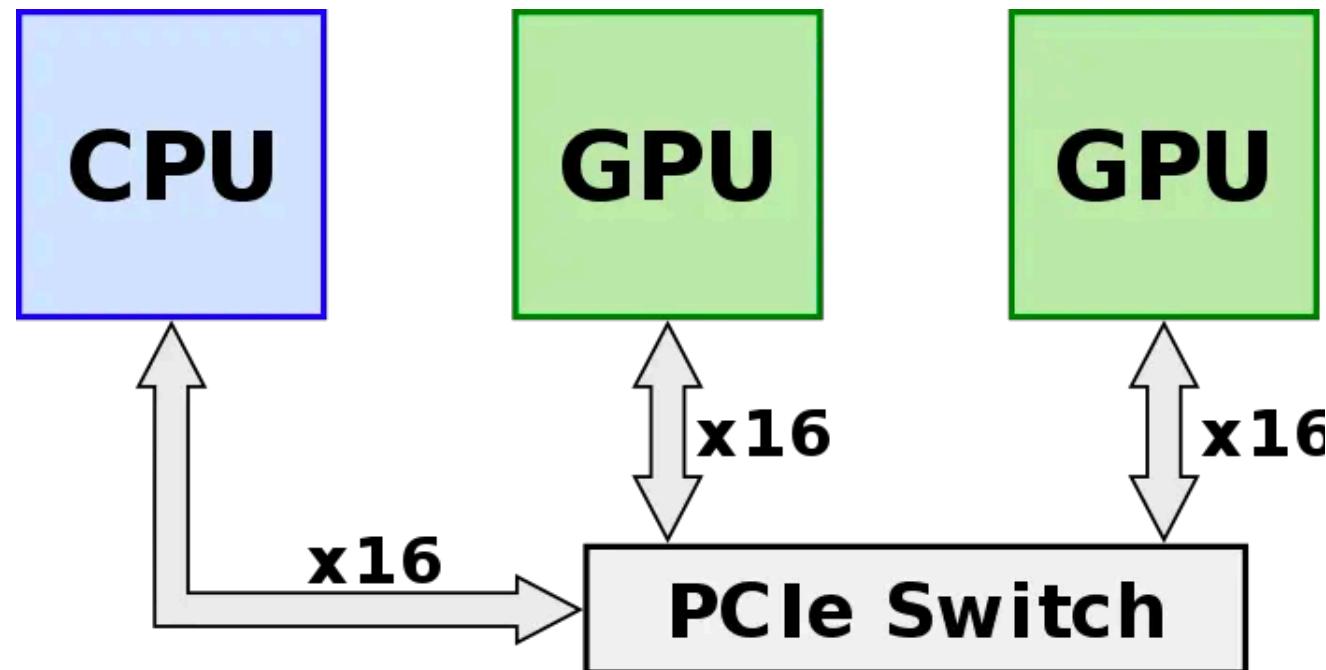
4



[www.hiascend.com](http://www.hiascend.com)

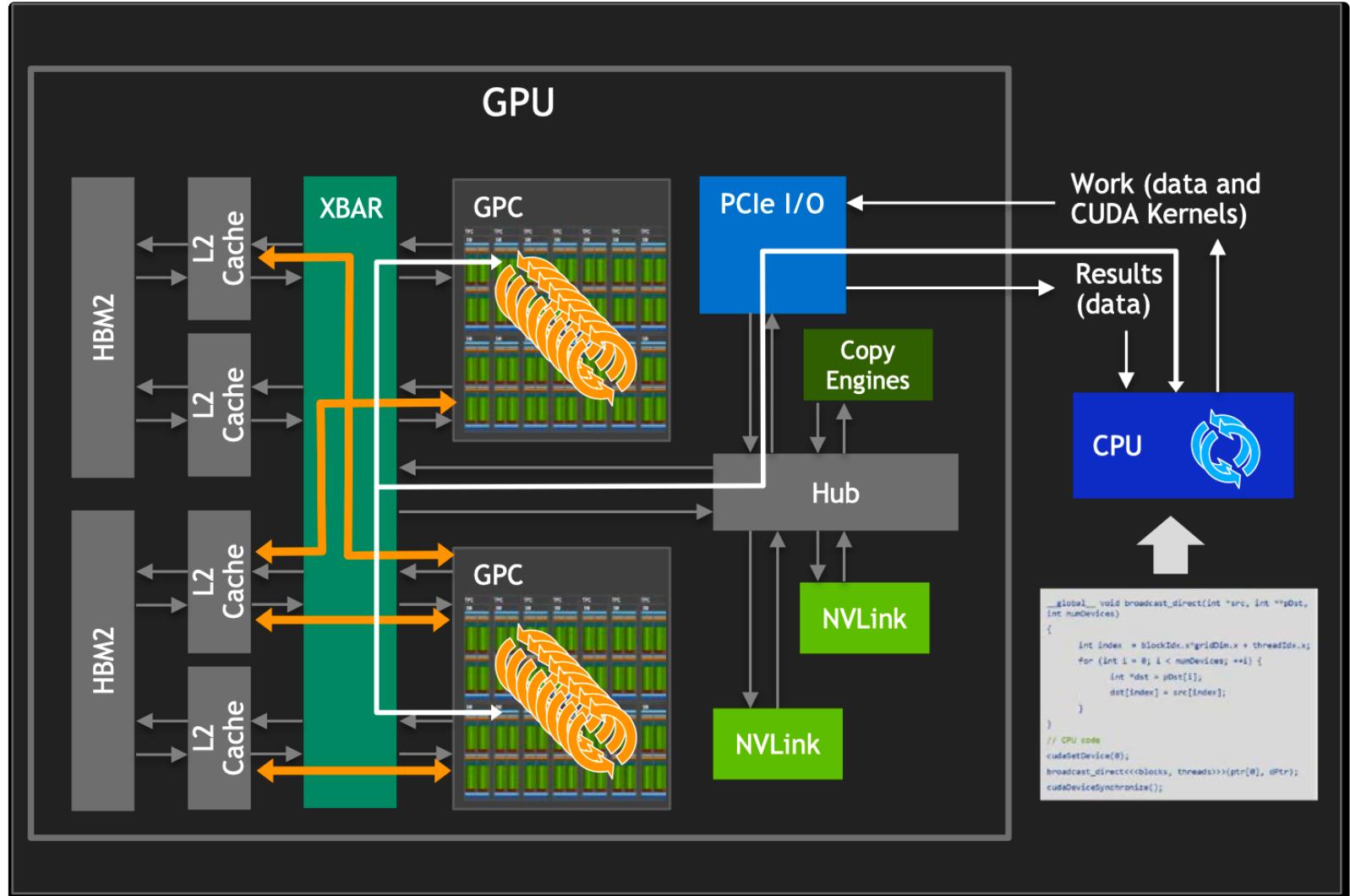
# PCIe

- 为了计算节点互联，多个 GPU 通过 PCIe Switch 直接与 CPU 相连。PCIe 3.0  $\times$  16 ~32GB/s 双向带宽，但是当训练数据不停增长的时候，互联方案称为系统瓶颈。如果不改进这个互联带宽，那么新时代GPU带来的额外性能就没法发挥出来，从而无法满足现实需求负载的增长。



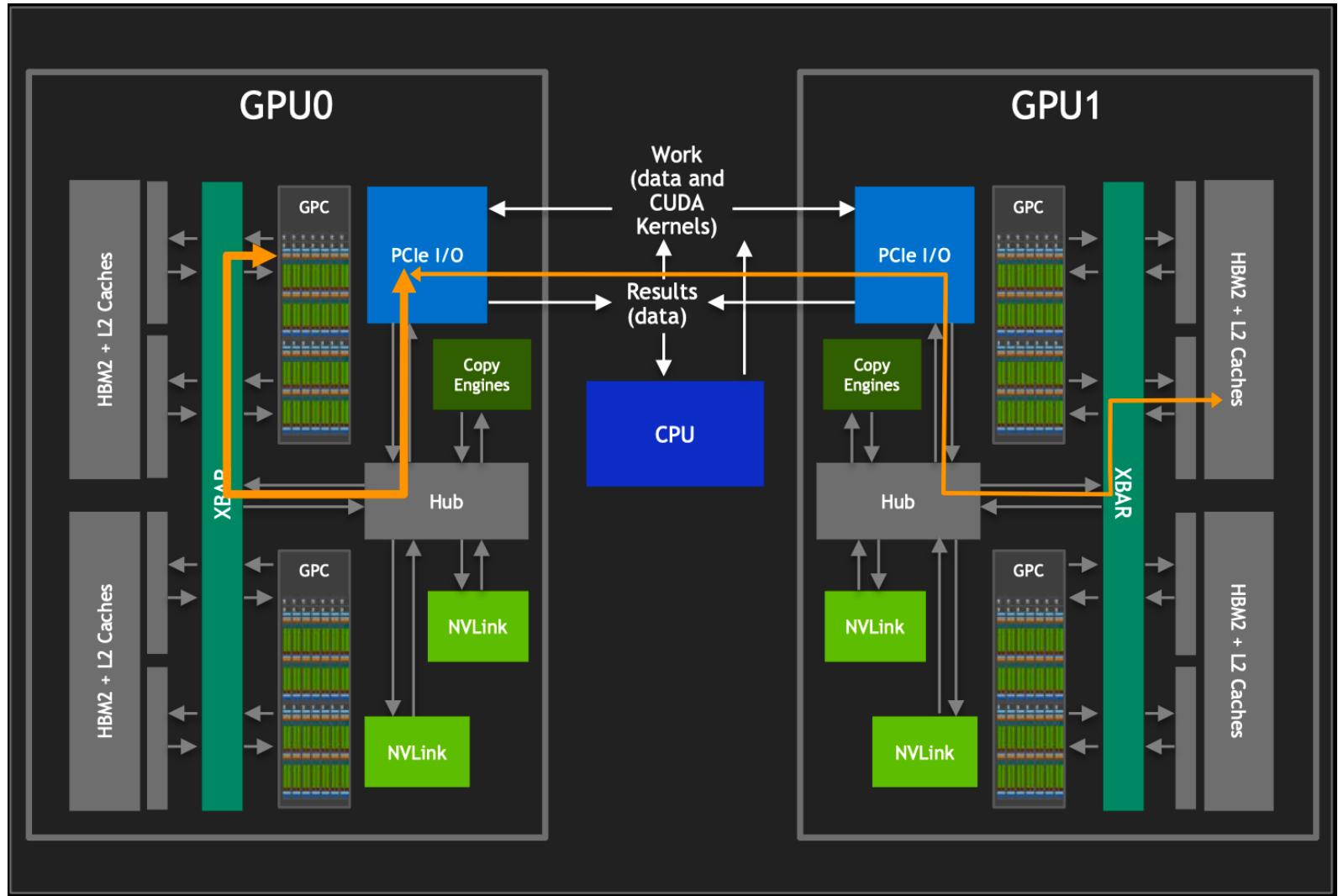
# 单 GPU 里面多个 SM 核心

- 使用 CUDA 来驱动硬件并行执行真正的计算；
- GPU 把线程工作分配给每个 GPC/SM cores；
- GPC/SM cores 利用 HBM2 中的数据进行计算；
- GPC/SM cores 之间可以共享 HBM2 中的数据；



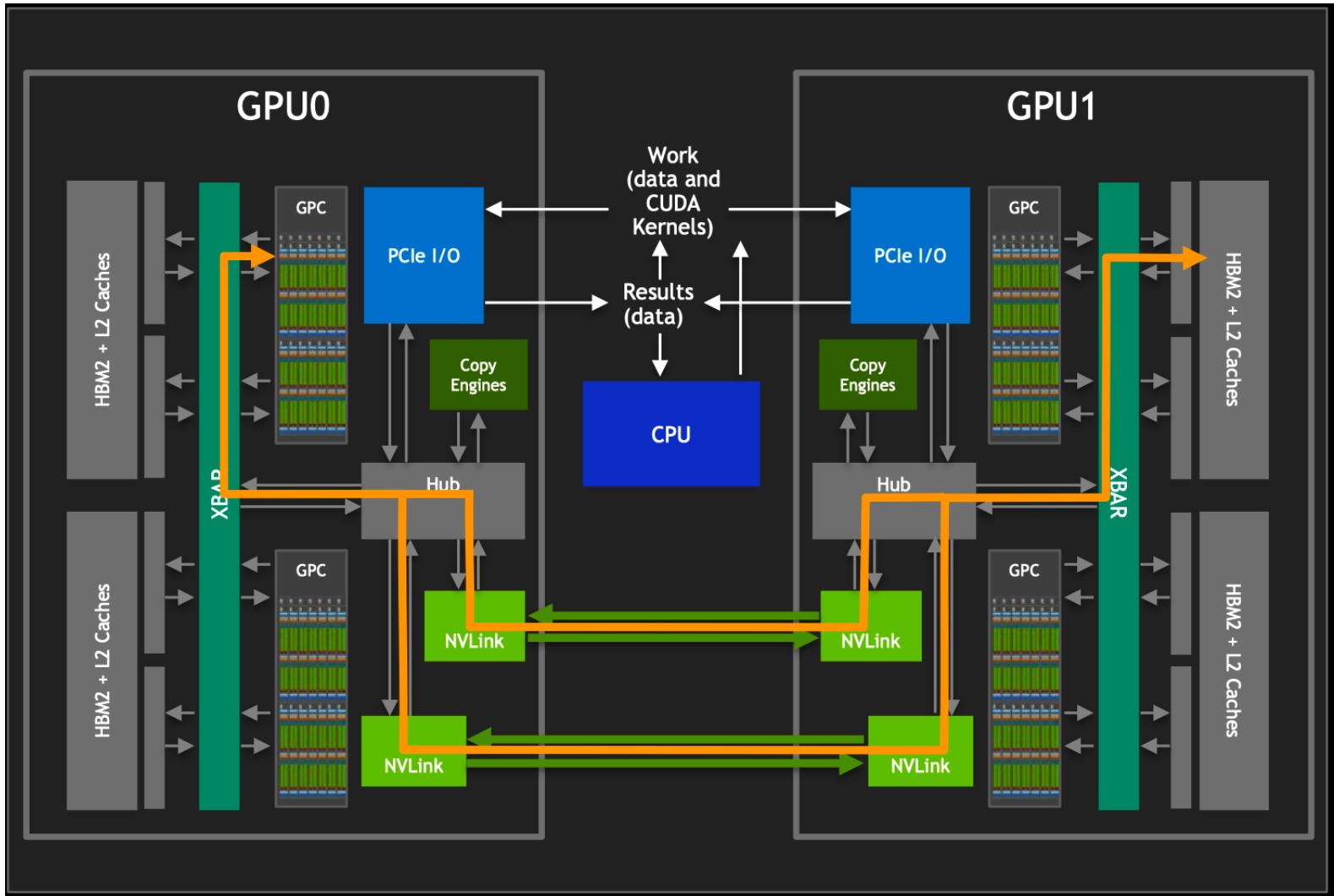
# GPU 间通过 PCIe 通信

- 如果要对其他 GPU 的 HBM2 进行访问，需要走 PCIe；
- GPU-to-GPU 之间的交互需要通过 CPU 进行分配调度；
- PCIe 的带宽限制了 GPU-to-GPU 的速率；



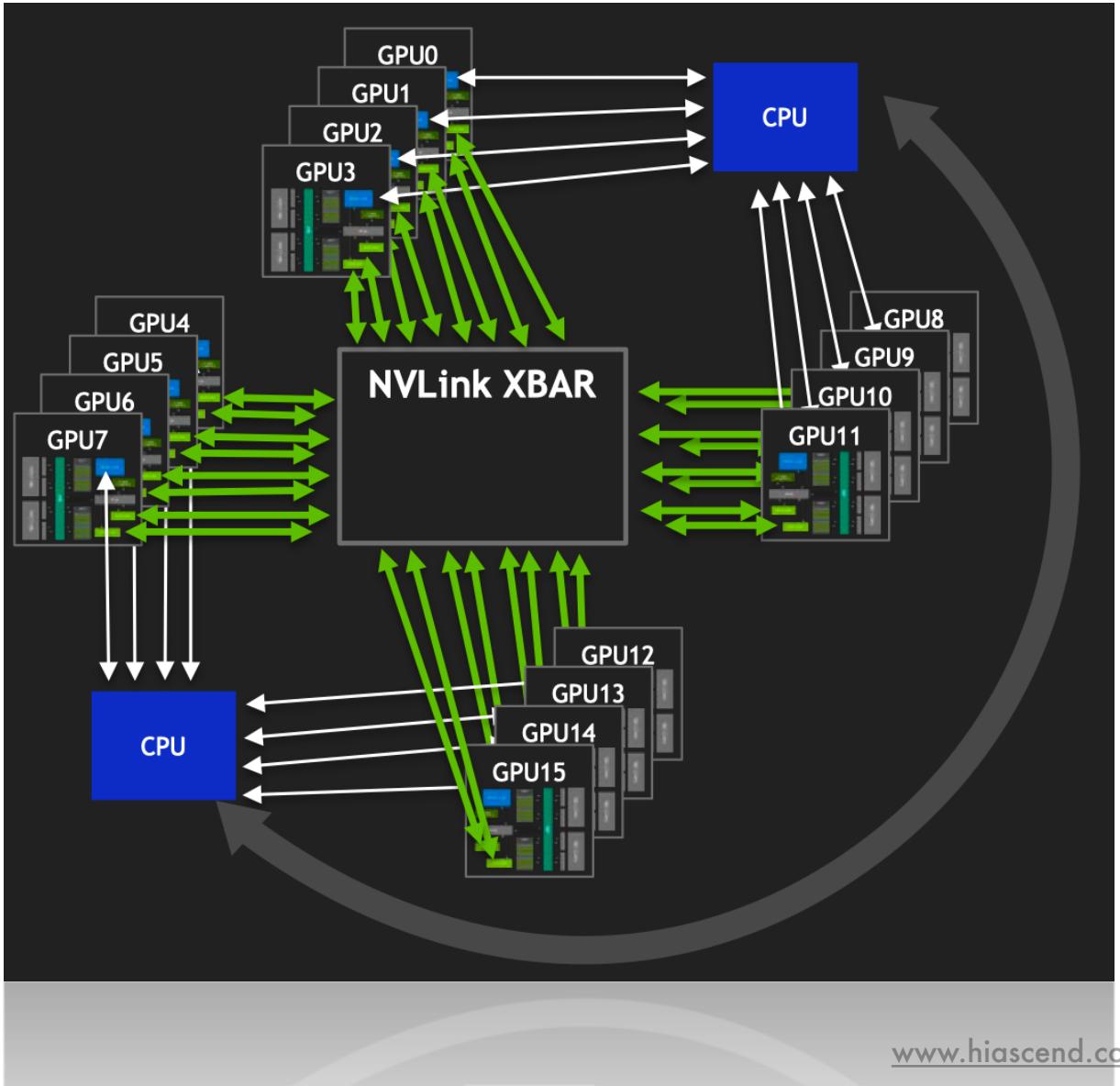
# GPU 间使用 NVLink

- GPCs 可以访问卡间 HBM2 内存数据；
- 通过多条 NVLink 来对其他 GPU 内的 HBM2 数据进行访问；
- 成为了 XBARs 的桥梁，并且与 PCIe 不冲突，作为互补方案；



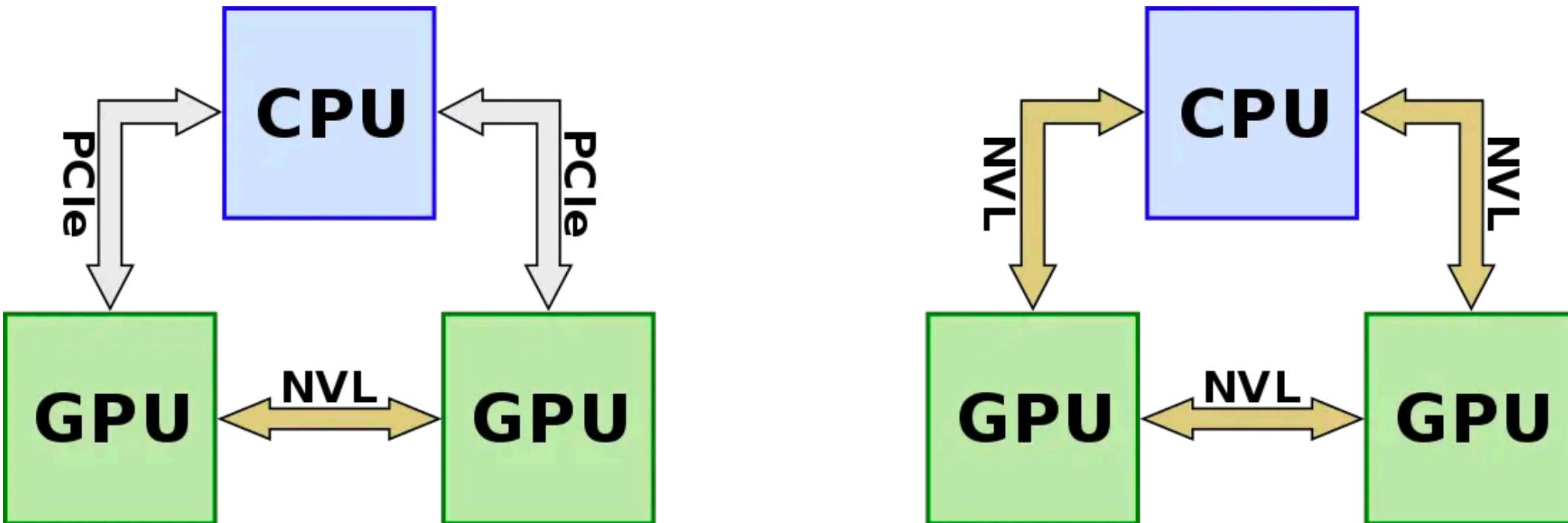
# GPU 间互联

- NVLink 可以使得更多的GPU间进行互联；
- 实现单个 GPU 驱动进程可以控制所有 GPU的计算任务；
- HBM2 可以在不受其他进程干扰下下访问（LD/ST指令、 RDMA）；
- XBAR 作为桥接器可以独立演进发展，提供更高的带宽；

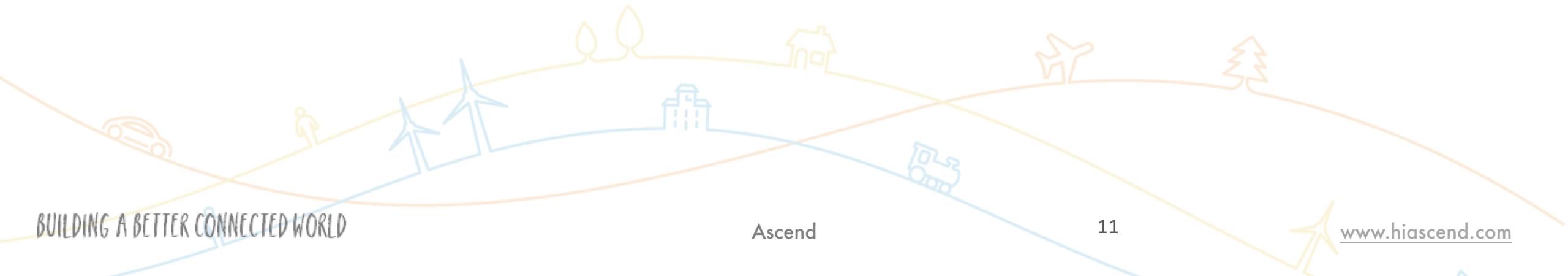


# PCIe

- 为了计算节点互联，多个 GPU 通过 PCIe Switch 直接与 CPU 相连。PCIe 3.0 × 16 ~32GB/s 双向带宽，但是当训练数据不停增长的时候，互联方案称为系统瓶颈。如果不改进这个互联带宽，那么新时代GPU带来的额外性能就没法发挥出来，从而无法满足现实需求负载的增长。



# NVLink 结构



BUILDING A BETTER CONNECTED WORLD

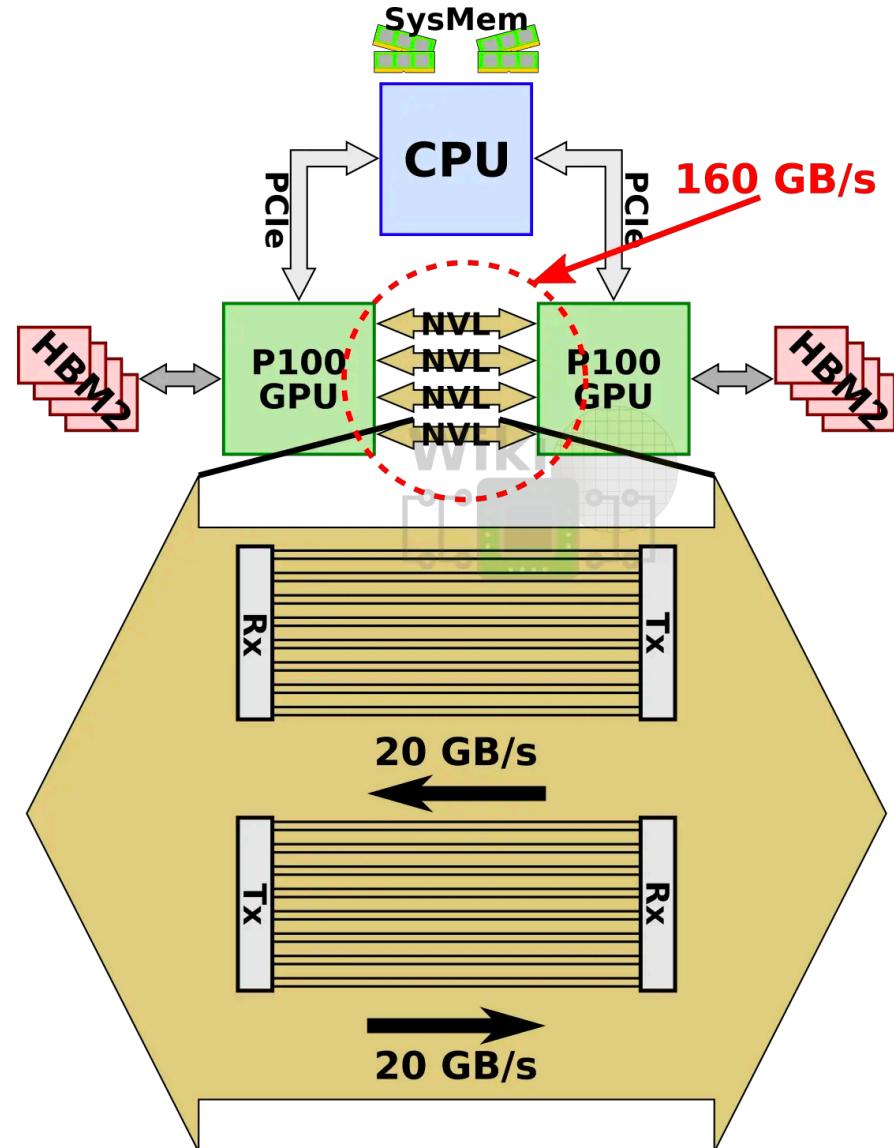
Ascend

11

 [www.hiascend.com](http://www.hiascend.com)

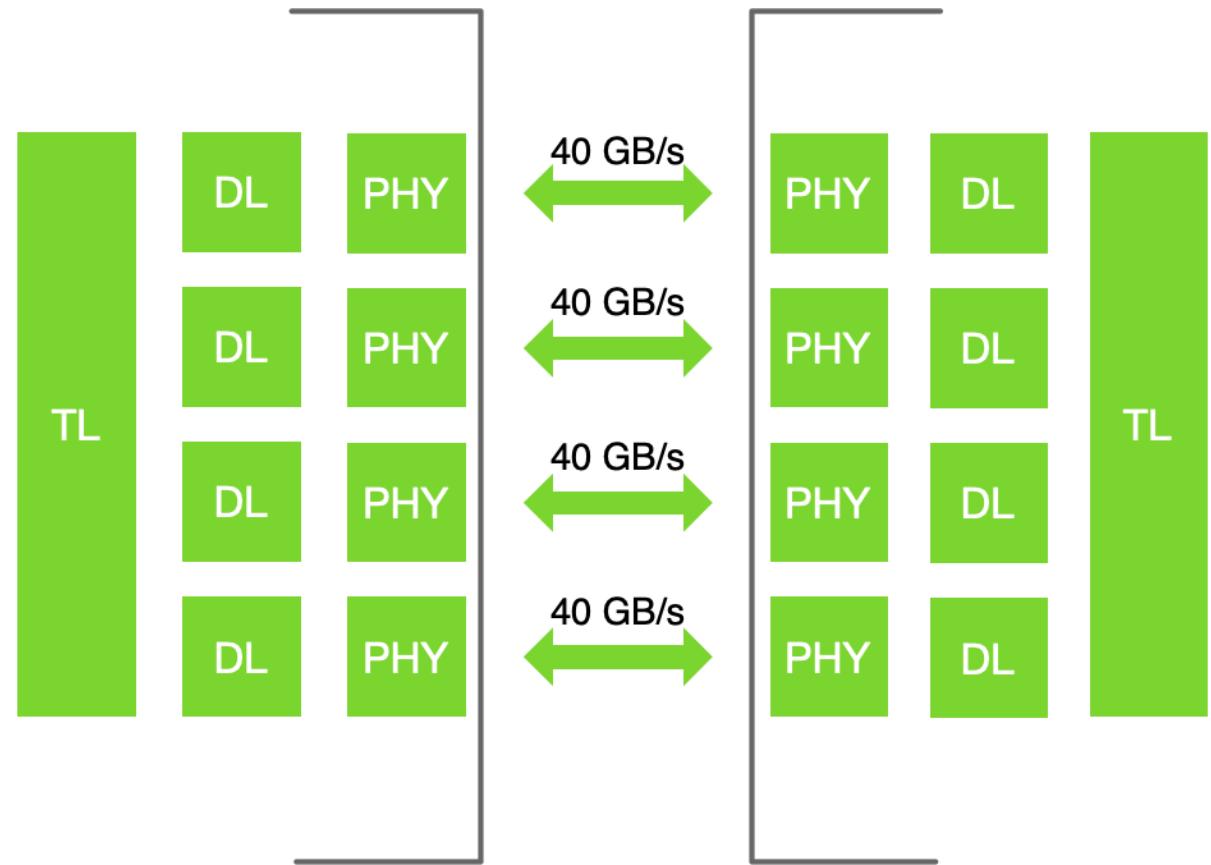
# 第一代 NVLink P100 DEMO

- 单条 NVLink 是一种双工双路信道，其通过组合 32 条配线，从而在每个方向上可以产生 8 对不同的配对 (  $2bi \times 8pair \times 2wire = 32wire$  )
- P100 上，集成了 4 条 nvlink。每条 link 具备双路共 40GB/s 的带宽，整个芯片具备整整 160GB/s 的带宽。



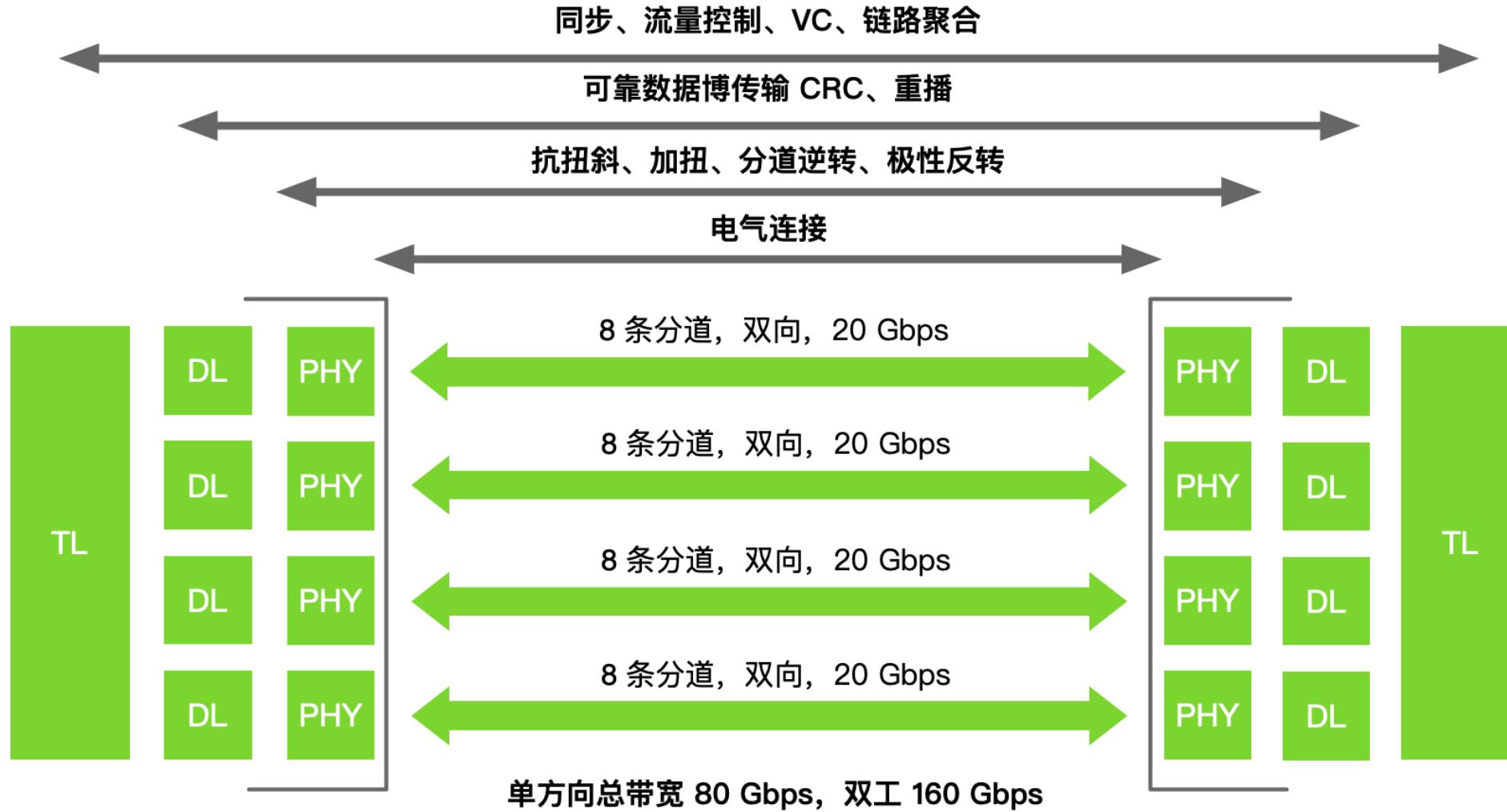
# NVLink 连接

- P100 supports 4 NVLinks
- Up to 94% bandwidth efficiency
- Supports read/writes/atomics to peer GPU
- Supports read/write to NVLink-enabled CPU
- Links can be ganged for higher bandwidth



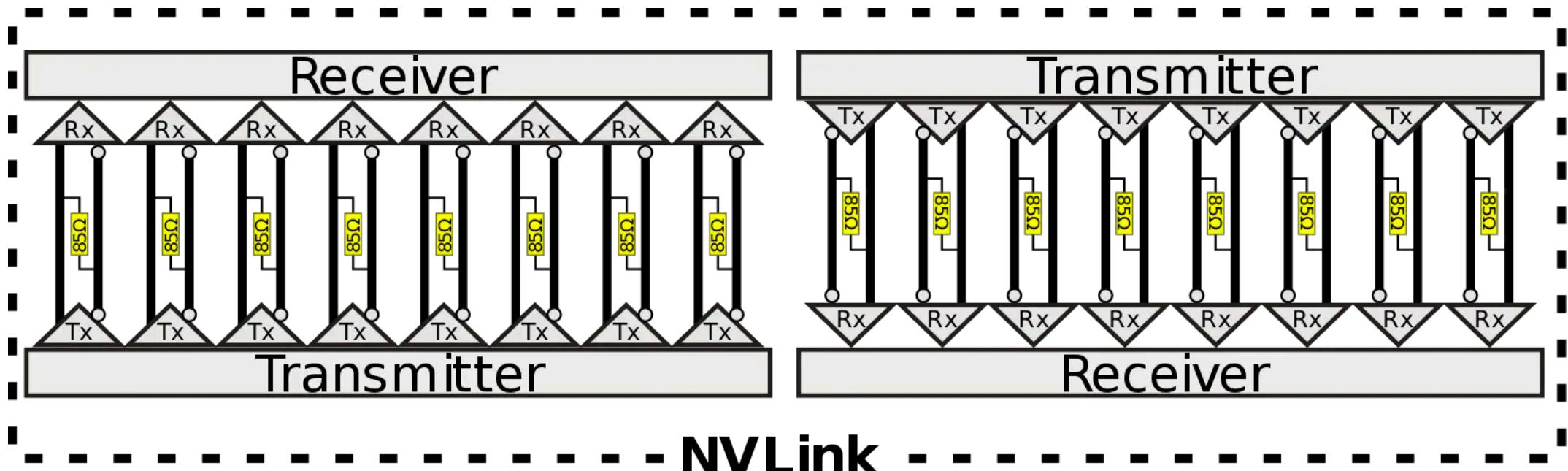
# NVLink 连接

物理层 (PHY)、数据链路层 (DL)、交易层 (TL)



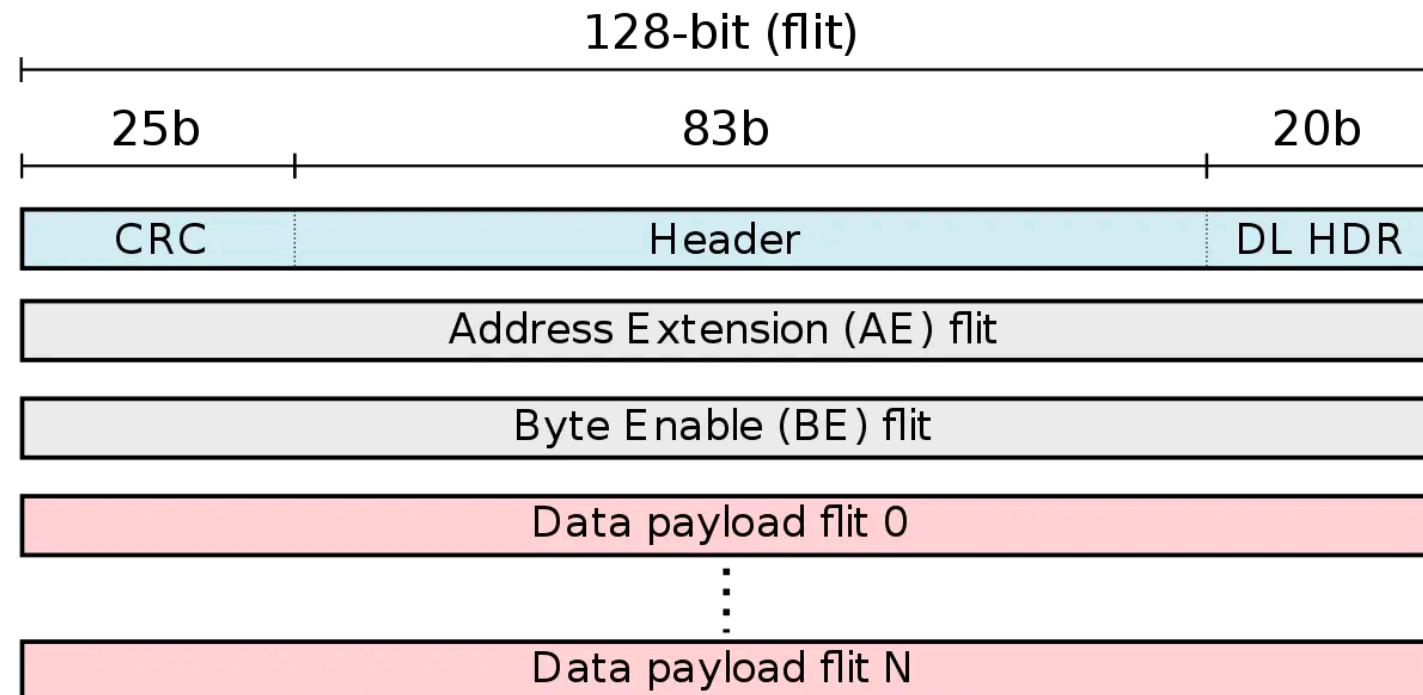
# NVLink 连接

- NVLink 通道称为Brick。单个 NVLink 是一个双向接口，每个方向包含 8 个差分对，总共 32 条线。这些电气连接线使用直流耦合 DC coupled，带 85 欧姆的差分终端。为了简化路由，NVLink 支持 **通道反转** 和 **通道极性**：设备间的物理通道顺序和极性可能会颠倒。



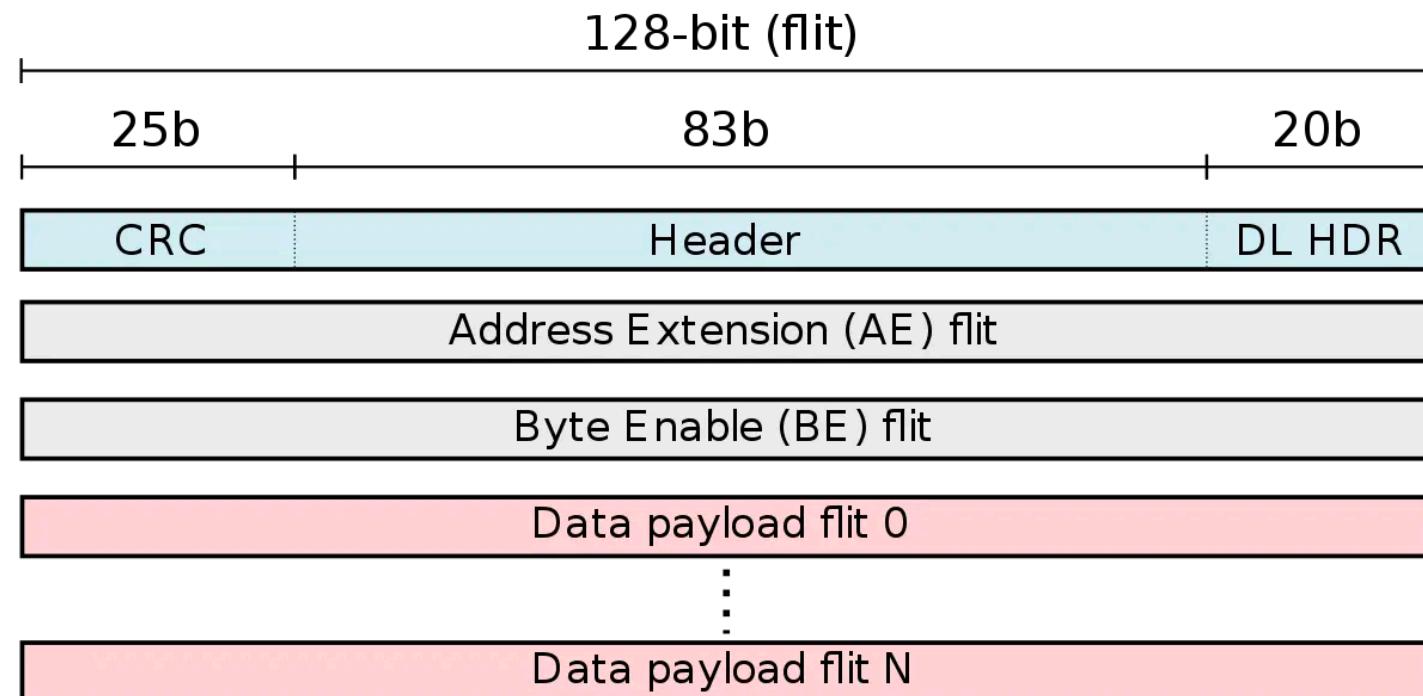
# NVLink 数据包

- 单 NVLink 数据包范围从 1~18 flit。每个 flit 128 bit , 1 header flit + 16 Data payload fit 单向可实现 256 bytes 传输，峰值带宽利用率为 94.12% ; 1 header flit + 4 Data payload fit 可以实现单向 64 bytes 传输，利用率到 80% ; 在双向传输分别下降到 88.9% 和 66.7%。



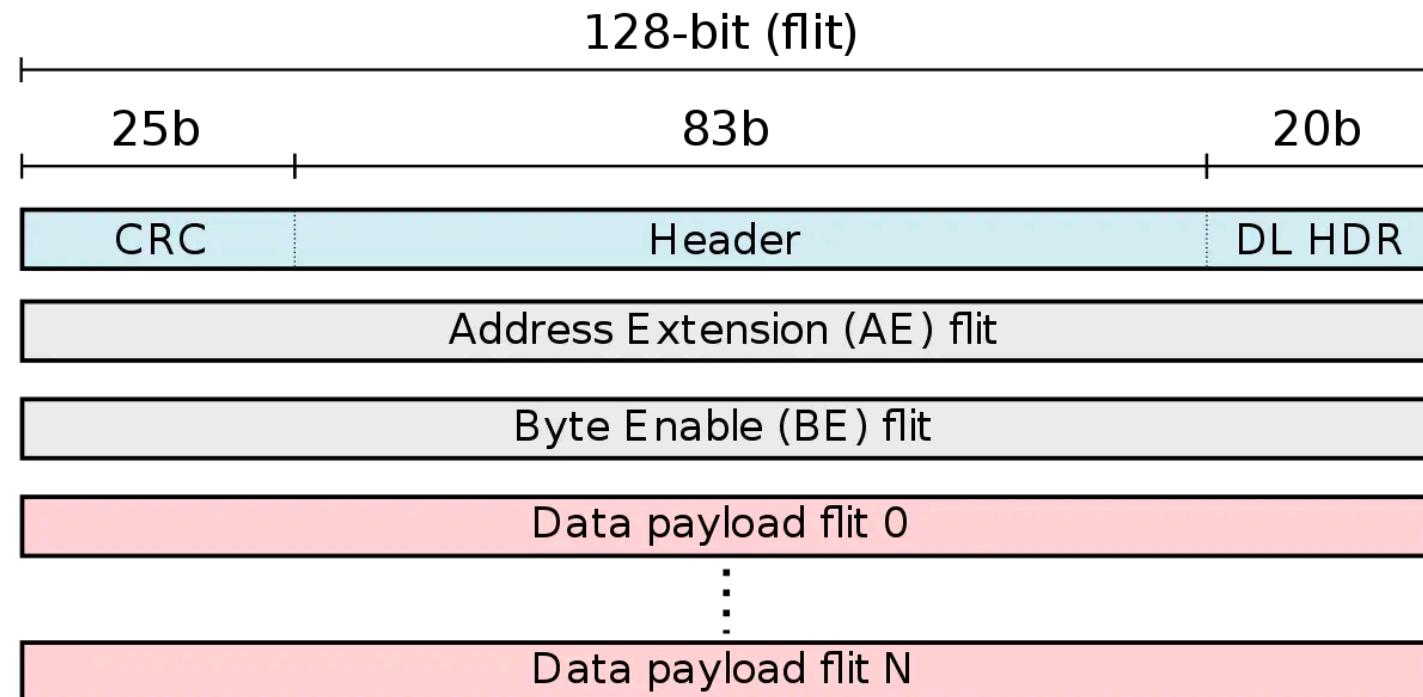
# NVLink 数据包

- Header flit 128-bit , 包括一个 25-bit CRC、 83-bit 传输字段 ( Transaction field ) 、 20-bit 数据链路 ( Data link ) 。
- **传输字段 HD** : 请求类型 request type、 地址 address 、 流量控制位 flow control bits 和标记标识符 tag identifier。
- **数据链路 DL** : 数据包长 packet length、 应用编号标签 application number tag 和确认标识符 acknowledge identifier。
- **地址扩展 AE** : 静态位保留 , 只传输位变化。

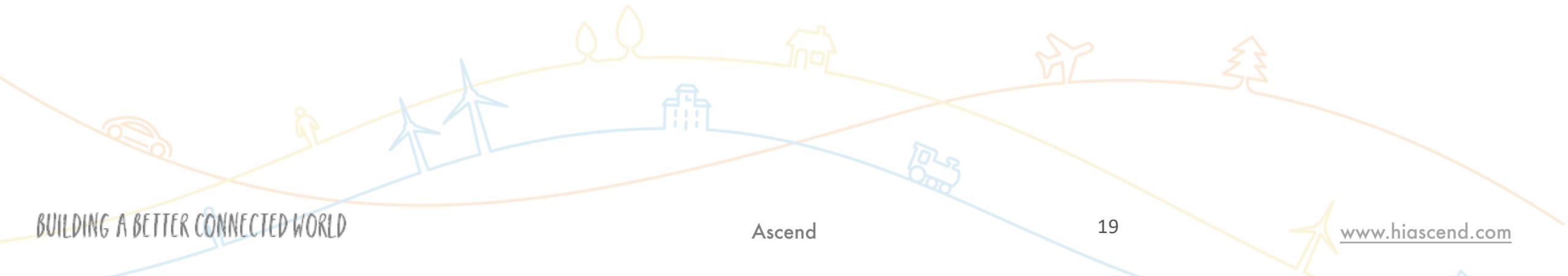


# NVLink 数据包

- 流程**：错误检测通过 25-bit 循环冗余校验报头字段完成。 Receiver 负责将数据保存在 Replay buffer，对传输数据包排序，并且在正确的 CRC 后将数据发送回源。
- 组成**：CRC 字段 25-bit，对于最大数据包最多允许 5 个随机位错误，或者对于差分对突发，支持最多 25 个连续位错误。CRC 实际上是有效载荷上计算，因此不需要为数据有效载荷单独 CRC 字段。



# NVLink 拓扑



BUILDING A BETTER CONNECTED WORLD

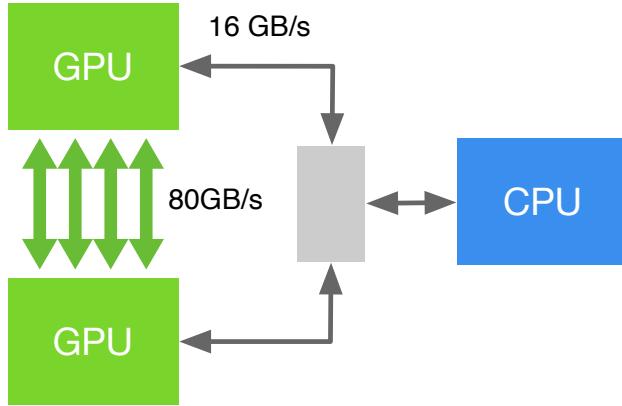
Ascend

19

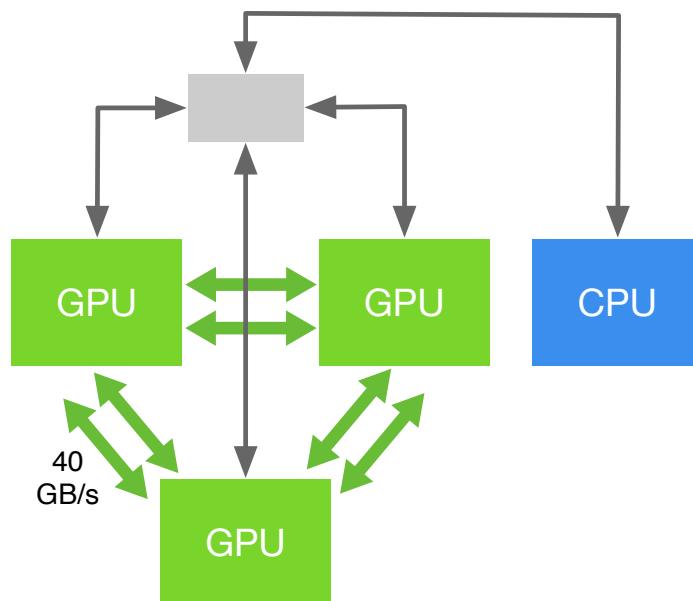
 [www.hiascend.com](http://www.hiascend.com)

# 第一代 NVLink P100 DEMO

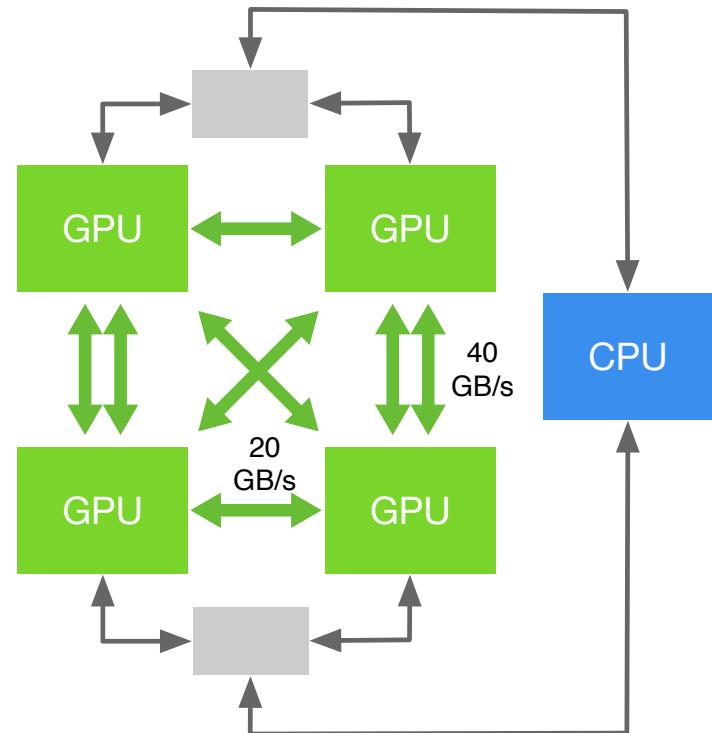
2 GPU per Nodes



3 GPU per Nodes



4 GPU per Nodes



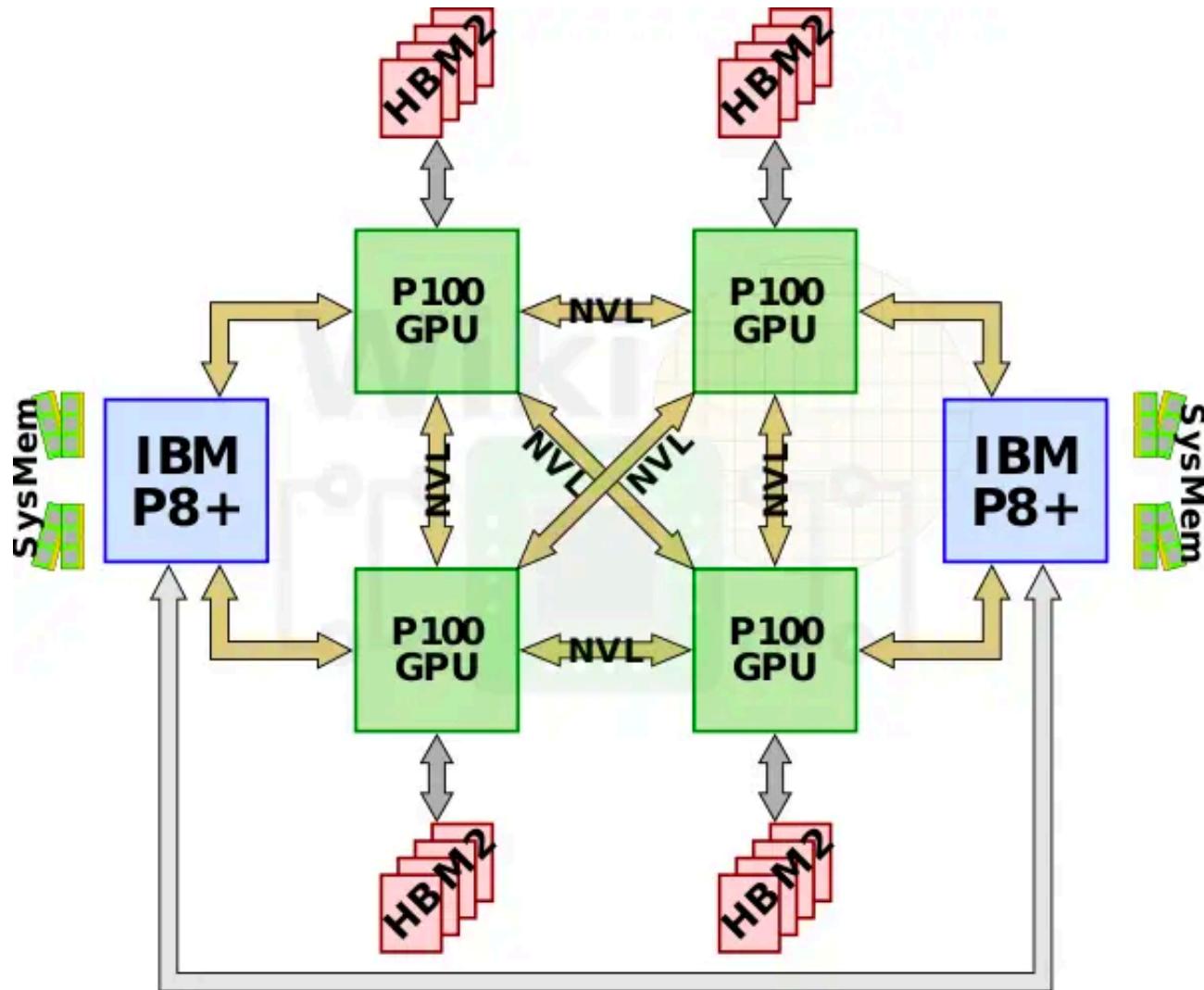
20 GB/s  
↔ NVLink

16 GB/s  
↔ PCIe Gen 3 x 16

PCIe Switch

# 第一代 NVLink P100 DEMO

- IBM 将 NVLink 1.0 添加到他们基于 Power8+ 微架构的 Power 处理器上，这一举措使得 P100 可以直接通过 NVLink 于 CPU 相连，而无需通过 PCIe。通过与最近的 power8+ CPU 相连，4GPU 的节点可以配置成一种全连接的 mesh 结构。

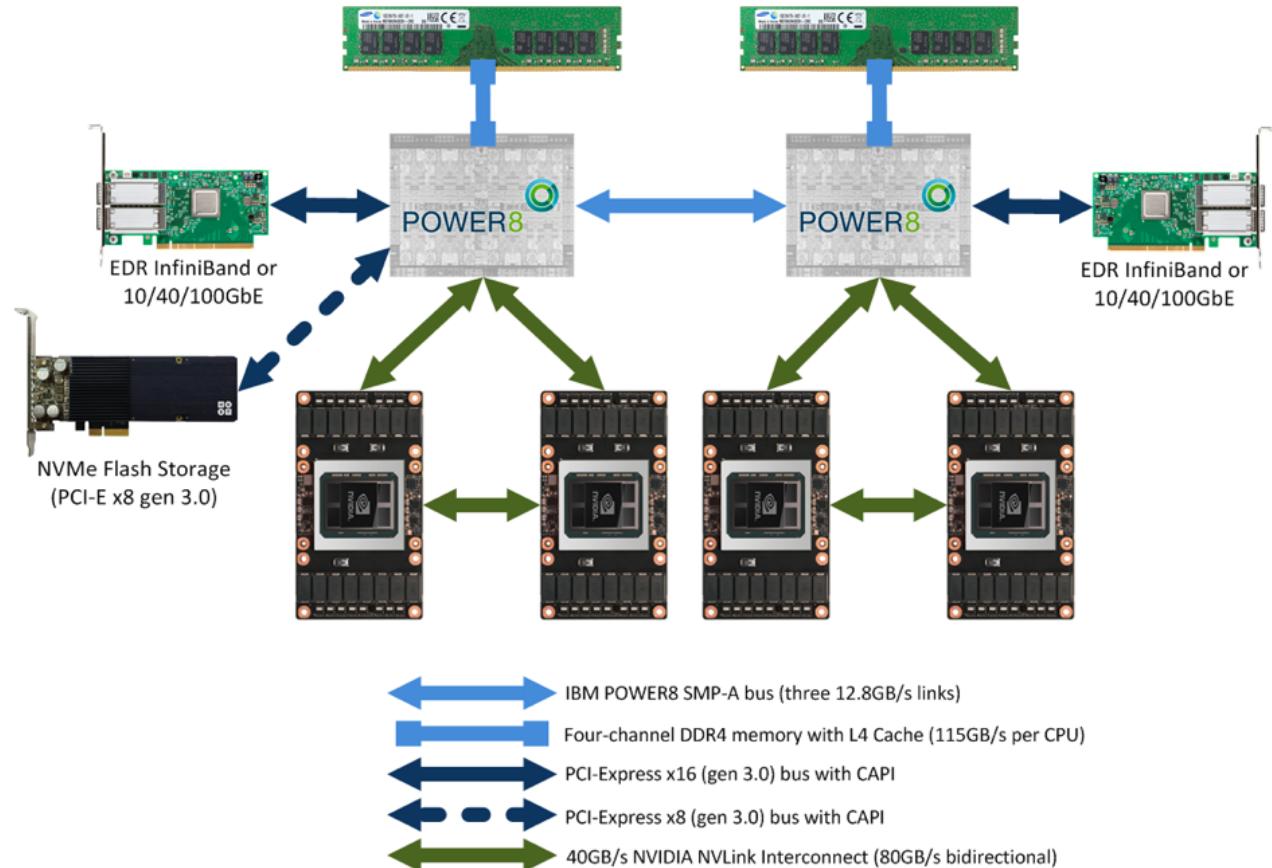


# 第一代 NVLink P100 DEMO

- IBM 将 NVLink 1.0 添加到他们基于 Power8+ 微架构的 Power 处理器上，这一举措使得 P100 可以直接通过 NVLink 于 CPU 相连，而无需通过 PCIe。通过与最近的 power8+ CPU 相连，4GPU 的节点可以配置成一种全连接的 mesh 结构。

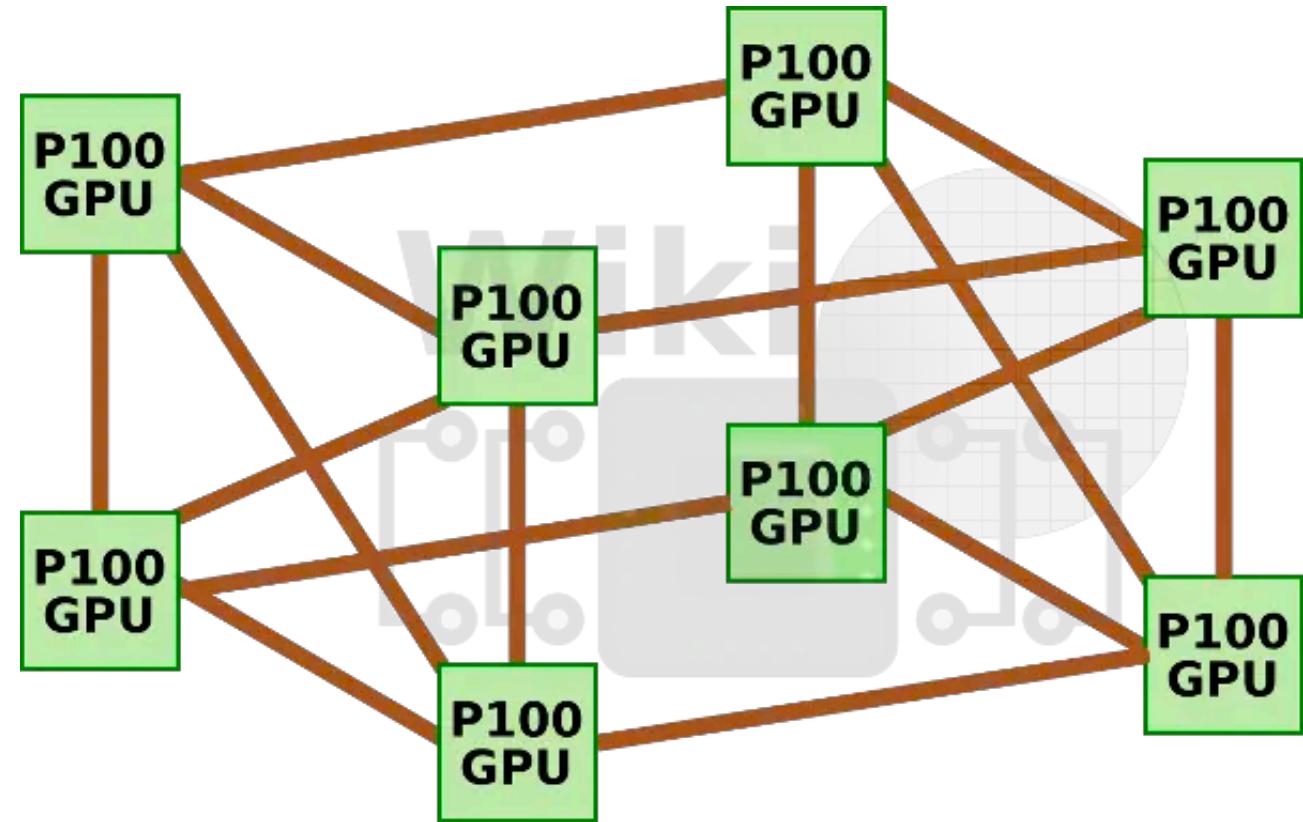
## Server Block Diagram

Microway OpenPOWER Server with NVIDIA Tesla P100 NVLink GPUs



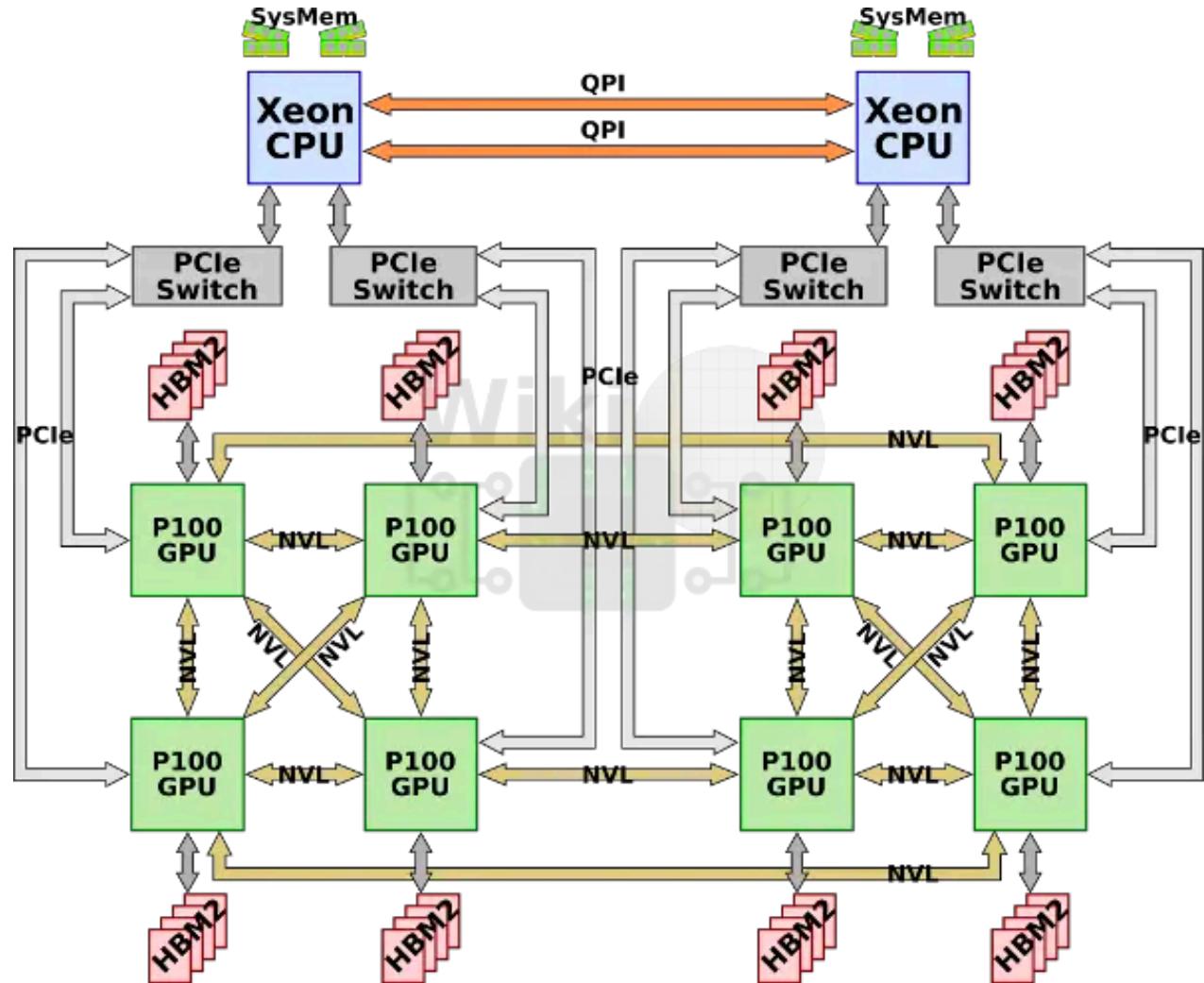
# 第一代 NVLink P100 DEMO

- dgx1：集成了八块P100与两块志强e5 2698v4，但是因为每块GPU只有4路NVLink，GPU构成了一种混合的cub e-mesh 网络拓扑结构，GPU被4块4块分为两组，然后在互相连接。



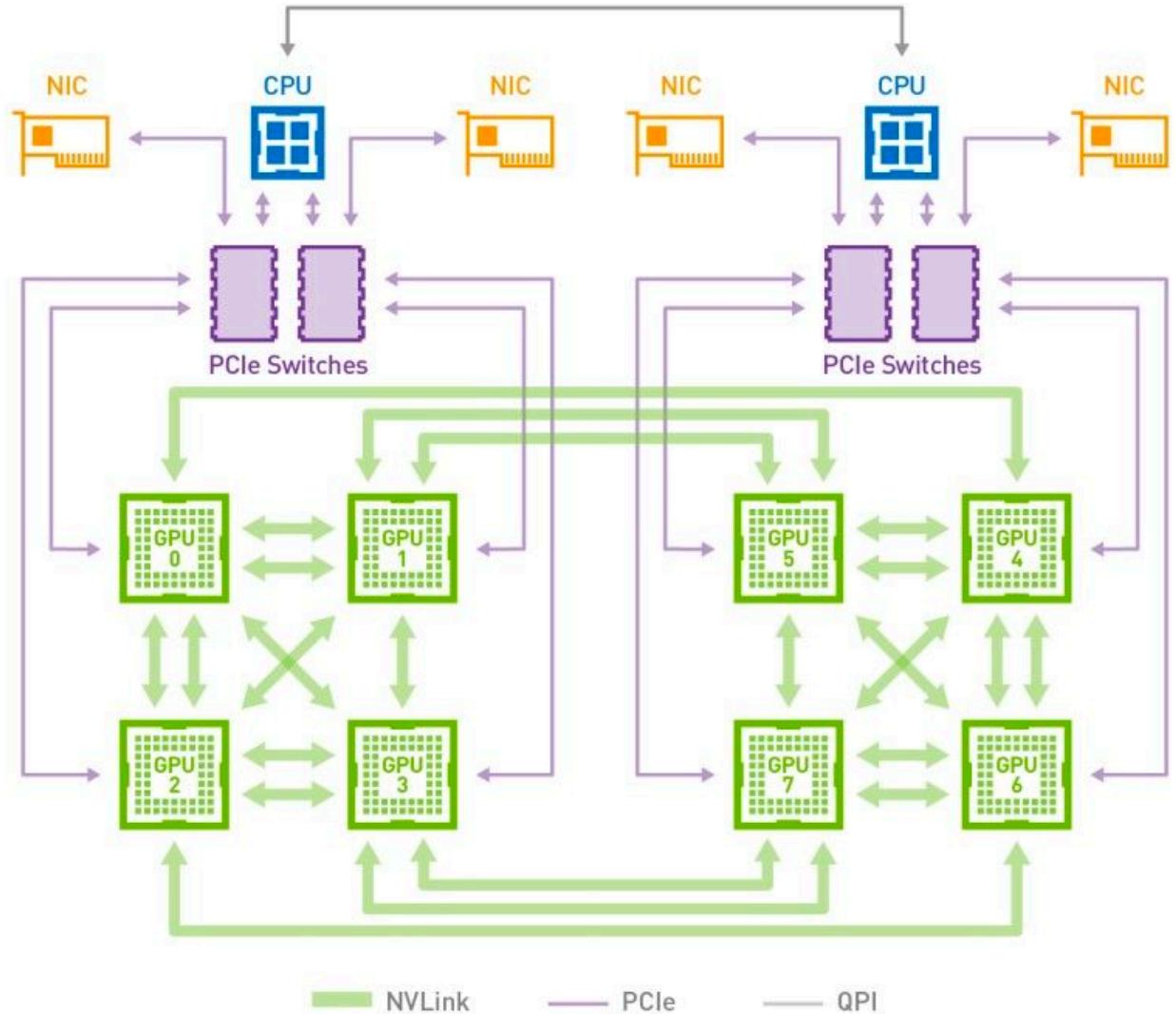
# 第一代 NVLink P100 DEMO

- 因为 GPU 需要的 PCIe 通道数量超过了芯片组所能提供的数量，所以每一对 GPU 将连接到一组 PCIe switch 上与志强相连，然后两块 Intel 再通过 QPI 总线连接

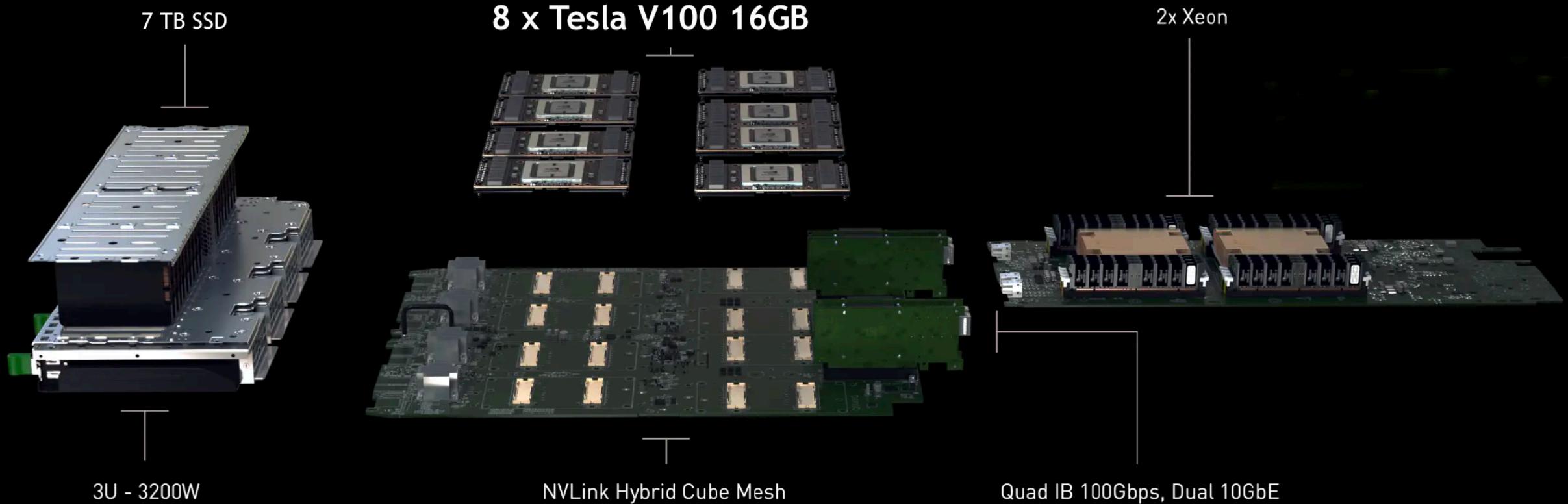


# 第一代 NVLink P100 DEMO

- 因为 GPU 需要的 PCIe 通道数量超过了芯片组所能提供的数量，所以每一对 GPU 将连接到一组 PCIe switch 上与志强相连，然后两块 Intel 再通过 QPI 总线连接



# Tesla V100 DGX-1 硬件渲染图



# Reference 引用&参考

1. <https://fuse.wikichip.org/news/1224/a-look-at-nvidias-nvlink-interconnect-and-the-nvswitch/>
2. <https://blog.csdn.net/BtB5e6Ns1g511Eg5XEg/article/details/86762135>
3. <https://developer.nvidia.com/blog/upgrading-multi-gpu-interconnectivity-with-the-third-generation-nvidia-nvswitch/>
4. <https://www.nextplatform.com/2016/05/04/nvlink-takes-gpu-acceleration-next-level/>
5. <https://www.servethehome.com/nvidia-nvlink4-nvswitch-at-hot-chips-34/>
6. <https://www.servethehome.com/nvidia-nvswitch-details-at-hot-chips-30/>  
<https://hc34.hotchips.org/>
7. [https://www.infoq.cn/article/3d4msrvs8zotgcj7\\*krt](https://www.infoq.cn/article/3d4msrvs8zotgcj7*krt)
8. <https://zhuanlan.zhihu.com/p/399405214>
9. <https://www.zhihu.com/question/63219175>
10. <https://developer.aliyun.com/article/591403>
11. <https://developer.aliyun.com/article/603617>
12. <https://developer.aliyun.com/article/599183>
13. [https://blog.csdn.net/tony\\_vip/article/details/117131380](https://blog.csdn.net/tony_vip/article/details/117131380)



# BUILDING A BETTER CONNECTED WORLD

# THANK YOU

Copyright©2014 Huawei Technologies Co., Ltd. All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.