

# 腾讯深度学习并行化实践

腾讯数据平台部高级工程师 金涛

## DTCC

### 2015中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2015

大数据技术探索和价值发现

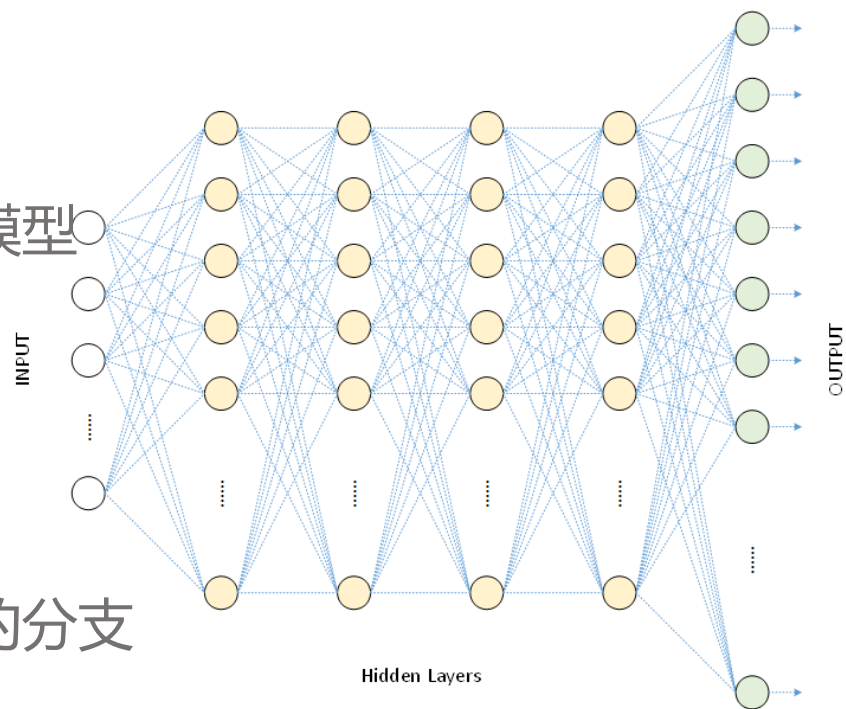


# 提纲

- 深度学习概述
- Mariana腾讯深度学习平台
  - Mariana DNN: DNN的GPU数据并行框架
  - Mariana CNN: CNN的GPU模型并行和数据并行框架
  - Mariana Cluster: DNN的CPU集群框架
- Mariana Cluster演进
- GPU Cluster的探索
- 深度学习并行化：系统和算法的双重视角

# 深度学习概况

- 深层网络是基于多层神经网络的复杂模型
  - 模型远复杂于当前的浅层模型
  - 是一定程度模拟人脑的模型
- 深度学习是近年机器学习中备受瞩目的分支
  - 端到端的学习
  - 在语音识别、图像识别等持续取得突破
  - E.g. ImageNet 1000类图像分类问题：准确率72%→85%→89%→93%
  - 各公司持续发力，Google，Microsoft，Facebook，百度，腾讯，阿里
- 深度学习的发展机遇
  - 海量的数据
  - 高速增长的计算能力
  - 算法改进



# 深度学习平台的挑战

- 大数据

- 大数据时代，可获取的数据量大大增加
- 可达数十亿样本

- 大模型

- 在大数据的支持下，更深更宽的网络能获得更好的结果
- 模型复杂：可达数万神经元，几千万至上亿参数。
- 以图像识别为例，增加卷积层filter数量，加深模型都有改善
- 内存消耗大

- 大计算量，耗时

- 大模型需要大计算量
- 大数据需要大计算量

- 非凸优化问题，超参数多，需要反复多次实验

- 非凸模型，倚重技巧和经验
- 超参数敏感：模型结构、输入数据处理方式、权重初始化方案、激活函数选择、参数配置等

# Mariana：腾讯深度学习平台概述

- 目标

通过并行加速计算

通过模型拆分支持大模型

通过框架简化应用代码

- 三个框架

**Mariana DNN:** 深度神经网络的GPU数据并行框架

**Mariana CNN:** 深度卷积神经网络的GPU模型并行和数据并行框架

**Mariana Cluster:** 深度神经网络的CPU集群框架

- 主要应用

语音  
识别

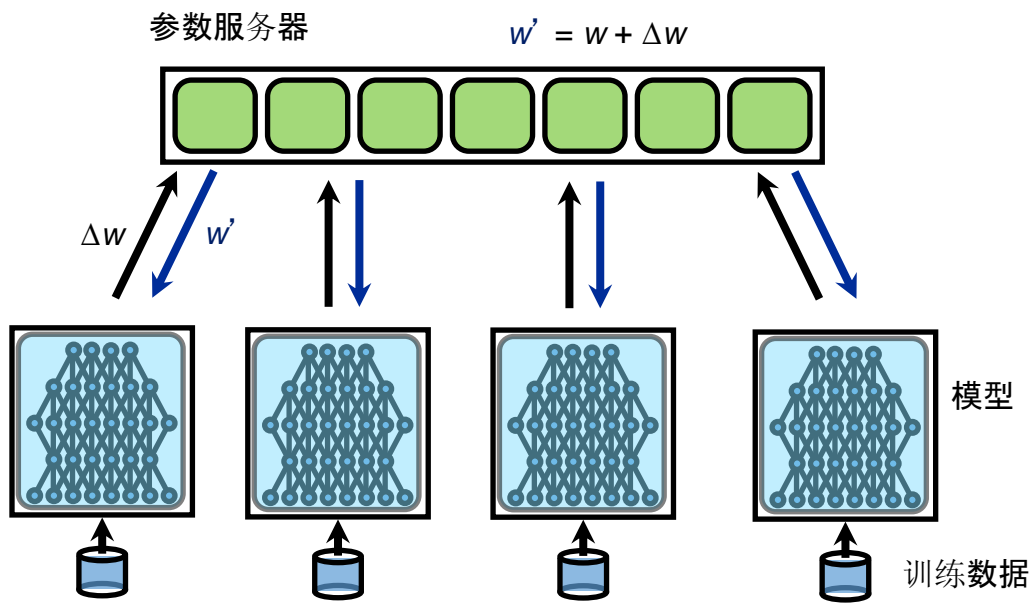
图像  
识别

广告  
推荐

# 深度学习模型训练的并行方法

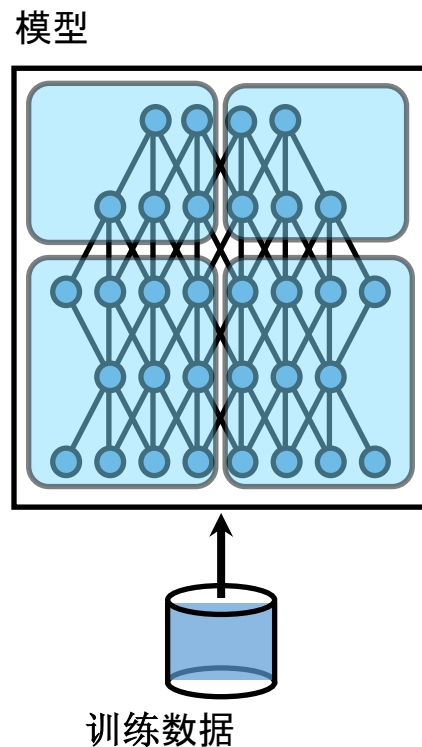
## ● 数据并行

- 划分训练数据
- 各Worker独自训练
- 交换参数



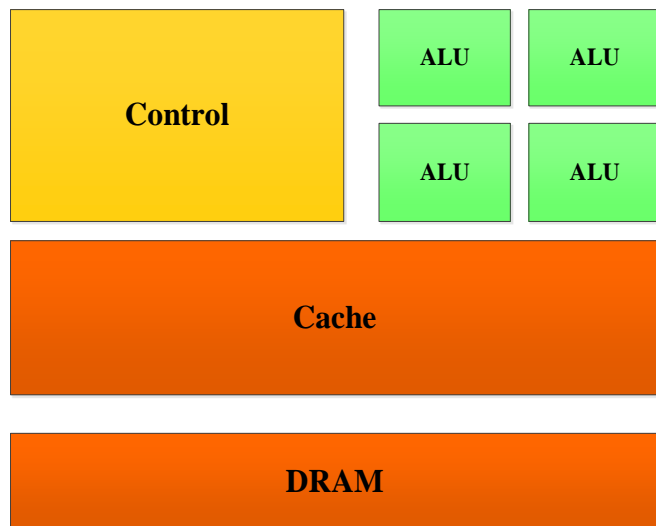
## ● 模型并行

- 模型拆分
- 多个Worker作为1组
- 同组Worker训练一个模型

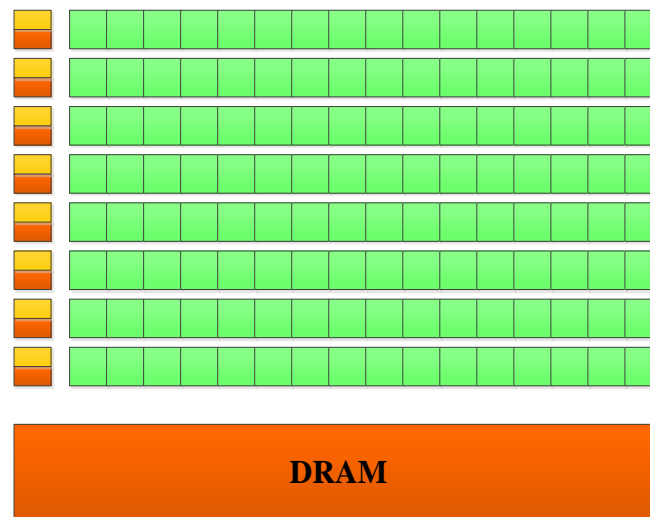


# GPU高性能计算

## ●体系结构：CPU vs GPU



- Multi-core CPU
- 巨大的缓存
- 复杂的控制逻辑
- 数十GB以上的内存



- Many-core GPU
- 数千流处理器，更多寄存器
- 轻量级线程共享控制逻辑
- 数GB的内存

## ● GPGPU (General Purpose GPU)

- 从图形处理（桌面级）到通用计算（专业级）



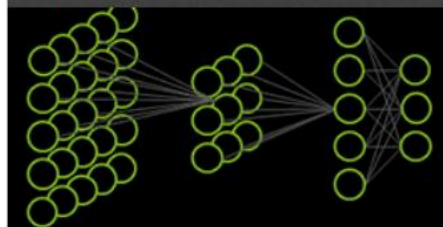


# GPU高性能计算应用领域

## ■ 适合GPU计算的应用类型

- 用同一计算方法对很大的数据进行并行计算
- 数据相关度低
- 计算密度大，计算所花的时间比数据存储的时间大得多

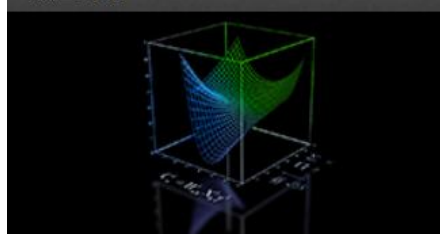
机器学习



数据科学



数值分析



成像与计算机视觉



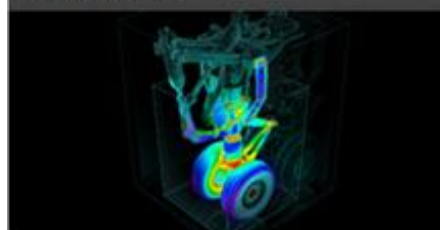
计算金融



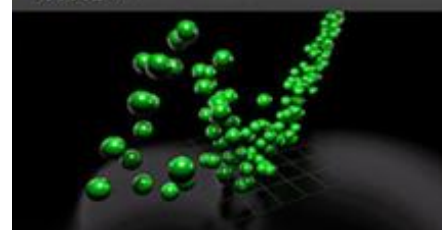
计算流体力学



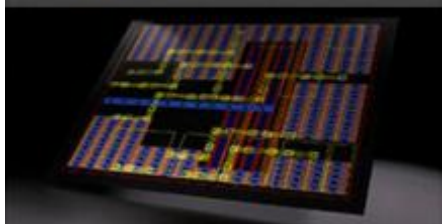
计算结构力学



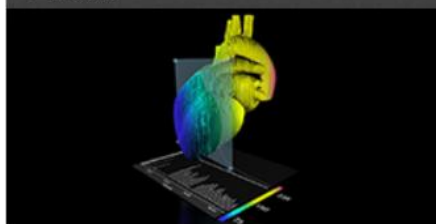
计算化学



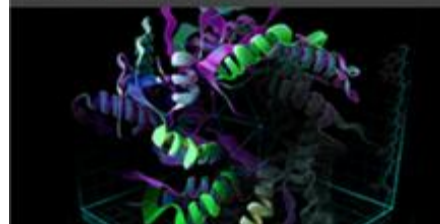
电子设计自动化



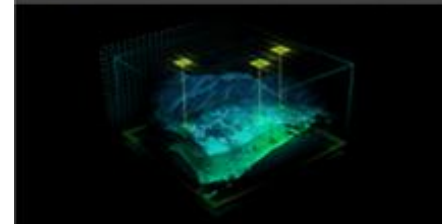
医学成像



生物信息学



国防



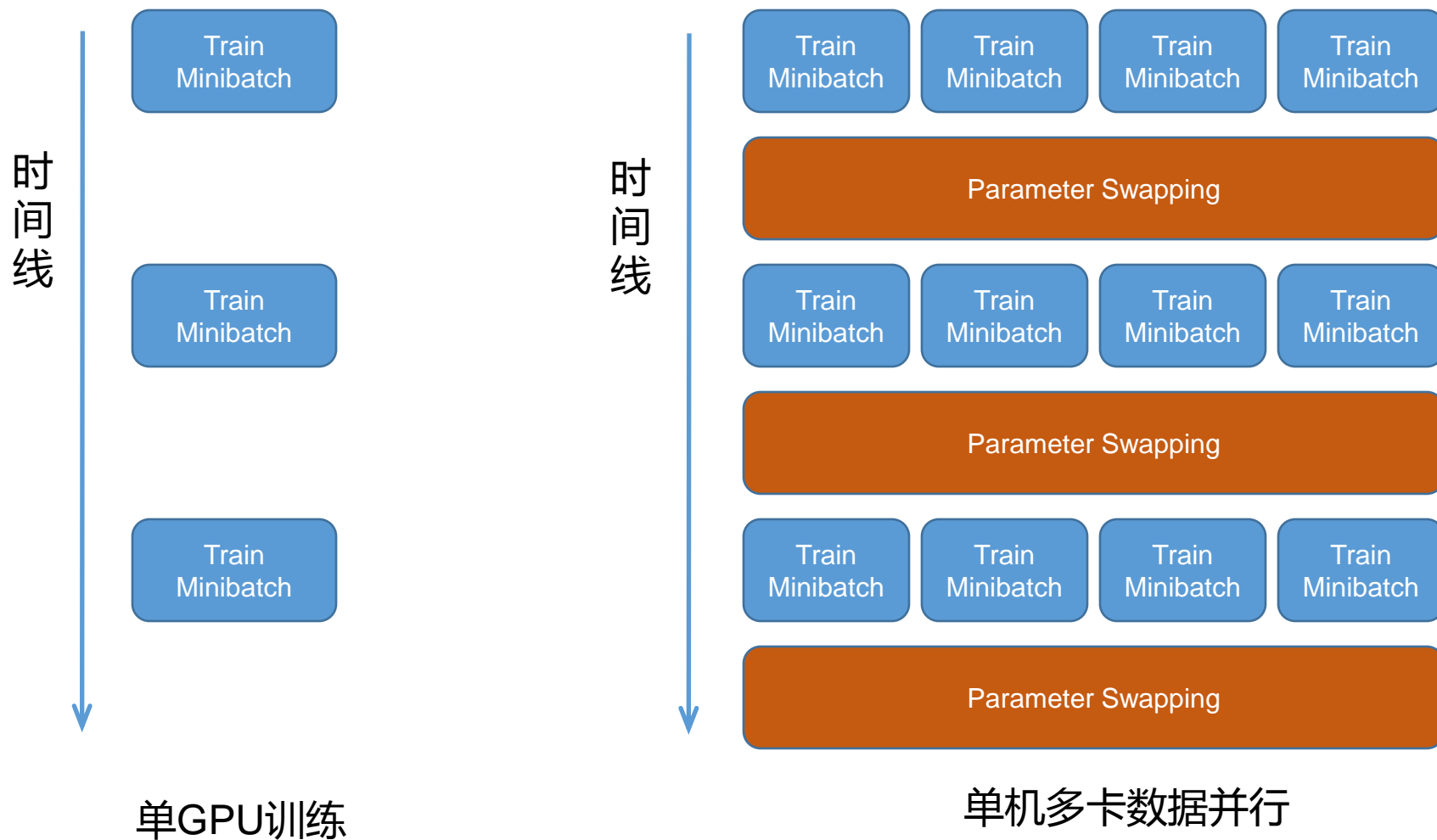


# Mariana的设计选择

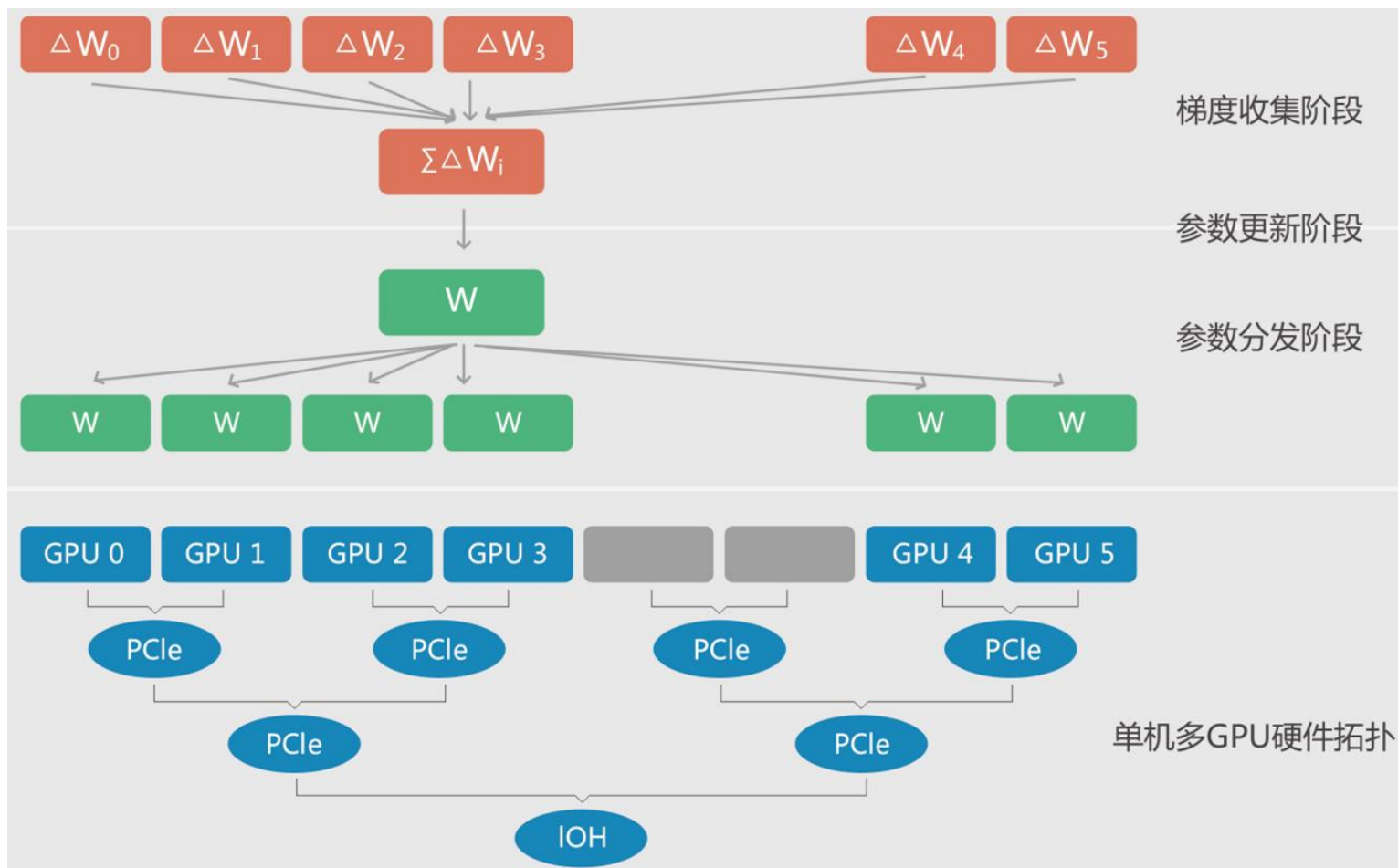
- CPU并行 vs GPU并行
  - CPU并行：适合稀疏连接模型
  - GPU并行：适合稠密连接模型
- 数据并行 vs 模型并行
  - 数据并行：适合参数交换相对较少的情况
  - 模型并行：适合输出值/残差交换相对较少的情况
- 同步SGD vs 异步SGD
  - Worker间参数更新方式：有同步点 vs 独立地进行
  - Worker的规模和计算/通信的同步性
- Mariana三个框架的设计选择

框架	目标业务	计算单元	数据并行	模型并行	SGD模式
Mariana DNN	语音识别	GPU	支持	不支持	同步
Mariana CNN	图像识别	GPU	支持	支持	同步
Mariana Cluster	广告推荐	CPU	支持	支持	异步

# Mariana DNN的多GPU数据并行：性能模型



# Mariana DNN的多GPU数据并行：参数交换架构

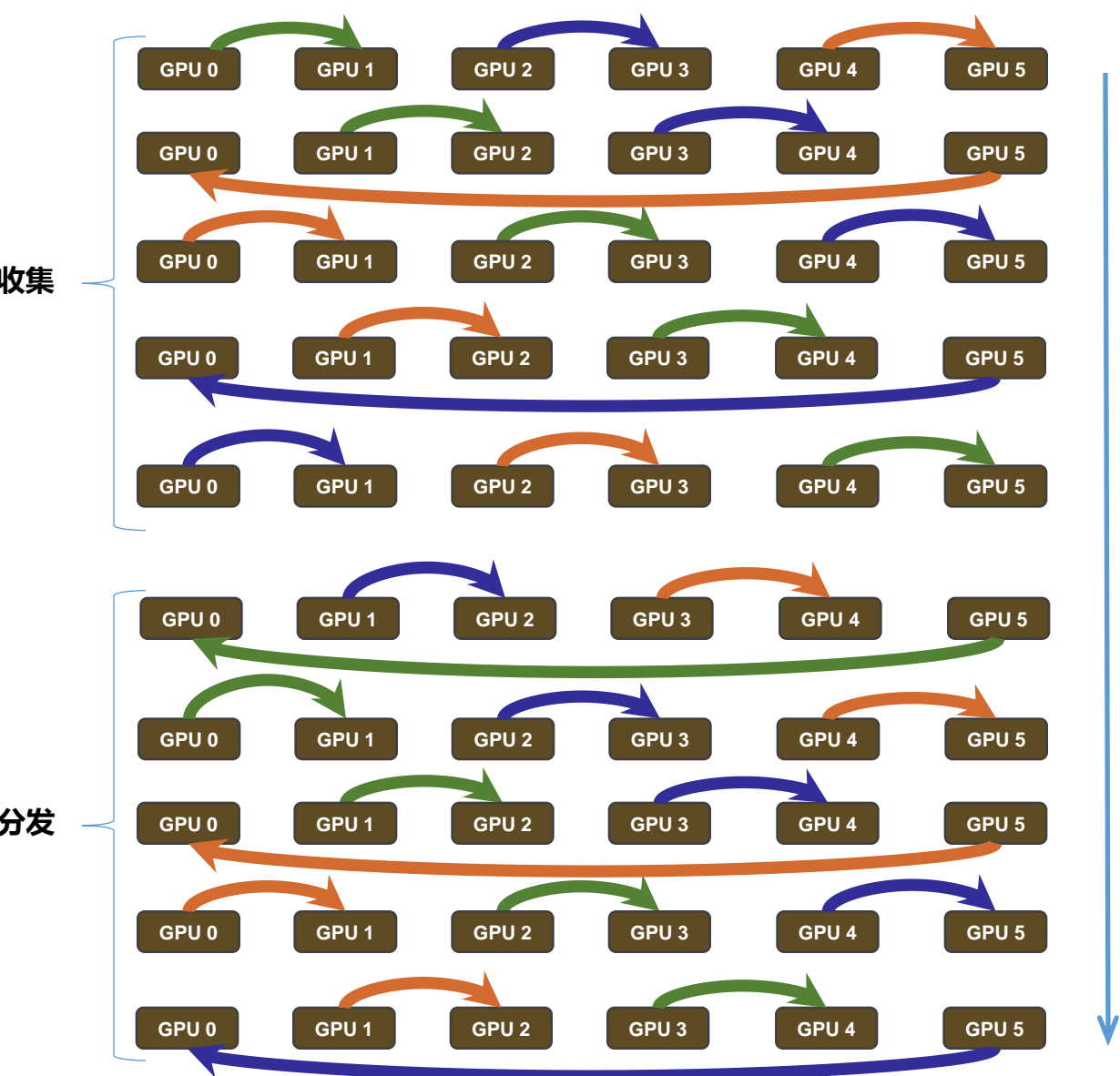


多GPU间参数交换: 基于PCIe互联的架构

# 多GPU卡通信性能模型

- 读取远程GPU内存计算（ Unified Virtual Addressing ），对小数据能简化编程；但对大块数据访问，其性能不如将数据Copy到GPU本地内存再计算。
- GPU计算和GPU间通信会产生干扰，单个GPU的多个通信也会产生干扰，任意一对GPU卡间的双向通信会干扰。除了引入不同的流隔离上述操作，在适当的地方加入同步点也可以提升效率。
- 任意两组GPU卡（ 如GPU1 GPU3和GPU2 GPU4 ）组间读写没有干扰，可并行，和GPU卡的位置无关，即使跨IOH也没有显著差别。

# Mariana DNN : 参数交换的线性拓扑结构



时间线

4GPU	6GPU
奇数编号GPU负责收集和分发模型的2/n	
n-1个周期收集Delta	
n-1个周期分发Param	
用时3s	用时4.6s

# 线形拓扑的性能及可扩展性分析

GPU数	带宽	模型大小	模型复制时间	模型分片	收集周期数	收集通信用时	单次参数交换通信用时（收集+分发）	Cache交换通信用时（MBS=2048）	Cache交换用时（实测，独占）
N	BW	MS	$T_0=MS/BW$	$n/2$	$n-1$	$2T_0(\frac{n-1}{n})$	$4T_0(\frac{n-1}{n})$	交换次数 * 单次参数交换用时	
4	6.6GB/s	186MB	28.2ms	2	3	42.3ms	84.6ms	1.35s	
6	4.5GB/s	186MB	41.3ms	3	5	68.8ms	137.7ms	2.2s	2.7s
8	4.5GB/s	186MB	41.3ms	4	7	72.3ms	144.6ms	2.3s	

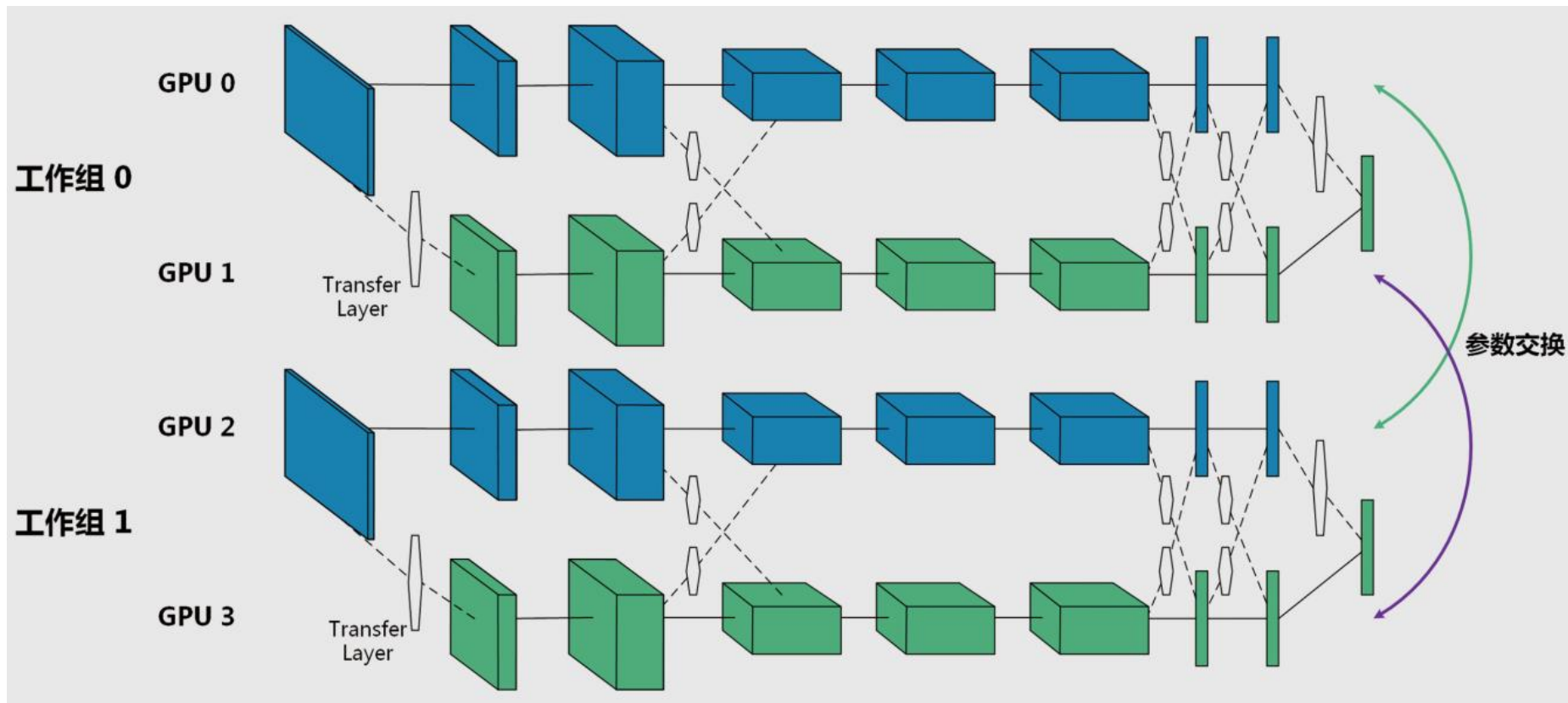
- 理论交换时间中的通信时间和实测时间差距不大，基本吻合。实测时间还包括了计算时间、同步时间等。
- 线形拓扑可以容易的扩展到偶数个GPU的参数交换（ $n=2, 4, 6, 8, \dots$ ）。
- 线形拓扑收集用时随GPU数增长缓慢，且有上界 $2T_0$ ，这说明线形拓扑非常适用于更多GPU卡做数据并行。

## 实测性能：语音识别的声学模型训练

- 超过10,000小时训练数据
- 超过4,000,000,000样本
- 超过50,000,000参数
- 6 GPU数据并行相对单GPU取得了4.6倍加速比



# Mariana CNN的多GPU并行架构



多GPU并行架构: Transfer Layer , IO/CPU/GPU pipeline

# Mariana CNN: 执行引擎

- 每个GPU配有一个独立的执行引擎
- Minibatch开始时，每个GPU的执行引擎同时启动
- 执行引擎在Layer具备执行前向或后向的条件时执行
- 执行引擎逐个Layer完成前向和后向计算的传递

# Mariana CNN : 三阶段流水线加速

任务：从磁盘读取样本  
类型：IO密集型作业  
方案：硬件RAID5提供单机6磁盘并行读

任务：图片预处理  
类型：CPU密集型作业  
方案：2路CPU multi-core提供多线程并行能力

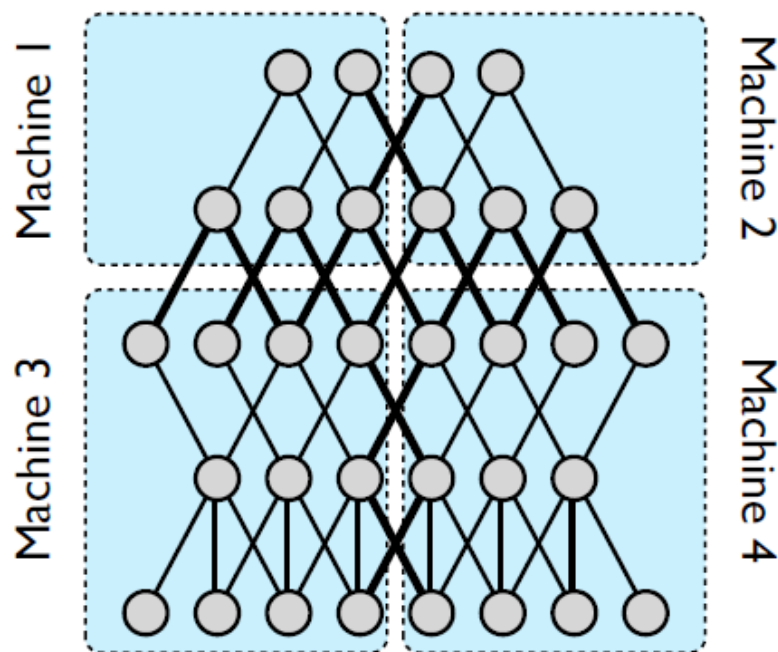
任务：CNN网络计算  
类型：GPU密集型  
方案：4GPU数据并行和模型并行计算

# Mariana CNN的应用：图像识别

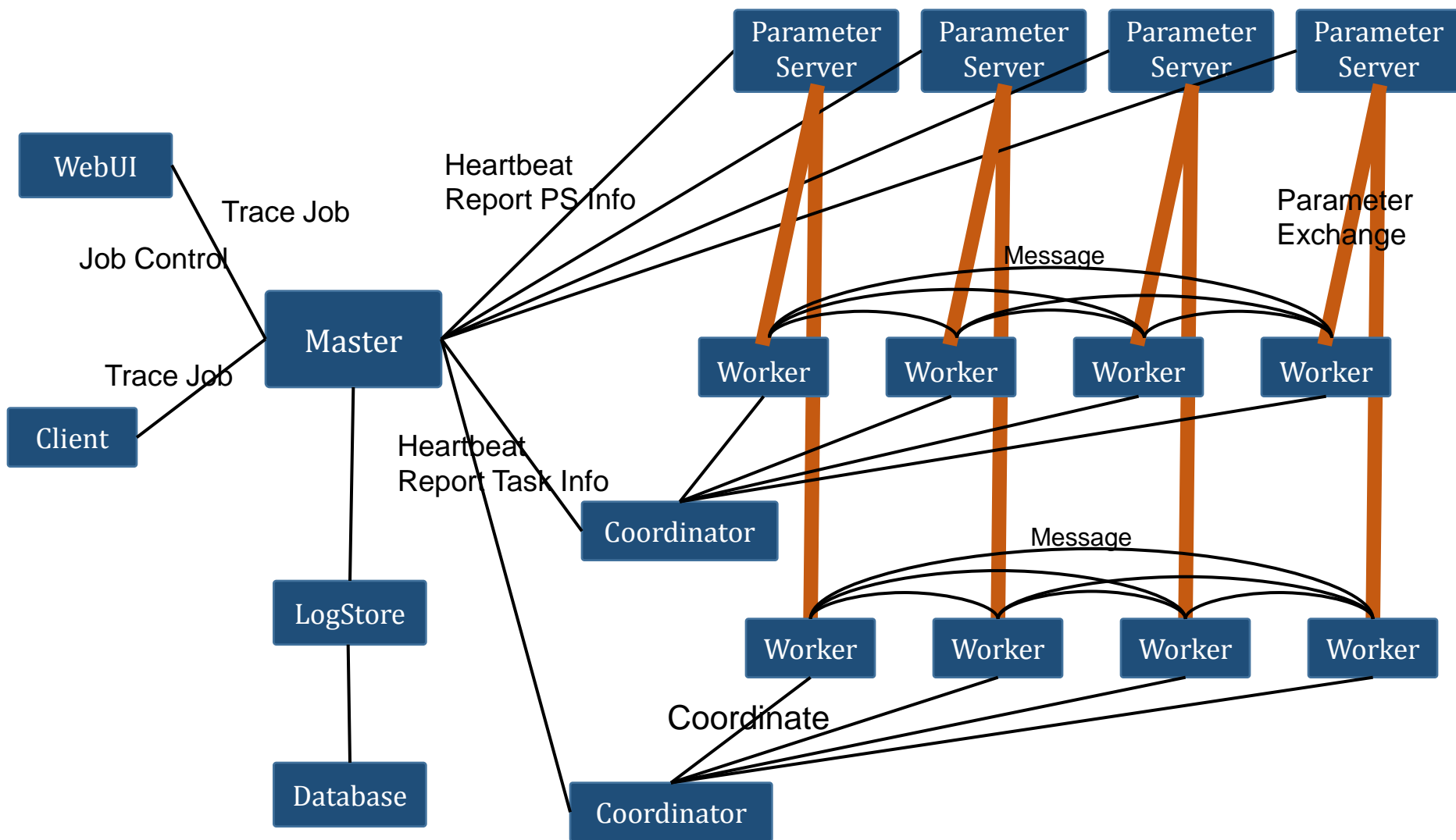
- ImageNet数据，AlexNet 2012
  - 超过1,000,000样本
  - 超过50,000,000参数
- 4 GPU模型并行+数据并行相对单GPU取得了2.5倍加速比
- 可支持更大模型

# Mariana Cluster

- DNNJob有多轮迭代
  - 训练、验证、测试模式
- 划分训练数据，多组节点并行
  - 数据并行
- 每个模型实例用组内多节点
  - 模型并行
- 应用开发
  - Vertex抽象：Compute方法
  - 应用写参数：序列化/反序列化
  - 框架完成DNNJob并行训练



# Mariana Cluster 架构：CPU集群框架



# Mariana Cluster的高可扩展、高可靠和高性能



## 高可扩展

- 每个Master进程管理一个DNNJob
- 数据并行+模型并行
- 可水平扩展的参数服务器



## 高可靠

- 借助分布式文件系统和资源管理系统做容错
- 框架设计容错
- 实现容错



## 高性能

- Worker计算性能调优
- 参数交换性能调优
- 支持Downpour SGD模式并行



# Mariana Cluster演进：支持广点通广告

- 支持广点通广告的点击率预估模型训练



走向实用的挑战：

每天**百亿级**点击率预估请求  
每个请求延迟小于**50 ms**

方式1：CNN/DNN提取特征

抽取图片中的用户点击相关特征

输出给Logistic Regression等浅层模型

方式2：DNN用于模型训练和预测

构造深层模型

# Mariana Cluster演进：针对广告的单机性能极致追求：方法

- 单机内性能优化

- C++版极简稀疏DNN网络内核实现，大幅砍掉开销
- 对象池，去除内存分配释放的开销
- 细节的持续tuning

- 单机内Hogwild! 模式的多线程并行

- 单机仅持有1份模型
- 多线程并行做稀疏计算
- 多线程并发的更新模型参数，每个样本仅更新少量参数，更新时不加锁

# GPU集群建设目标

## ● 目标

- 建立多业务共享的GPU集群
- 灵活管理和调度多个业务作业
- 支持大规模机器学习模型训练



## ● 集群建设：硬件

- 定型GPU服务器，实现性能、成本、功耗的平衡
- 构建高速网络，连接GPU服务器

## ● 集群建设：软件

- 统一资源管理和调度，灵活部署应用框架和软件库
- 实现Mariana GPU Cluster框架支持多机多卡并行

# GPU Cluster软件栈

- 深度学习应用
- 深度学习并行框架
  - 通过Docker images预置常见深度学习并行框架
  - 多机多卡**并行框架的定制与优化
- Gaia
  - GPU集群资源管理和调度
- CUDA-aware MPI library
  - MVAPICH-GDR v2.0
- Docker container
  - 支持通过Docker container运行应用
  - 已测试GPU和高速网络无性能损失
- 基础软件
  - 基于CentOS 6.5
  - 基于CUDA 6.5
  - 基于恰当的网卡OFED驱动

DL Applications

DL frameworks  
In Docker images

Gaia

CUDA-aware MPI

Docker container

基础软件  
OS/CUDA/OFED

# GPU Cluster的Docker image仓库

- 构建共享的Docker images仓库
  - 一键部署深度学习并行框架，简化应用部署
  - 完美支持单机多个版本的库并存
  - images共享，持续扩充可用images
  - 单个image支持千兆/万兆/RoCE和CPU/GPU等环境
- 通过Gaia Portal & API完成基于Docker image的作业提交和管理
- 主要Docker images一览
  - cuda-convnet & cuda-convnet 2
  - Caffe
  - Petuum v0.9 & Petuum v1.0
  - Minerva
  - Mariana DNN
  - Mariana CNN
  - Mariana Cluster



# 深度学习并行化：系统视角

- 系统视角：并行计算的层级（"parallel hierarchy"）
  - CPU指令级并行：SIMD指令，CPU L1/L2 cache
  - GPU并行：many-core的GPU架构，扩展的SIMD
  - 多线程并行：multi-core的CPU
  - 单机多GPU卡并行：利用PCI-e通信
  - 多CPU服务器并行：利用千兆以太网通信
  - 多GPU服务器并行：利用RoCE 40Ge通信
- 系统视角：高性能计算问题，非大数据处理问题
  - 性能远重于扩展性
  - 关注通信时的网络拓扑
  - RPC vs MPI：从编程易用性到性能
  - RoCE高速网络，而非千兆以太网

# 深度学习并行化：算法视角

- 算法视角：算法对系统的影响
  - 数据并行 vs 模型并行：两种的有机组合
  - Hogwild! vs 异步SGD vs 同步SGD：不同应用的选择
  - Scalability有多重要？并行度与收敛性
  - 可以容忍的失败：允许简化系统可靠性设计
- 算法视角：算法本身的改进
  - 近似算法的作用：E.g. Mariana DNN中的近似AdaGrad算法，Hogwild! 的无锁参数更新
  - 非精确的计算：Double or Float?
  - 结果的非唯一性：爬山过程的N种路径





THANKS